symmetry

MDPI

Article

Error State Extended Kalman Filter Localization for Underground Mining Environments

Igor Brigadnov [1], Aleksandr Lutonin [2] and Kseniia Bogdanova [1,*]

[1] Department of Information Systems and Computer Engineering, St. Petersburg Mining University, St. Petersburg 199106, Russia
[2] Department of Informatics and Computer Technology, St. Petersburg Mining University, St. Petersburg 199106, Russia
* Correspondence: bogdanova_ka@pers.spmi.ru

Abstract: The article addresses the issue of mobile robotic platform positioning in GNSS-denied environments in real-time. The proposed system relies on fusing data from an Inertial Measurement Unit (IMU), magnetometer, and encoders. To get symmetrical error gauss distribution for the measurement model and achieve better performance, the Error-state Extended Kalman Filter (ES EKF) is chosen. There are two stages of vector state determination: vector state propagation based on accelerometer and gyroscope data and correction by measurements from additional sensors. The error state vector is composed of the velocities along the x and y axes generated by combining encoder data and the orientation of the magnetometer around the axis z. The orientation angle is obtained from the magnetometer directly. The key feature of the algorithm is the IMU measurements' isolation from additional sensor data, with its further summation in the correction step. Validation is performed by a simulation in the ROS (Robot Operating System) and the Gazebo environment on the grounds of the developed mathematical model. Trajectories for the ES EKF, Extended Kalman Filter (EKF), and Unscented Kalman Filter (UKF) algorithms are obtained. Absolute position errors for all trajectories are calculated with an EVO package. It is shown that using the simplified version of IMU's error equations allows for the achievement of comparable position errors for the proposed algorithm, EKF and UKF.

Keywords: localization; EKF; data fusion; ESEKF; error state; odometry; IMU; encoder

Citation: Brigadnov, I.; Lutonin, A.; Bogdanova, K. Error State Extended Kalman Filter Localization for Underground Mining Environments. *Symmetry* **2023**, *15*, 344. https://doi.org/10.3390/sym15020344

Academic Editors: Beloglazov Ilya, Rudolf Kawalla and Jan Awrejcewicz

Received: 5 December 2022
Revised: 17 January 2023
Accepted: 20 January 2023
Published: 26 January 2023

1. Introduction

Despite the technological advancement in indoor positioning over recent years, real-time localization in underground mining environments remains an important issue [1]. The underground mining operation is an intensive process that requires constant updating of the layout plan, as well as roof subsidence monitoring. Automation of this work with the help of robotic equipment and novel technologies, such as computer vision [2], predictive model control [3], machine learning [4] and neural networks [5,6], will increase the efficiency of the mining process, in general, [7–10], and mine surveying, in particular, [11–13].

Sensor fusion is the de facto standard technique for outdoor and indoor localization and mapping [14,15]. This method increases the reliability of the system along with the accuracy of estimated parameters by merging information from a collection of sensors. On the other hand, by providing redundant measurements this method results in significant computational costs and requires the use of powerful CPUs or special optimization techniques [16,17].

GNSS/IMU systems [18] are one of the most common solutions for outdoor positioning as GNSS signals correct IMU observations prone to drift. This paper focuses on localization in GNSS-denied environments, such as underground mine work. The absence of GNSS signals leads to alternative sensor combinations for indoor positioning. There are three core categories of indoor sensor fusion techniques based on types of main sensors: LiDAR-based, visual-based, and multi-sensor approach.

The work [19] proposes LiDAR-IMU data integration in real-time for UAVs localization in environments with poor lighting conditions. In addition to IMU, LiDARs and cameras [20] are often used together for IMU data correction in indoor environments. If the primary objective is only localization IMU data can be corrected by means of magnetometer and wheel encoder observation [21].

In [22,23], a scheme of a single-channel vertical with an adjustable pendulum based on a three-axis accelerometer and one free gyroscope is proposed, which ensures the unperturbation of navigation reckoning. The main difference between the proposed scheme and the IMU is the accelerometers' data compensation through an external linear speed source (e.g., a log designed to measure the ship's speed). The possibility of using the proposed scheme to estimate the trajectory of dynamic objects seems scarcely probable for the author [23].

EKF is the most prevalent method for position estimation through sensor fusion although its convergence is not assured [24]. Issues of integrating data from IMU and encoders using the extended Kalman filter are presented in the paper [25]. Another common solution for localization is the sigma point approach which is the UKF algorithm [26,27]. UKF overcomes EKF linearization errors using a deterministic sampling approach by representing the state distribution as a set of sample points. Consequently, it achieves better accuracy [28]. However, study [29] is based on real data and shows that the performances of the EKF and UKF are substantially identical.

Indirect formulation of the Kalman filter, error-state extended Kalman filter (ES EKF) for attitude estimation is given in [30]. Several forms of ES EKF with GNSS and WiFi data have been presented recently for outdoor [31] and indoor applications [32]. The study [33] focuses on the ES EKF algorithm for orientation estimation by means of IMU and magnetometer data. The proposed method provides accurate results even when magnetic fluctuations occur. In [34], the authors combine ES EKF with the smooth variable structure filter for attitude estimation based on IMU observations. Attitude estimation of IMU under dynamic conditions using ES EKF is presented in [35]. Nevertheless, the amount of work aimed at ES EKF implementation for indoor environments is relatively small compared to EKF localization although it has significant potential. Low-cost indoor solutions based on IMU, wheel encoders, and magnetometer observations are studied in [36,37].

A comprehensive description of quaternion kinematics for the error-state Kalman filter is provided by [24]. Its findings served as a reference point in the formulation of the proposed approach. Previously, authors implemented a system for integrating data from IMU and encoders for a robotic platform using EKF and assessed linear and angular trajectory displacements [38]. The main contributions of this paper are as follows. The ES EKF is adapted to provide localization in a GNSS-denied environment by means of IMU, encoders, and magnetometer data fusion. In addition, tightly-coupled integration of encoders and magnetometer data is introduced to calculate the errors between prediction and measurements. An extensive description of the algorithm is also provided. Furthermore, the approach is verified through Gazebo simulation by obtaining trajectories from the ES EKF, EKF, and UKF localization algorithms based on the same sensors data.

2. Simulation Methodology

2.1. Reference Frame Decomposition

Before localization algorithm designing, the correlative coordinate frames should be described. According to Figure 1, there are four main coordinate frames which are commonly used for localisation purposes:

- Body frame $\boldsymbol{B}$,
- Local frame $\boldsymbol{L}$,
- Earth-North-UP frame $\boldsymbol{ENU}$,
- Earth-Centered, Earth-Fixed frame $\boldsymbol{ECEF}$

ECEF is a cartesian coordinate frame with the origin at the center of the Earth's mass. The X-axis in this frame represents the intersection of the Earth's equator at zero degrees latitude and longitude accordingly. The Z-axis is orthogonal to X and points to the north pole. The Y-axis is orthogonal to X and the Z-axis is according to the right-hand rule. ***ENU*** is a coordinate system with the origin at a specified point from the ***ECEF*** frame, where the X and Y axes point to geodesic east and north respectively. The Z-axis faces upward according to the right-hand rule. ***L*** is a local frame of reference with an origin at the starting point of system movement. ***B*** is the local coordinate system fixed to the vehicle's center of gravity. Therefore, all measurements from sensors, which are mounted on the robotic platform, must be transformed into the body frame. Then, to obtain sensor data in ***L***, ***ENU***, ***ECEF*** reference frames, it must be converted by transformation matrices $\boldsymbol{T}_B^L$, $\boldsymbol{T}_L^{ENU}$,$\boldsymbol{T}_{ENU}^{ECEF}$ which are explicitly presented in [39,40].

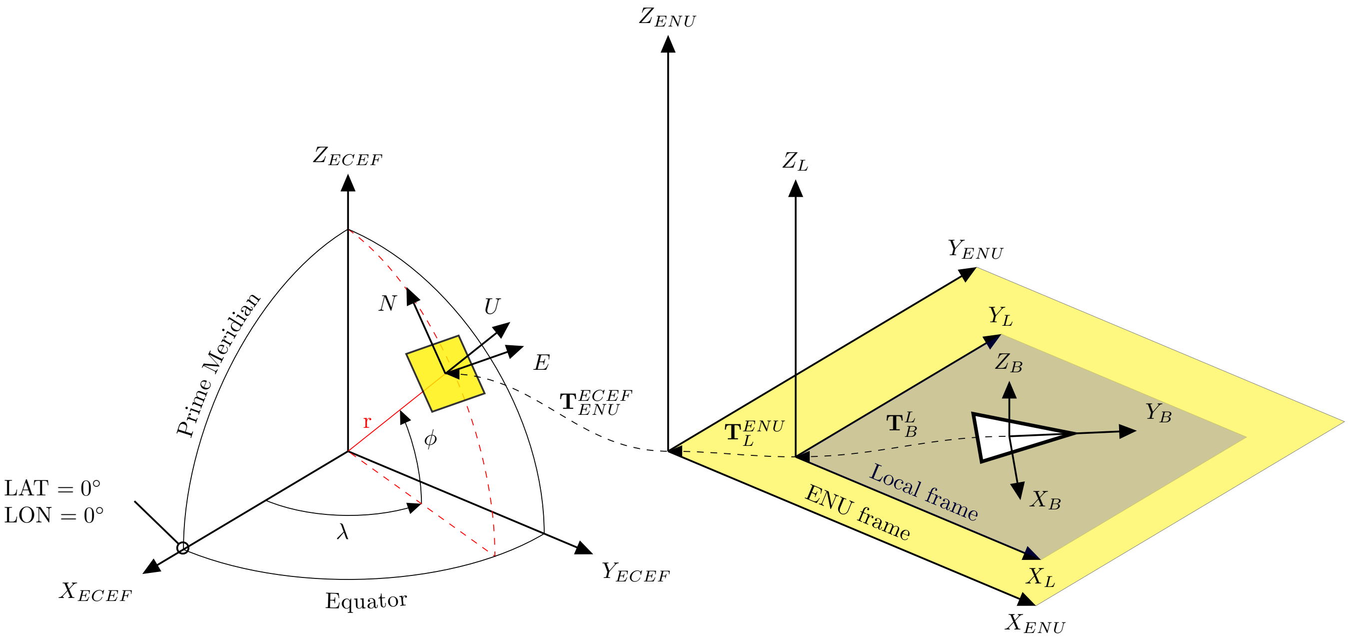


Figure 1. Coordinate frames transformation.

As a matter of convenience in our system, the IMU is firmly attached to the platform and its coordinate frame is aligned to the body frame B, consequently, they have the same orientation and origin and are considered as ENU convention. Data from encoders are aligned to the body frame origin as well. Therefore, in the considered system frame ***L*** is equal to ***ENU*** frame.

2.2. ES EKF with IMU and Encoder Data Fusion

The main concept of the Error-state Extended Kalman filter (ES EKF) consists of the separation of the state vector into two parts: nominal state $\boldsymbol{x}$ and error state $\delta\boldsymbol{x}$.

$$\boldsymbol{x}_t = \boldsymbol{x} + \delta\boldsymbol{x}$$

2.3. Nominal State

The nominal state is represented by a motion model and in most cases composed of IMU observations. Errors related to process noise accumulation and motion model imperfections constitute the error state vector. Therefore, the nominal state vector has the following components:

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{p} & \boldsymbol{v} & \boldsymbol{q} \end{bmatrix}^T,$$

where $\boldsymbol{p} = \begin{bmatrix} p_x & p_y & p_z \end{bmatrix}$—represents position of the platform; $\boldsymbol{v} = \begin{bmatrix} v_x & v_y & v_z \end{bmatrix}$—platform's velocity; $\boldsymbol{q} = \begin{bmatrix} q_w & q_x & q_y & q_z \end{bmatrix}$—orientation in the form of quaternion. They can be obtained with high-frequency IMU data. The IMU contains a collection of three-axis accelerometers and three-axis gyroscopes whose output data is linear acceleration $\boldsymbol{a} = \begin{bmatrix} a_x & a_y & a_z \end{bmatrix}$ and angular velocities $\boldsymbol{\omega} = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}$ accordingly. Integration of the linear acceleration yields the velocity of the platform and angular velocity represents the orientation. With the known initial values of velocity, position, and orientation, the coordinates and attitude of the platform can be evaluated at any time. However, integration of the biases and sensor noise inherent to IMU measurements leads to the unlimited growth of errors with time [41] and as a consequence of this fact, position and orientation estimates are not precise. Hence, applying only mechanization equations does not guarantee satisfactory results in terms of accuracy.

The continuous motion model based on IMU observations, according to [24] is described by the following equations:

$$\dot{\boldsymbol{p}} = \boldsymbol{v}, \qquad \dot{\boldsymbol{v}} = \boldsymbol{a}, \qquad \dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{q} \otimes \boldsymbol{\omega}, \tag{1}$$

where $\otimes$—quaternion product operator.

Because of IMU sensor signal frequency limitations, Equation (1) need to be discretized according to the transmitting time interval Δt. Therefore, the discrete-time motion model can be defined by the equations:

$$\begin{gathered} \boldsymbol{p}_k = \boldsymbol{p}_{k-1} + \boldsymbol{v}_{k-1} \cdot \Delta t + (\boldsymbol{R}\{\boldsymbol{q}_{k-1}\} \cdot \boldsymbol{a}_k + \boldsymbol{g}) \cdot \frac{\Delta t^2}{2}, \\ \boldsymbol{v}_k = \boldsymbol{v}_{k-1} + (\boldsymbol{R}\{\boldsymbol{q}_{k-1}\} \cdot \boldsymbol{a}_k + \boldsymbol{g}) \cdot \Delta t, \\ \boldsymbol{q}_k = \boldsymbol{q}_{k-1} \otimes \boldsymbol{q}(\boldsymbol{\omega}_k \cdot \Delta t), \end{gathered} \tag{2}$$

where k, $k-1$—indexes of current and previous sample step; $\boldsymbol{g} = \begin{bmatrix} 0 & 0 & -9.81 \end{bmatrix}$—gravity vector; $\boldsymbol{R}\{\boldsymbol{q}\}$—quaternion to rotation matrix conversion which can be defined as:

$$\mathbf{R} = \begin{bmatrix} q_w^2 + q_x^2 - q_y^2 - q_z^2 & 2(q_x q_y - q_w q_z) & 2(q_x q_z + q_w q_y) \\ 2(q_x q_y + q_w q_z) & q_w^2 - q_x^2 + q_y^2 - q_z^2 & 2(q_y q_z - q_w q_x) \\ 2(q_x q_z - q_w q_y) & 2(q_y q_z + q_w q_x) & q_w^2 - q_x^2 - q_y^2 + q_z^2 \end{bmatrix}.$$

2.4. Error State

The error state kinematic equations are considered linear due to small magnitudes of error dynamics. Error state vector consists of the following vectors:

$$\delta \boldsymbol{x} = \begin{bmatrix} \delta \boldsymbol{p} & \delta \boldsymbol{v} & \delta \boldsymbol{\theta} \end{bmatrix}^T,$$

where $\delta \boldsymbol{p} = \begin{bmatrix} \delta p_x & \delta p_y & \delta p_z \end{bmatrix}$—represents the position error of the platform; $\delta \boldsymbol{v} = \begin{bmatrix} \delta v_x & \delta v_y & \delta v_z \end{bmatrix}$—velocity error of the platform; $\delta \boldsymbol{\theta} = \begin{bmatrix} \delta \theta_x & \delta \theta_y & \delta \theta_z \end{bmatrix}$—angular error of the platform. Conversion between quaternion and angular representations is obtained by the following equations [42]:

$$\boldsymbol{q}\{\boldsymbol{\theta}\} = \begin{bmatrix} cos(\phi/2) \cdot cos(\theta/2) \cdot cos(\psi/2) + sin(\phi/2) \cdot sin(\theta/2) \cdot sin(\psi/2) \\ sin(\phi/2) \cdot cos(\theta/2) \cdot cos(\psi/2) - cos(\phi/2) \cdot sin(\theta/2) \cdot sin(\psi/2) \\ cos(\phi/2) \cdot sin(\theta/2) \cdot cos(\psi/2) + sin(\phi/2) \cdot cos(\theta/2) \cdot sin(\psi/2) \\ cos(\phi/2) \cdot cos(\theta/2) \cdot sin(\psi/2) - sin(\phi/2) \cdot sin(\theta/2) \cdot cos(\psi/2) \end{bmatrix},$$

$$\boldsymbol{\theta}\{\boldsymbol{q}\} = \begin{bmatrix} \arctan(\frac{2 \cdot (q_w \cdot q_x + q_y \cdot q_z)}{1 - 2 \cdot (q_x^2 + q_y^2)}) \\ \arcsin(2 \cdot (q_w \cdot q_y - q_z \cdot q_x)) \\ \arctan(\frac{2 \cdot (q_w \cdot q_z + q_x \cdot q_y)}{1 - 2 \cdot (q_y^2 + q_z^2)}) \end{bmatrix}. \tag{3}$$

The simplified version of the IMU error state equations with excluded sensor biases in the continuous state is represented by

$$\delta\dot{\boldsymbol{p}} = \delta\boldsymbol{v}, \qquad \delta\dot{\boldsymbol{v}} = -\mathbf{R}[\boldsymbol{a}]_\times\delta\boldsymbol{\theta}, \qquad \delta\dot{\boldsymbol{\theta}} = -[\boldsymbol{\omega}]_\times\delta\boldsymbol{\theta}, \tag{4}$$

where $[\boldsymbol{a}]_\times$—skew symmetric matrix operator, which can be defined by

$$[\boldsymbol{a}]_\times = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}.$$

Error state kinematics equations in discrete time are as follows

$$\begin{aligned} \delta\boldsymbol{p} &= \delta\boldsymbol{p}_{k-1} + \delta\boldsymbol{v}_{k-1}\cdot\Delta t \\ \delta\boldsymbol{v} &= \delta\boldsymbol{v}_{k-1} + (-\boldsymbol{R}\{\boldsymbol{q}_{k-1}\}\cdot[\boldsymbol{a}_{k-1}]_\times\cdot\delta\boldsymbol{\theta}_{k-1})\cdot\Delta t \\ \delta\dot{\boldsymbol{\theta}} &= \boldsymbol{R}\{\boldsymbol{\omega}_{k-1}\cdot\Delta t\}^T\cdot\delta\boldsymbol{\theta}_{k-1} \end{aligned} \tag{5}$$

The ES EKF, like the classic Kalman filter, includes two recursive steps. The first one considers the propagation of the state vector and covariance (prediction step), and the second carries out correction or update. It is important to point out that EKF generates an output state only after measurement update step [43], while ES EKF output state vector can be obtained with prediction data. It increases the reliability of the system in case of additional sensor failure.

2.5. ES EKF Prediction Step

Propagation deals with the state vector implementing mechanization equations mentioned above (2) which should be rewritten as follows:

$$\begin{aligned} \hat{\boldsymbol{p}}_k^- &= \hat{\boldsymbol{p}}_{k-1}^+ + \hat{\boldsymbol{v}}_{k-1}^+\cdot\Delta t + (\boldsymbol{R}\{\hat{\boldsymbol{q}}_{k-1}^+\}\cdot\boldsymbol{a}_k + \boldsymbol{g})\cdot\frac{\Delta t^2}{2}, \\ \hat{\boldsymbol{v}}_k^- &= \hat{\boldsymbol{v}}_{k-1}^+ + (\boldsymbol{R}\{\hat{\boldsymbol{q}}_{k-1}^+\}\cdot\boldsymbol{a}_k + \boldsymbol{g})\cdot\Delta t, \\ \hat{\boldsymbol{q}}_k^- &= \hat{\boldsymbol{q}}_{k-1}^+ \otimes \boldsymbol{q}(\boldsymbol{\omega}_k\cdot\Delta t), \end{aligned} \tag{6}$$

where $\hat{\boldsymbol{x}}^-$—a priori and $\hat{\boldsymbol{x}}^+$— posteriori state estimation for each vactor accordingly.

Likewise, the error covariance matrix of the state estimate $\boldsymbol{P}^-$ is computed at that stage. Unlike the standard variations of KF, provided that additional sensor data is available, the ES EKF algorithm performs a correction step, otherwise—it continues propagation with time.

The error covariance matrix of the state estimate can be obtained with the following equations:

$$\boldsymbol{P}_k^- = \boldsymbol{F}_x\cdot\boldsymbol{P}_{k-1}^+\cdot\boldsymbol{F}_x^T + \boldsymbol{F}_i\cdot\boldsymbol{Q}_i\cdot\boldsymbol{F}_i^T, \tag{7}$$

where $\boldsymbol{F}_x$—the transition matrix which can be obtained as Jacobian of error state kinematic Equation (5); $\boldsymbol{F}_i$— the perturbation vector Jacobian matrix; $\boldsymbol{Q}_i$—measurement covariance matrix for the prediction step; $\boldsymbol{P}_{k-1}^+$—posteriori error covariance matrix of the state estimate at the previous step.

In the first step matrix $\boldsymbol{P}_k^-$ is defined as a unit matrix multiplied by the initial covariance value, for example:

$$\boldsymbol{P}_k^- = \boldsymbol{P}_0 = \boldsymbol{I}_{9\times 9}\cdot 0.01$$

The corresponding error state transition matrix $\boldsymbol{F}_x$ is derived via the Taylor series block-wise truncation as follows:

$$\boldsymbol{F}_x = \begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{I}_{3\times3}\cdot\Delta t & -0.5\cdot\boldsymbol{R}\{\hat{\boldsymbol{q}}_{k-1}^+\}\cdot[\boldsymbol{a}_k]_\times\cdot\Delta t^2 \\ 0 & \mathbf{I}_{3\times3} & -[\boldsymbol{R}\{\hat{\boldsymbol{q}}_k^+\}\cdot\boldsymbol{a}_{k-1}]_\times\cdot\Delta t \\ 0 & 0 & \boldsymbol{R}\{\hat{\boldsymbol{q}}_{k-1}^+\}\cdot(\boldsymbol{\omega}\cdot\Delta t)^T \end{bmatrix} \tag{8}$$

Perturbation $\boldsymbol{F}_i$ matrix is defined by

$$\boldsymbol{F}_i = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \boldsymbol{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \boldsymbol{I}_{3\times3} \end{bmatrix} \quad (9)$$

The error covariance matrix can be obtained as a multiplication of the unit matrix $\boldsymbol{I}_{6\times6}$ multiplied by integrated covariances of linear acceleration $\sigma^2_{a_x}, \sigma^2_{a_y}, \sigma^2_{a_z}$ and angular velocities $\sigma^2_{\omega_x}, \sigma^2_{\omega_y}, \sigma^2_{\omega_z}$ accordingly:

$$\boldsymbol{Q}_i = Diag(\sigma^2_{a_x} \cdot \Delta t^2, \quad \sigma^2_{a_y} \cdot \Delta t^2, \quad \sigma^2_{a_z} \cdot \Delta t^2, \quad \sigma^2_{\omega_x} \cdot \Delta t^2, \quad \sigma^2_{\omega_y} \cdot \Delta t^2, \quad \sigma^2_{\omega_z} \cdot \Delta t^2) \quad (10)$$

2.6. ES EKF Measurement Update Step

Correction implies the calculation of the error state vector via Kalman gain and innovation computation along with process covariance updating. Instead of using IMU observations as measurements for innovation calculation, additional sensor data is applied. Thereby, ES EKF innovation is the discrepancy between the prediction based on IMU equations and measurements from the additional sensor.

This paper represents a measurement update by fused wheel encoders and magnetometer data, described in Section 3.1. Therefore, the measurement output is defined as

$$\boldsymbol{y}_k = \begin{bmatrix} 0 & 0 & 0 & v_x & v_y & 0 & \theta_x & \theta_y & \theta_z \end{bmatrix}^T \quad (11)$$

Depending on the availability of measurements from additional sensors, the algorithm performs a correction step or continues updating the nominal state vector based only on the IMU data. If measurements are available, the gain matrix is calculated according to the equation:

$$\boldsymbol{K}_k = \boldsymbol{P}_k^- \cdot \boldsymbol{H}_k^T \cdot (\boldsymbol{H}_k \cdot \boldsymbol{P}_k^- \cdot \boldsymbol{H}_k^T + \boldsymbol{R})^{-1}, \quad (12)$$

where $\boldsymbol{P}_k^-$—state vector covariance matrix, defined by (7); $\boldsymbol{R}$—measurement noise covariance matrices; $\boldsymbol{H}$—observation matrix. Measurement noise covariance matrices can be defined as

$$\boldsymbol{R} = Diag(\boldsymbol{\sigma}^2_{\boldsymbol{p}}, \quad \boldsymbol{\sigma}^2_{\boldsymbol{v}}, \quad \boldsymbol{\sigma}^2_{\boldsymbol{\theta}}), \quad (13)$$

where $\boldsymbol{\sigma}^2_{\boldsymbol{p}}, \boldsymbol{\sigma}^2_{\boldsymbol{v}}, \boldsymbol{\sigma}^2_{\boldsymbol{\theta}}$ —measurement dispersion matrices for the position, velocity, and orientation in coordinates x, y, z.

The observation matrix demonstrates what types of measurements are taken into account when calculating innovation. If the state of a system is directly measurable then the H matrix for the according element is 1. As the measurement matrix (11) consists of $u_x, u_y, \theta_x, \theta_y, \theta_z$ elements, $\boldsymbol{H}$ can be defined as follows:

$$\boldsymbol{H} = Diag(0, \quad 0, \quad 0, \quad 1, \quad 1, \quad 0, \quad 1, \quad 1, \quad 1) \quad (14)$$

The updated error state vector is a 1×9 matrix computed using the difference between additional sensor data and predicted nominal state (that is innovation) and Kalman gain

$$\delta\hat{\boldsymbol{x}}_k^+ = \boldsymbol{K}_k(\boldsymbol{y}_k - \hat{\boldsymbol{x}}_k^-) = \begin{bmatrix} \delta\hat{\boldsymbol{p}}_k^+ & \delta\hat{\boldsymbol{v}}_k^+ & \delta\hat{\boldsymbol{\theta}}_k^+ \end{bmatrix}^T \quad (15)$$

where $\hat{\boldsymbol{x}}_k^-$—a priori vector of the estimated state which can be found as

$$\hat{\boldsymbol{x}}_k^- = \begin{bmatrix} \hat{\boldsymbol{p}}_k^- & \hat{\boldsymbol{v}}_k^- & \boldsymbol{\theta}\{\hat{\boldsymbol{q}}_k^-\} \end{bmatrix}^T$$

Then on the grounds of error state vector total state is corrected

$$\begin{aligned} \hat{\boldsymbol{p}}_k^+ &= \hat{\boldsymbol{p}}_k^- + \delta\hat{\boldsymbol{p}}_k^+ \\ \hat{\boldsymbol{v}}_k^+ &= \hat{\boldsymbol{v}}_k^- + \delta\hat{\boldsymbol{v}}_k^+ \\ \hat{\boldsymbol{q}}_k^+ &= \hat{\boldsymbol{q}}_k^- \otimes \boldsymbol{q}\{\delta\hat{\boldsymbol{\theta}}_k^+\} \end{aligned} \tag{16}$$

The Joseph formula is used for the covariance update to provide a symmetric and positive matrix.

$$\boldsymbol{P_k}^+ = (\boldsymbol{I} - \boldsymbol{K}_k \cdot \boldsymbol{H}_k) \cdot \boldsymbol{P}_k^- \cdot (\boldsymbol{I} - \boldsymbol{K}_k \cdot \boldsymbol{H}_k)^T + \boldsymbol{K}_k \cdot \boldsymbol{R} \cdot \boldsymbol{K}_k^T \tag{17}$$

Thus, the ES EKF itself estimates only errors by taking into account supplementary sensor data, e.g., LiDARs, cameras, or wheel encoders. In this way, by means of an estimated error state, the total state is improved.

2.7. ES EKF Error Reset

After vector state updating performed by (16) as well as the error covariance matrix of the state estimation (17), the error should be set to zero:

$$\begin{bmatrix} \delta\hat{\boldsymbol{p}}_k^+ & \delta\hat{\boldsymbol{v}}_k^+ & \delta\hat{\boldsymbol{q}}_k^+ \end{bmatrix} = \boldsymbol{0} \tag{18}$$

Error covariance matrix $\boldsymbol{P_k}^+$ also should be reset according to the following equation:

$$\boldsymbol{P_k}^+ = \boldsymbol{G} \cdot \boldsymbol{P_k}^+ \cdot \boldsymbol{G}^T, \tag{19}$$

where $\boldsymbol{G}$ is the Jacobian matrix which can be found by

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{I} & 0 & 0 \\ 0 & \boldsymbol{I} & 0 \\ 0 & 0 & \boldsymbol{I} - [\frac{1}{2}\delta\hat{\boldsymbol{\theta}}_k^+]_\times \end{bmatrix}, \tag{20}$$

To structurize the error state extended Kalman filter algorithm, a combined block diagram is shown in Figure 2.

According to Figure 2, the ES EKF algorithm first got measurements from the accelerometer and magnetometer. Then, if it is a first measurement, the initial values of $\boldsymbol{P}_0, \boldsymbol{p}_0, \boldsymbol{v}_0$, and $\boldsymbol{q}_0$ should be set, as well as matrices $\boldsymbol{F}_x, \boldsymbol{F}_i, \boldsymbol{Q}_i$ initialized. After parameter initialization, the prediction step should be performed by computing $\hat{\boldsymbol{p}}_k^-, \hat{\boldsymbol{v}}_k^-, \hat{\boldsymbol{q}}_k^-$, and $\boldsymbol{P}^-$. Then, if the measurements of wheel encoders $\boldsymbol{v}$ and magnetometer $\boldsymbol{q}$ are available, the error calculation block computes $\delta\hat{\boldsymbol{p}}_k^+, \delta\hat{\boldsymbol{v}}_k^+, \delta\hat{\boldsymbol{q}}_k^+$, and $\boldsymbol{P}_k^+, \boldsymbol{G}_k^+$. At the final stage, the error values of position, velocity, and orientation are injected into the nominal state and go to «Output block» which provides an error reset for $\boldsymbol{P}_k^+$ and sends updated nominal values back to the prediction block.

The main advantage of that approach is the robustness as the filter's malfunction does not influence the system algorithm. In the case of filter failure, position and orientation estimation will be obtained by motion model propagation.

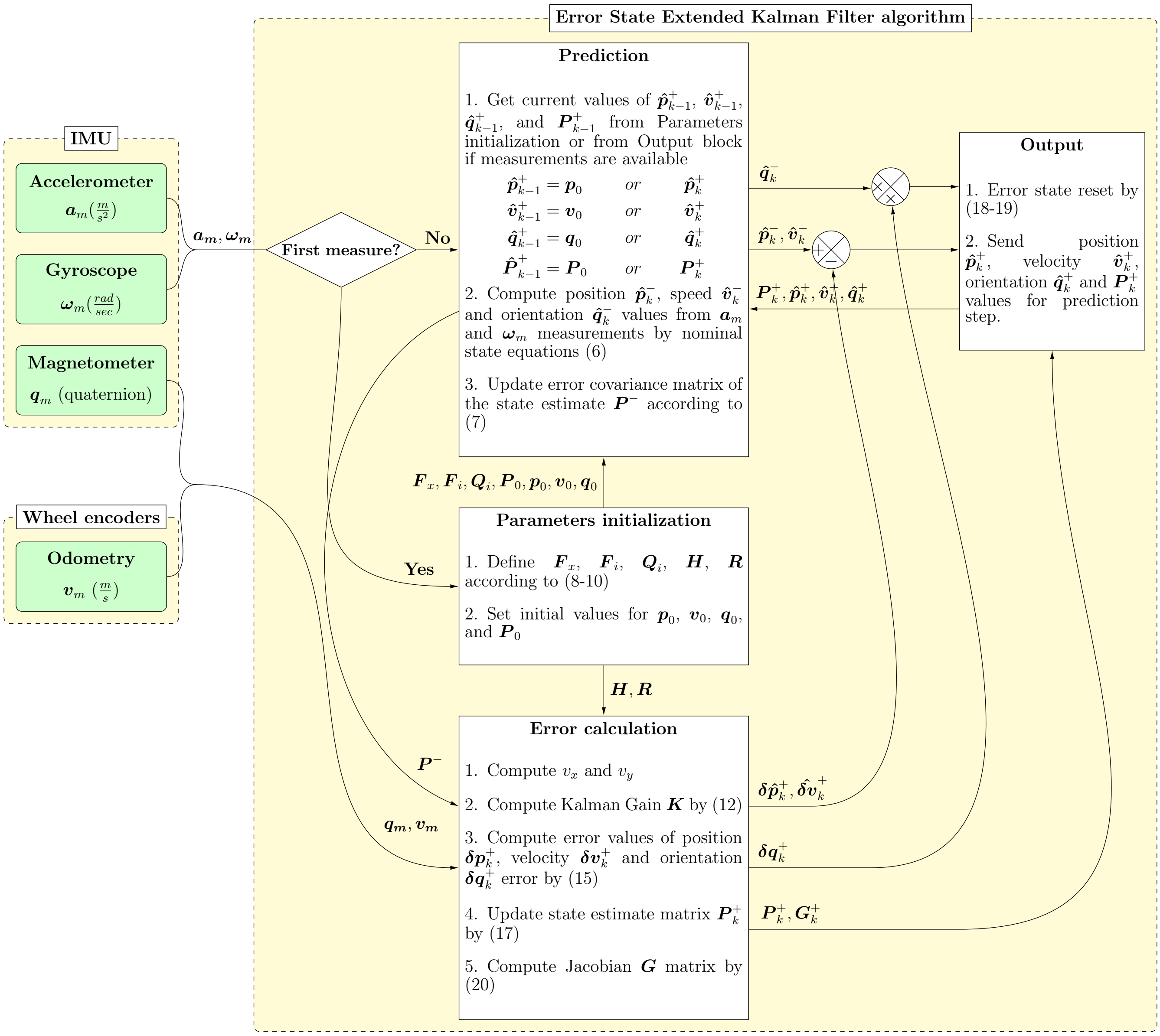


Figure 2. ES EKF algorithm based on IMU, magnetometer and wheel encoders data.

3. Simulation Results

All simulations are performed by a Robot operating system (ROS) package [44], with Gazebo simulation software [45]. A four-wheeled differential drive robot «Jackal» was chosen as the main platform with an onboard mounted IMU sensor and wheel encoders. The simulated robotic platform has the following specifications, presented in Table 1.

Table 1. «Jackal» robot platform specifications

Context	Item	Value	Unit
Platform wheelbase	L	0.262	m
Robot speed limit	v_{max}	0.62	m/s
Linear acceleration RMSE	$\sigma_a = [\sigma_{a_x} \quad \sigma_{a_y} \quad \sigma_{a_z}]$	[0.005 0.005 0.005]	m/s^2
Angular velocity RMSE	$\sigma_\omega = [\sigma_{\omega_x} \quad \sigma_{\omega_y} \quad \sigma_{\omega_z}]$	[0.005 0.005 0.005]	rad/s
Magnetometer RMSE	$\sigma_\omega = [\sigma_{\omega_x} \quad \sigma_{\omega_y} \quad \sigma_{\omega_z}]$	[0.005 0.005 0.005]	µT^2
Platform speed RMSE	$\sigma_v = [\sigma_{v_x} \quad \sigma_{v_y} \quad \sigma_{v_z}]$	[0.001 0.001 10000]	m/s

The platform speed RMSE in z reference frame was set high because of a lack of measurements.

3.1. Magnetometer and Odometer Sensor Fusion

Vector $\boldsymbol{y}_k$ from Equation (11) is composed of the wheel encoders and magnetometer measurements. Since the ROS framework has a specific sensor data transmission format, odometry data from the wheel encoders are sent as linear v and angular ω speed. Therefore, the v_x, v_y values can be obtained by the following equations:

$$v_x = \frac{v_L + v_R}{2 \cdot \cos(\theta_m)}, v_y = \frac{v_L + v_R}{2 \cdot \sin(\theta_m)}, \tag{21}$$

where v_L, v_R—left and right wheel speed which can be calculated in accordance with the formulas (22); θ_m—orientation angle on z reference frame provided from magnetometer data after conversion source orientation vector from quaternion to Euler's angle according to (3).

$$v_L = v - \frac{\omega \cdot L}{2}, v_R = v + \frac{\omega \cdot L}{2} \tag{22}$$

where L—the distance between the left and right wheels.

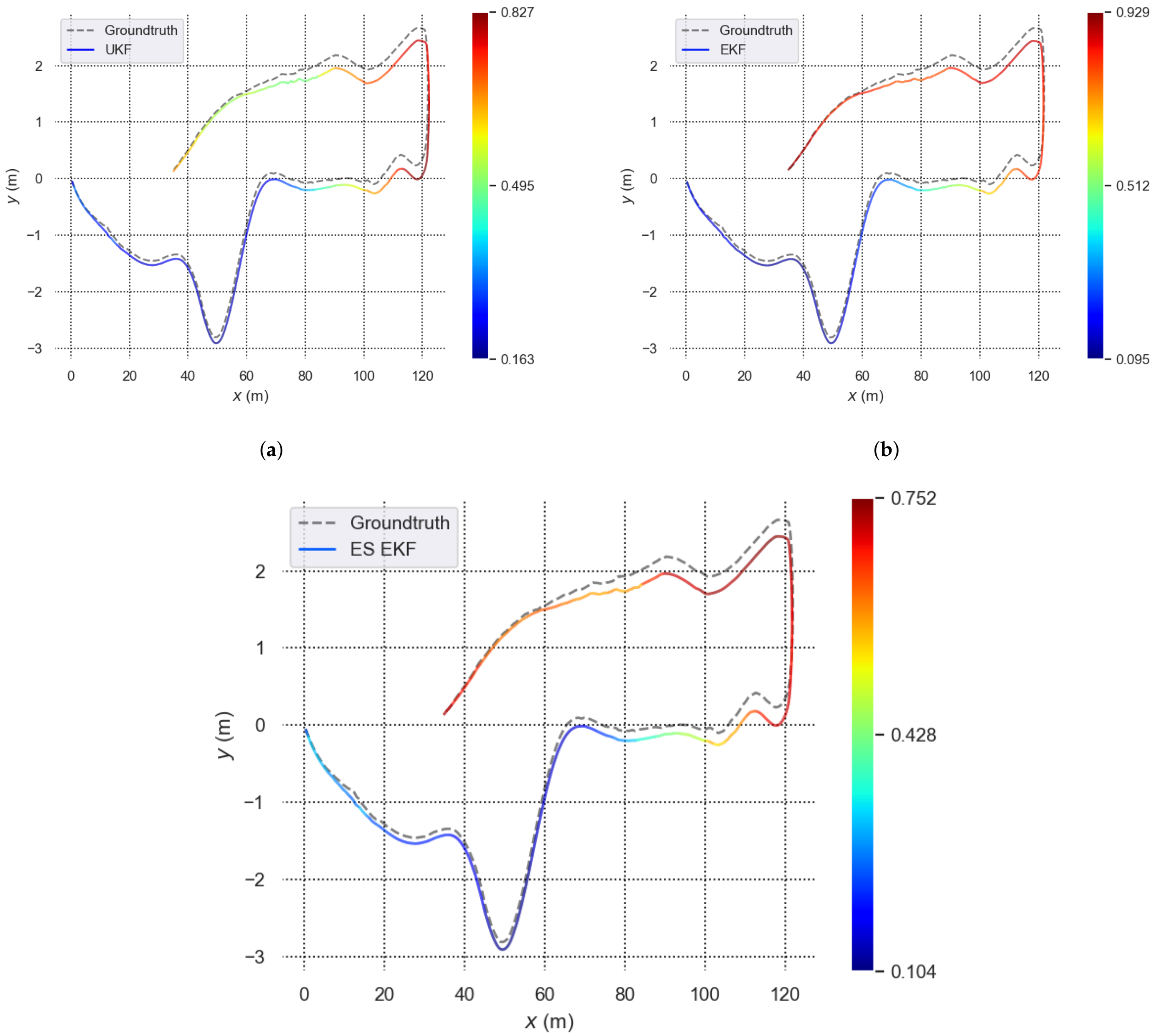
Therefore, the measurement vector state (11) consists of the velocities v_x and v_y obtained by encoder and magnetometer data fusion (21) as well as magnetometer orientation data θ_x, θ_y, θ_z.

4. Discussion

The algorithm test is performed in a simulated underground mine environment. On the account of the simulation, two trajectories are derived: a reference trajectory and a trajectory based on sensor fusion data (Figure 3). Using the evo package [46], differences between corresponding points of trajectories are calculated. Achieved results were compared with EKF and UKF by using a "robot localization" ROS package [47]. The simulation results are presented in Table 2. According to Table 2, the mean error values for the ES EKF and UKF are almost the same (0.517 cm and 0.516 cm, accordingly), while the EKF is 0.13 cm worse. The maximum values are also close to each other with values 0.752, 0.929, and 0.827 for the ES EKF, EKF and UKF accordingly. The trajectory built using the proposed algorithm has similar accuracy to the trajectories obtained by the Extended Kalman Filter and the Unscented Kalman Filter. However, all three algorithms need to increase the accuracy of localization.

Table 2. Statistics of calculated differences.

Error	ES EKF	EKF	UKF
RMSE	0.517	0.620	0.516
Mean	0.469	0.543	0.475
Median	0.570	0.679	0.512
Min	0.104	0.095	0.163
Max	0.752	0.929	0.827



(**a**) (**b**)

(**c**)

Figure 3. ES EKF, EKF, and UKF trajectories comparison. (**a**) UKF reference trajectory; (**b**) EKF reference trajectory; (**c**) ES EKF reference trajectory.

5. Conclusions

In this paper, the ES EKF algorithm is proposed for real-time localization in GNSS-denied environments. To correct the vector state calculated through motion model equations based on the IMU observations, the heading angle measured by magnetometer and the speed determined by means of encoder data were added. The proposed system is validated through a Gazebo simulation. The algorithm minimizes IMU drift by means of additional sensor data and increases the accuracy of positioning.

The algorithm achieves accuracy comparable to the results of the classic Extended Kalman Filter and Unscented Kalman Filter. At the same time, the proposed system is more resistant to failures of the auxiliary sensors. In the future, 3D LiDAR data will be integrated into the system to increase localization accuracy.

Author Contributions: Conceptualization, I.B. and K.B.; methodology, K.B.; software, A.L.; validation, A.L., K.B. and I.B.; formal analysis, K.B.; investigation, A.L.; writing—original draft preparation, K.B.; writing—review and editing, A.L.; supervision, I.B.; project administration, I.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

Abbreviations

The following abbreviations are used in this manuscript:

MDPI	Multidisciplinary Digital Publishing Institute
IMU	Inertial Measurement Units
EKF	Extended Kalman Filter
ES EKF	Error State Extended Kalman Filter
RMSE	Root Mean Square Error

References

1. Baek, J.; Park, J.; Cho, S.; Lee, C. 3D Global Localization in the Underground Mine Environment Using Mobile LiDAR Mapping and Point Cloud Registration. *Sensors* **2022**, *22*, 2873. [CrossRef]
2. Vasilyeva, N.V.; Boikov, A.V.; Erokhina, O.O.; Trifonov, A.Y. Automated digitization of radial charts. *J. Min. Inst.* **2021**, *247*, 82–87. [CrossRef]
3. Vasilyeva, N.; Fedorova, E.; Kolesnikov, A. Big Data as a Tool for Building a Predictive Model of Mill Roll Wear. *Symmetry* **2021**, *13*, 859. [CrossRef]
4. Brilliant, L.S.; Zavialov, A.S.; Danko, M.U.; Andronov, K.A.; Shpurov, I.V.; Bratkova, V.G.; Davydov, A.V. Integration of machine learning methods and geological and hydrodynamic modeling in field development design (Russian). *Neft. Khozyaystvo Oil Ind.* **2022**, *2022*, 48–53. [CrossRef]
5. Pshenin, V.; Liagova, A.; Razin, A.; Skorobogatov, A.; Komarovsky, M. Robot Crawler for Surveying Pipelines and Metal Structures of Complex Spatial Configuration. *Infrastructures* **2022**, *7*, 75. [CrossRef]
6. Grishchenkova, E. Development of a Neural Network for Earth Surface Deformation Prediction. *Geotech. Geol. Eng.* **2018**, *36*, 1953–1957. [CrossRef]
7. Men'Shikov, S.N.; Dzhaljabov, A.A.; Vasiliev, G.G.; Leonovich, I.A.; Ermilov, O.M. Spatial models developed using laser scanning at gas condensate fields in the northern construction-climatic zone. *J. Min. Inst.* **2019**, *238*, 430–437. [CrossRef]
8. Mazakov, E.; Matrokhina, K.; Trofimets, V. Traffic management at the enterprises of the mineral industry. In *Advances in Raw Material Industries for Sustainable Development Goals*; CRC Press: Boca Raton, FL, USA, 2020; pp. 397–405.
9. Kosarev, O.V.; Tcvetkov, P.S.; Makhovikov, A.B.; Vodkaylo, E.G.; Zulin, V.A.; Bykasov, D.A. Modeling of Industrial IoT complex for underground space scanning on the base of Arduino platform. In *Topical Issues of Rational Use of Natural Resources: Proceedings of the International Forum-Contest of Young Researchers*; CRC: New York, NY, USA, 2018; pp. 407–412.
10. Trushko, V.L.; Protosenya, A.G. Prospects of geomechanics development in the context of new technological paradigm. *J. Min. Inst.* **2019**, *236*, 162–166. [CrossRef]
11. Vystrchil, M.; Sukhov, A.; Novozhenin, S.; Popov, A.; Guba, S. Quality analysis of digital photogrammetric models obtained in low light conditions. In *Journal of Physics: Conference Series*; IOP Publishing: Bristol, UK, 2020; Volume 1661, p. 012089.
12. Gusev, V.; Puporevich, F. Improving accuracy of navigation using gyroscopes with regard to gyro drift and azimuth error. *Min. Informational Anal. Bull. (Sci. Tech. J.)* **2021**, 134–145. [CrossRef]
13. Gusev, V.N.; Blishchenko, A.A.; Sannikova, A.P. Study of a set of factors influencing the error of surveying mine facilities using a geodesic quadcopter. *J. Min. Inst.* **2022**, *254*, 173–179. [CrossRef]
14. El-Sheimy, N.; Li, Y. Indoor navigation: State of the art and future trends. *Satell. Navig.* **2021**, *2*, 7. [CrossRef]
15. He, X.; Pan, S.; Gao, W.; Lu, X. LiDAR-Inertial-GNSS Fusion Positioning System in Urban Environment: Local Accurate Registration and Global Drift-Free. *Remote Sens.* **2022**, *14*, 2104. [CrossRef]
16. Mustafin, M.; Bykasov, D. Adjustment of Planned Surveying and Geodetic Networks Using Second-Order Nonlinear Programming Methods. *Computation* **2021**, *9*, 131. [CrossRef]
17. Beloglazov, I.; Krylov, K. An Interval-Simplex Approach to Determine Technological Parameters from Experimental Data. *Mathematics* **2022**, *10*, 2959. [CrossRef]
18. Chi, C.; Zhan, X.; Wang, S.; Zhai, Y. Enabling robust and accurate navigation for UAVs using real-time GNSS precise point positioning and IMU integration. *Aeronaut. J.* **2021**, *125*, 87–108. [CrossRef]

19. Yang, J.; Lin, C.; You, B.; Yan, Y.; Cheng, T. Rtlio: Real-time lidar-inertial odometry and mapping for uavs. *Sensors* **2021**, *21*, 3955.
20. Rego, G.; Bazhenov, N.; Korzun, D. Trajectory Construction for Autonomous Robot Movement based on Sensed Physical Parameters and Video Data. In Proceedings of the 2021 30th Conference of Open Innovations Association FRUCT, Oulu, Finland, 27–29 October 2021; pp. 200–206. [CrossRef]
21. Murthy, S.D.; Krishnan, S.; Sundarrajan, G.; S, K.K.; Bhagwanth, R.; Balasubramanian, V. A Robust Approach for Improving the Accuracy of IMU based Indoor Mobile Robot Localization. In Proceedings of the ICINCO 2016, Lisbon, Portugal, 29–31 July 2016; pp. 436–445.
22. Binder, Y.I. Dead reckoning using an attitude and heading reference system based on a free gyro with equatorial orientation. *Gyroscopy Navig.* **2017**, *8*, 104–114. [CrossRef]
23. Binder, Y.I.; Paderina, T.; Litmanovich, Y.A. Method errors of the dead reckoning schemes based on a single free gyroscope. *Gyroscopy Navig.* **2019**, *10*, 292–302. [CrossRef]
24. Solà, J. Quaternion kinematics for the error-state Kalman filter. *arXiv* **2017**, arXiv: 1711.02508.
25. Brossard, M.; Bonnabel, S. Learning Wheel Odometry and IMU Errors for Localization. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20-24 May 2019; pp. 291–297. [CrossRef]
26. Lasmadi, L.; Kurniawan, F.; Dermawan, D.; Pratama, G.N. Mobile robot localization via unscented kalman filter. In Proceedings of the 2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), Yogyakarta, Indonesia, 5–6 December 2019; pp. 129–132.
27. Ullah, I.; Shen, Y.; Su, X.; Esposito, C.; Choi, C. A localization based on unscented Kalman filter and particle filter localization algorithms. *IEEE Access* **2019**, *8*, 2233–2246. [CrossRef]
28. Yang, C.; Shi, W.; Chen, W. Comparison of unscented and extended Kalman filters with application in vehicle navigation. *J. Navig.* **2017**, *70*, 411–431. [CrossRef]
29. D'Alfonso, L.; Lucia, W.; Muraca, P.; Pugliese, P. Mobile robot localization via EKF and UKF: A comparison based on real data. *Robot. Auton. Syst.* **2015**, *74*, 122–127. [CrossRef]
30. Roumeliotis, S.; Sukhatme, G.; Bekey, G. Circumventing dynamic modeling: Evaluation of the error-state Kalman filter applied to mobile robot localization. In Proceedings of the 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C), Detroit, MI, USA, 10–19 May 1999; Volume 2, pp. 1656–1663. [CrossRef]
31. D'Adamo, T.; Phillips, T.; McAree, P. LiDAR-Stabilised GNSS-IMU Platform Pose Tracking. *Sensors* **2022**, *22*, 2248. [CrossRef] [PubMed]
32. Li, Z.; Zhang, Y. Constrained ESKF for UAV Positioning in Indoor Corridor Environment Based on IMU and WiFi. *Sensors* **2022**, *22*, 391. [CrossRef] [PubMed]
33. Vitali, R.V.; McGinnis, R.S.; Perkins, N.C. Robust error-state Kalman filter for estimating IMU orientation. *IEEE Sens. J.* **2020**, *21*, 3561–3569. [CrossRef]
34. Youn, W.; Gadsden, S.A. Combined quaternion-based error state Kalman filtering and smooth variable structure filtering for robust attitude estimation. *IEEE Access* **2019**, *7*, 148989–149004. [CrossRef]
35. Sun, W.; Wu, J.; Ding, W.; Duan, S. A robust indirect Kalman filter based on the gradient descent algorithm for attitude estimation during dynamic conditions. *IEEE Access* **2020**, *8*, 96487–96494. [CrossRef]
36. Panich, S. Indirect Kalman Filter in Mobile Robot Application. *J. Math. Stat.* **2010**, *6*, 381–384. [CrossRef]
37. Zunaidi, I.; Kato, N.; Nomura, Y.; Matsui, H. Positioning system for 4-wheel mobile robot: Encoder, gyro and accelerometer data fusion with error model method. *CMU. J.* **2006**, *5*, 1., 1–14
38. Lutonin, A.S.; Bogdanova, K.A. Development of a robotic platform for underground geomonitoring. *News Tula State Univ. Tech. Sci.* **2021**, *12*, 209–216.
39. Gong, Z.; Ying, R.; Fei, W.; Qian, J.; Liu, P. Tightly Coupled Integration of GNSS and Vision SLAM Using 10-DoF Optimization on Manifold. *IEEE Sens. J.* **2019**, *19*, 12105–12117. [CrossRef]
40. Cai, G.; Chen, B.M.; Lee, T.H. *Unmanned Rotorcraft Systems*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2011.
41. Scholte, W.J.; Rodrigo Marco, V.; Nijmeijer, H. Experimental Validation of Vehicle Velocity, Attitude and IMU Bias Estimation. *IFAC-PapersOnLine* **2019**, *52*, 118–123.
42. Diebel, J. Representing attitude: Euler angles, unit quaternions, and rotation vectors. *Matrix* **2006**, *58*, 1–35. Available online: https://www.astro.rug.nl/software/kapteyn/_downloads/fa29752e4cd69adcfa2fc03b1c020f4e/attitude.pdf (accessed on 19 January 2023).
43. Titterton, D.; Weston, J. *Strapdown Inertial Navigation Technology*; IET: London, UK, 2004; Volume 17.
44. Quigley, M.; Conley, K.; Gerkey, B.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; Ng, A.Y.; et al. ROS: An open-source Robot Operating System. In Proceedings of the ICRA Workshop on Open Source Software, Kobe, Japan, 12–17 March 2009; Volume 3, p. 5.
45. Koenig, N.; Howard, A. Design and use paradigms for gazebo, an open-source multi-robot simulator. In Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No. 04CH37566), Sendai, Japan, 28 Sepetember–2 October 2004; Volume 3, pp. 2149–2154.

46. Grupp, M. evo: Python Package for the Evaluation of Odometry and SLAM, 2017. Available online: https://github.com/MichaelGrupp/evo (accessed on 19 January 2023).
47. Moore, T.; Stouch, D. A Generalized Extended Kalman Filter Implementation for the Robot Operating System. In Proceedings of the Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13), Padova, Italy, 15–18 July 2014; Springer: Berlin/Heidelberg, Germany, 2014.