sensors

MDPI

Article

Online Calibration of Inertial Sensors Based on Error Backpropagation

Vojtech Simak, Jan Andel, Dusan Nemec * and Juraj Kekelak

Department of Control and Information Systems, Faculty of Electrical Engineering and Information Technology, University of Žilina, 010 26 Žilina, Slovakia; vojtech.simak@uniza.sk (V.S.); andel.jano@gmail.com (J.A.); juraj.kekelak@feit.uniza.sk (J.K.)

* Correspondence: dusan.nemec@uniza.sk

Abstract: Global satellite navigation systems (GNSSs) are the most-used technology for the localization of vehicles in the outdoor environment, but in the case of a densely built-up area or during passage through a tunnel, the satellite signal is not available or has poor quality. Inertial navigation systems (INSs) allow localization dead reckoning, but they have an integration error that grows over time. Inexpensive inertial measurement units (IMUs) are subject to thermal-dependent error and must be calibrated almost continuously. This article proposes a novel method of online (continuous) calibration of inertial sensors with the aid of the data from the GNSS receiver during the vehicle's route. We performed data fusion using an extended Kalman filter (EKF) and calibrated the input sensors through error backpropagation. The algorithm thus calibrates the INS sensors while the GNSS receiver signal is good, and after a GNSS failure, for example in tunnels, the position is predicted only by low-cost inertial sensors. Such an approach significantly improved the localization precision in comparison with offline calibrated inertial localization with the same sensors.

Keywords: global satellite navigation; inertial sensors; online calibration; error backpropagation

check for updates

Citation: Simak, V.; Andel, J.; Nemec, D.; Kekelak, J. Online Calibration of Inertial Sensors Based on Error Backpropagation. *Sensors* **2024**, *24*, 7525. https://doi.org/10.3390/s24237525

Academic Editor: José Luis Lázaro-Galilea

Received: 18 October 2024
Revised: 15 November 2024
Accepted: 24 November 2024
Published: 25 November 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The localization of mobile objects is an integral part of modern life. Global navigation satellite systems (GNSSs) [1] are most often used for localization in the outdoor environment. However, they have their limitations, especially due to bad weather conditions, densely built-up areas (e.g., larger cities), or when passing through tunnels. Therefore, mobile object localization systems need to be a combination of multiple localization sources. A common approach is to combine GNSSs with an inertial measurement unit (IMU), which contains inertial sensors (accelerometer, gyroscope), and, optionally, other sensors (e.g., magnetic compass). Navigation-grade inertial navigation systems (INSs) are usually very expensive since they contain optical gyroscopes due to their high accuracy. Another possibility is the use of LIDAR technology [2] or cameras [3]. For localization in buildings or halls, GNSS localization may be combined with localization based on the measurement of relative signal strength (RSSI) between multiple fixed transmitters and a mobile receiver, or vice versa [4,5]. The most frequently used wireless technologies are BLE, Wi-Fi, or proprietary UVW (Ultra-Wide Band) protocols. The development and rapid deployment of low-cost MEMS IMU units that are integrated into wearable electronics opens new possibilities for more accurate position prediction, even for small and affordable devices. However, the biggest disadvantage is the inaccuracy of MEMS IMU sensors, whose errors (noise, bias) are greater than that of optical IMUs, and it is necessary to calibrate them almost continuously.

When combining different sensors, we must solve the problem of sensor fusion. The Kalman filter or complementary filter is most often used for the fusion [6]. Many articles have discussed the fusion of readings from GNSSs and IMUs that use traditional

mathematical methods based on complementary filters and Kalman filters [6–12]. The EKF and its derivatives is the de facto standard approach for sensor fusion, but there are a few disadvantages as follows:

- The model of the system needs to be known (including its parameters).
- The EKF assumes the sensors' noise is normally distributed.
- The vanilla version of the EKF assumes the noise covariance matrices are constant and a priori known. For time-variant systems, this issue can be addressed by an adaptive EKF, which estimates the covariance of inputs from previous samples; see, e.g., [13,14].
- The vanilla EKF does not handle sensor bias well. The augmented EKF appends unknown parameters to the state vector (seem e.g., [15]); hence, it can be used to estimate and compensate for input bias.
- For highly non-linear systems, the sensor noise needs to be reasonably low since the EKF is a first-order linear approximation. Otherwise, the filter becomes unstable. This issue can be partially addressed by the error state extended Kalman filter (ES-EKF), which does not estimate the state directly but estimates only the correction of the state with respect to the incrementally predicted value, which can be less non-linear [16].

In real-world scenarios, the above conditions are usually not met. Especially, in continuous long-term applications, the noise characteristics of input sensors and the parameters of the system may change over time due to degradation, malfunction, and external stimuli. Especially, changes in the sensors' calibration constants have a significant negative impact on the overall performance of the localization system.

We may also perform data fusion through advanced methods using artificial intelligence (AI) [17,18]. There are many approaches in which we can combine traditional methods with AI. In such an approach, the adaptive extended Kalman filter (AEKF) is most often used for data fusion, which is adapted based on the output of a trained neural network [19]. However, we can also use neural networks themselves for data fusion. The use of recurrent neural networks, specifically gated recurrent units (GRUs) and long short-term memory units (LSTMs) [20], has proven to be the most effective among the neural-network-based data fusion techniques. There are also approaches via traditional convolution neural networks (CNNs), which proved to be effective only for determining the direction of rotation, but the input was uncalibrated data [21]. The main drawback of AI-based sensor fusion is its low predictability; hence, we cannot guarantee the behavior of the system for all possible inputs. The main challenge of sensor fusion is the estimation of quality for each individual sensor subsystem and the optimal combination of their readings by decreasing the influence of low-quality sources in favor of higher-quality sources. The first GNSS was a GPS, which reached tens of meters of accuracy. With the deployment of other navigation satellite systems (GLONASS, Galileo, BeiDou) and their supporting SBAS systems, accuracy has increased, especially with modern dual-band GNSS receivers. GNSS precision can be further improved by multiple approaches, such as troposphere estimation [22], obtaining the clock offset [23], or we may use a distributed localization system containing several GNSS receivers combined using a linear Kalman filter (KF) [24].

Before the data fusion itself, it is necessary to eliminate the systematic errors (gain/bias deviations) of the input, commonly by the calibration of the input sensors. The Zero Velocity Potential Update (ZUPT) algorithm is most often used to calibrate the gyroscope, which calibrates the gyroscope offset whenever the IMU is motionless [25]. Such an algorithm is used especially in wearable electronics where the device spends a lot of time in idle mode. To calibrate the accelerometer, turning the object to multiple sides is often used, but this approach is not always possible and cannot be performed continuously. There are several approaches to calibrating a magnetometer. The most common is through the search for an ellipsoid. We can also use this approach in the case of an accelerometer [25–27]. Some approaches solve the ellipsoid problem via maximum likelihood (ML) estimation [28]. Some calibration algorithms estimate also the misalignment between the axes of different sensors [29,30]. To estimate the attitude (rotation) of the vehicle, we may use multichannel phase-locked GNSS receivers [31,32].

There are several possible approaches to calibration and, subsequently, to data fusion, and each of them increases the accuracy of the localization of mobile objects, especially in the event of a GNSS communication failure. The best approach is probably to use a combination of several approaches. It is advantageous to use the EKF for data fusion, but its adaptation should be solved through a recurrent neural network that looks for anomalies in the data. The calibration of the sensors should be solved through the knowledge of movement with a good signal from GNSSs. A large amount of data is needed to train a recurrent network that would adapt the EKF. We focused on the calibration of the sensors based on the knowledge of the motion vector through AI tools. We designed and built a module for logging data from the vehicle, and we subsequently processed and determined the accuracy of position prediction through various approaches of sensor fusion and calibration.

2. Data Logger

We designed and built a module for logging data from a passenger vehicle. The module has several sensors, specifically, a MEMS IMU unit, a barometer, and a GNSS receiver. The main microcontroller is ESP32, which has a Wi-Fi and Bluetooth wireless connection. It detects information from the car's control unit via the ODBII connector via Bluetooth wireless communication. The module logs all data on the SD card to a CSV file. After arriving at a certain location, it connects to the Wi-Fi network and sends the measured data to the server for further processing. The module was described in more detail in [33,34]. The ground-truth location has been captured by SPAN-CPT from NovAtel, which is a navigation-grade GPS+IMU system incorporating fiber optic gyroscopes (FOGs) and MEMS accelerometers, which are tightly coupled with a GNSS receiver. We placed the data logger module on the dashboard in a 3D-printed box, aligned with the axes of the vehicle (Figure 1). The location on the dashboard is not optimal since there is the structure of the vehicle under the module, which strongly deforms Earth's magnetic field near the logger. On the other hand, this setup is very close to the real situation in the vehicle when the user mounts the navigation hardware on the dashboard or windshield. We placed the SPAN-CPT system above the rear axle of the vehicle and configured it for our purposes. The GNSS receiver antenna for the SPAN-CPT system was placed on the roof rails of the passenger car (see Figure 2).

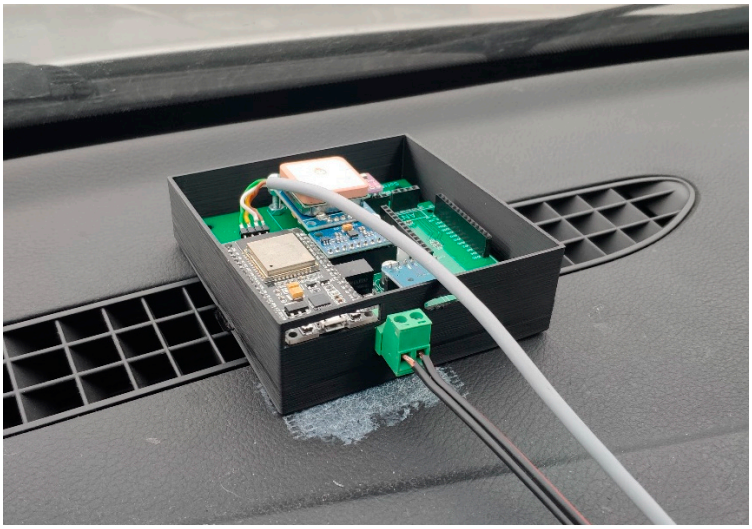


Figure 1. Integration of the module into a passenger vehicle.



Figure 2. GNSS antenna placed on a passenger vehicle.

We logged data during normal routes of a passenger vehicle and also routes directly determined for the purpose of data logging. We have laid out a route in which there are underground objects, such as garages and highway tunnels, in which a failure of GNSS communication is guaranteed. In total, we completed 12 trials, from which we had enough data to analyze several algorithms.

3. Data Fusion Through AEKF

When navigating in 3D space, it is important to set the coordinate system in which the location of the vehicle is determined. In GNSS localization, each position is represented by latitude, longitude, and altitude, which is the height above the mean sea level (MSL). However, when using INSs, a local world Cartesian (orthonormal) x, y, and z coordinate system oriented tangentially to the normalized Earth's surface is used. The origin (0, 0, 0) of the local world coordinate system corresponds to the initial position of the vehicle, and the curvature of the Earth is neglected. We can use this approach only in the case when mobile objects move at a low speed, or for a short time; hence, the traveled distance is negligible with respect to the Earth's curvature radius. The body's frame of reference is oriented according to NED convention (x—forward, y—right, z—down).

Single GNSS receiver estimates only the position but not the attitude (rotation) of the vehicle. If the vehicle is moving, we can estimate the rotation around the z-axis (yaw-γ) and the body's y-axis (pitch-β) from the GNSS data, while the body's x-axis (Roll-α) rotation is unknown. INS systems use gyroscopes and accelerometers, hence providing the measurement of angular velocity and linear acceleration in all three axes of the vehicle's body.

The gyroscope measures rotational acceleration in individual axes. It does not matter where it is located on the mobile object, as the rotation of the object is the same everywhere. We can calculate the changes of Euler angles α, β, γ from the angular velocity vector $[\omega_x, \omega_y, \omega_z]$ measured by a gyroscope using the transformation (1). This approach is suitable, especially for small angles and a short time Δt [35].

$$\begin{bmatrix} \Delta\alpha \\ \Delta\beta \\ \Delta\gamma \end{bmatrix} = \begin{bmatrix} 1 & \frac{\sin(\alpha)\sin(\beta)}{\cos(\beta)} & \frac{cos(\alpha)sin(\beta)}{\cos(\beta)} \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \frac{\sin(\alpha)}{\cos(\beta)} & \frac{\cos(\alpha)}{\cos(\beta)} \end{bmatrix} \cdot \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \Delta t \quad (1)$$

If the sensor unit is not accelerating, it only measures gravitational acceleration and noise, which can determine the direction of the center of the Earth. After subtracting the gravitational acceleration from the readings of the accelerometer and subsequent integration, we obtain the speed of the object; and after integrating the angular velocity measured by gyroscope, we obtain a rotation in individual axes. From the given values,

we can calculate the direction vector, which, after integration, creates the traveled path. Theoretically, we may read the value of the gravitational acceleration from the accelerometer data, but we must know the attitude of the mobile object and its linear acceleration. The measured value of the accelerometer is expressed by relation (2), where $\boldsymbol{g}$ is the gravitational acceleration, $\boldsymbol{a}_0$ is the acceleration of the vehicle, and $\boldsymbol{a}_\mathrm{n}$ is the noise, which also includes the vibrations of the vehicle. All vectors are measured in the local coordinates of the vehicle. Due to the large noise and the strong effect of gravitational acceleration, the accelerometer is usually used to detect downward direction, hence estimating the roll and pitch of the vehicle. The rotation around the vertical (z) axis does not affect accelerometer; hence, the yaw angle cannot be determined.

$$\boldsymbol{a} = -\boldsymbol{g} + \boldsymbol{a}_0 + \boldsymbol{a}_n \tag{2}$$

The magnetometer measures the Earth's magnetic field, which, however, is affected by the surrounding environment, so it is necessary to calibrate the magnetometer. Knowing the location of a mobile object, we can determine the inclination and declination (Figure 3) in the area, thanks to which we can determine the geographical north from the estimated magnetic north direction. The magnetometer is not sensitive to the rotation of the object around the axis parallel to the magnetic vector. By fusing the data from the magnetometer and the accelerometer, we can more accurately obtain information about the rotation of the vehicle.

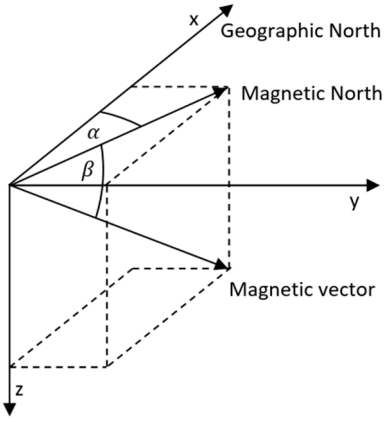


Figure 3. Inclination and declination of the magnetic field.

We can estimate the speed of a mobile object from an accelerometer, but, as mentioned, this method is highly inaccurate due to the vibrations of the vehicle. For a ground vehicle, it is more precise to obtain the speed through an odometer. We calculate the direction vector of the mobile object using Equations (3)–(5). First, we calculate the rotation matrix ($\boldsymbol{R}$) from the Euler angles. Symbols c_x and s_x are shorthand for $\cos(x)$ and $\sin(x)$, respectively. The forward direction vector $\boldsymbol{n_w}$ in world coordinates is simply the first column of the rotation matrix. Subsequently, we multiply the direction vector by the scalar speed of the mobile object (v). By the numerical integration of the velocity vector in the world coordinates, we obtain the position of the vehicle ($\boldsymbol{D}$) (6).

$$\boldsymbol{R} = \begin{bmatrix} c_\beta c_\gamma & s_\alpha s_\beta c_\gamma - c_\alpha s_\gamma & c_\alpha s_\beta c_\gamma + s_\alpha s_\gamma \\ c_\beta s_\gamma & s_\alpha s_\beta s_\gamma + c_\alpha c_\gamma & c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma \\ -s_\beta & s_\alpha c_\beta & c_\alpha c_\beta \end{bmatrix} \tag{3}$$

$$\boldsymbol{n}_w = \boldsymbol{R} \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{4}$$

$$\boldsymbol{v}_w = v\boldsymbol{n}_w \tag{5}$$

$$\boldsymbol{D} = \sum \boldsymbol{v}_w \Delta t \tag{6}$$

The above equations assume that the IMU axes are aligned with the vehicle's axes. The process of compensating for the misalignment of individual sensors will be described in the next section.

We may compensate for the altitude error (z coordinate) using a barometric sensor. It senses the atmospheric pressure, which is directly proportional to the altitude. The conversion between altitude and atmospheric pressure is carried out using the well-known formula as follows:

$$h^{bar} = \frac{RTln\left(\frac{P}{P_0}\right)}{-gm} \tag{7}$$

where P_0 is the sea level pressure, P is the gauge pressure, m is the molecular weight, g is the gravitational acceleration, R is the universal gas constant, and T is the thermodynamic temperature. Some atmospheric pressure sensors directly output altitude, while the calculation is carried out directly by the sensor's built-in CPU. Such a reading of the altitude contains a non-constant offset, as the atmospheric pressure at sea level varies over time.

We can use the EKF to fuse data from the IMU unit. Figure 4 shows the EKF block diagram where the input is data from the IMU unit and the output is Euler angles. During the prediction cycle, the data from the gyroscope are added according to (1). During the update process, based on the prediction, α, β, and γ are calculated, which should be the values from the accelerometer and magnetometer; they are subtracted from the real data, which creates an error vector.

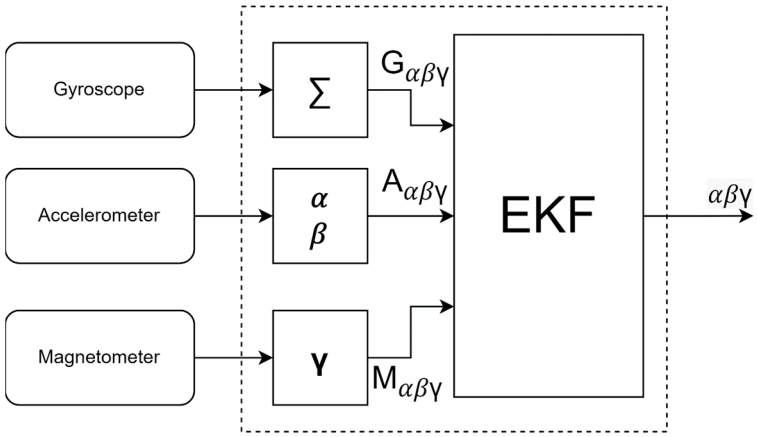


Figure 4. Extended Kalman filter for the IMU unit.

Prediction phase of the EKF:

$$\hat{\boldsymbol{s}}_k = f(\boldsymbol{s}_{k-1}, \boldsymbol{u}_k) \tag{8}$$

$$\begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix}_{k+1} = \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix}_{k} + \Delta t \begin{bmatrix} 1 & \frac{s_\alpha s_\beta}{c_\beta} & \frac{c_\alpha s_\beta}{c_\beta} \\ 0 & c_\alpha & -s_\alpha \\ 0 & \frac{s_\alpha}{c_\beta} & \frac{c_\alpha}{c_\beta} \end{bmatrix} \cdot \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{9}$$

$$\hat{P}_k = F_k P_{k-1} F_k^T + G_k U_k G_k^T \tag{10}$$

$$F_k = \frac{\partial f(\boldsymbol{s}_k, \boldsymbol{u}_k)}{\partial \boldsymbol{s}_k};\ G_k = \frac{\partial f(\boldsymbol{s}_k, \boldsymbol{u}_k)}{\partial \boldsymbol{u}_k} \tag{11}$$

The input $\boldsymbol{u}_k$ of the prediction cycle is the data from the gyroscope, the angular velocity vector, and the state $\boldsymbol{s}_k$ of the filter contains all Euler angles. Matrices F and G are partial derivatives of the function f in Formula (8). P is a 3×3 matrix representing the

covariance of the state vector $\boldsymbol{s}$. The matrix U determines the variance of the input data of the gyroscope.

Updated phase of the EKF:

$$\boldsymbol{e}_k = \boldsymbol{q}_k - h(\hat{\boldsymbol{s}}_k) \tag{12}$$

$$\begin{bmatrix} ea_x \\ ea_y \\ ea_z \\ eM_x \\ eM_y \\ eM_z \end{bmatrix} = \begin{bmatrix} qa_x \\ qa_y \\ qa_z \\ qM_x \\ qM_y \\ qM_z \end{bmatrix} - \begin{bmatrix} gs_\beta \\ gc_\beta s_\alpha \\ gc_\beta c_\alpha \\ Mc_\beta c_\delta c_\gamma c_\theta - Ms_\beta s_\theta + Mc_\beta c_\theta s_\delta s_\gamma \\ Mc_\beta s_\alpha s_\theta - Mc_\delta c_\theta \left(c_\alpha s_\gamma - c_\gamma s_\alpha s_\beta\right) + Mc_\theta s_\delta \left(c_\alpha c_\gamma + s_\alpha s_\beta s_\gamma\right) \\ Mc_\alpha c_\beta s_\theta + Mc_\delta c_\theta \left(s_\alpha s_\gamma + c_\alpha c_\gamma s_\beta\right) - Mc_\theta s_\delta \left(c_\gamma s_\alpha - c_\alpha s_\beta s_\gamma\right) \end{bmatrix} \tag{13}$$

$$S_k = H_k P_k H_k^T + R_k \tag{14}$$

$$K_k = H_k P_k S_k^{-1} \tag{15}$$

$$\boldsymbol{s}_k = \hat{\boldsymbol{s}}_k + K_k \boldsymbol{e}_k \tag{16}$$

$$P_k = (I - K_k H_k)\hat{P}_k \tag{17}$$

$$H_k = \frac{\partial h(\boldsymbol{s}_k)}{\partial \boldsymbol{s}_k} \tag{18}$$

During the update, the error vector $\boldsymbol{e}$ is first calculated, which is equal to the difference between the measured values from the accelerometer and magnetometer sensors and the calculated value from the prediction of the Euler angles from the knowledge of inclination-θ and declination-δ. The matrix R determines the noise variance of individual sensors. The K matrix determines the Kalman gain. The output of the update is the most probable value of the Euler angles from the IMU unit.

To predict the trajectory, we need to know the speed of the mobile object. We can obtain it from the odometer or accelerometer after subtracting the gravitational acceleration. We can use a linear Kalman filter for data synthesis. We assume that the mobile object and the accelerometer have aligned axes and that the mobile object can only accelerate in the x-axis direction. From the KF output, we can calculate the effect of gravitational acceleration in the direction of the x-axis. In prediction, the input is the acceleration sensor, and in the update, the error is corrected using velocity v_k^{odo} obtained from odometer data. Figure 5 shows the block diagram of the described KF.

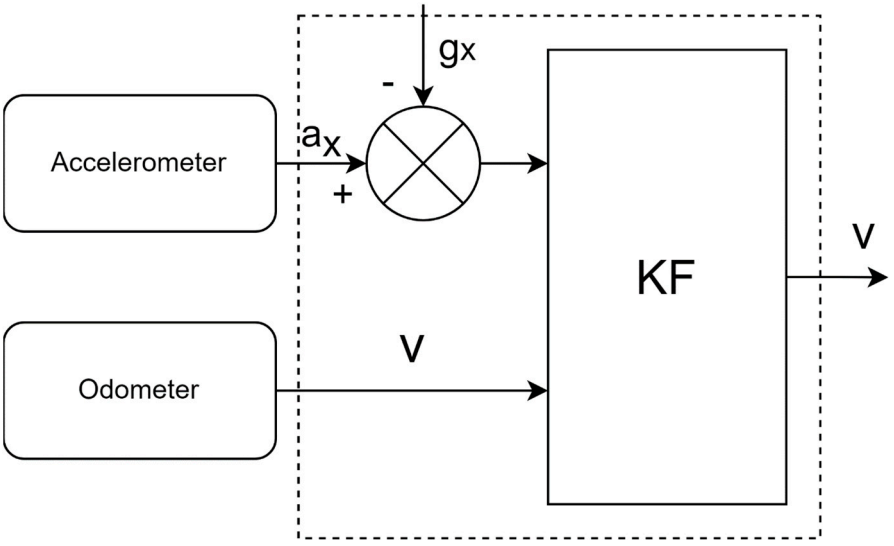


Figure 5. Kalman filter for velocity update.

Prediction:

$$\hat{x}_k = Ax_{k-1} + Bu_k \tag{19}$$

$$\hat{v}_k = 1 \cdot v_{k-1} + \Delta t \cdot (a_x - g_x) \tag{20}$$

$$\hat{P}_k = A_k P_{k-1} A_k^T + Q_k = P_{k-1} + Q_k \tag{21}$$

Update:

$$\delta v_k = v_k^{odo} - H \cdot \hat{v}_k \tag{22}$$

$$K_k = \frac{\hat{P}_k H_k^T}{H_k \hat{P}_k H_k^T + R_k} \tag{23}$$

$$v_k = \hat{v}_k + K_k \delta v_k \tag{24}$$

$$P_k = (I - K_k H_k) \hat{P}_k \tag{25}$$

Such a KF is simple and easy to calculate, and all values are scalar. The Q value expresses the noise variance of the accelerometer and R the noise variance of the odometer. The output of the synthesis is the most likely speed of the mobile object. Knowing Euler angles and speed, we can calculate the direction vector through the already described Equations (3) and (5). By adding the direction vector, we obtain the vehicle's traveled path (6). With this approach, we created the INS inertial navigation system.

We also used the KF for the fusion between the INS and GNSS. The system is linear, so we did not have to use the extended Kalman filter. We also added values from the barometer to the input. The output from the KF was the most likely position of the mobile object in global coordinates. The KF equations are shown in (26)–(33). Matrices A and B are unitary, so during prediction, data from the INS are only added to the currently predicted value. During the update process, an error vector $\boldsymbol{e}$ is calculated, in which there are four values, e_x, e_y, e_z, and height correction e_h. The Kalman gain matrix K will have the form 4×3.

Prediction:

$$\hat{\mathbf{s}}_k = A_k \boldsymbol{s}_{k-1} + B_k \boldsymbol{u}_k \tag{26}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{k-1} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \Delta x_{INS} \\ \Delta y_{INS} \\ \Delta z_{INS} \end{bmatrix} \tag{27}$$

$$\hat{P}_k = A_k P_k A_k^T + Q_k \tag{28}$$

Update:

$$\boldsymbol{e}_k = \boldsymbol{q}_k - H \cdot \hat{\mathbf{s}}_k \tag{29}$$

$$\begin{bmatrix} e_x \\ e_y \\ e_z \\ e_h \end{bmatrix}_k = \begin{bmatrix} x^{GNSS} \\ y^{GNSS} \\ z^{GNSS} \\ h^{bar} \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{30}$$

$$K_k = \frac{\hat{P}_k H_k^T}{H_k \hat{P}_k H_k^T + R_k} \tag{31}$$

$$\boldsymbol{s}_k = \hat{\mathbf{s}}_k + K_k \boldsymbol{e}_k \tag{32}$$

$$P_k = (I - K_k H_k) \hat{P}_k \tag{33}$$

In the described way, we can determine the most likely position of the vehicle, but we must have knowledge of the noise variance of the sensors before running the algorithm, and all sensors must be calibrated. We can perform sensor calibration using traditional methods. However, our approach is to calibrate the sensors based on the position from the GNSS at a time when the GNSS signal is good. Calibration takes place through the backpropagation of the error through the entire algorithm. At the output of the sensors,

we added neurons that serve as a calibration matrix, and their parameters are adjusted based on the backpropagation of the error. With such an approach, we guarantee a constant calibration of the sensors, and in the event of a GNSS failure, the position is predicted based on the INS. Kalman filter functions are written as classes under the "PyTorch" library, which uses tensors through which it can calculate the backpropagation of the error. Individual neural networks are placed between the sensors and the position prediction calculation. The neural network for the gyroscope, magnetometer, and accelerometer consists of three neurons (Figure 6), the inputs of which are connected to each neuron, and thanks to which, we can obtain dependencies between individual axes; for example, if the sensor is rotated not completely in the direction of travel. Such a calculation creates a 4×3 calibration matrix (34) and (35). For one neuron, we obtain only offset and gain. We set the initial parameters of individual neurons to $w_{\mathrm{nk}} = 1$ and $b_{\mathrm{n}} = 0$ so that the sensor readings are unaltered. This approach considers only first-order calibration. Using a more complex neural network, we could also compensate for certain non-linearity, but the training is more computationally demanding, and the calibration would take longer to converge. The block diagram of the described method is shown in Figure 7.

$$o_1 = b_1 + \sum_{i=1}^{n} w_{1n} i_n \tag{34}$$

$$\begin{bmatrix} o_1 \\ o_2 \\ o_3 \end{bmatrix} = \begin{bmatrix} w_{11} & w_{21} & w_{31} & b_1 \\ w_{12} & w_{22} & w_{32} & b_2 \\ w_{13} & w_{23} & w_{33} & b_3 \end{bmatrix} \cdot \begin{bmatrix} i_1 \\ i_2 \\ i_3 \\ 1 \end{bmatrix} \tag{35}$$

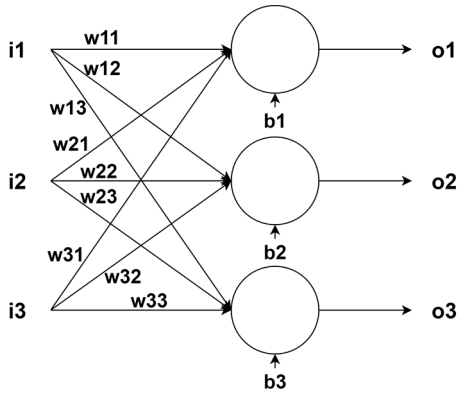


Figure 6. Illustration of a neural network.

The algorithm processes the data if the HDOP value from the GNSS receiver is lower than 10, which represents barely acceptable quality of a GNSS lock. The backpropagation calculation is applied through the loss function. Through optimization, new neuron constants are recalculated, symbolizing our calibration constants. An error vector is calculated between the predicted position and the value from the GNSS. We use the criterion for the loss error function "MSEloss", which calculates the quadratic error. We use the Adam algorithm [36] as an optimization method with a learning rate of 0.0001. The algorithm changes the calibration constants during the runs. The longer we have a good signal from the GNSS, the more accurate the calibration constants the algorithm will find.

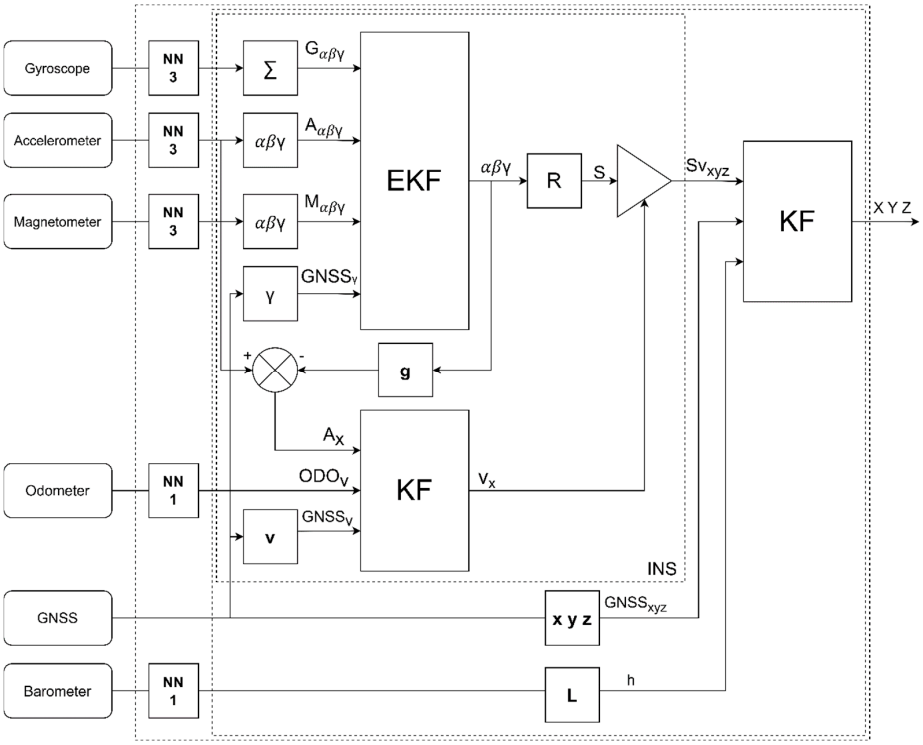


Figure 7. Fusion of input data through neural networks.

4. Achieved Results and Discussion

We wanted to compare the evaluation of the achieved results against the SPAN-CPT system. In Figure 8, we can see a comparison of the vehicle speed during a GNSS signal failure. The algorithm adjusted the calibration constants for the accelerometer and odometer that entered the KF. In Figure 9, we can see the elevation comparison. During the GNSS outage, altitude was predicted based on INS and barometer data. We compared the data with the reference sample from the SPAN-CPT system. We displayed the resulting data from the prediction on maps and compared it to the position from SPAN-CPT. As we can see in Figure 10, position prediction from the SPAN-CPT system was not accurate, even though it contained high-quality inertial sensors.

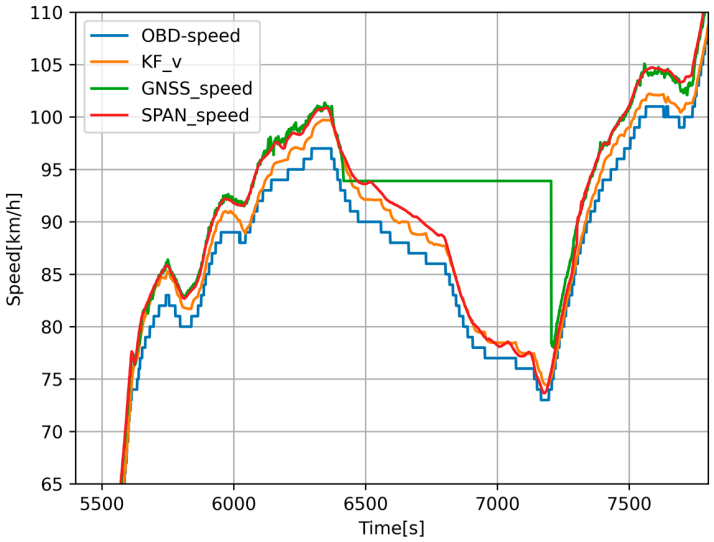


Figure 8. Comparison of estimated speed from odometry (blue), low-cost GNSS (green), navigation-grade GNSS (red), and Kalman filter (orange).

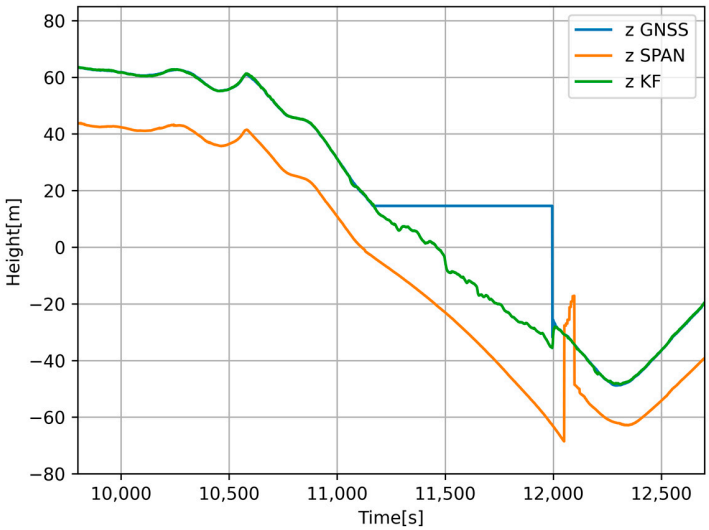


Figure 9. Comparison of estimated relative elevation from low-cost GNSS (blue), navigation-grade GNSS (orange), and Kalman filter (green).

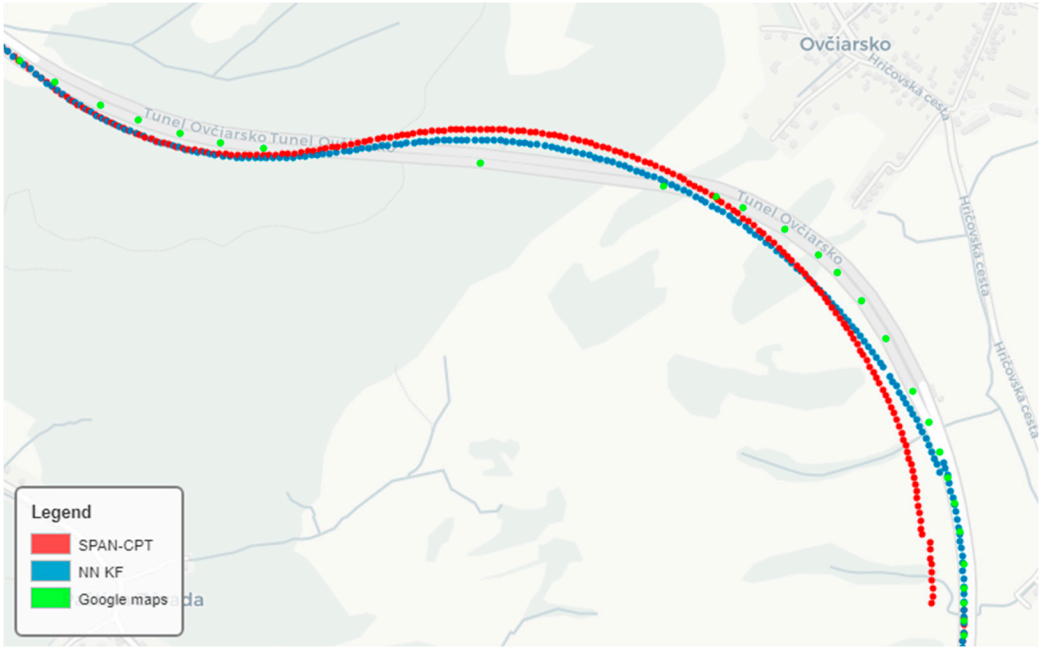


Figure 10. GNSS signal failure in tunnels (red), calibration through error backpropagation (blue).

Hence, the position prediction from SPAN-CPT dead reckoning was not accurate enough to be considered as a reference value during an outage. That is why we decided to compare the data against the route generated from the map documents. But, we ran into a problem, where the maps of the tunnels from several sources did not correspond to reality since they do not match with each other. Online maps may show only a rough estimate of the underground road trajectory. Each of the tested map documents had the trajectory of the tunnels drawn slightly differently. The most accurate was the maps provided by Google. We have compared Google maps with the official maps provided by the Slovak Road Administration agency [37]. Hence, we generated a trajectory along which we walked and compared the data. We created a function that calculates the tangent to the trajectory from map data. The trajectory is shown in Figure 10, and the summary statistics are in Table 1. However, the comparison is not completely accurate, as the generated path was not the real road along which the vehicle traveled; also, when passing certain right-angled turns, the trajectory was curved, while in the generated route, the turn was sharp without

shortening the trajectory. We have compared the errors of the proposed neural-calibrated Kalman filter system with the same system, where the neural networks were replaced by standard calibration parameters (gain, bias). The system was assembled in the same way, but at the input, there were no neural networks but calibration matrices. We have calibrated the gyroscope through the ZUPT algorithm (bias calibration was performed whenever the vehicle was stationary). Calibration of the magnetometer was performed based on the knowledge of the traveled path [33]. We did not calibrate the accelerometer in place because it was not possible to rotate the object (passenger vehicle) in all directions.

Table 1. Comparison of positioning accuracy.

	SPAN-CPT [m]	KF [m]	KF with Backpropagation [m]
Mean average error	6.12	9.19	4.14
RMSE	14.07	34.14	8.92

The results in Figures 11 and 12 show that calibration through error backpropagation is feasible and has better results than traditional methods, including the navigation grade GNSS + INS system. The proposed method takes advantage of the readings from the odometer; hence, it is primarily applicable to wheeled vehicles. Although data processing took place offline, the entire algorithm works in online mode (processing data sample by sample, without the need to know all future values), and it is possible to integrate it into the software of an onboard signal processing microcomputer. The main disadvantage is that the algorithm requires slightly more computing power compared to traditional approaches. Nowadays, however, this is not such a problem. The main advantage of the algorithm is its adaptability and continuous sensor calibration, thanks to which we can use low-cost sensors with lower accuracy and predict the position of the vehicle during GNSS outages more accurately than expensive systems.

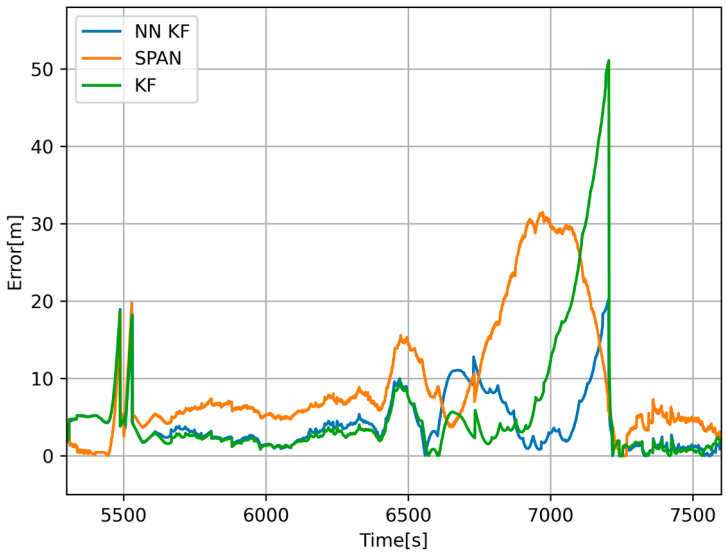


Figure 11. Error of the estimated position from NN-calibrated KF (blue), navigation-grade GNSS+INS (orange), and offline-calibrated KF (green) with respect to the position obtained from map documents while traveling through the first tunnel.

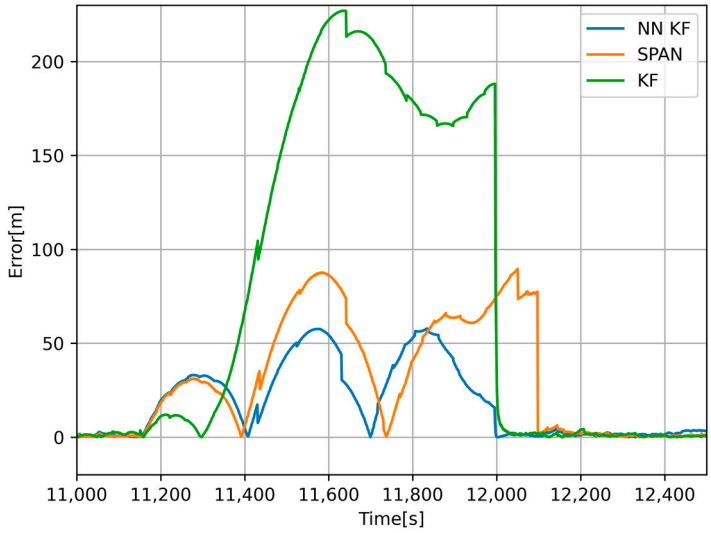


Figure 12. Error of the estimated position from NN-calibrated KF (blue), navigation-grade GNSS+INS (orange), and offline-calibrated KF (green) with respect to the position obtained from map documents while traveling through the first tunnel.

5. Conclusions

More accurate localization of mobile objects is still an open topic. GNSS systems or their combination with INSs are most often used for localization. However, INS systems require precisely calibrated sensors for the calculation to be correct. There are several options for calibrating the sensors within INS systems. We focused on the possibilities of using AI tools for the task, while we proposed an algorithm that consists of three KFs estimating the position and attitude of the vehicle. Instead of sensor calibration constants, we connected neural networks (NNs) between the filter input and the sensors. The parameters of the NNs are adjusted using error backpropagation. Backpropagation always takes place when the GNSS signal is good (HDOP value is below a threshold). In this way, the algorithm can update the calibration constants of individual sensors and at the same time find dependencies between the axes, which calibrates the inaccuracies of the axis alignment. We verified the algorithm on our own measured data from a passenger vehicle and verified the values against the same algorithm via the KF but with calibration via traditional methods. At the same time, we also compared the prediction with the navigation-grade GNSS+INS system. We expected such a system to be a reference system, but it turned out that after the GNSS failure, the position prediction had an increasing error, which was significant. Therefore, we compared the position estimate against the route generated from the online map documents. The results show that our algorithm achieved the best results.

The result of this work is a clear conclusion that artificial intelligence tools can improve the prediction of the position of mobile objects, especially in combination with traditional mathematical procedures that solve the fusion between sensors. Artificial intelligence tools can then be used to calibrate the sensors, adapt the KF parameters, or detect anomalies in the data from the sensors and then remove them from the fusion process. Such approaches are especially suitable for affordable MEMS sensors, whose calibration requirement strongly affects the accuracy of position prediction.

Future work may focus on implementing the proposed approach to different types of vehicles (e.g., aerial drones) that cannot take advantage of the wheel odometry. It is also a viable option to test the ES-EKF (error-state EKF) instead of the EKF, which may improve the precision of the localization even further.

Author Contributions: Conceptualization, V.S. and J.A.; methodology, D.N.; software, J.A.; validation, V.S., J.A. and J.K.; formal analysis, J.K.; data curation, J.A.; writing—original draft preparation, J.A., V.S. and D.N.; funding acquisition, D.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been supported by the project APVV SK-IL-RD-23-0002: Advanced Localization Sensors and Techniques for Autonomous Vehicles and Robots.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are available upon request from the corresponding author.

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

References

1. Cernohorsky, J.; Jandura, P.; Mach, O. Mobile robot localization and object description. In Proceedings of the 2017 18th International Carpathian Control Conference (ICCC), Sinaia, Romania, 28–31 May 2017; pp. 503–506. [CrossRef]
2. He, G.; Yuan, X.; Zhuang, Y.; Hu, H. An Integrated GNSS/LiDAR-SLAM Pose Estimation Framework for Large-Scale Map Building in Partially GNSS-Denied Environments. *IEEE Trans. Instrum. Meas.* **2020**, *70*, 7500709. [CrossRef]
3. Wen, W.; Bai, X.; Zhang, G.; Chen, S.; Yuan, F.; Hsu, L.-T. Multi-Agent Collaborative GNSS/Camera/INS Integration Aided by Inter-Ranging for Vehicular Navigation in Urban Areas. *IEEE Access* **2020**, *8*, 124323–124338. [CrossRef]
4. Dinh-Van, N.; Nashashibi, F.; Thanh-Huong, N.; Castelli, E. Indoor Intelligent Vehicle localization using WiFi received signal strength indicator. In Proceedings of the 2017 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM), Nagoya, Japan, 19–21 March 2017; pp. 33–36. [CrossRef]
5. Greco, G.; Lucianaz, C.; Bertoldo, S.; Allegretti, M. Localization of RFID tags for environmental monitoring using UAV. In Proceedings of the 2015 IEEE 1st International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI), Turin, Italy, 16–18 September 2015; pp. 480–483. [CrossRef]
6. Islam, A.; Saha, S. Loosely Coupled GPS/INS Integrated Navigation System based on Kalman Filter and Complementary Filter for Aircraft. In Proceedings of the 2017 2nd International Conference on Electrical & Electronic Engineering (ICEEE), Rajshahi, Bangladesh, 27–29 December 2017; pp. 1–4. [CrossRef]
7. Caron, F.; Duflos, E.; Pomorski, D.; Vanheeghe, P. GPS/IMU data fusion using multisensor Kalman filtering: Introduction of contextual aspects. *Inf. Fusion* **2006**, *7*, 221–230. [CrossRef]
8. Qi, H.; Moore, J. Direct Kalman filtering approach for GPS/INS integration. *IEEE Trans. Aerosp. Electron. Syst.* **2002**, *38*, 687–693. [CrossRef]
9. Wang, G.; Han, Y.; Chen, J.; Wang, S.; Zhang, Z.; Du, N.; Zheng, Y. A GNSS/INS Integrated Navigation Algorithm Based on Kalman Filter. *IFAC-PapersOnLine* **2018**, *51*, 232–237. [CrossRef]
10. Ndjeng, A.N.; Gruyer, D.; Glaser, S.; Lambert, A. Low cost IMU–Odometer–GPS ego localization for unusual maneuvers. *Inf. Fusion* **2011**, *12*, 264–274. [CrossRef]
11. M.G., D.; Arun, A. Analysis of INS Parameters and Error Reduction by Integrating GPS and INS Signals. In Proceedings of the 2018 International Conference on Design Innovations for 3Cs Compute Communicate Control (ICDI3C), Bangalore, India, 25–28 April 2018; pp. 18–23. [CrossRef]
12. Raveena, C.; Sravya, R.; Kumar, R.; Chavan, A. Sensor Fusion Module Using IMU and GPS Sensors For Autonomous Car. In Proceedings of the 2020 IEEE International Conference for Innovation in Technology (INOCON), Bangaluru, India, 6–8 November 2020; pp. 1–6. [CrossRef]
13. Zhou, W.; Qiao, X.-W.; Meng, F.; Zhang, H. Study on SINS/GPS tightly integrated navigation based on Adaptive Extended Kalman Filter. In Proceedings of the 2010 IEEE International Conference on Information and Automation, Harbin, China, 20–23 June 2010; pp. 2344–2347. [CrossRef]
14. Kailath, T.; Sayed, A.H.; Hassibi, B. *Linear Estimation*; Prentice Hall: Englewood Cliffs, NJ, USA, 2000.
15. Kang, J.-G.; Choi, W.-S.; An, S.-Y.; Oh, S.-Y. Augmented EKF based SLAM method for improving the accuracy of the feature map. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 3725–3731. [CrossRef]
16. T, M.H.; Akram, V.; Reddy, S.; Kn, M.G.; Ld, U.K. An Error-State Extended Kalman Filter Based State Estimation and Localization Algorithm for Autonomous Systems. In Proceedings of the 2023 International Conference on Smart Systems for applications in Electrical Sciences (ICSSES), Tumakuru, India, 7–8 July 2023; pp. 1–6. [CrossRef]
17. Noureldin, A.; El-Shafie, A.; Bayoumi, M. GPS/INS integration utilizing dynamic neural networks for vehicular navigation. *Inf. Fusion* **2010**, *12*, 48–57. [CrossRef]
18. Dai, H.-F.; Bian, H.-W.; Wang, R.-Y.; Ma, H. An INS/GNSS integrated navigation in GNSS denied environment using recurrent neural network. *Def. Technol.* **2020**, *16*, 334–340. [CrossRef]
19. Brossard, M.; Barrau, A.; Bonnabel, S. AI-IMU Dead-Reckoning. *IEEE Trans. Intell. Veh.* **2020**, *5*, 585–595. [CrossRef]

20. Tang, Y.; Jiang, J.; Liu, J.; Yan, P.; Tao, Y.; Liu, J. A GRU and AKF-Based Hybrid Algorithm for Improving INS/GNSS Navigation Accuracy during GNSS Outage. *Remote Sens.* **2022**, *14*, 752. [CrossRef]
21. Huang, J.; Huang, Z.; Chen, K. Combining low-cost Inertial Measurement Unit (IMU) and deep learning algorithm for predicting vehicle attitude. In Proceedings of the 2017 IEEE Conference on Dependable and Secure Computing, Taipei, Taiwan, 7–10 August 2017; pp. 237–239. [CrossRef]
22. Guo, J.; Yang, F.; Shi, J.; Xu, C. An Optimal Weighting Method of Global Positioning System (GPS) Troposphere Tomography. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2016**, *9*, 5880–5887. [CrossRef]
23. Kumar, N.A.; Rao, G.S.; Arasavali, N. Development of Advanced Extended Kalman Filter for Precise Estimation of GPS Receiver Position. In Proceedings of the 2019 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET), Chennai, India, 21–23 March 2019; pp. 213–216. [CrossRef]
24. Zhang, M.; Jia, X.; Yang, B.; Chi, X.; Peng, D. A GPS Positioning Algorithm Based on Distributed Kalman Filter Data Fusion with Feedback. In Proceedings of the 2018 37th Chinese Control Conference (CCC), Wuhan, China, 25–27 July 2018; pp. 7359–7363. [CrossRef]
25. Suresh, R.P.; Sridhar, V.; Pramod, J.; Talasila, V. Zero Velocity Potential Update (ZUPT) as a Correction Technique. In Proceedings of the 2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU), Bhimtal, India, 23–24 February 2018; pp. 1–8. [CrossRef]
26. Renaudin, V.; Afzal, M.H.; Lachapelle, G. Complete Triaxis Magnetometer Calibration in the Magnetic Domain. *J. Sens.* **2010**, *2010*, 967245. [CrossRef]
27. Vasconcelos, J.F.; Elkaim, G.; Silvestre, C.; Oliveira, P.; Cardeira, B. Geometric approach to strapdown magnetometer calibration in sensor frame. *IEEE Trans. Aerosp. Electron. Syst.* **2011**, *47*, 1293–1306. [CrossRef]
28. Wu, Y.; Shi, W. On Calibration of Three-Axis Magnetometer. *IEEE Sens. J.* **2015**, *15*, 6424–6431. [CrossRef]
29. Salehi, S.; Mostofi, N.; Bleser, G. A practical in-field magnetometer calibration method for IMUs. In Proceedings of the IROS Workshop on Cognitive Assistive Systems: Closing the Action-Perception Loop, Vilamoura, Portugal, 7–12 October 2012; pp. 39–44.
30. Li, X.; Li, Z. A new calibration method for tri-axial field sensors in strap-down navigation systems. *Meas. Sci. Technol.* **2012**, *23*, 105105. [CrossRef]
31. Henkel, P.; Berthold, P.; Kiam, J.J. Calibration of magnetic field sensors with two mass-market GNSS receivers. In Proceedings of the 2014 11th Workshop on Positioning, Navigation and Communication (WPNC), Dresden, Germany, 12–13 March 2014; pp. 1–5. [CrossRef]
32. Henkel, P. Calibration of Magnetometers with GNSS Receivers and Magnetometer-Aided GNSS Ambiguity Fixing. *Sensors* **2017**, *17*, 1324. [CrossRef] [PubMed] [PubMed Central]
33. Andel, J.; Šimák, V.; Kanálikova, A.; Pirník, R. GNSS Based Low-Cost Magnetometer Calibration. *Sensors* **2022**, *22*, 8447. [CrossRef] [PubMed]
34. Andel, J.; Šimák, V.; Škultéty, F.; Nemec, D. IoT-based Data Acquisition Unit for aircraft and road vehicle. *Transp. Res. Procedia* **2021**, *55*, 969–976. [CrossRef]
35. Nemec, D.; Janota, A.; Hrubos, M.; Simak, V. Intelligent Real-Time MEMS Sensor Fusion and Calibration. *IEEE Sens. J.* **2016**, *16*, 7150–7160. [CrossRef]
36. Kingma, D.; Ba, J. Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015; pp. 1–15.
37. Map of Road Network, Slovak Road Administration. Available online: https://www.cdb.sk/sk/Vystupy-CDB/Mapy-cestnej-siete-SR.alej (accessed on 16 January 2023).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.