

Article

Not peer-reviewed version

Validation of an AI-Assisted Terrain Aided Navigation Algorithm Using Real-World Flight Test Instrumentation Data

[Ümit Can Bekar](#)*, [Bilgehan Tanyeri](#), Ibrahim Enes Uslu, [N.Arda Güngör](#), [Gökhan İnalhan](#)

Posted Date: 30 May 2025

doi: 10.20944/preprints202505.2427.v1

Keywords: avionics; AI; Terrain Aided Navigation; navigation



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

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.

Article

Validation of an AI-Assisted Terrain Aided Navigation Algorithm Using Real-World Flight Test Instrumentation Data

Ümit Can Bekar ^{1,2,*}, Bilgehan Tanyeri ^{1,2}, Ibrahim Enes Uslu ^{1,2}, N. Arda Güngör ^{1,2} and Gökhan İnalhan ^{1,3}

¹ Istanbul Technical University
² Turkish Aerospace
³ Altay Aerospace
* Correspondence: can.bekar@tai.com.tr

Abstract: This study introduces an enhanced artificial intelligence (AI)-assisted Terrain Aided Navigation (TAN) for a sophisticated jet trainer, building upon our prior research[1] by incorporating real-flight test validation. The proposed TAN integrates a high-performance terrain server, a digital elevation model, and an efficient line-of-sight algorithm to facilitate terrain-aided navigation. The system utilizes an advanced search algorithm in conjunction with two filter designs, including adaptive filters that dynamically optimize navigation precision and operational efficiency. A significant development is the AI model's capacity to independently alternate between the resource-intensive search algorithm and a set of filters, thereby maintaining navigational accuracy while facilitating in-flight execution without supplementary hardware requirements. Comprehensive Monte Carlo calculations, validated by flight test instrumentation (FTI) data, indicate that the proposed TAN consistently facilitates low-altitude navigation across diverse operational settings. The incorporation of actual flight data not only substantiates the system's efficacy but also offers novel perspectives on practical implementation obstacles and improvements. These findings signify an advancement in autonomous terrain-aided navigation, connecting simulation with actual flight performance.

Keywords: avionics; AI; Terrain Aided Navigation; navigation

1. Introduction

This study presents an AI-enhanced Terrain Aided Navigation (TAN) designed for Hürjet, an advanced jet trainer aircraft seen in Figure 1, with the aim of improving navigation accuracy and operational efficiency. The system utilizes an advanced terrain server combined with a high-resolution digital elevation map (DEM) called digital terrain elevation data (DTED) to produce accurate height-on-terrain (HoT) and height-above-terrain (HaT) data, which are crucial for precise low-altitude aircraft navigation.



Figure 1. Hürjet flying over the Bosphorus Strait.

The TAN is fundamentally based on a sophisticated technique that includes intricate filter setups. We specifically implement and evaluate banks of adaptive filters, which dynamically react to varying terrain roughness and other environmental factors. This adaptive filtering method enhances the system’s tracking accuracy and considerably improves its resilience under diverse flying situations. Additionally, the TAN incorporates an AI-driven supervisory module that can automatically switch among various software designs, providing a warrant that the system selects the most efficient mode’s output at all times. A simplified diagram of the algorithm is given in Figure 2 where latitude δL , longitude $\delta \lambda$, and altitude δh errors in geodetic coordinate system. Also, h_{radalt} is the radalt clearance measurement and $\hat{x}$ is the state vector that contains positions. A sophisticated search algorithm, designed for the acquisition phase, effectively analyzes topographical data to determine safe flight paths while reducing processing demands.

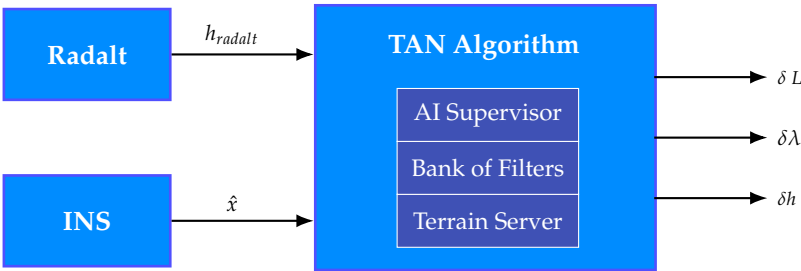


Figure 2. Basic schematic of the TAN algorithm.

To assess the efficacy and resilience of the proposed system, we use empirical analysis bolstered by FTI data and comprehensive simulation outcomes. This dual approach highlights the practical application of our AI-enhanced TAN and illustrates its capability to satisfy the stringent certification standards necessary for sophisticated avionic systems. Our study establishes a foundation for a next-generation terrain-aided navigation system that is both safe and reliable by connecting simulation with real-world validation.

While the proposed system builds upon recent advances in adaptive filtering and AI-assisted supervisory control, it is important to acknowledge the foundational contributions in the field. Notably, the Terrain Contour Matching (TERCOM) algorithm [1] and the Sandia Inertial Terrain Aided Navigation (SITAN) system [2,3] have laid the groundwork for batch and recursive TAN architectures, respectively. Our study builds upon these classical methodologies, extending them through hybrid filter banks and artificial intelligence to support robust navigation in GPS-denied environments.

Real-world accuracy of the proposed TAN algorithm to achieve %89 better performance in comparison to just standalone traditional filter solutions. Our extended Kalman filter has better performance than inertial navigation system only solution by %38.

Our contributions in this study are summarized as follows: Development of a hybrid TAN framework and demonstration of reduced navigational drift in environments lacking GPS signals. Our main contribution is the validation of an artificial intelligence-based supervisory system that dynamically selects appropriate navigation algorithms according to terrain characteristics and filter attributes, against real world flight test instrumentation data.

The rest of the paper is organized as follows: Section 2 provides a brief history of previous works. Section 3 explains the TAN methodology. Section 4 introduces our proposed TAN algorithms and Section 5 presents the simulation results. Section 6 concludes our paper.

2. Previous Work

TAN is one of the methods that produces a navigation solution independently of GPS. There are many TAN algorithms in the literature and the most famous ones are terrain contour matching (TERCOM)[2], Sandia inertial terrain aided navigation (SITAN) [3,4] algorithms. TAN algorithms can be examined under three headings: batch, recursive, and hybrid algorithms. Although the first studies on TAN focused on batch or recursive algorithms, current studies generally consist of hybrid

algorithms that combine different algorithms, as in this article. Moreover, gravity-aided and magnetic navigation studies are among the alternative methods that can work without GPS.

2.1. Batch TAN Algorithms

In the batch TAN algorithms, the data is processed as a single group. In the TERCOM [2] basically height error comparison is made using different comparison methods. In [5], it is stated that mean absolute difference (MAD) is the most efficient and most correct combination among comparison method for the real time applications. [6] added maximum a posterior (MAP) estimator to TERCOM and SITAN, and noted improvements were achieved.

Probability-based TAN (PTAN) [7] that examines the probability distribution differences is another research. The solution with the minimum probability distribution difference gives the best match. However, [5] stated that the computational load of this algorithm is very high.

[8] observed an increase in system performance using probabilistic data association filter (PDAF) compared to TERCOM. However there are questions marks about running this system in real time [5].

[9] proposes a computer vision based TAN application by using a charge-coupled camera. However, this method has the disadvantage of giving poor results in cases such as adverse weather conditions [5].

2.2. Recursive TAN Algorithms

In the recursive TAN algorithms, the data is processed recursively when new measurement data is received. For SITAN algorithm different filters are compared as nonlinear Kalman filter (KF), extended Kalman filter (EKF), parallel Kalman filter, particle filter (PF). Among these filter structures, PF has the most computational cost. Additionally, it has been observed that a divergence problem may occur in some trials. In this method, terrain linearization is one of the most important factors [4]. In [10] system models have been improved with colored noises for SITAN. Also by adding error models to SITAN, better and more realistic results are achieved [11]. In a study, terrain altitude linearization is made [12]. It is stated that SITAN is used for AFTI/F16 [13] and US Army UH-1 Helicopter [14]. Multi model adaptive estimation (MMAE) and KF is used for this application [14].

In [15] Inertial Terrain Aided Navigation (BITAN)-II method is presented where position and speed errors are examined. Also for navigation solution Bank of KF are used. BITAN [16] which can work in both acquisition and track modes, is actually an improved version of SITAN.

In [17], the point mass filter reached the Cramer-Rao lower limit after convergence, and good results were achieved with this Bayesian method on both flat and variable terrain. However, there are computational problems due to dense meshes [5]. [18] proposed Rao-Backwellisation technique that uses recursive Monte Carlo (MC) runs and particle filter for position estimation.

Viterbi algorithm terrain aided navigation (VATAN) [19] used Viterbi algorithm and contains dynamic programming for position estimate.

Landmark detection is made [20] by using probabilistic data association filter (PDAF). Moreover, a batch implementation is made for PDAF [8].

There are also classical TAN algorithms that perform INS correction using DTED [21,22]. In one study, it was mentioned that it provides a high level of localization accuracy [23]. In addition, there are some image based TAN applications [24,25].

2.3. Hybrid TAN Algorithms

Hybrid TAN algorithms are compose of batch and recursive algorithms. Since it contains recursive algorithms, it is also possible to consider this as recursive algorithms. Hybrid TAN algorithms uses KF and comparison techniques together, thus taking advantage of both batch and recursive methods. With this method it is possible to achieve better navigation solutions [26].

In article [27] written on improved TERCOM, it is mentioned that velocity correction is made as well as position correction using Kalman filter. Via incorporating this method, improvement in

accuracy and stability in the navigation solution has been achieved. It has also been stated that this method is a solution to the divergence problem encountered in TERCOM.

TERPROM[®] [28], which is used in military systems, produces navigation solutions using INS and radar measurements and digital terrain data. TERPROM[®], which can also operate at low altitudes, has functions such as advanced terrain avoidance cueing (ATAC), predictive ground collision system (PGCAS), and terrain awareness display (TAD).

Recent studies have been proposed to enhance classical TAN systems, including hybrid filters such as batch + APF structures [29], visual TAN using structure-from-motion [30], or deep learning-based object recognition with aerial maps [31]. In particular, marginalized particle filtering (MPF) has become a popular approach to address the nonlinear nature of TAN, with several studies introducing moment-matched kernels [32], boundary flow mechanisms [33], and computationally efficient filter variants [34] to ensure robustness and efficiency. Moreover, adaptively modeling radar altimeter errors [35] and employing LSTM-based measurement models [36] have proven effective in maintaining filter consistency, especially over flat terrain.

High-altitude applications, such as aircraft-based TAN, have also benefited from slant-range measurement strategies using steerable lasers [37], which allow selective targeting of terrain gradients to maximize navigation information. These methods outperform traditional nadir TAN, both in CRLB analysis and simulation, and have been validated with real flight data. Broader perspectives, such as all-source PNT architectures [38,39], highlight the necessity of combining multiple navigation modalities—TAN, SOOP, vision, and sensor fusion—to ensure resilient performance under jamming or spoofing conditions.

TAN is not only used for airborne applications, there are some underwater TAN applications [40,41]. By using sonar and inertial measurement unit (IMU) autonomous navigation of underwater vehicle is investigated. In [42] it is stated that DTED map is not used, terrain height is detected thanks to sonar sensor.

2.4. Gravity-Aided and Magnetic Navigation Algorithms

Apart from TAN, efforts have been made to investigate GPS independent solutions using gravity-aided and magnetism based methods [43–45]. Gravity navigation approaches, often implemented in underwater or simulation environments, require precise, often bulky gradiometers and rely on high-resolution gravity maps, making their application on aircraft challenging in terms of both cost and platform dynamics [40,46–48]. Moreover, many studies in this domain still focus on improving basic matching or filtering algorithms, indicating that gravity-aided navigation remains in a relatively early stage compared to TAN systems [49]. Similar to this, magnetic navigation techniques rely on anomaly maps and are highly susceptible to magnetic disturbances caused by platforms; this is especially problematic on aircraft because of the complex and dynamic electromagnetic environments present [44,50,51]. While some successful demonstrations have been reported, including real flight tests on fighter aircraft like the F-16 [44], the calibration complexity and variability between platforms make operational deployment difficult.

Biological and marine studies further show how magnetic navigation can be effective in controlled or natural environments, but they also highlight limitations in sensor placement and environmental stability for airborne use [38,52]. However, with off-the-shelf sensors and globally accessible, current terrain databases, TAN systems that use radar altimeters and digital terrain elevation maps provide a more practical option for airplanes. Table 1 provides a comparison of algorithm types.

Table 1. Comparison of Algorithm Types.

Algorithm Type	Terrain Type Sensitivity	Computational Complexity Performance	Divergence Characteristic	Accuracy	Feasibility (Airborne Use)
Batch TAN	Medium	High	No	Medium - High	High
Recursive TAN	Low	Medium – High	Yes	Medium	Medium
Hybrid TAN	High	Medium	No	Medium - High	High
Gravity-Aided Navigation	Low	High	Potentially Yes	Low - Medium	Low
Magnetic Navigation	Low	High	Yes	Low - Medium	Low–Medium
Proposed Algorithm	High	Low	No	High	High

3. TAN Methodology

3.1. Modeling INS Errors Using Linearized State Equations

In an Inertial Navigation System (INS), the mechanization equations determine the system’s position, velocity, and orientation (attitude) within a specified reference frame. These equations exhibit significant nonlinearity, and sensor flaws, such as bias and noise, eventually degrade navigation accuracy. A typical method to model and mitigate these errors is to linearize the INS error dynamics around a nominal (presumed) trajectory. The resulting set of linear time-varying equations enables the implementation of filtering techniques, including the Kalman filter, for state estimation and error corrections [53].

Given that it is significantly easier for individuals on the Earth’s surface to conceptualize a navigation solution within the local level frame, and considering that the computational errors in the horizontal frame are bounded (Schuler effect), it is deemed more suitable to resolve the error dynamics equations in the local level frame for practical applications.

To formulate the error-state equations, applying a small perturbation to the state variables of the INS in its nominal trajectory, while excluding higher-order terms and negligible terms such as the terms that are multiplied by the square of the Earth’s radius, a system of linear differential equations governing the INS error dynamics can be derived. Furthermore, integrating the sensor error characteristics into this INS error dynamics with specific assumptions, such as first-order Gauss-Markov and random walk, provides the augmented state space equation. Now, using this knowledge, we will derive a linear system model of the errors of INS. Comprehensive knowledge about linearization, INS mechanization and error dynamics derivations are available in [53], Chapters 2, 5 and 6.

The INS mechanization equations can be explained as a set of nonlinear differential equations constructed using IMU measurements taken by a 6-degree-of-freedom system with respect to a specified reference coordinate frame. The general form of the nonlinear ODE can be given as below:

$$\dot{x}(t) = f(x(t), u(t)) \tag{1}$$

Where $x(t)$ is the INS states and $u(t)$ is the IMU inputs.

Let the state vector can be composed as

$$\mathbf{x}_{\text{true}} = \mathbf{x}_{\text{nominal}} + \delta \mathbf{x}, \tag{2}$$

where $\mathbf{x}_{\text{nominal}}$ is the major part and $\delta \mathbf{x}$ is the minor and stochastic part. It can be thought $\mathbf{x}_{\text{nominal}}$ as the best guess of the true state (given our motion model and IMU inputs). The model we use for estimation is always subject to some modeling errors and random noises; $\delta \mathbf{x}$ is attributed to the small, stochastic components of the solution.

By taking this into account, linearizing the INS mechanization equations around a nominal solution, we establish that:

$$\delta \dot{x}(t) = \frac{\partial f(x(t), u(t))}{\partial x} \delta x + G(t) = \frac{\partial f(x(t), u(t))}{\partial u} \delta u \tag{3}$$

By letting

$$F(t) = \frac{\partial f[x(t)]}{\partial x} \quad \text{and} \quad G(t) = \frac{\partial f[x(t), u(t)]}{\partial u} \quad (4)$$

we obtain the desired linearized form.

In the local-level frame, the mechanization equations yield a 15-dimensional error-state vector. The position error vector, denoted by $\delta \mathbf{r} = [\delta L, \delta \lambda, \delta h]^T$, represents errors in geodetic latitude δL , longitude $\delta \lambda$, and altitude δh . The velocity error vector, $\delta \mathbf{v} = [\delta v_n, \delta v_e, \delta v_d]^T$, accounts for errors in the north δv_n , east δv_e , and down δv_d velocity components. The attitude error vector, $\delta \boldsymbol{\epsilon} = [\delta r, \delta p, \delta a]^T$, captures the small-angle errors in roll δr , pitch δp , and azimuth δa . The gyroscope error vector, $\delta \boldsymbol{\omega} = [\delta \omega_x, \delta \omega_y, \delta \omega_z]^T$, corresponds to gyroscope biases, and the accelerometer error vector, $\delta \mathbf{f} = [\delta f_x, \delta f_y, \delta f_z]^T$, represents accelerometer biases resolved in x-y-z triad. Concatenating these sub-vectors results in the 15-dimensional error-state vector $\delta \mathbf{x}_{15 \times 1}^T = [\delta \mathbf{r} \ \delta \mathbf{v} \ \delta \boldsymbol{\epsilon} \ \delta \boldsymbol{\omega} \ \delta \mathbf{f}]^T$. After linearizing the INS trajectory along nominal path, the linear relationship between error states and the change of error states can be expressed as follows.

$$\begin{bmatrix} \delta \dot{L} \\ \delta \dot{\lambda} \\ \delta \dot{h} \\ \delta \dot{v}_n \\ \delta \dot{v}_e \\ \delta \dot{v}_d \\ \delta \dot{r} \\ \delta \dot{p} \\ \delta \dot{a} \\ \delta \dot{\omega}_x \\ \delta \dot{\omega}_y \\ \delta \dot{\omega}_z \\ \delta \dot{f}_x \\ \delta \dot{f}_y \\ \delta \dot{f}_z \end{bmatrix} = \begin{bmatrix} 0_{3 \times 3} & \begin{bmatrix} \frac{1}{(R_N+h)\cos(L)} & 0 & 0 \\ 0 & \frac{1}{R_M+h} & 0 \\ 0 & 0 & -1 \end{bmatrix} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & \begin{bmatrix} 0 & f_n & -f_e \\ -f_d & 0 & f_n \\ f_e & -f_n & 0 \end{bmatrix} & 0_{3 \times 3} & \begin{bmatrix} R_b^l & R_b^l & R_b^l \\ R_b^l & R_b^l & R_b^l \\ R_b^l & R_b^l & R_b^l \end{bmatrix} \\ 0_{3 \times 3} & \begin{bmatrix} 0 & -\frac{1}{R_N+h} & 0 \\ \frac{1}{R_M+h} & 0 & 0 \\ 0 & \frac{\tan(L)}{R_N+h} & 0 \end{bmatrix} & 0_{3 \times 3} & \begin{bmatrix} R_b^l & R_b^l & R_b^l \\ R_b^l & R_b^l & R_b^l \\ R_b^l & R_b^l & R_b^l \end{bmatrix} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & \begin{bmatrix} -\beta_{\omega x} & 0 & 0 \\ 0 & -\beta_{\omega y} & 0 \\ 0 & 0 & -\beta_{\omega z} \end{bmatrix} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & \begin{bmatrix} -\beta_{f_x} & 0 & 0 \\ 0 & -\beta_{f_y} & 0 \\ 0 & 0 & -\beta_{f_z} \end{bmatrix} \end{bmatrix} \cdot \begin{bmatrix} \delta L \\ \delta \lambda \\ \delta h \\ \delta v_n \\ \delta v_e \\ \delta v_d \\ \delta r \\ \delta p \\ \delta a \\ \delta \omega_x \\ \delta \omega_y \\ \delta \omega_z \\ \delta f_x \\ \delta f_y \\ \delta f_z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \sqrt{2\beta_{\omega x}\sigma_{\omega x}^2} \\ \sqrt{2\beta_{\omega y}\sigma_{\omega y}^2} \\ \sqrt{2\beta_{\omega z}\sigma_{\omega z}^2} \\ \sqrt{2\beta_{f_x}\sigma_{f_x}^2} \\ \sqrt{2\beta_{f_y}\sigma_{f_y}^2} \\ \sqrt{2\beta_{f_z}\sigma_{f_z}^2} \end{bmatrix} w(t) \quad (5)$$

Where $\beta_{\omega x}, \beta_{\omega y}, \beta_{\omega z}$ and $\beta_{f_x}, \beta_{f_y}, \beta_{f_z}$ are the autocorrelation time constants of the gyroscope and accelerometer, respectively, resolved in the x-y-z triad., $\sigma_{\omega x}, \sigma_{\omega y}$ and $\sigma_{\omega z}$ are the standard deviations associated with gyroscope, $\sigma_{f_x}, \sigma_{f_y}$ and σ_{f_z} are the standard deviations associated with accelerometer, R_b^l represents the rotation matrix elements which converts body frame to local level frame, f_n, f_e and f_d are the accelerometer readings resolved in local NED frame and $w(t)$ is the unit variance Gaussian white noise.

We can express this set of linear differential equations in an compact way for filtering tasks like state estimation.

$$\delta \dot{\mathbf{x}}(t) = \mathbf{F}_x(t) \delta \mathbf{x}(t) + \mathbf{G} w(t), \quad (6)$$

$\mathbf{F}_x(t)$ is called the *dynamic matrix* which propagates the INS errors over time, $\mathbf{G}$ is the noise distribution matrix.

3.2. TAN Fundamentals

Terrain-Aided Navigation (TAN) entails determining the platform's position by comparing the measured terrain profile with the onboard elevation data. These topographical profiles are generated utilizing data from the platform's barometric altimeter and radar altimeter (RADALT). As depicted

in Figure 3, the barometric altimeter provides altitude information relative to mean sea level ($\hat{h}_{\text{baro}}$), whilst the RADALT measures the platform's elevation above the ground (h_{radalt}). The terrain elevation at the measurement location is determined by subtracting the RADALT value from the barometer altitude.

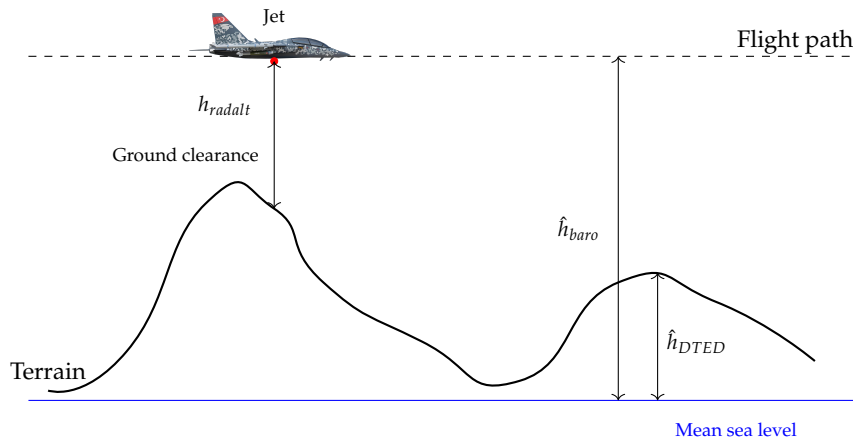


Figure 3. Visualization of measurements.

Batch and recursive solutions have become popular in academia and industry for providing navigation solutions with predictive algorithms using terrain data. The most popular of these algorithms are TERCOM and SITAN, respectively explained in references [2,4]. The SITAN method employs EKF and local terrain linearization algorithms to analyze the terrain profile, with position updates performed as measurements become available. While favorable outcomes may be achieved in certain terrain types and mission profiles, local linearization may not comprehensively address the issue due to the discontinuous and highly nonlinear nature of terrain data in certain areas.

3.3. Terrain Server

Terrain server is a database query system that interpret DTED and provides digital elevation models (DEM). DTED models that is interpreted is a DTED-2 model that has 30m spacing between grid lines. It also has 1 second latitude and longitude spacing. The database for the terrain server is stored onboard computer of the aircraft. It can provide responses with user datagram protocol (UDP) and distributed interactive simulation (DIS). To It has several different functions that handles the LoS, HaT and HoT.

For HaT and HoT queries server uses the provided data to respond with the terrain elevation of a location. It can provide this service for a single location or for a $V \times V$ matrix centered around a location, which is mostly used for our case. LoS is used to predicted the collisions between the aircraft path and the terrain. A straight line is calculated between the aircraft's location and the stated location, and checked against the elevation of the terrain along the path. To provide these services in a real time environment cache system is used to decrease the time required to load the data into the memory. It stores the DTED in the database to the local memory and retrieves information based on these data.

3.4. Filter Formulations

The formulations of the filters used during implantation are based on the formulations described in this section and are included in the text for completeness.

3.4.1. Kalman Filter Formulation

The Kalman filter is a recursive solution to the linear quadratic estimation problem. It estimates the state $\mathbf{x}_k \in \mathbb{R}^n$ of a discrete-time controlled process governed by the linear stochastic difference equation

$$\mathbf{x}_k = \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{G}_{k-1}\mathbf{u}_{k-1} + \mathbf{w}_{k-1}, \quad (7)$$

where $\mathbf{x}_k$ is the state vector at time step k , $\mathbf{F}_{k-1}$ is the state transition matrix, $\mathbf{G}_{k-1}$ is the control input matrix, $\mathbf{u}_{k-1}$ is the control vector, and $\mathbf{w}_{k-1}$ denotes the process noise, assumed to be zero mean Gaussian with covariance $\mathbf{Q}_{k-1}$.

The measurement vector $\mathbf{z}_k \in \mathbb{R}^m$ is defined by

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k, \quad (8)$$

where $\mathbf{H}_k$ represents the observation matrix and $\mathbf{v}_k$ is the measurement noise, also assumed to be zero mean Gaussian with covariance $\mathbf{R}_k$.

The Kalman filter comprises two phases: prediction and update.

In the prediction step, the state and error covariance matrix is forecasted using

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_{k-1} \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{G}_{k-1} \mathbf{u}_{k-1}, \quad (9)$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_{k-1} \mathbf{P}_{k-1|k-1} \mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1}. \quad (10)$$

During the update step, the Kalman gain is computed by

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T \left(\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k \right)^{-1}, \quad (11)$$

after which the state estimate is updated with the new measurement $\mathbf{z}_k$ according to

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \left(\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1} \right), \quad (12)$$

and finally, the error covariance is updated as

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1}. \quad (13)$$

3.4.2. Unscented Kalman Filter (UKF) Formulation

The Unscented Kalman Filter addresses nonlinear state and measurement models by employing the unscented transform. Let the state vector be $\mathbf{x}_k \in \mathbb{R}^n$, with a prior estimate $\hat{\mathbf{x}}_{k-1|k-1}$ and covariance $\mathbf{P}_{k-1|k-1}$. A set of $2n + 1$ sigma points is generated as below, where λ is a scaling parameter.

$$\chi_{k-1}^{(0)} = \hat{\mathbf{x}}_{k-1|k-1}, \quad (14)$$

$$\chi_{k-1}^{(i)} = \hat{\mathbf{x}}_{k-1|k-1} + \left(\sqrt{(n + \lambda) \mathbf{P}_{k-1|k-1}} \right)_i, \quad i = 1, \dots, n, \quad (15)$$

$$\chi_{k-1}^{(i)} = \hat{\mathbf{x}}_{k-1|k-1} - \left(\sqrt{(n + \lambda) \mathbf{P}_{k-1|k-1}} \right)_{i-n}, \quad i = n + 1, \dots, 2n, \quad (16)$$

In the Prediction step, each sigma point is propagated through the nonlinear state transition function $f(\cdot)$:

$$\chi_{k|k-1}^{(i)} = f\left(\chi_{k-1}^{(i)}, \mathbf{u}_{k-1}\right), \quad i = 0, \dots, 2n. \quad (17)$$

The predicted state and covariance are computed as

$$\hat{\mathbf{x}}_{k|k-1} = \sum_{i=0}^{2n} W_m^{(i)} \chi_{k|k-1}^{(i)}, \quad (18)$$

$$\mathbf{P}_{k|k-1} = \sum_{i=0}^{2n} W_c^{(i)} \left(\chi_{k|k-1}^{(i)} - \hat{\mathbf{x}}_{k|k-1} \right) \left(\chi_{k|k-1}^{(i)} - \hat{\mathbf{x}}_{k|k-1} \right)^T + \mathbf{Q}_{k-1}, \quad (19)$$

where $W_m^{(i)}$ and $W_c^{(i)}$ are the weights for the mean and covariance, and $\mathbf{Q}_{k-1}$ is the process noise covariance.

In the Update step, each predicted sigma point is transformed through the measurement function $h(\cdot)$:

$$\gamma_k^{(i)} = h(\chi_{k|k-1}^{(i)}), \quad i = 0, \dots, 2n. \quad (20)$$

The predicted measurement and its covariance are obtained by

$$\hat{\mathbf{z}}_k = \sum_{i=0}^{2n} W_m^{(i)} \gamma_k^{(i)}, \quad (21)$$

$$\mathbf{S}_k = \sum_{i=0}^{2n} W_c^{(i)} (\gamma_k^{(i)} - \hat{\mathbf{z}}_k) (\gamma_k^{(i)} - \hat{\mathbf{z}}_k)^T + \mathbf{R}_k, \quad (22)$$

with $\mathbf{R}_k$ representing the measurement noise covariance. The cross-covariance between the state and the measurement is computed as

$$\mathbf{P}_{xz,k} = \sum_{i=0}^{2n} W_c^{(i)} (\chi_{k|k-1}^{(i)} - \hat{\mathbf{x}}_{k|k-1}) (\gamma_k^{(i)} - \hat{\mathbf{z}}_k)^T. \quad (23)$$

The Kalman gain is then given by

$$\mathbf{K}_k = \mathbf{P}_{xz,k} \mathbf{S}_k^{-1}. \quad (24)$$

Finally, the state estimate and error covariance are updated as

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{z}_k - \hat{\mathbf{z}}_k), \quad (25)$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^T. \quad (26)$$

3.4.3. Particle Filter (PF) Formulation

The particle filter approximates the posterior density

$$p(\mathbf{x}_k | \mathbf{z}_{1:k})$$

by a set of N weighted particles

$$\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}_{i=1}^N,$$

where $\mathbf{x}_k^{(i)} \in \mathbb{R}^n$ is a state candidate and $w_k^{(i)}$ is the corresponding weight.

Assuming a proposal distribution $q(\mathbf{x}_k | \mathbf{x}_{0:k-1}, \mathbf{z}_{1:k})$, the weights are updated as

$$w_k^{(i)} \propto w_{k-1}^{(i)} \frac{p(\mathbf{z}_k | \mathbf{x}_k^{(i)}) p(\mathbf{x}_k^{(i)} | \mathbf{x}_{k-1}^{(i)})}{q(\mathbf{x}_k^{(i)} | \mathbf{x}_{0:k-1}, \mathbf{z}_{1:k})}. \quad (27)$$

A common choice is

$$q(\mathbf{x}_k | \mathbf{x}_{0:k-1}, \mathbf{z}_{1:k}) = p(\mathbf{x}_k | \mathbf{x}_{k-1}^{(i)}), \quad (28)$$

which simplifies the weight update to

$$w_k^{(i)} \propto w_{k-1}^{(i)} p(\mathbf{z}_k | \mathbf{x}_k^{(i)}). \quad (29)$$

The weights are normalized by

$$w_k^{(i)} = \frac{w_k^{(i)}}{\sum_{j=1}^N w_k^{(j)}}. \quad (30)$$

The effective sample size is defined as

$$N_{\text{eff}} = \frac{1}{\sum_{i=1}^N \left(w_k^{(i)}\right)^2}. \quad (31)$$

Resampling is performed when $N_{\text{eff}} < N_{\text{th}}$, where N_{th} is a predefined threshold, by drawing N particles with replacement according to the weights, and resetting the weights to

$$w_k^{(i)} = \frac{1}{N}, \quad i = 1, \dots, N. \quad (32)$$

4. Proposed Algorithms

4.1. Bank of Kalman Filters

In our TAN system, in order to prevent problems caused by high initial errors, different terrain structures affect our altimeter readings and linearization errors, a parallel Kalman filter structure was used, and an appropriate navigation solution was provided by making hard decision making between these filters by artificial intelligence. Here, when the three filter structures that have been explained calculate their separate solutions, the supervisor decides which filter will give better results by looking at certain metrics from the covariance matrices of these solutions, such as trace, determinant, eigenvalue, and the roughness of the terrain profile. The supervisor makes this decision by separating the latitude and longitude solutions of the filters. In other words, in each iteration, supervisor can choose the longitudinal error of one filter while choosing the latitudinal error of the other filter. In this way, by decoupling, errors that may differentially affect the latitude and longitude solutions can be eliminated. Details of EKF, stochastic linearization and parallel filter structures can be examined in [4,5]. Details of UKF and PF are given in [5,54,55].

The terrain data used was converted into a continuous data set using the spatial interpolation method.

4.2. Our TAN Algorithm

We choose the open-loop architecture due to the algorithm's solution being significantly influenced by the terrain profile, which can produce inaccurate results in steep or non-unique terrains. Moreover, improperly correcting errors in a reliable INS solution, which is already equipped with high-precision, military-grade sensors, creates a risk in flight-critical scenarios. The TAN algorithm (see Figure 4) corrects the errors of INS in an open-loop configuration. It makes positional and velocity update by subtracting the estimated error states from the INS predictions in the iteration TAN runs. In this configuration, unlike a closed loop, the navigation solution at the output has no direct relationship with the next iteration. Bank of filters produce error state estimation outputs in the iteration where the radar measurement is received. The AI supervisor also decides which filter structures output to use. As parallel filter structures, EKF with numeric jacobian, classic unscented Kalman filter and particle filter were preferred.

The primary objective of creating this structure is to minimize the risk of finding incorrect solutions in situations where error states are assumed to follow a Gaussian distribution, thereby allowing the numerical Jacobian retrieval process and the unscented transform to perform well in varying contexts, while simultaneously approximating the distribution of particles affected by noise across various terrain profiles. The measurement of the filter can be expressed as:

$$y_k = h_{radalt} = \hat{h}_{baro} - h(\hat{L}_k, \hat{\lambda}_k) \quad (33)$$

In Equation (33), y_k is the estimated terrain height, $h(\hat{L}_k, \hat{\lambda}_k)$ is the estimated height of the terrain obtained from the aircraft's latitude and longitude using the onboard terrain server database. Our proposed measurement function inserts the adjusted latitude and longitude values into the terrain

server to estimate the height of terrain at the true position hypothesis. The input can be formulated as below:

$$\begin{pmatrix} \hat{L} \\ \hat{\lambda} \end{pmatrix} = \begin{pmatrix} L_{\text{INS}} + \delta \hat{L}_{\text{Filter}} \\ \lambda_{\text{INS}} + \delta \hat{\lambda}_{\text{Filter}} \end{pmatrix} \quad (34)$$

In Equation (34), $\begin{pmatrix} \hat{L} \\ \hat{\lambda} \end{pmatrix}$ is the measurement innovation vector between the INS predictions and the estimates of the filter. Where, $\delta \hat{L}$ and $\delta \hat{\lambda}$ are the latitude and longitude measurement error estimations of the error state Kalman Filter.

4.3. AI Supervisor

One way to have more precise navigation is to have better sensors or better algorithms; however, if sensors and algorithms are fixed beforehand, one can cleverly choose which algorithm serves better in which terrain conditions. This is how our AI supervisor is conceived.

As it can be observed from Figure 4, our AI supervisor takes two series of inputs: the covariance matrices of bank of filters and the roughness values of the terrain. Supervisor is a classifier model which selects the predicted best rms value returning filter.

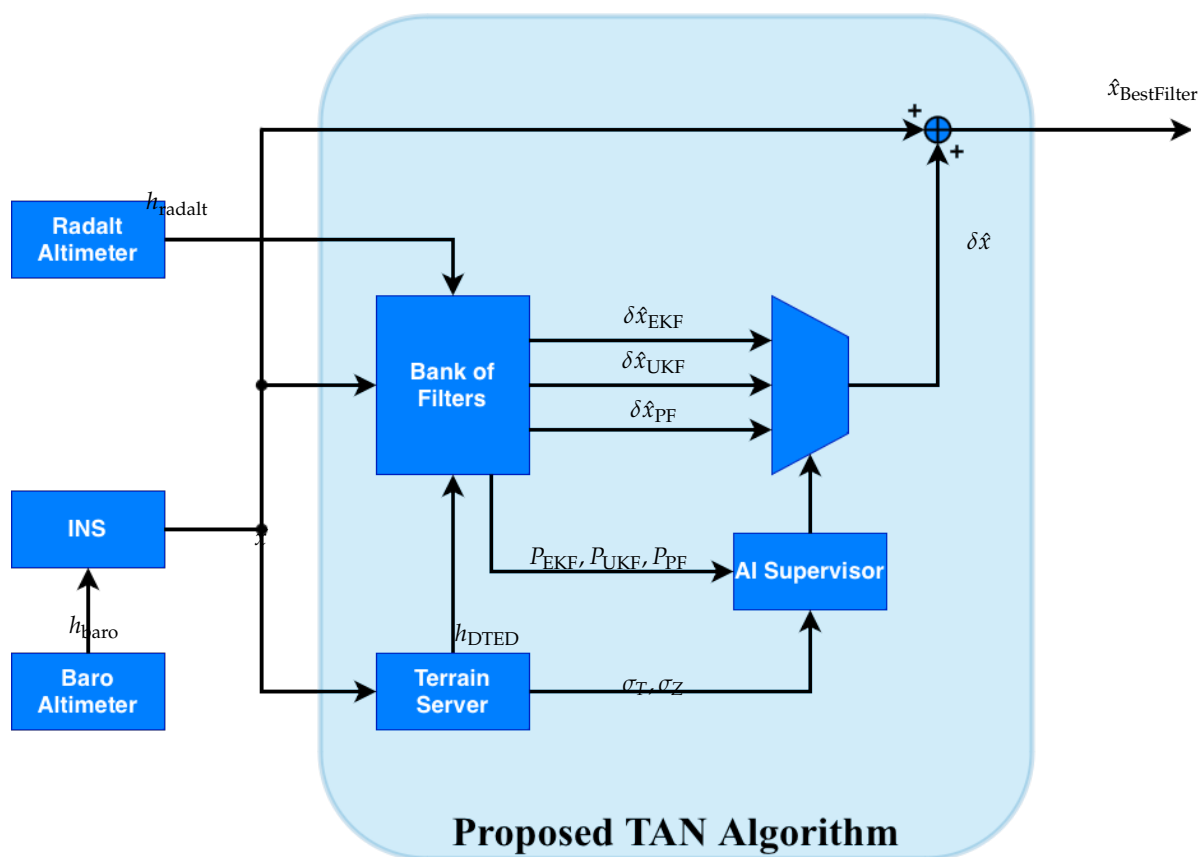


Figure 4. Schematic of the TAN algorithm.

The output of the model are two label structures consisting of three classes separately on the latitude and longitude axes. Thus, a total of nine different filter configurations are obtained, and the model selects the most suitable one from these nine.

To select our model architecture, we used hyper parameter optimization tool Optuna. Layer count, dropout layer batch size, hidden state size are all given as a parameter to our optimization pipeline. Our final model has 3 linear layers to process the input. Each layer has a hidden with fully connected (FC) layers size of 1024 and has RELU non linear activation function. The architecture of the supervisor model is given in Figure 5.

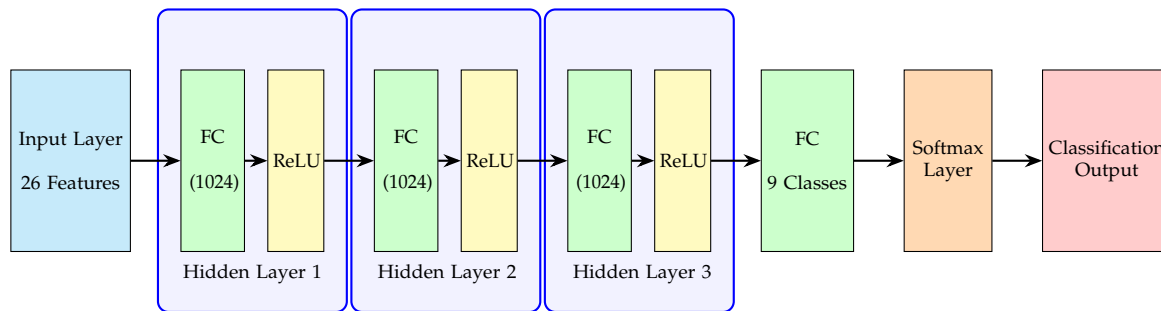


Figure 5. Supervisor neural network architecture.

13 different terrains were selected, including rough, moderate and flat terrain. 10 Monte Carlo flights of 1000 seconds were performed on these terrains. Filter error metrics, covariance data and roughness data where the flight took place were recorded. The model was then trained on this dataset collected from simulations in the environment mentioned in the Section 5.1. 85% of the data was used as training data and 15% was used as test data. Training was carried out for 30 epochs using a cross-entropy loss function applied to both outputs. The model achieved a validation accuracy of 0.88, computed over the joint label space consisting of all nine class combinations. The Figure 6 illustrates the learning dynamics of the proposed architecture.

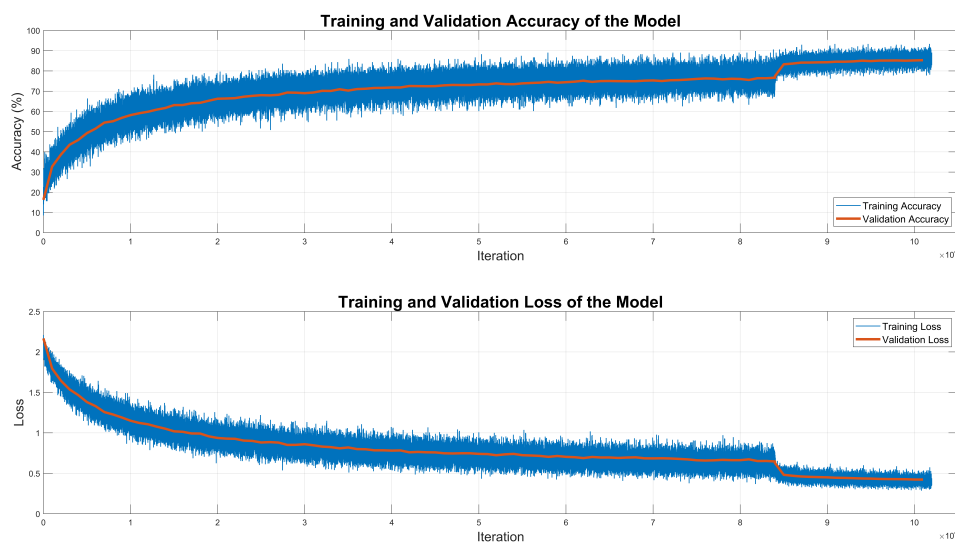


Figure 6. Learning dynamics of the training.

5. Simulations and Results

5.1. Simulations

In this section, simulation environment designed to provide AI training by generating the results of the aforementioned methods and filters will be explained. First of all, three different terrain models were selected for simulations: rough, moderate and flat terrain. Then, flight trajectories were determined. While determining these trajectories path height standard deviation σ_T and path height slope standard deviation σ_Z parameters are examined. Even though both parameters are a metric of roughness, σ_Z is more significant metric in such filtering methods since algorithms such as KF use linearization methods [5]. σ_T corresponds to the standard deviation of the terrain heights on the trajectory. N is the samples, h_i is the terrain height measurement and $\bar{h}$ is the mean height of that terrain.

$$\bar{h} = \frac{1}{N} \sum_{i=1}^N h_i \quad (35)$$

By using h_i and $\bar{h}$ path height standard deviation σ_T can be calculated as below.

$$\sigma_T = \sqrt{\frac{1}{N} \sum_{i=1}^N (h_i - \bar{h})^2} \tag{36}$$

In order to calculate the path height slope standard deviation σ_Z , slopes d_i should be found first and $\bar{d}$ is the mean of the slopes.

$$d_{i-1} = h_{i-1} - h_i \tag{37}$$

$$\bar{d} = \frac{1}{N-1} \sum_{i=1}^{N-1} d_i \tag{38}$$

By using d_i and $\bar{d}$ path height slope standard deviation σ_Z can be calculated as below.

$$\sigma_Z = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N-1} (d_i - \bar{d})^2} \tag{39}$$

Sample σ_T and σ_Z values of chosen terrains are given in the Table 2.

Table 2. Roughness Values of Terrains.

Terrain	σ_T	σ_Z
Rough Terrain	163.48 m	1.29 m
Moderate Terrain	63.93 m	0.41 m
Flat Terrain	$3.21 \times 10^{-13} \text{ m} \sim 0 \text{ m}$	$7.32 \times 10^{-13} \text{ m} \sim 0 \text{ m}$

After the flight trajectories were selected, 1000-second flights were made on the determined paths. A trajectory sample is given in Figure 7. A path was defined for the aircraft such that there was a random acceleration on the horizontal axis. Then, the IMU was modeled to make measurements. The characteristics of the accelerometer, gyroscope and barometric altimeter used in the simulations are given in Table 3.

Table 3. Simulation Sensor Characteristics.

Property	Accelerometer	Gyroscope	Barometric Altimeter
Frequency	100 Hz	100 Hz	100 Hz
Constant Bias	10^{-4} m/s^2	$250^\circ/\text{s}$	0.01 Pa
Noise Density	$10^{-4} \text{ m/s}^2/\sqrt{\text{Hz}}$	$10^{-3}^\circ/\text{s}/\sqrt{\text{Hz}}$	$2 \text{ Pa}/\sqrt{\text{Hz}}$

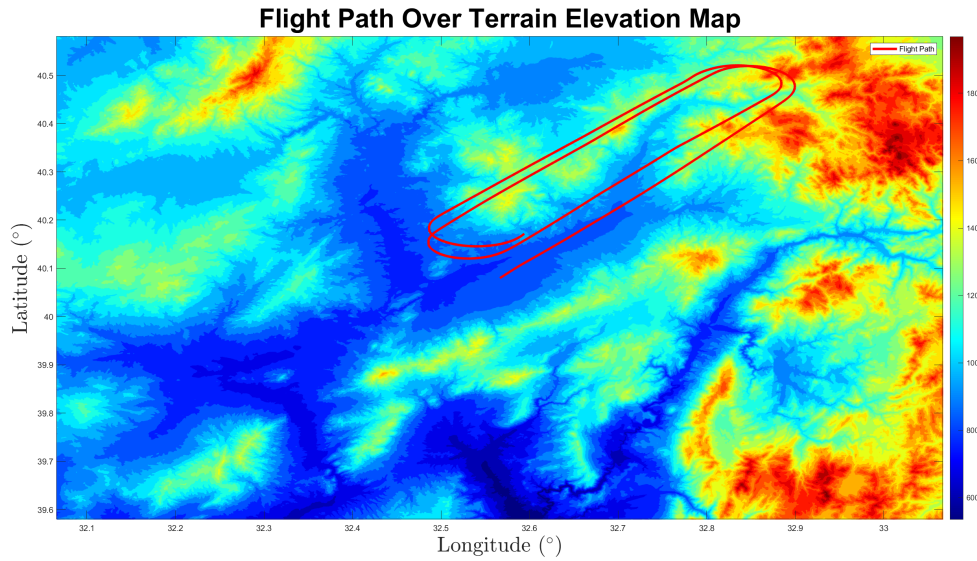


Figure 7. Flight trajectory.

5.2. Validation with Real World FTI Data

The real flight test data (FTI) of Hürjet advanced jet trainer platform projects within Turkish Aerospace was used. The TAN algorithm was run at a frequency of 20 Hz. The data were recorded with a sampling frequency of 128 Hz. There are three different navigation solutions: free inertial solution (FI), GPS solution, and hybrid solution. The hybrid solution is a filtered version of the FI and GPS solutions and produces more accurate and reliable results compared to both solutions. Therefore, hybrid solution was accepted as "ground truth" while evaluating the performance of TAN algorithm. FI navigation solution shows drift over time due to various errors as explained in Section 3.1 and constitutes the input data that are being tried to improve. Similarly, radalt measurements were also taken from the aircraft. When the aircraft makes high angles of roll or yaw, radalt cannot measure the HaT correctly and produces invalid data. For this reason, when the radalt measurement could not be produced, only the prediction step was applied and no correction was made.

5.3. Results

In order to compare to results of the trials two parameters are examined: absolute error and circular error probable (CEP). Absolute error and CEP [56] are computed as below.

$$\text{Error}_i = \sqrt{e_{x_i}^2 + e_{y_i}^2} \quad (40)$$

$$C_0 = \begin{cases} 0.562\sigma_x + 0.615\sigma_y, & \sigma_y < \sigma_x \text{ and } \frac{\sigma_y}{\sigma_x} < 0.28 \\ 0.562\sigma_y + 0.615\sigma_x, & \sigma_x < \sigma_y \text{ and } \frac{\sigma_x}{\sigma_y} < 0.28 \\ 1.1774 \frac{(\sigma_x + \sigma_y)}{2}, & \text{else} \end{cases} \quad (41)$$

$$CEP = \sqrt{C_0^2 + \mu_x^2 + \mu_y^2} \quad (42)$$

/comment/textbfReviewer 2, Comment 3:The experimental results are based on only a few tests, and there is no information about variability or uncertainty in the results. Important metrics like std or confidence intervals are missing.

/comment/textbfAnswer to Reviewer 2, Comment 3Positional uncertainty is already expressed through the Circular Error Probable (CEP) metric. This paper targets the concept's first experimental demo; the flight-test campaign is still in progress and the complete V&V statistics will be presented in a follow-up publication.

Here e_{x_i} and e_{y_i} are the position errors between the ground truth and the TAN output. The standard deviations of the position errors between the ground truth and the TAN output is denoted as σ_x and σ_y . Also μ_x and μ_y are the means of the position errors between the ground truth and the TAN output. CEP is calculated by using coefficient C_0 .

The FTI data absolute error results are given in Figure 8 and 9 for two different flights. Parallel filter solutions give better navigation solutions than INS only solutions as proposed.

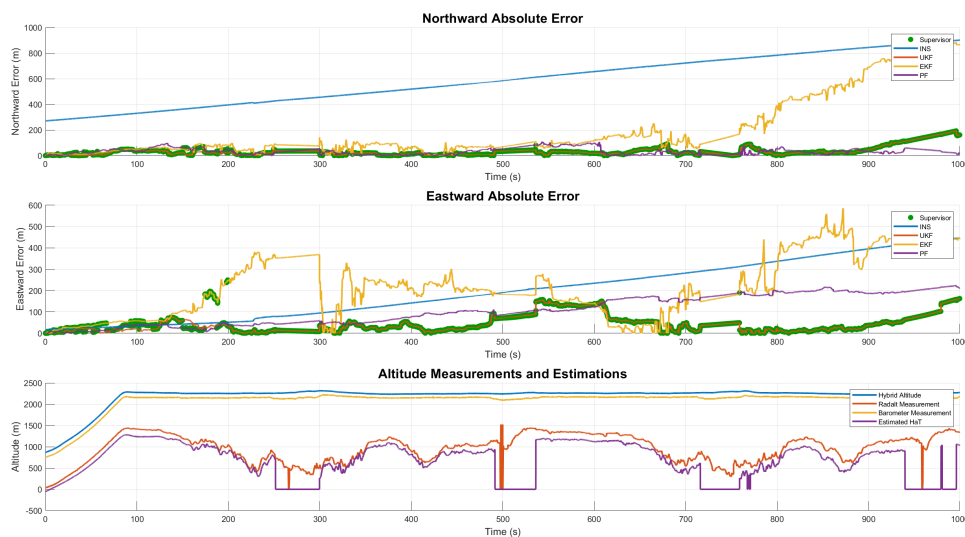


Figure 8. Absolute Error results.

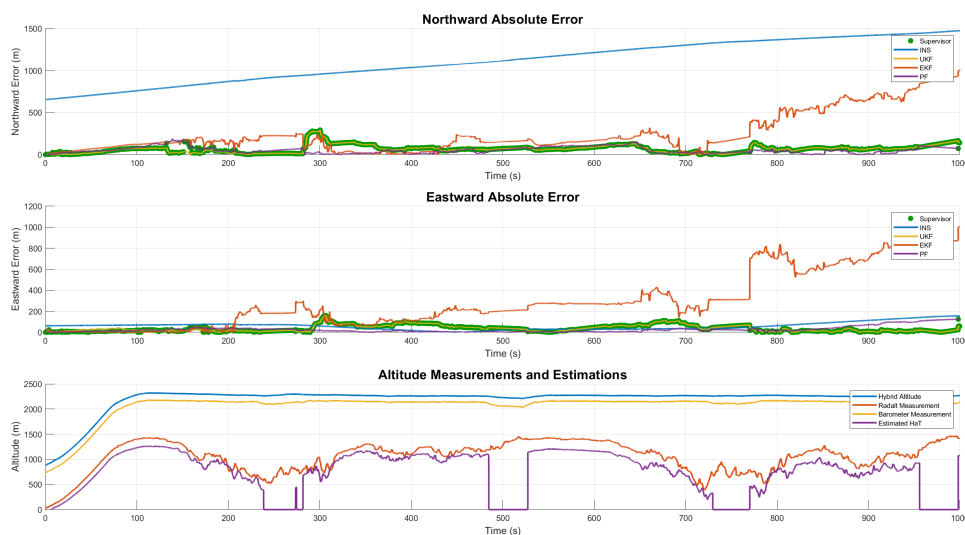


Figure 9. Absolute Error results.

In Figure 8 and 9 The reason for the dips in the Estimated Line between 250-300, 480-540, 720-760 and 930-1000 seconds is that the aircraft is performing an agile maneuver at these moments. This can be called radalt shadowing due to roll and pitch maneuvers. Moreover, there is a variable called "radalt attitude angle" that is dependent on the roll and pitch angles of the aircraft. When this variable is below a certain value, measurements from radalt can be accepted as correct. On the specified times that listed above, the attitude angle is above threshold and the correction to the bank of filters can not be accepted. In order to visualize this type of maneuvers, estimated HaT value is set to zeros.

This situations can also be validated from the dip to 0 in the Radalt measurements in the intervals mentioned above.

Table 4. CEP Results of Simulations.

CEP_{INS}	CEP_{EKF}	CEP_{UKF}	CEP_{PF}	$CEP_{Supervisor}$
184.782 m	115.211 m	21.158 m	40.654 m	21.988 m

6. Conclusions

Designed for the Hürjet advanced jet trainer, the paper presents a new artificial intelligence-enhanced Terrain-Aided Navigation (TAN) system. The TAN system improves navigation precision and computation efficiency by including adaptive filtering techniques, an advanced terrain server, and an artificial intelligence supervisory module that dynamically alternately between multiple filtering schemes (EKF, UKF, PF). Comparatively to conventional navigation methods (INS-only solutions), the proposed hybrid TAN method significantly increases low-altitude navigation efficacy. Emphasizing improved navigational accuracy even in GPS-denied environments, Monte Carlo simulations and empirical flight test instrumentation (FTI) data support the efficacy and viability of the AI-driven methodology. With an accuracy of around 82% in appropriate filter selection, the AI supervisor efficiently finds acceptable filters based on terrain roughness and ambient factors, hence significantly lowering absolute error values. This study provides important new insights on the issues and operational aspects related to avionics certification by using real flight data from the Hürjet platform to link theoretical modeling with pragmatic application. We understand the importance of upcoming projects and stress the change from simulation to real-time onboard applications as well as a complete certification process for the TAN integration with aircraft control systems. Appropriate for inclusion into advanced avionics systems, the work represents a notable advancement in autonomous, trustworthy, and accurate terrain-aided navigation procedures.

Author Contributions: Conceptualization, Ü.C.B. and G.İ.; methodology, Ü.C.B.; software, B.T and İ.E.U.; validation, İ.E.U., investigation, B.T. and İ.E.U.; resources, N.A.G.; data curation, İ.E.U.; writing—original draft preparation, Ü.C.B.; writing—review and editing, Ü.C.B., B.T., İ.E.U. and G.İ.; visualization, İ.E.U.; supervision, G.İ.; project administration, Ü.C.B.; funding acquisition, Ü.C.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding

Acknowledgments: The project was realized as part of Hürjet advanced jet trainer platform’s artificial intelligence concept projects within Turkish Aerospace. The domain experts were employees of the company. The simulation environment and flight dynamics model of the jet were provided from the company.

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

References

1. Bekar, U.C.; Tanyeri, B.; Canoglu, A.S.; Uslu, I.E.; Gungor, N.A.; Inalhan, G. AI-assisted Digital Terrain System for an Advanced Jet Trainer. In Proceedings of the 2024 AIAA DATC/IEEE 43rd Digital Avionics Systems Conference (DASC), 2024, pp. 1–10. <https://doi.org/10.1109/DASC62030.2024.10749144>.
2. Golden, J.P. Terrain contour matching (TERCOM): a cruise missile guidance aid. In Proceedings of the Image processing for missile guidance. SPIE, 1980, Vol. 238, pp. 10–18.
3. Hostetler, L.; Beckmann, R. Sandia inertial terrain-aided navigation system. *Unknown* **1977**.
4. Hostetler, L.; Andreas, R. Nonlinear Kalman filtering techniques for terrain-aided navigation. *IEEE Transactions on Automatic Control* **1983**, 28, 315–323.
5. Ekütekin, V. Navigation and control studies on cruise missiles. PhD thesis, Middle East Technical University, 2007.

6. Johnson, N.; Tang, W.; Howell, G. Terrain aided navigation using maximum a posteriori estimation. In Proceedings of the IEEE Symposium on Position Location and Navigation. A Decade of Excellence in the Navigation Sciences. IEEE, 1990, pp. 464–469.
7. Erhui, W.; Guohua, G.; Lincheng, S.; Wensen, C. A probability-based terrain-aided navigation approach and its relative terrain navigability analysis. In Proceedings of the Proceedings of the IEEE International Conference on Industrial Technology (ICIT'96). IEEE, 1996, pp. 781–785.
8. Qingtang, F.; Lincheng, S.; Wenseng, C. Terrain aided navigation using PDAF. In Proceedings of the IEEE International Conference on Robotics, Intelligent Systems and Signal Processing, 2003. Proceedings. 2003. IEEE, 2003, Vol. 2, pp. 1063–1068.
9. Zhou, H.; Zhang, C.y. Terrain aided navigation based on computer vision. In Proceedings of the Proceedings of the 4th World Congress on Intelligent Control and Automation (Cat. No. 02EX527). IEEE, 2002, Vol. 3, pp. 2164–2166.
10. Chen, Z.; Yu, P. Model study for terrain aided navigation systems. In Proceedings of the [1992] Proceedings of the IEEE International Symposium on Industrial Electronics. IEEE, 1992, pp. 848–849.
11. Wang, W.; Chen, Z. Error model identification on digital map of TAN system based on EKF. In Proceedings of the Proceedings of 1994 IEEE International Conference on Industrial Technology-ICIT'94. IEEE, 1994, pp. 818–822.
12. Yu, P.j.; Chen, Z.; Hung, J.C. Performance evaluation of six terrain stochastic linearization techniques for TAN. In Proceedings of the Proceedings of the IEEE 1991 National Aerospace and Electronics Conference NAECON 1991. IEEE, 1991, pp. 382–388.
13. Baird, C.A.; Snyder, F.B.; Beierle, M. Terrain-aided altitude computations on the AFTI/F-16. In Proceedings of the IEEE Symposium on Position Location and Navigation. A Decade of Excellence in the Navigation Sciences. IEEE, 1990, pp. 474–481.
14. Hollowell, J. Heli/SITAN: A terrain referenced navigation algorithm for helicopters. Technical report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 1990.
15. Pei, Y.; Chen, Z.; Hung, J. BITAN-II: an improved terrain aided navigation algorithm. In Proceedings of the Proceedings of the 1996 IEEE IECON. 22nd International Conference on Industrial Electronics, Control, and Instrumentation. IEEE, 1996, Vol. 3, pp. 1675–1680.
16. Chen, Z. BUAA Inertial Terrain Aided Navigation. *Icas92* **1992**.
17. Bergman, N.; Ljung, L. Point-mass filter and Cramer-Rao bound for terrain-aided navigation. In Proceedings of the Proceedings of the 36th IEEE Conference on Decision and Control. IEEE, 1997, Vol. 1, pp. 565–570.
18. Nordlund, P.J.; Gustafsson, F. Recursive estimation of three-dimensional aircraft position using terrain-aided positioning. In Proceedings of the 2002 IEEE International Conference on Acoustics, Speech, and Signal Processing. IEEE, 2002, Vol. 2, pp. II–1121.
19. Enns, R.; Morrell, D. Terrain-aided navigation using the Viterbi algorithm. *Journal of Guidance, Control, and Dynamics* **1995**, *18*, 1444–1449.
20. Dezert, J. Improvement of strapdown inertial navigation using PDAF. *IEEE transactions on Aerospace and Electronic Systems* **1999**, *35*, 835–856.
21. Madhavan, R.; Durrant-Whyte, H.; Dissanayake, G. Natural landmark-based autonomous navigation using curvature scale space. In Proceedings of the Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292). IEEE, 2002, Vol. 4, pp. 3936–3941.
22. Bruder, S.B.; Wedeward, K. Terrain aided INS robot navigation: a deferred decision making approach. In Proceedings of the 42nd Midwest Symposium on Circuits and Systems (Cat. No. 99CH36356). IEEE, 1999, Vol. 1, pp. 135–139.
23. Morisue, F.; Ikeda, K. Evaluation of map-matching techniques. In Proceedings of the Conference Record of papers presented at the First Vehicle Navigation and Information Systems Conference (VNIS'89). IEEE, 1989, pp. 23–28.
24. Hongbo, X.; Yan, T.; JianZhong, S.; Jinwen, T.; Jian, L. Terrain matching based on imaging laser radar. In Proceedings of the 6th International Conference on Signal Processing, 2002. IEEE, 2002, Vol. 2, pp. 1043–1046.
25. Bevington, J.E.; Marttila, C.A. Precision aided inertial navigation using sar and digital map data. In Proceedings of the IEEE Symposium on Position Location and Navigation. A Decade of Excellence in the Navigation Sciences. IEEE, 1990, pp. 490–496.
26. Metzger, J.; Wendel, J.; Trommer, G.F.; Tumbregel, F.; Taddiken, B. Hybrid terrain referenced navigation system using a bank of Kalman filters and a comparison technique. In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, 2004, pp. 1–12.

27. Yoo, Y.; Lee, W.; Lee, S.; Park, C.; Kwon, J. Improvement of TERCOM aided inertial navigation system by velocity correction. In Proceedings of the Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium. IEEE, 2012, pp. 1082–1087.
28. Wilkinson, N.; Brookes, T.; Price, A.; Godfrey, M. Latest developments of the TERPROM® digital terrain system 2009. In Proceedings of the Joint Navigation Conference, Orlando, Florida, USA, 2009.
29. Lee, J.; Sung, C.K.; Oh, J.; Han, K.; Lee, S.; Yu, M.J. A Pragmatic Approach to the Design of Advanced Precision Terrain-Aided Navigation for UAVs and Its Verification. *Remote Sensing* **2020**, *12*. <https://doi.org/10.3390/rs12091396>.
30. Navon, D.; Rivlin, E.; Rotstein, H. A Robust Approach to Vision-Based Terrain-Aided Localization. *NAVIGATION: Journal of the Institute of Navigation* **2025**, *72*, [<https://navi.ion.org/content/72/1/navi.683.full.pdf>]. <https://doi.org/10.33012/navi.683>.
31. Bijjahalli, S.; Sabatini, R.; Gardi, A. Advances in intelligent and autonomous navigation systems for small UAS. *Progress in Aerospace Sciences* **2020**, *115*, 100617.
32. Lee, J.; Sung, C.; Nam, S. CRLB analysis for a robust TRN based on a combination of RNN and PF. *International Journal of Aeronautical and Space Sciences* **2020**, *21*, 265–276.
33. González-García, J.; Gómez-Espinoza, A.; Cuan-Urquizo, E.; García-Valdovinos, L.G.; Salgado-Jiménez, T.; Escobedo Cabello, J.A. Autonomous underwater vehicles: Localization, navigation, and communication for collaborative missions. *Applied sciences* **2020**, *10*, 1256.
34. Kim, Y.; Back, S.; Song, D.; Lee, B.Y. Aerial Map-Based Navigation by Ground Object Pattern Matching. *Drones* **2024**, *8*, 375.
35. Kim, S.; Park, J.; Bang, H. Enhanced Terrain-Referenced Navigation Through Adaptive Radar Altimeter Error Estimation with Monte Carlo Sampling. *International Journal of Aeronautical and Space Sciences* **2025**, pp. 1–21.
36. Choe, Y.; Song, J.W.; Park, C.G. Lightweight marginalized particle filtering with enhanced consistency for terrain referenced navigation. *IEEE Transactions on Aerospace and Electronic Systems* **2021**, *58*, 2493–2504.
37. Canciani, A.J. Magnetic navigation on an F-16 aircraft using online calibration. *IEEE Transactions on Aerospace and Electronic Systems* **2021**, *58*, 420–434.
38. Wang, B.; Ma, Z.; Huang, L.; Deng, Z.; Fu, M. A filtered-marine map-based matching method for gravity-aided navigation of underwater vehicles. *IEEE/ASME Transactions on Mechatronics* **2022**, *27*, 4507–4517.
39. Yuan, G.; Zhang, H.; Yuan, K.; Zhu, L. Improved SITAN algorithm in the application of aided inertial navigation. In Proceedings of the Proceedings of 2012 International Conference on Measurement, Information and Control. IEEE, 2012, Vol. 2, pp. 922–926.
40. Wang, R.; Wang, J.; Li, Y.; Ma, T.; Zhang, X. Research Advances and Prospects of Underwater Terrain-Aided Navigation. *Remote Sensing* **2024**, *16*, 2560.
41. Keller, B.A.; Putman, N.F.; Grubbs, R.D.; Portnoy, D.S.; Murphy, T.P. Map-like use of Earth's magnetic field in sharks. *Current Biology* **2021**, *31*, 2881–2886.
42. Newman, P.; Durrant-Whyte, H. Using sonar in terrain-aided underwater navigation. In Proceedings of the Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146). IEEE, 1998, Vol. 1, pp. 440–445.
43. Sung, C.K.; Lee, S.J. Moment matched Gaussian kernel and region representative likelihood for performance improvement of PMF-based TRN. *International Journal of Control, Automation and Systems* **2020**, *18*, 1691–1704.
44. Luque, I.D. Planes de contingencia GNSS a largo plazo. *Unknown* **2020**.
45. Rodriguez, S.; Brashar, C.; Haydon, T.C.E.; Luong, A.; Pihlaja, C. PNT Resilience RFI Response. Technical report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2020.
46. Choe, Y.; Park, C.G. Point-mass filtering with boundary flow and its application to terrain referenced navigation. *IEEE Transactions on Aerospace and Electronic Systems* **2021**, *57*, 3600–3613.
47. Li, R.K. *All Source Positioning, Navigation and Timing*; Artech House, 2020.
48. Carroll, J.D.; Canciani, A.J. Terrain-referenced navigation using a steerable-laser measurement sensor. *Navigation* **2021**, *68*, 115–134.
49. Han, Y.; Wang, B.; Deng, Z.; Fu, M. An improved TERCOM-based algorithm for gravity-aided navigation. *IEEE Sensors Journal* **2016**, *16*, 2537–2544.
50. Wang, B.; Yu, L.; Deng, Z.; Fu, M. A particle filter-based matching algorithm with gravity sample vector for underwater gravity aided navigation. *IEEE/ASME Transactions on Mechatronics* **2016**, *21*, 1399–1408.
51. Wu, L.; Ma, J.; Tian, J. A self-adaptive unscented Kalman filtering for underwater gravity aided navigation. In Proceedings of the Ieee/ion position, location and navigation symposium. IEEE, 2010, pp. 142–145.

52. Bao, B.; Hua, Y.; Wang, R.; Li, D. Quantum-Based Magnetic Field Sensors for Biosensing. *Advanced Quantum Technologies* **2023**, *6*, 2200146.
53. Noureldin, A.; Karamat, T.; Georgy, J. *Fundamentals of Inertial Navigation, Satellite-based Positioning and their Integration*; Springer Berlin Heidelberg, 2012.
54. Yiğit, H. Arazi verisine dayalı konumlandırma ve seyrüsefer sistemi tasarlanması. Master's thesis, Havacılık ve Uzay Teknolojileri Enstitüsü, 2012.
55. Turan, B. Comparison of nonlinear filtering methods for terrain referenced aircraft navigation. In Proceedings of the 2020 IEEE/ION Position, Location and Navigation Symposium (PLANS). IEEE, 2020, pp. 144–149.
56. Wang, Y.Y.; Jia, X.R.; Yang, G.L.; Wang, Y.M. Comprehensive CEP evaluation method for calculating positioning precision of navigation systems. *Applied Mechanics and Materials* **2013**, *341*, 955–960.

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.