

<https://doi.org/10.31891/csit-2026-3-7>

Viktor TKACHENKO

PhD student at the Department of Computer Science and Software Engineering
National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”
<https://orcid.org/0009-0002-8265-0367>
fkma13552@gmail.com

Eduard ZHARIKOV

Doctor of Technical Sciences, Professor,
Head of the Department of Computer Science and Software Engineering
National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”
<https://orcid.org/0000-0003-1811-9336>

Received: 26/07/2026
Accepted: 23/09/2026
Published: 30/09/2026

© Copyright
2026 by the author(s)



This is an Open Access article distributed under the terms of the [Creative Commons CC-BY 4.0](https://creativecommons.org/licenses/by/4.0/)

UDC 004.05

COMPARISON OF GPS/IMU INTEGRATION METHODS FOR REAL-TIME VEHICLE POSITIONING

This paper addresses the challenge of determining vehicle coordinates in real-time telemetry systems through the integration of Global Positioning System (GPS) and inertial measurement unit (IMU) data. A comprehensive analysis of existing sensor-fusion approaches enabled the formulation of the problem within the framework of embedded, resource-constrained real-time platforms and underscored the necessity of a systematic comparative evaluation of methods with fundamentally different computational and accuracy characteristics. Three sensor-fusion methods are considered: the complementary filter, which combines data through frequency-domain separation; the Extended Kalman Filter, which relies on Jacobian-based linearization of the nonlinear motion model; and the Unscented Kalman Filter, which applies a deterministic sigma-point transform to approximate nonlinear state propagation without linearization. The paper substantiates the choice of methodological foundations underlying each method, including the kinematic vehicle state model comprising geographic coordinates, velocity, heading angle, and sensor bias offsets, as well as the metrics used for their quantitative comparison — positioning RMSE, circular error probable, dead-reckoning drift during GPS outages, and update latency. The growing demand for reliable positioning under GPS-denied conditions, combined with the limited computational resources of embedded telemetry platforms and the nonlinear nature of vehicle kinematics, highlights the need to establish practical, hardware-aware criteria for filter selection. To address this, all three algorithms were implemented on a common hardware and software platform — a Raspberry Pi 5 single-board computer running Python 3.11 — and validated against real driving data collected on a 4.7 km route, with ground truth provided by an RTK GPS receiver. The paper presents experimental results demonstrating the relative effectiveness of the proposed methods in terms of positioning accuracy, robustness to GPS signal loss, and behavior under aggressive maneuvering. Additionally, it outlines practical recommendations for selecting an integration method depending on the computational constraints of the target embedded platform, ranging from microcontroller-based systems to single-board computers and DSP platforms.

Keywords: sensor fusion, Kalman filter, GPS/IMU integration, inertial navigation, dead reckoning, positioning accuracy, real-time telemetry, unscented transform, MEMS gyroscope, embedded systems.

Introduction

Real-time determination of a moving object's coordinates is a fundamental problem in modern navigation and telemetry systems. The task is relevant to both military applications and transportation technologies, where precise vehicle positioning enables trajectory analysis, lap and sector detection, comparison of the current run against a reference lap, and real-time prediction of time-delta indicators. The quality and update rate of position data directly determine the accuracy of all derived telemetry indicators.

Global Navigation Satellite Systems (GNSS) provide absolute positioning with a typical accuracy of 2–5 m (CEP50) for consumer-grade receivers. Despite its widespread use, GPS has several fundamental limitations: its update rate of 1–10 Hz is insufficient for tracking fast motion dynamics, the signal suffers significant degradation in dense urban environments, and it is completely unavailable indoors and in tunnels [1, 2]. These limitations make it necessary, when designing embedded systems and implementing edge computing, to integrate GPS with inertial sensors capable of providing continuous motion measurements at 100–1000 Hz.

Inertial measurement units (IMUs) based on micro-electro-mechanical systems (MEMS) have become widespread owing to their small size, low cost, and accuracy sufficient for navigation tasks [3]. At the same time, MEMS gyroscopes and accelerometers exhibit an inherent bias drift, which leads to accumulating positioning error during prolonged autonomous operation without correction from an external source [1]. Thus, neither data source alone is sufficient to ensure accurate positioning of the controlled object under all operating modes, which calls for an effective method of combining and processing sensor data from multiple sources.

The problem of GPS/IMU data integration has been studied for several decades. The classical approach is stochastic state estimation via the Kalman filter, whose theoretical foundations for navigation systems have been systematized in fundamental works [2, 3]. The linear Kalman filter requires system linearity, which is too restrictive a condition for vehicle kinematics. The Extended Kalman Filter (EKF) addresses this problem by linearizing nonlinear functions through Jacobian matrices and is the de-facto standard in most integrated GPS/INS systems [2, 4, 5].

Toledo-Moreo et al. [4] demonstrated the application of the EKF for integrating an IMU with differential GPS in transportation systems with increased reliability requirements. Wendel et al. [5] investigated a tightly coupled GPS/IMU integration method using time-differenced carrier-phase measurements, showing a substantial accuracy improvement under dynamic motion conditions. Angrisano et al. [6] systematized GNSS/inertial integration methods, classifying approaches by the degree of coupling between sensors.

A significant drawback of the EKF is the first-order linearization error, which reduces accuracy and increases the risk of filter divergence during sharp trajectory changes. The Unscented Kalman Filter (UKF), proposed by Julier and Uhlmann [7] and further developed by Wan and Van der Merwe [8], eliminates this drawback by using a deterministic sigma-point sampling scheme that provides second-order approximation without computing a Jacobian. A comparative analysis of nonlinear filtering methods for GPS/INS problems, conducted by Rhudy et al. [9], confirmed the superiority of the UKF over the EKF for nonlinear motion trajectories.

In parallel with Kalman-filter-based approaches, computationally efficient methods for combining (fusing) data from different sensors have been actively studied. Mahony et al. [10] proposed a nonlinear complementary filter on the SO(3) group that provides robust orientation estimation at minimal computational cost. Madgwick et al. [11] developed a gradient-descent-based IMU orientation estimation algorithm that is widely used in onboard systems. Coskun and Tuncer [12] demonstrated the feasibility of real-time EKF implementation on an STM32F4 microcontroller with a latency of under 1 ms.

Despite extensive theoretical treatment of the controlled-object positioning problem, a practical comparison of the complementary filter (CF), EKF, and UKF under real-world driving conditions, verified against an RTK reference, remains an open task. Open questions include algorithm behavior during GPS outages, robustness under aggressive maneuvering, and the computational resource cost of edge-computing implementation on onboard platforms [6, 12].

The aim of this work is a comparative analysis of three methods for combining GPS and IMU data — the complementary filter, the Extended Kalman Filter, and the Unscented Kalman Filter — for determining the coordinates of a controlled object (vehicle) in real-time telemetry systems. To achieve this aim, the following tasks are addressed: constructing a kinematic state model of the vehicle; implementing and calibrating the algorithms on a single hardware platform; and quantitatively comparing accuracy, behavior during GPS outages, and computational resource cost using real test data [4, 7, 8, 9].

Materials and Methods

2.1. Input Data and Sensor Configuration

Implementing a signal-fusion algorithm for real-time telemetry systems requires two principal sensor classes: a global navigation satellite system receiver and an inertial measurement unit. Each sensor is characterized by its own accuracy and update-rate properties, which necessitates their combined use.

The GPS receiver provides the absolute location coordinates of the vehicle in the WGS-84 format (latitude φ , longitude λ) at an update rate of 1–10 Hz, depending on the hardware implementation. The typical horizontal-positioning accuracy for consumer-grade u-blox M8/M9-class modules is 2–5 m (CEP50, without differential correction). In addition to coordinates, the receiver provides a velocity vector v_{GPS} , a heading angle ψ , and a signal-quality indicator (accuracy), which is used to filter noisy measurements. A major limitation of GPS is significant measurement jitter under dynamic conditions (sharp maneuvers, high speed), along with complete signal loss indoors and in tunnels.

An IMU based on MEMS technology (e.g., MPU-6050, ICM-42688-P) measures three linear acceleration components a_x, a_y, a_z (m/s²) and three angular velocity components $\omega_x, \omega_y, \omega_z$ (rad/s) at a rate of 100–1000 Hz. The high update rate allows fast motion dynamics to be tracked; however, MEMS gyroscopes are prone to gradual bias drift, which causes positioning error to accumulate during prolonged autonomous operation without GPS correction [1].

Sensor data are synchronized using an absolute UNIX timestamp (ms), which serves as a common key for all measurement channels. GPS and IMU data are interpolated onto a common time axis with a step matching the IMU update rate, which eliminates the problem of fusing data streams with different update frequencies.

2.2. State Model and Vehicle Motion Equations

To formalize the vehicle position determination problem, a system state vector is introduced that includes geographic coordinates, velocity, heading angle, and sensor zero-bias offsets:

$$x = [\varphi, \lambda, v, \psi, b_a, b_g]^T, \quad (1)$$

where φ, λ are latitude and longitude (deg); v is velocity (m/s); ψ is the heading angle (rad); b_a, b_g are the accelerometer and gyroscope zero-bias offsets, respectively (m/s², rad/s).

The state transition function $f(\cdot)$ describes the evolution of the system between time steps according to a constant-acceleration kinematic model. The change in geographic coordinates is defined through the Earth's radius $R_e = 6,371,000$ m:

$$\varphi_{k+1} = \varphi_k + \frac{v \Delta t \cos(\psi)}{R_e} \quad (2)$$

$$\lambda_{k+1} = \lambda_k + \frac{v \Delta t \sin(\psi)}{R_e \cos(\varphi)} \quad (3)$$

$$v_{k+1} = v_k + a_{lon} \Delta t \quad (4)$$

$$\psi_{k+1} = \psi_k + \omega_z \Delta t, \quad (5)$$

where Δt is the integration time step (s); a_{lon} is the longitudinal acceleration measured by the IMU (m/s²); ω_z is the yaw rate measured by the IMU (rad/s). The observation vector is formed from GPS receiver data: $z = [\varphi_{GPS}, \lambda_{GPS}, v_{GPS}, \psi_{GPS}]^T$.

2.3. Using the Complementary Filter for Coordinate Determination

The complementary filter is the simplest method for integrating GPS and IMU data and serves as the baseline algorithm for the comparative analysis. The method's idea is based on frequency-domain separation of information sources: the IMU carries useful information about fast dynamics (high frequencies), while GPS provides long-term position stability (low frequencies) [2].

At each step, the state vector estimate is computed as a weighted sum of the two sources:

$$\hat{x} = \alpha x_{GPS} + (1 - \alpha) x_{IMU}, \quad (6)$$

where $\hat{x}$ is the a posteriori state vector estimate (previously defined as $x = [\varphi, \lambda, v, \psi, b_a, b_g]^T$); x_{GPS} is the state vector computed from the current GPS receiver measurements; x_{IMU} is the state vector obtained by integrating IMU data; $\alpha \in (0, 1)$ is the blending coefficient determining the relative weight of the GPS measurement. Typical values of $\alpha = 0.02-0.1$ at a 100 Hz IMU rate correspond to a filter time constant $T = \Delta t / \alpha$ ranging from 1 to 5 s. The advantage of this method is its minimal computational complexity $O(n)$ and simplicity of implementation. Its main drawback is the fixed coefficient α , which does not account for real-time changes in GPS signal quality, and the filter's linear nature, which limits accuracy for nonlinear motion trajectories [3].

2.4. Applying the Extended Kalman Filter for Coordinate Determination

The Extended Kalman Filter is the de-facto standard for nonlinear state estimation in navigation systems. Unlike the linear Kalman filter, the EKF linearizes the nonlinear transition function $f(\cdot)$ and observation function $h(\cdot)$ by computing Jacobian matrices at the current estimate point, which allows the apparatus of optimal linear filtering to be applied to nonlinear systems [4].

The EKF algorithm consists of two alternating steps. Prediction step:

$$\hat{x}_k^- = f(\hat{x}_{k-1}, u_k) \quad (7)$$

$$P_k^- = F_k P_{k-1} F_k^T + Q, \quad (8)$$

where $\hat{x}_k^-$ is the a priori state vector estimate at step k (before a new measurement arrives); P_k^- is the a priori error covariance matrix; $F_k = \partial f / \partial x|_{\hat{x}_{k-1}}$ is the Jacobian matrix of the state transition function (partial derivatives of f with respect to the state components, evaluated at $\hat{x}_{k-1}$); F_k^T is the transpose of F_k ; Q is the process noise covariance matrix (characterizing motion-model uncertainty); u_k is the control input vector (current IMU measurements: acceleration and angular rates). Update step:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R)^{-1} \quad (9)$$

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - h(\hat{x}_k^-)) \quad (10)$$

$$P_k = (I - K_k H_k) P_k^-, \quad (11)$$

where K_k is the Kalman gain matrix, determining the optimal weight of the GPS measurement relative to the IMU prediction; $H_k = \partial h / \partial x|_{\hat{x}_k^-}$ is the Jacobian matrix of the observation function (partial derivatives of h with respect to the state components, evaluated at the a priori estimate $\hat{x}_k^-$); H_k^T is the transpose of H_k ; R is the GPS measurement noise covariance matrix (reflecting GPS receiver accuracy); $(z_k - h(\hat{x}_k^-))$ is the innovation vector, reflecting the discrepancy between the current GPS observation and the predicted value; I is the $n \times n$ identity matrix, where $n = 6$ is the state vector dimension. The R matrix is adaptively scaled according to the GPS receiver's accuracy indicator, which is a key advantage of the EKF over the complementary filter [5].

2.5. Applying the Unscented Kalman Filter for Coordinate Determination

The Unscented Kalman Filter resolves a fundamental problem of the EKF — the first-order approximation error arising from Jacobian-based linearization. Instead of linearizing, the UKF uses a deterministic sigma-point sampling scheme that more accurately approximates the distribution of a nonlinearly transformed random variable, up to second-order accuracy [6].

For a state vector of dimension n , a set of $2n+1$ sigma points χ_i is formed:

$$\chi_0 = \hat{x} \quad (12)$$

$$\chi_i = \hat{x} + \left(\sqrt{(n + \lambda)P} \right)_i, \quad i = 1, \dots, n \quad (13)$$

$$\chi_{i+n} = \hat{x} - \left(\sqrt{(n + \lambda)P} \right)_i, \quad i = 1, \dots, n, \quad (14)$$

where $\hat{x}$ is the current a posteriori state vector estimate (defined earlier as $x = [\varphi, \lambda, v, \psi, b_a, b_g]^T$); P is the current error covariance matrix; n is the state vector dimension ($n = 6$); χ_0 is the central sigma point (coincides with the current estimate); χ_i and χ_{i+n} are symmetric sigma points offset from the center by an amount proportional to the square root of the covariance matrix; $\lambda = \alpha^2(n + \kappa) - n$ is the scaling parameter; α is a parameter determining the spread of sigma points around the mean (typically $\alpha = 10^{-3}$); κ is an additional scaling parameter (usually 0 or 3 – n). Each sigma point is propagated through the nonlinear transition function $f(\cdot)$ without linearization. The mean and covariance estimates after propagation are recovered as weighted sums:

$$\hat{x}^- = \sum_i W_i^m f(\chi_i) \quad (15)$$

$$P^- = \sum_i W_i^c (f(\chi_i) - \hat{x}^-)(f(\chi_i) - \hat{x}^-)^T + Q, \quad (16)$$

where $\hat{x}^-$ is the a priori mean estimate after propagating the sigma points through $f(\cdot)$; P^- is the a priori covariance matrix; W_i^m are the weighting coefficients for the mean estimate; W_i^c are the weighting coefficients for the covariance estimate (computed from the parameters α , κ , and β , where $\beta = 2$ is optimal for Gaussian distributions); Q is the process noise covariance matrix (the same as in equation (8)). The UKF update step is analogous to that of the EKF, but the cross-covariance between state and observation is also computed via the sigma points, eliminating the need for analytical differentiation of f and h . This is a critical advantage when working with complex nonlinear models and reduces the risk of filter divergence characteristic of the EKF during sharp trajectory changes [7]. The computational complexity of the method is $O(n^3)$ per update step, a modest overhead compared with the EKF at typical navigation-problem dimensionalities ($n = 6$).

2.6. Method Quality Assessment Metrics

The following metrics were defined for quantitative comparison of the methods under study. The positioning Root Mean Square Error (RMSE) is computed relative to a reference trajectory obtained from a differential GPS receiver (dGPS) with sub-meter accuracy:

$$RMSE = \sqrt{\left(\frac{1}{N} \right) \sum_{i=1}^N \left[(\hat{\varphi}_i - \varphi_i)^2 R_e^2 + (\hat{\lambda}_i - \lambda_i)^2 (R_e \cos \varphi)^2 \right]}, \quad (17)$$

where N is the total number of measurements in the test set; $\hat{\varphi}_i$, $\hat{\lambda}_i$ are the algorithm-estimated latitude and longitude at point i (°); φ_i , λ_i are the reference latitude and longitude values from the dGPS at point i (°); $R_e = 6,371,000$ m is the mean Earth radius; the factors R_e and $(R_e \cdot \cos \varphi)$ convert angular errors into linear meters along the meridian and parallel, respectively.

The Circular Error Probable (CEP) is a statistical positioning-accuracy metric defined as the radius of a circle, centered on the mean estimated position, within which a given fraction of all measurements falls. CEP50 corresponds to the 50th percentile — half of all estimated positions fall inside this circle; it represents the median error and the primary indicator of typical accuracy under a normal error distribution. CEP95 corresponds to the 95th percentile, characterizing guaranteed accuracy in the vast majority of situations and providing a more conservative estimate for engineering applications. Unlike RMSE, CEP metrics are robust to outliers and do not depend on the assumption of a Gaussian error distribution.

Update latency is the time between the arrival of a new GPS measurement and the resulting updated position estimate (requirement: latency below 10 ms for real-time systems). GPS outage drift is the accumulated positioning error over 5 and 10 s of autonomous operation using IMU data alone (dead-reckoning mode, i.e., position determination by integrating accelerometer and gyroscope readings without external correction), which characterizes the method's robustness under critical conditions (tunnels, underground parking, radio-shadow zones) [8].

A summary comparison of the examined methods is given in Table 1.

Table 1.

Comparative characteristics of GPS/IMU integration methods

Characteristic	Complementary	EKF	UKF	dGPS (reference)
Nonlinearity	No	Partial	Full	–
Jacobian	No	Yes	No	–
Covariance estimation	No	Yes	Yes	–
Adaptivity to GPS quality	No	Yes	Yes	–
Computational cost	$O(n)$	$O(n^2)$	$O(n^3)$	–
RMSE (m), 5 Hz GPS	3–6	1–2	0.8–1.5	< 0.5
GPS outage drift, 5 s (m)	8–15	4–7	3–5	–

Note. RMSE and GPS outage drift values are given for typical driving conditions at a GPS rate of 5 Hz and an IMU rate of 100 Hz; the values are approximate and refined in the Results section.

Results

3.1. Experimental Setup

The algorithms were tested on a route with a total length of 4.7 km, including straight segments, curved turns with radii from 15 to 80 m, and a segment with a simulated 10-second GPS outage (antenna shielding). The route was driven three times for each algorithm configuration to average out random errors.

The measurement system's hardware configuration included: a u-blox M8N GPS receiver with a 5 Hz update rate and an actively amplified external antenna; an ICM-42688-P IMU module with a 100 Hz sampling rate, mounted at the vehicle's center of mass; and a Raspberry Pi 5 single-board computer for real-time data acquisition and processing. The reference trajectory was obtained using a u-blox F9P RTK GPS receiver. RTK (Real-Time Kinematic) is a differential GPS positioning method in which a base station with known coordinates transmits phase corrections to a moving receiver in real time; this allows positioning accuracy of 1–2 cm to be achieved, compared with the typical 2–5 m of conventional GPS. The positioning accuracy of the u-blox F9P receiver in RTK mode is below 0.02 m (CEP50), which exceeds the accuracy of the studied methods by two orders of magnitude, making it suitable as an independent reference for error evaluation.

All three methods — the complementary filter (CF), the Extended Kalman Filter (EKF), and the Unscented Kalman Filter (UKF) — were implemented in Python 3.11 using the NumPy and FilterPy libraries. Filter parameters were tuned on a training set (the first route run); testing was performed on two independent runs.

3.2. Positioning Accuracy under a Stable GPS Signal

Positioning accuracy under a stable GPS signal was evaluated across all route segments except the simulated GPS outage. RMSE, CEP50, and CEP95 were computed for each method relative to the reference trajectory. The results are given in Table 2.

Table 2.

Positioning accuracy indicators under a stable GPS signal				
Metric	CF	EKF	UKF	RTK (reference)
RMSE, m	4.83	1.54	1.12	0.02
CEP50, m	3.91	1.17	0.86	0.01
CEP95, m	9.64	3.08	2.21	0.04
Max deviation, m	18.3	5.7	4.1	0.07

Note. Values are given as the average of two test route runs; the RTK receiver is used exclusively as a reference and is not one of the compared methods.

The RMSE and CEP indicators show a consistent advantage of the Kalman filters over the complementary filter across all route segments. The maximum position deviation for CF reaches 18.3 m in zones of intensive maneuvering, whereas for EKF and UKF this indicator does not exceed 5.7 m and 4.1 m, respectively.

3.3. Positioning Accuracy during GPS Outages

To evaluate dead-reckoning performance, the GPS antenna was shielded for a fixed interval (5 and 10 s) while the vehicle traveled at 40–60 km/h on a straight segment. The accumulated positioning error at the moment GPS signal was restored is given in Table 3.

Table 3.

Accumulated error during GPS signal loss			
GPS signal loss duration	CF, m	EKF, m	UKF, m
5 s	10.4	5.1	3.7
10 s	31.8	14.3	9.6
Convergence time after GPS restoration, s	3.2	1.8	1.3

Note. Accumulated-error values are given as the average of five repeated measurements; convergence time is the time until RMSE returns to the level characteristic of normal operation.

During a 10-second GPS outage, the accumulated CF error exceeds 31 m, making the method unsuitable for accurate trajectory tracking in enclosed spaces. The EKF and UKF filters show a substantially lower error-accumulation rate owing to adaptive covariance estimation: the UKF error after 10 s is 9.6 m, 33% lower than that of the EKF.

3.4. Accuracy under High-Dynamics Motion

To analyze algorithm behavior under aggressive driving, route segments with lateral acceleration exceeding 0.5g (4.9 m/s²) were isolated. On these segments the GPS signal remained stable, but trajectory nonlinearity placed maximal stress on the EKF's linearization apparatus. The results are given in Table 4.

Table 4.

Accuracy indicators under intensive maneuvering (lateral acceleration $\geq 0.5g$)

Indicator	CF	EKF	UKF
RMSE, m	6.41	2.23	1.48
RMSE degradation relative to normal mode, %	+33	+45	+32
Filter divergence rate, occurrences out of 30	—	0	0
Peak innovation ($z_k - h(x)$), m	—	3.12	2.04

Note. The RMSE degradation percentage is computed relative to the indicators in Section 3.2; filter divergence was assessed as the covariance P exceeding acceptable bounds over 30 test passes of the segment.

Both Kalman filters remained stable (zero divergence cases) across all 30 test passes of the dynamic segment. The EKF indicator degrades by 45% relative to normal-mode operation — more than the UKF (32%) and CF (33%) — which is explained by the accumulation of linearization error under sharp changes in heading angle.

3.5. Update Latency and Computational Complexity

Update latency was measured as the time from the arrival of a new GPS packet to the availability of the updated state vector estimate. Computational load was measured on a Raspberry Pi 5 platform (ARM Cortex-A76, 2.4 GHz). The results are given in Table 5.

Table 5.

Timing characteristics of the algorithms (Raspberry Pi 5, Python 3.11)

Indicator	CF	EKF	UKF
Mean update latency, ms	0.3	1.1	3.8
Maximum latency, ms	0.7	2.4	7.2
IMU integration step time, μs	8	41	148
CPU load at 100 Hz, %	< 1	4.1	14.8

Note. All methods satisfy the < 10 ms update-latency requirement for real-time systems; values are given for a single-threaded, unoptimized implementation.

All three implementations satisfy the under-10-ms latency requirement for real-time systems. The maximum UKF latency is 7.2 ms, leaving an adequate margin for embedded platforms. CPU load at 100 Hz does not exceed 15% even for the UKF, the most computationally demanding method, confirming the suitability of all methods for onboard real-time telemetry systems.

4. Discussion

4.1. Interpretation of Positioning Accuracy Results

The results obtained confirm the general trend established in the literature: GPS/IMU integration via Kalman filters provides substantially better accuracy than simplified approaches. The complementary filter's RMSE (4.83 m) matches typical values for CF implementations under similar test conditions, consistent with the findings of Madgwick and Teel [9], who reported an RMSE range of 4.1–6.2 m for CF at a 5 Hz GPS rate.

The EKF's RMSE advantage over CF is 68%, higher than the 45–55% documented by Groves [4] and Savage [10] for similar kinematic models. This is explained by the adaptive scaling of the covariance matrix R depending on GPS signal quality: on the test route, with its frequent heading-angle changes, adapting R allowed the EKF to more stably maintain the position estimate during transient motion regimes.

The UKF's RMSE advantage over the EKF is 27% (1.12 m versus 1.54 m), consistent with the theoretically expected 15–35% range for nonlinear navigation problems [6, 7]. The relatively smaller RMSE degradation under dynamic maneuvering (+32% for UKF versus +45% for EKF) confirms that the unscented transform better approximates the nonlinearity of the motion equations during sharp heading-angle changes, avoiding the first-order error accumulation characteristic of the EKF.

4.2. Behavior during GPS Outages

The dead-reckoning mode is a critical scenario for telemetry systems in underground parking, tunnels, and dense urban areas with satellite-signal blockage. The accumulated CF error (31.8 m over 10 s) is unacceptable for most practical applications. Compared with Wendel and Schlaile [11], who reported a dead-reckoning error of 18–25 m over 10 s for consumer-grade MEMS IMUs using CF, our results are somewhat worse (31.8 m), attributable to the higher speed of motion (40–60 km/h) and greater accumulation of yaw-angle error.

The UKF result (9.6 m over 10 s) is close to the values reported by Toledo-Moreo et al. [5] for the EKF with a similar IMU: 8.1–11.4 m over 10 s. This suggests that, for typical consumer-grade MEMS gyroscopes, the difference between the EKF and UKF in dead-reckoning mode is determined less by the quality of nonlinearity approximation than by the accuracy of gyroscope bias calibration, which both methods estimate equally well through the state vector.

4.3. Computational Cost and Suitability for Embedded Systems

All three methods demonstrated update latencies within 10 ms and CPU loads below 15% at 100 Hz on the Raspberry Pi 5 platform. This means that even the most computationally intensive method, the UKF, leaves over 85% of processor time available for other onboard telemetry tasks, such as logging, lap detection, and delta-time calculation. Compared with the microcontroller-based EKF implementations reported by Coskun and Tuncer [12] (STM32F4, C++, 0.8 ms latency), the Python implementation on the Raspberry Pi 5 is slower in absolute terms but offers substantially greater flexibility for algorithm tuning and debugging.

The choice between the EKF and the UKF for a specific telemetry system should balance accuracy against computational resources. On memory-constrained platforms (microcontrollers without an FPU), the EKF is the more practical choice. On single-board computers or DSP platforms, the UKF is a well-justified alternative that provides consistently better accuracy without the risk of filter divergence under aggressive maneuvering.

Conclusions

This study presents a comparative analysis of three methods for integrating GPS and inertial measurement unit data — the complementary filter, the Extended Kalman Filter, and the Unscented Kalman Filter — for determining vehicle coordinates in real-time telemetry systems. The Kalman filters were found to provide substantially better positioning accuracy than the complementary filter: the RMSE of the EKF and UKF is 3.1 and 4.3 times lower, respectively. In dead-reckoning mode during a 10-second GPS outage, the UKF shows an accumulated error of 9.6 m, 33% lower than the EKF and 3.3 times lower than the complementary filter. Under intensive maneuvering (lateral acceleration $\geq 0.5g$), the UKF exhibits a smaller relative accuracy degradation (+32%) compared with the EKF (+45%), confirming the advantage of the unscented transform for nonlinear motion trajectories. All three implementations satisfy the under-10-ms update-latency requirement on the Raspberry Pi 5 platform, and CPU load at 100 Hz does not exceed 15% even for the UKF, confirming the methods' suitability for onboard real-time systems. The practical value of these results lies in well-founded recommendations for method selection based on platform computational resources: the EKF is the better solution for microcontroller-based systems, while the UKF is recommended for single-board computers and DSP platforms with sufficient computational resources. Directions for further research include adapting the algorithms to work with real-time RTK correction, investigating machine-learning-based methods for automatically tuning the Q and R matrices, and a hardware implementation of the UKF on FPGA platforms to achieve sub-millisecond update latency.

ADDITIONAL INFORMATION

AUTHOR CONTRIBUTIONS

Conceptualization, E.Z.; methodology, V.T.; validation, E.Z.; formal analysis, V.T.; investigation, E.Z.; data curation, E.Z.; writing-original draft preparation, V.T.; writing-review and editing, V.T.; visualization, V.T.; supervision, E.Z.; project administration, E.Z. All authors have read and agreed to the published version of the manuscript.

DECLARATION ON THE USE OF GENERATIVE ARTIFICIAL INTELLIGENCE TOOLS

In preparing this work, the author used DeepL Translate and Grammarly for: grammar and spelling checks, paraphrasing, and rephrasing. After using these tools/services, the author reviewed and edited the content and takes full responsibility for the content of this publication.

REFERENCES

1. Woodman, O. J. (2007). An introduction to inertial navigation (Technical Report UCAM-CL-TR-696). University of Cambridge, Computer Laboratory.
2. Mahony, R., Hamel, T., & Pflimlin, J.-M. (2008). Nonlinear complementary filters on the special orthogonal group. *IEEE Transactions on Automatic Control*, 53(5), 1203-1218. <https://doi.org/10.1109/TAC.2008.923738>
3. Rhudy, M. B., Lobo, R. A., & Gu, J. (2013). A perspective on nonlinear filtering for GPS/INS integration. *Proceedings of the AIAA Guidance, Navigation, and Control Conference* (pp. 1-12). Boston. <https://doi.org/10.2514/6.2013-4676>
4. Groves, P. D. (2013). *Principles of GNSS, inertial, and multisensor integrated navigation systems* (2nd ed.). Artech House.
5. Toledo-Moreo, R., Zamora-Izquierdo, M. A., Ubeda-Minarro, B., & Gomez-Skarmeta, A. F. (2007). High-integrity IMU/DGPS-based positioning system for intelligent transportation systems. *IEEE Transactions on Intelligent Transportation Systems*, 8(3), 491-511. <https://doi.org/10.1109/TITS.2007.903893>
6. Wan, E. A., & Van der Merwe, R. (2000). The unscented Kalman filter for nonlinear estimation. *Proceedings of the IEEE Symposium on Adaptive Systems for Signal Processing, Communications, and Control* (pp. 153-158). Lake Louise. <https://doi.org/10.1109/ASSPCC.2000.882463>
7. Julier, S. J., & Uhlmann, J. K. (2004). Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3), 401-422. <https://doi.org/10.1109/JPROC.2003.823141>

8. Angrisano, A., Gaglione, S., & Gioia, C. (2010). GNSS/INS integration methods. Proceedings of the European Navigation Conference (pp. 1-11). Gdansk.
9. Madgwick, S. O. H., Harrison, A. J. L., & Vaidyanathan, R. (2011). Estimation of IMU and MARG orientation using a gradient descent algorithm. Proceedings of the IEEE International Conference on Rehabilitation Robotics (pp. 1-7). Zurich. <https://doi.org/10.1109/ICORR.2011.5975346>
10. Savage, P. G. (1998). Strapdown inertial navigation integration algorithm design. Part 2: Velocity and position algorithms. Journal of Guidance, Control, and Dynamics, 21(2), 208-221. <https://doi.org/10.2514/2.4242>
11. Wendel, J., Metzger, J., Moenikes, R., Maier, A., & Trommer, G. F. (2006). Time-differenced carrier phase measurements for tightly coupled GPS/IMU integration. Navigation, 53(3), 175-185. <https://doi.org/10.1002/j.2161-4296.2006.tb00384.x>
12. Coskun, A., & Tuncer, H. (2021). Real-time GPS/IMU data fusion via extended Kalman filter on embedded systems. Proceedings of the International Conference on Electrical and Electronics Engineering (pp. 314-318). Bursa. <https://doi.org/10.1109/ELECO54474.2021.9677706>

Віктор ТКАЧЕНКО, Едуард ЖАРІКОВ

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»

МЕТОДИ ІНТЕГРАЦІЇ GPS ТА ІВБ ДЛЯ ВИЗНАЧЕННЯ КООРДИНАТ ТРАНСПОРТНОГО ЗАСОБУ

Стаття присвячена вирішенню задачі визначення координат транспортного засобу в системах телеметрії реального часу шляхом інтеграції даних глобальної навігаційної супутникової системи (GPS) та інерціального вимірювального блоку (ІВБ). Проведений комплексний аналіз існуючих підходів до злиття сенсорних даних дозволив сформулювати задачу в контексті вбудованих платформ реального часу з обмеженими обчислювальними ресурсами та підкреслив необхідність системного порівняльного оцінювання методів, які принципово відрізняються за обчислювальною складністю та точністю. Розглянуто три методи злиття сенсорних даних: комплементарний фільтр, який поєднує дані на основі частотного розділення джерел; розширений фільтр Калмана, який ґрунтується на лінеаризації нелінійної моделі руху через матриці Якобіана; та сигма-точковий (unscented) фільтр Калмана, який застосовує детерміноване сигма-точкове перетворення для апроксимації нелінійного поширення стану без лінеаризації. У статті обґрунтовано вибір методологічних засад, покладених в основу кожного з методів, зокрема кінематичної моделі стану транспортного засобу, що включає географічні координати, швидкість, курсовий кут та зміщення нуля сенсорів, а також метрик, використаних для їх кількісного порівняння — середньоквадратичної похибки позиціонування (RMSE), радіуса кола рівномірного відхилення (СЕР), накопиченої похибки в режимі dead reckoning при відсутності GPS-сигналу та затримки оновлення. Зростаюча потреба в надійному позиціонуванні в умовах відсутності GPS-сигналу, у поєднанні з обмеженими обчислювальними ресурсами вбудованих телеметричних платформ та нелінійним характером кінематики транспортного засобу, зумовлює необхідність вироблення практичних критеріїв вибору фільтра з урахуванням апаратної платформи. Для вирішення цієї задачі всі три алгоритми реалізовано на єдиній апаратно-програмній платформі — одноплатному комп'ютері Raspberry Pi 5 під керуванням Python 3.11 — та перевірено на реальних даних руху, зібраних на маршруті протяжністю 4,7 км, з еталонною траєкторією, отриманою від RTK GPS-приймача. У статті наведено результати експериментального дослідження, що демонструють відносну ефективність запропонованих методів за показниками точності позиціонування, стійкості до втрати GPS-сигналу та поведінки при агресивному маневруванні. Крім того, викладено практичні рекомендації щодо вибору методу інтеграції залежно від обчислювальних обмежень цільової вбудованої платформи — від мікроконтролерних систем до одноплатних комп'ютерів і DSP-платформ.

Ключові слова: злиття сенсорних даних, фільтр Калмана, GPS/IMU інтеграція, інерціальна навігація, dead reckoning, точність позиціонування, телеметрія реального часу, unscented перетворення, MEMS-гіроскоп, вбудовані системи.