Peer-reviewed version available at *Remote Sens.* **2020**, *12*, 1396; doi:10.3390/rs12091396

Article

A Pragmatic Approach to the Design of Advanced Precision Terrain Aided Navigation for UAVs and Its Verification

Jungshin Lee [1]*, Chang-Ky Sung [1], Juhyun Oh [1], Kyungjun Han [1], Sangwoo Lee [1] and Myeong-Jong Yu [1]

[1] The 3rd R&D Institute 3rd Center, Agency for Defense Development, Daejeon 34061, Republic of Korea; einstein@add.re.kr (C.-K.S.); juhyunoh@add.re.kr (J.O.); kjhan@add.re.kr (K.H.); 245489@add.re.kr (S.L.); mjyu@add.re.kr (M.-J.Y.)

* Correspondence: jslee0534@add.re.kr (J.L.); Tel.: +82-42-821-3728

Abstract: Autonomous unmanned aerial vehicles (UAVs) require highly reliable navigation information. Generally, navigation systems with the inertial navigation system (INS) and global navigation satellite system (GNSS) have been widely used. However, the GNSS is vulnerable to jamming and spoofing. The terrain referenced navigation (TRN) technique can be used to solve this problem. In this study, to obtain reliable navigation information even if a GNSS is not available or the degree of terrain roughness is not determined, we propose a federated filter based INS/GNSS/TRN integrated navigation system. we also introduce a TRN system that combines batch processing and an auxiliary particle filter to ensure stable flight of UAVs even in a long-term GNSS-denied environment. As an altimeter sensor for the TRN system, we use an interferometric radar altimeter (IRA) to obtain reliable navigation accuracy in high altitude flight. In addition, a parallel computing technique with general-purpose computing on graphics processing units (GPGPU) is applied to process a high resolution terrain database and a nonlinear filter in real time on board. Finally, we verify the performance of the proposed system through software-in-the-loop (SIL) tests and captive flight tests in a GNSS unavailable environment.

Keywords: Terrain Referenced Navigation (TRN); Federated Filter; Interferometric Radar Altimeter (IRA); Batch Processing; Auxiliary Particle Filter; Digital Elevation Model (DEM); Captive Flight Test

---

1. Introduction

Recently, much research has been conducted on surveillance and reconnaissance technologies for autonomous unmanned aerial vehicles (UAVs). These UAVs require navigation information with utmost reliability and survivability. Traditionally, a navigation algorithm integrated with an inertial navigation system (INS) and a global navigation satellite system (GNSS) has been widely used [1], [2]. However, the GNSS cannot operate independently as it provides information with outside help and is vulnerable to hostile jamming and spoofing. To overcome such weaknesses, the vision-based navigation technique is commonly used for UAVs [3]-[5], but a problem with the camera adapting to the changing conditions of a real flight is not completely solved yet, so vision-based navigation is mainly used in drone systems that fly at low altitude for a relatively short time [6] and autonomous driving vehicles [7]. As another technique that can resolve the problems of GNSS, the terrain referenced navigation (TRN) can be used.

TRN is a navigation technology that estimates the aircraft's precise position by comparing the altitude measured by an altimeter with the loaded digital elevation model (DEM) [8], [9]. Existing TRN techniques use a radio altimeter to measure the distance between the ground level and the aircraft [10]. If a radio altimeter (RA) is used at a low altitude, a precise altitude measurement can be obtained because the radio wave reflection area is narrow, but if the altitude is higher than 800m, the radio wave reflection area is wide, and that increases the uncertainty of the reflected location of the wave, and errors in the measured altitude increase. This makes the RA unsuitable for UAVs with reconnaissance missions at high altitudes. To overcome such deficiencies, Honeywell developed a TRN technique called precision terrain aided navigation (PTAN) that employed an interferometric radar altimeter (IRA) and was designed for the purpose of applying to the Tomahawk cruise missiles [11], [12]. The IRA outputs the range and cross-track angle from aircraft to the nearest point on the zero Doppler line using two or more antennas [11]. Since the output of the IRA can be converted to three-dimensional relative distances on an east-north-up (ENU) coordinate system, TRN based on an IRA can be used at high altitudes. In this paper, we present a domestically developed IRA that can operate up to an altitude of 10km.

On the other hand, to ensure the survivability and high reliability of UAVs, a TRN system should be able to obtain its current position even when a GNSS is unavailable for a long time and still guarantee a precise navigation performance similar to that of the INS/GNSS integrated navigation system. In this study, to ensure the survivability of UAVs when a GNSS is unavailable for a long time, we applied batch processing for the acquisition mode. The acquisition mode is a TRN technique that roughly estimates the current position when it has a large initial position error. To guarantee high reliability, we used a particle filter (PF) for the tracking model and a high resolution DEM. The tracking mode is a TRN technique that preciously estimates the current position. To estimate precise location information from irregularly changing terrains, nonlinear estimation problems must be solved in real time. Since the extended Kalman filter (EKF)-based TRN may diverge due to local linearization [13], we selected the PF for the tracking mode. In addition, a parallel computing technology using general-purpose computing on graphics processing units (GPGPU) was applied to implement the TRN algorithm based on PF and process the high resolution DEM in real time on board.

As mentioned above, unlike the GNSS, TRN is not affected by external interference, but its performance can be degraded in flat and repetitive terrain [14]. In order to mitigate individual deficiencies of TRN and GNSS and to ensure the high stability of UAVs, INS/GNSS/TRN integrated navigation is constantly being studied [15], [16]. In Refs. [15] and [16], the INS/GNSS/TRN integrated navigation based on centralized filter was presented. However, it is very difficult to determine the logic of switching aiding sources or the conditions of determining priorities in a real flight environment. In this study, we designed a INS/GNSS/TRN based on a federated filter that does not need to consider these problems. The federated filter is widely used in information fusion navigation systems because of its high computing efficiency [17]. All navigation sources are processed by local filters before they are merged into a master filter. The master filter of the INS/GNSS/TRN integrated navigation system uses the estimates and covariances of two local filters – INS/GNSS and INS/TRN aided filters. The local filters are designed with an EKF, which is a feedforward type of filter and is composed of the 18st state variables, and the INS/GNSS aided navigation includes the barometer error compensation method.

Finally, before the proposed advanced precision terrain aided navigation system (AP-TAN) is applied to UAVs, it is necessary to complete a performance verification step on a real aircraft platform. Therefore, in this study, the captive flight tests were performed under various conditions to verify the proposed system. Since it is expensive to carry out a captive flight test, various simulations and software-in-the-loop (SIL) experiments using the IRA, GNSS, Barometer simulators were performed in advance with the flight tests.

In section 2, we introduce the design of the INS/GNSS/TRN integrated navigation algorithm based on the federated filter. First, we introduce the overall design of the proposed AP-TAN. Next, we explain how the master filter of the integrated navigation that uses the estimates and covariances of two local filters - INS/GNSS and INS/TRN aided filters. We also describe the barometer error compensation method for the INS/GNSS aided navigation as well as the design of a semi-tightly coupled INS/TRN aided navigation. In section 3, we propose the design of a hybrid TRN based on the combination of batch processing and a PF using an IRA and a high resolution DEM. First, the three-dimensional relative distance from the aircraft to a nearest point was derived from an IRA measurement. Next, TRN algorithm based on batch processing for the acquisition mode and a PF for the tracking mode is explained. We also briefly describe the parallel computing method for implementing the PF and processing the high resolution DEM in real time on board. In section 4, we show simulation results in various terrain conditions with the SIL experiments to verify the proposed three-dimensional terrain based integrated navigation system. Finally, we describe the captive flight tests that were conducted to analyze the navigation performance of the proposed system.

2. Design of the proposed AP-TAN System

2.1. Overall Design of the AP-TAN System

As shown in Figure 1, the proposed AP-TAN consists of:

- A inertial measurement unit (IMU) consists of 3 ring laser gyros (RLGs) and 3 silicon pendulum accelerometers to measure the motion of the vehicle without external helps.
- A parallel process unit (PPU) processes the outputs of a IMU and auxiliary sensors. It also performs navigation algorithms including pure navigation, INS/GNSS aided navigation, TRN, INS/TRN aided navigation and INS/GNSS/TRN integrated navigation.
- A GNSS receiver provides the navigation information of the vehicle using GNSS outputs
- A barometer provide the altitude information of the vehicle by measuring the air pressure from the average sea level.
- A IRA provides 3D terrain elevation information for the nearest point.
- A control and display unit (CDU) gives control commands to the PPU and monitors the navigation results in real time.
- A signal and power distributor (SPD) delivers the commands from the CDU and the outputs of the auxiliary sensors to the PPU. It also delivers the navigation results from the PPU to the CDU. In addition, it provides adequate power for the entire system components.

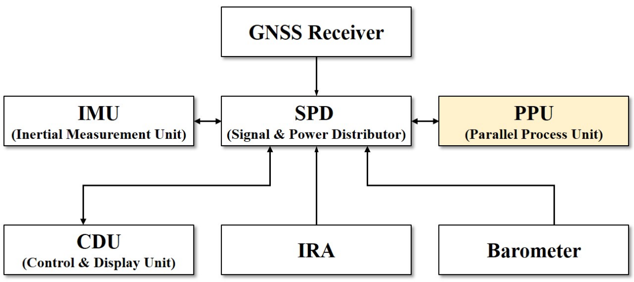


Figure 1. Overview of an AP-TAN

In this study, we mainly deal with the software of the proposed AP-TAN. However, since pure navigation and GNSS navigation are already widely known, they are omitted. In this section, we introduce the INS/GNSS/TRN integrated navigation system.

2.2. Design of an INS/GNSS/TRN Integrated Navigation System

The conventional INS has the advantage of providing continuous navigation information without external assistance. However, there is a disadvantage in that the navigation error increases due to sensor errors accumulated over time. The INS/GNSS or INS/TRN aided navigation can overcome this disadvantage. However, GNSS is vulnerable to jamming and spoofing, while TRN performance can be degraded in flat or repetitive terrain. In this study, in an effort to improve the performance

Peer-reviewed version available at Remote Sens. 2020, 12, 1396; doi:10.3390/rs12091396

and ensure high reliability of the navigation system, we designed an INS/GNSS/TRN integrated navigation system based on a federated filter.

To achieve integrated navigation, an appropriate design of an information fusion filter is necessary, it can be divided into centralized and federated filters according to the method of processing measurements. The centralized filter is a method of selectively using the measurements of sensors in one filter, while the federated filter is composed of local filters using the measurements of each sensor and fuses the weighted estimates of local filters using their covariances [18]-[19]. With a centralized filter, an optimal solution can be found, but if a sensor fails, the performance degrades. Moreover, implementing the logic to properly select measurements in one filter is not easy. In addition, proper fusion of information from auxiliary sensors in one filter requires much experience. On the other hand, with a federated filter, only estimates and covariances are used, which reduces the number of variables passed to the master filter and reduces the burden of designing selective decision logic of measurements. Therefore, in this section, we describe our design of a federated filter-based INS/GNSS/TRN integrated navigation using INS/GNSS aided navigation and INS/TRN aided navigation filters as local filters. The local filters are designed with an EKF, which is an indirect feedforward type and composed of the 18th state variables. Along with two local filters, the INS/GNSS aided navigation includes the barometer error compensation method.

2.2.1. Design of the Master Filter based on the Federated Filter

The federated filter consists of local filters and a master filter, and has a structure that obtains the master filter estimate by fusing the estimates of the local filters. This structure allows local filters to be designed independently to prevent cross contamination between them and to reduce the amount of computation since there is no need to calculate their mutual covariance [18]. The federated filter can be divided into the fusion-reset (FR) mode for distributing the fused estimates and covariances of the master filter to the local filters and the fault tolerant (FT) mode for independently designing the local filters [19]. In this system, we designed a federated filter-based integrated navigation system in the FT mode that does not guarantee an optimal solution but is robust against sensor failure. Figure 2 shows a block diagram of the federated filter-based INS/GNSS/TRN integrated navigation with INS/GNSS aided navigation and INS/TRN aided navigation as local filters.

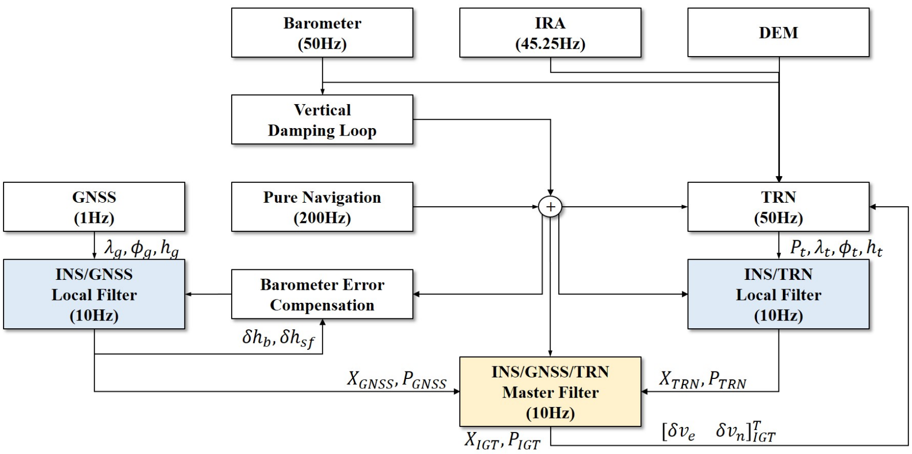


Figure 2. Block diagram of federated filter-based INS/GNSS/TRN integrated navigation

The pure navigation in Figure 2 is a navigation algorithm that calculates the position, velocity, and attitude using the outputs of the gyro and accelerometer. The position output of pure navigation is used in the time propagation of the TRN filter. In addition, the position, velocity and attitude information of pure navigation is used to update the system matrix of the local filters. However, since the altitude error estimated by pure navigation increases exponentially with time, it is replaced by the output of the vertical damping loop with the barometer to obtain a stable vertical navigation

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 March 2020 doi:10.20944/preprints202003.0442.v1

Peer-reviewed version available at Remote Sens. 2020, 12, 1396; doi:10.3390/rs12091396

performance. As shown in Figure 2, the local and master filters are updated at 10Hz periods. In INS/GNSS aided navigation, GNSS measurements are updated every 1Hz, so the filter is designed to perform only the time propagation in the intervals between measurement updates. The longitude, latitude, and altitude estimates of GNSS, λ_g, ϕ_g, and h_g, are used as measurements for the INS/GNSS aided navigation, and the diagonal elements of the measurement noise covariance matrix are set at fixed values. In this system, the IRA and barometer were used to obtain altimeter measurements to compare with the DEM altitude. As shown in Figure 2, the IRA measurement is updated at 45.25Hz cycles, so it is synchronized by latching pure navigation results. When the measurement is not updated within the TRN filter period of 50Hz, it is designed to perform only time propagation. As mentioned in the previous section, the TRN of this system used batch processing as an acquisition mode for estimating the approximate position when the initial position error was large, and an APF as a tracking mode for estimating the precise position. INS/TRN aided navigation is designed to be performed only in the APF-based tracking mode. The INS/TRN aided navigation uses the longitude, latitude, and altitude estimates of TRN λ_t, ϕ_t, and h_t and their covariance matrix P_t as the measurement and the measurement noise covariance matrix, respectively.

The INS/GNSS/TRN integrated navigation algorithm using INS/GNSS aided navigation and INS/TRN aided navigation as local filters is as follows:

Case 1. The state variables and covariances when GNSS is not available

$$P_{IGT}(k) = P_{TRN}(k), \qquad X_{IGT}(k) = X_{TRN}(k) \tag{1}$$

Case 2. The state variables and covariances when TRN is not available

$$P_{IGT}(k) = P_{GNSS}(k), \qquad X_{IGT}(k) = X_{GNSS}(k) \tag{2}$$

Case 3. The state variables and covariances when both GNSS and TRN are available

$$P_{IGT}(k) = \left[(P_{GNSS}(k))^{-1} + (P_{TRN}(k))^{-1} \right]^{-1} \tag{3}$$

$$X_{IGT}(k) = P_{IGT}(k) \left[P_{GNSS}(k)^{-1} X_{GNSS}(k) + P_{TRN}(k)^{-1} X_{TRN}(k) \right] \tag{4}$$

Case 4. The state variables and covariance when both GNSS and TRN are not available

$$X_{IGT}(k) = (I + A_{IGT}\Delta t)\, X_{IGT}(k-1) = \Phi_{IGT} X_{IGT}(k-1) \tag{5}$$

$$P_{IGT}(k) = \Phi_{IGT} P_{IGT}(k-1) \Phi_{IGT}^{T} + Q_{IGT} \tag{6}$$

Here, $X_{IGT}(k)$ and $P_{IGT}(k)$ are the estimated state variable vector and the estimated covariance matrix of the master filter, respectively. The estimated state variable vectors and estimated covariance matrices of the INS/GNSS and INS/TRN local filters are represented as $X_{GNSS}(k)$, $P_{GNSS}(k)$, $X_{TRN}(k)$, and $P_{TRN}(k)$, respectively, and Q_{IGT} and R_{IGT} represent the process noise covariance and measurement noise covariance of the master filter. A_{IGT} and Δt are the system matrix and filter update period. The first-order Taylor expansion of the system matrix, A_{IGT}, is computed for the discrete system matrix, Φ_{IGT}. The master filter and local filters use the 18th state variables composed of the errors of latitude, longitude, altitude, velocity and attitude, accelerometer bias, gyro bias, barometer error compensation acceleration, barometer bias, and barometer scale factor. If both TRN and GNSS are not available, as in Case 4, the master filter only performs time propagation in equations (5) and (6). If there is any available local filter, the master filter performs not only the time propagation but also the appropriate measurement update for each case. The unavailability of GNSS and TRN in Case 1 and 2 means that the system itself is inoperable. Case 3 applies when GNSS or TRN is temporarily unavailable.

Peer-reviewed version available at Remote Sens. 2020, 12, 1396; doi:10.3390/rs12091396

2.2.2. Design of the Local Filters based on the EKF

As mentioned in the previous section, the local filters of this system are composed of 18th state variables as follows:

$$X_{GNSS\ or\ TRN} = \left[\delta\lambda\ \delta\phi\ \delta h\ \delta v_e\ \delta v_n\ \delta v_u\ \delta\varphi_e\ \delta\varphi_n\ \delta\varphi_u\ \delta B_x^a\ \delta B_y^a\ \delta B_z^a\ \delta B_x^w\ \delta B_y^w\ \delta B_z^w\ \delta a\ \delta h_b\ \delta h_{sf}\right]^T \tag{7}$$

Here, $[\delta\lambda\ \delta\phi\ \delta h]$ is the error vector of longitude, latitude, and altitude, and $[\delta v_e\ \delta v_n\ \delta v_u]$ is the velocity error vector in the ENU coordinate system. $[\delta\varphi_e\ \delta\varphi_n\ \delta\varphi_u]$ is the error vector of roll, pitch, and yaw, and $\left[\delta B_x^a\ \delta B_y^a\ \delta B_z^a\right]$ and $\left[\delta B_x^w\ \delta B_y^w\ \delta B_z^w\right]$ are the accelerometer bias error and the gyro bias error vector in the body frame, δa, δh_b, and δh_{sf} denote the barometer error compensation acceleration, barometer bias, and barometer scale factor. The state variable configuration is identical to the two local filters and the master filter, but the barometer bias and scale factor error compensation is designed to be performed only in the local filter corresponding to the INS/GNSS aided navigation. As mentioned in the previous section, APF-based TRN was designed to compensate for the vertical bias error in the IRA and barometer when the likelihood function is calculated. Therefore, if the barometer error is additionally compensated for the INS/TRN aided navigation, the vertical bias error estimation of the APF, the filter used in the TRN tracking mode, may not normally be performed. Considering this point, our system is designed not to perform barometer error compensation in INS/TRN aided navigation.

First, we describe the barometer error estimation and compensation technique as well as the overall 18th local filter design. Barometer errors include bias errors modeled in the first-order Markov process, scale factor errors in the form of random constants, and white Gaussian noise. Applying the linear perturbation method to a system with a third-order vertical damping loop, the following vertical error model is derived [20].

$$\delta\dot{h} = \delta v_u - K_1\left(\delta h - \delta h_b - h\delta h_{sf}\right) \tag{8}$$

$$\delta\dot{v}_u = \delta A_z - K_2\left(\delta h - \delta h_b - h\delta h_{sf}\right) - \delta a + 2\omega_s^2\delta h \tag{9}$$

$$\delta\dot{a} = K_3\left(\delta h - \delta h_b - h\delta h_{sf}\right) \tag{10}$$

$$\delta\dot{h}_b = -\frac{1}{\tau}\delta h_b + \omega_B \tag{11}$$

$$\delta\dot{h}_{sf} = 0 \tag{12}$$

Here, ω_B is the white noise of the barometer. If the 2nd and 1st gains and the bias of the damping loop characteristic equation are K_1, K_2, and K_3, respectively, it can be calculated as follows:

$$K_1 = \frac{3}{\tau}, \qquad K_2 = 2\omega_s^2 + \frac{3}{\tau^2}, \qquad K_3 = \frac{1}{\tau^3} \tag{13}$$

Here, τ is the time constant of the system and ω_s is the Sculler frequency. The above barometer bias and scale factor error estimation and compensation are applied only to INS/GNSS aided navigation. However, in the initialization step of INS/TRN aided navigation, the barometer bias and scale factor error estimated by the INS/GNSS aided navigation are taken once. The discrete system and measurement model of the local filter is as follows:

$$X\left(k+1\right) \approx \left(I + A\Delta t\right) X(k) + v(k) \tag{14}$$

$$z(k) = HX(k) + w(k) \tag{15}$$

For, convenience, subscripts were omitted from the state variables in equation (7). Here $v(k)$ and $w(k)$ is the process and measurement white noise. I is the identity matrix. The system matrix, $A(k)$

Peer-reviewed version available at Remote Sens. 2020, 12, 1396; doi:10.3390/rs12091396

and measurement matrix, H are described in Ref. [21]. The system matrix is derived from the error model of the INS, but the detailed derivation process is out of the scope of this paper, so it is omitted. Two local filters were designed with an indirect feedforward EKF using the system matrix $A(k)$ and the measurement matrix H.

Unlike INS/GNSS aided navigation, we designed semi-tightly coupled structure when combining TRN and INS. Therefore, an aircraft movement input vector is required for the prior pdf for time propagation of APF-based TRN. In this study, the following three cases were compared and analyzed for the aircraft movement input vector.

- Case 1. Use only the INS position increments
- Case 2. Use only the INS/GNSS/TRN integrated navigation position increments
- Case 3. Use the INS position increments and the integrated navigation velocity increments

In Case 1, the filter has a smooth transient characteristic, but the overall filter performance is slightly degraded. On the other hand, in Case 2, the performance is better than Case 1, but the filter initially had high ripples. Therefore, in this study, Case 3, which has the excellent initial transient characteristic and overall performance, is applied as shown in Figure 2.

3. Hybrid TRN Algorithm based on an Interferometric Radar Altimeter

Three key parameters that determine the performance of TRN are a precision altimeter, an optimal TRN algorithm, and a high-resolution terrain database. In addition to developing a precision altimeter, it is also important to design an appropriate error compensation method and a measurement validity check logic for an altimeter. Next, to enhance the performance of TRN even when GNSS is unavailable for a long time, an optimal TRN algorithm should be designed. The TRN algorithm should include the validity check logic of the terrain roughness and uniqueness to prevent performance degradation on flat or repetitive terrain. Finally, we need to build a high-resolution DEM and develop appropriate matching technology with altimeter measurements for precise TRN. Moreover, the TRN algorithm with the high-resolution DEM must be capable of real-time processing on board. Considering these requirements, we propose the hybrid TRN algorithm that uses batch processing for the acquisition mode and a PF for the tracking mode. We used an IRA as the altimeter and a multi-resolution terrain database with a combination of a 3m DEM level 4 and a 10m resolution DEM level 3. To guarantee the real-time implementation on board, we used a parallel computing technology based on GPGPU in the proposed TRN system.

3.1. Error Compensation Method of IRA Measurements

An IRA outputs not only the relative distance (slant range, R) but also the cross-track angle (look angle, θ) from an aircraft to the nearest point on the zero Doppler line using the principle of interferometry [11]. The zero Doppler line is perpendicular to the velocity vector of the aircraft [11]. When the direction cosine matrix (DCM) required for converting IRA measurements to relative positions in the ENU coordinate system was calculated using Euler angles as shown in Ref. [22], there was performance degradation in a real flight environment [11]. This degradation was due to the angle of attack and the side slip angle caused by wind during flight. Therefore, in this study, to extract precision 3D position information, the effective look angle ξ, should be calculated with the IRA look angle θ, and the virtual attitude calculated by the velocity of the aircraft, as shown in Figure 3 [23].

The relative position vector $x_{IRA,k}$ of the nearest point, the virtual azimuth angle β, the virtual pitch angle α, and the effective look angle ξ can be determined as follows:

$$x_{IRA,k} = \begin{bmatrix} p_{e,k} \\ p_{n,k} \\ p_{u,k} \end{bmatrix} = \begin{bmatrix} \rho\cos\xi\sin\alpha\sin\beta + \rho\sin\xi\cos\beta \\ \rho\cos\xi\sin\alpha\cos\beta - \rho\sin\xi\sin\beta \\ -\rho\cos\xi\cos\alpha \end{bmatrix} \tag{16}$$

doi:10.20944/preprints202003.0442.v1

Peer-reviewed version available at *Remote Sens.* **2020**, *12*, 1396; doi:10.3390/rs12091396

$$\beta = \begin{cases} \frac{\pi}{2} - \tan^{-1}\frac{V_n}{V_e}\,, & \left|\frac{V_e}{V_n}\right| > 1 \\ \tan^{-1}\frac{V_e}{V_n}\,, & \left|\frac{V_e}{V_n}\right| \leq 1 \end{cases} \tag{17}$$

$$\alpha = \tan^{-1}\frac{V_u}{\sqrt{V_e^2 + V_n^2}} \tag{18}$$

$$\xi = \sin^{-1}\frac{\sin\theta}{\sqrt{\alpha^2\sin^2(\psi-\beta) + \cos^2(\psi-\beta)}} - \tan^{-1}\frac{-\alpha\sin(\psi-\beta)}{\cos(\psi-\beta)} \tag{19}$$

Here, ψ is the yaw of the aircraft. A detailed derivation process is given in section 2 of Ref. [23]. $[V_e,\ V_n,\ V_u]^T$ is the velocity of the aircraft in the ENU frame. Typically, $[V_e,\ V_n]^T$ is the estimates of the INS/GNSS/TRN integrated navigation and V_u is the estimate of the barometer damping loop. This is because the upward velocity error estimated by the INS/GNSS/TRN integrated navigation increases exponentially when neither a GNSS nor TRN is available. This misleads the IRA measurements as unavailable, further degrading the integrated navigation performance. Therefore, in this study, we use an upward velocity estimate of the barometer damping loop that shows stable changes during flight.

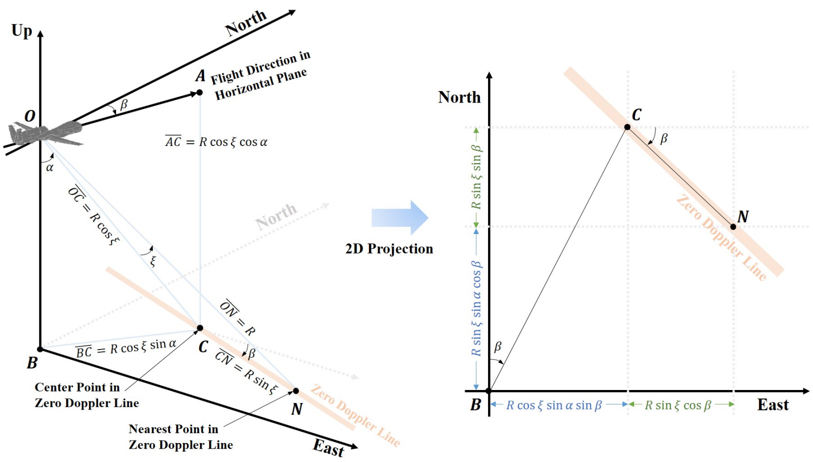


Figure 3. Relative position from the aircraft to the nearest point

3.2. Acquisition Mode based on Batch Processing

TRN based on batch processing calculates the navigation solution by comparing the terrain elevation profile collected during a certain correction period with the reference matrix [24]. Here, the reference matrix refers to a matrix with the terrain elevations extracted from the DEM around the current position of the aircraft as elements. When the correction period of TRN is reached, the matching reference elevation set can be acquired by comparing the measured terrain elevation profile with the DEM reference matrix. In that case, the mean absolute difference (MAD) and the mean square difference reference matrix (MSD) are defined as follows [11]:

$$MAD_{i,j} = \frac{1}{N}\sum_{k=1}^{N}\left|(h_{baro,k} - p_{u,k}) - h_{DB}\left(\hat{\lambda} + i\Delta\lambda + p_{lon,k},\ \hat{\varphi} + j\Delta\varphi + p_{lat,k}\right)\right| \tag{20}$$

$$MSD_{i,j} = \frac{1}{N}\sum_{k=1}^{N}\left[(h_{baro,k} - p_{u,k}) - h_{DB}\left(\hat{\lambda} + i\Delta\lambda + p_{lon,k},\ \hat{\varphi} + j\Delta\varphi + p_{lat,k}\right)\right]^2 \tag{21}$$

Here, N is the number of samples in the measured terrain elevation profile. h_{DB} is the terrain elevation value of the DEM. $\hat{\lambda}$ and $\hat{\varphi}$ are the longitude and latitude estimated by INS. $\Delta\lambda$ and $\Delta\varphi$ are the longitudinal and latitudinal cell size of the reference matrix. i and j are the column and row

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 March 2020 doi:10.20944/preprints202003.0442.v1

Peer-reviewed version available at Remote Sens. 2020, 12, 1396; doi:10.3390/rs12091396

index for the reference matrix. $p_{lon,k}$ and $p_{lat,k}$ are the east and north direction relative positions to the nearest point. These are calculated using Equation (16) as follows:

$$p_{lon,k} = \frac{p_{e,k}}{R_{ew}\cos\hat{\varphi}}, \qquad p_{lat,k} = \frac{p_{n,k}}{R_{ns}} \tag{22}$$

Here, R_{ew} and R_{ns} represent the radius of the curvature of the Earth ellipsoid in the east-west and north-south directions, respectively. As shown in Ref. [24], the TRN output based on batch processing was determined as the position with the minimum MAD or MSD value in the reference matrix. Therefore, in this study, we designed the following conditions where the acquisition mode is converted to a tracking mode.

$$\sqrt{MSD_{min}} < \sigma_{MSD1}, \qquad \sqrt{MSD_{sec,min}} - \sqrt{MSD_{min}} > \sigma_{MSD2} \tag{23}$$

$$|MSD_{min} - MAD_{min}| \leq \sigma_{MSD3} \tag{24}$$

Here, MSD_{min} and $MSD_{sec,min}$ represent the minimum and the second minimum values of the MSD. MAD_{min} represents the minimum value of the MAD. In this study, σ_{MSD1}, σ_{MSD2} and σ_{MSD3} are the thresholds for the conditions, respectively. Also, we designed to update the correction period when a certain number of measurements are collected without fixing the period. In other words, the acquisition is completed quickly in rough and unique terrain and the acquisition is performed for a long time in a flat and repetitive area. This batch processing has the disadvantage of having a long update period, but the calculation is relatively simple, which is suitable for the purpose of finding a coarse position in a wide search area. Therefore, in this study, we applied batch processing for the acquisition mode to estimate the current position roughly when the initial position error is large. The size of the reference matrix was set to 1001 × 1001 so that it can be searched for even when there is an initial position error of up to 2.5km.

3.3. Tracking Mode based on a Particle Filter

To estimate the precise position by TRN, nonlinear estimation problems have to be solved in real time [14]. Since the EKF locally linearizes the highly nonlinear characteristics of terrains, the EKF based TRN algorithm can diverge or degrade performance. Recently, to solve this problem, PF-based TRN techniques have been actively studied [25]-[28].

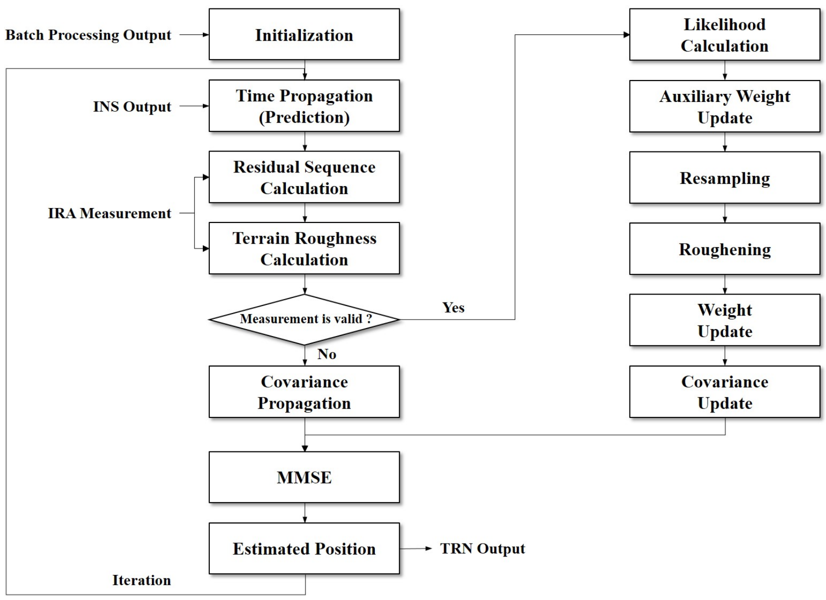


Figure 4. Execution flowchart of the APF-based TRN algorithm

Among the various PFs, the auxiliary particle filter (APF) was applied for the tracking mode of TRN in this study. The APF was first introduced by Pitt and Shephard in Ref. [26] to improve deficiencies in the general sequential importance resampling (SIR) algorithm. In the general SIR algorithm, the last measurement is ignored in the derivation of the weight update. Since this makes it impossible to optimize the proposal density, many efforts have been made. In this study, we used the APF, which is easy to implement and shows stable performance in IRA-based TRN. The APF-based TRN algorithm execution flowchart is shown in Figure 4.

Assume the state vector of the APF is $x_{k|k} = [\delta x_e,\ \delta x_n,\ \delta x_u]^T$. It is a state vector composed of the position errors on the ENU coordinate at the time step k. First, the state vector and the weight of the particles must be initialized using the estimates by the batch processing. Next, the time propagation of the states and weights is calculated as follows:

$$x_{k|k-1}^{[i]} \approx q\left(x_{k|k-1}^{[i]} | x_{k-1|k-1}^{[i]},\ y_{k-1}\right) \tag{25}$$

$$w_{k|k-1}^{[i]} \approx \frac{p\left(y_k | x_{k|k-1}^{[i]}\right) p\left(x_{k|k-1}^{[i]} | x_{k-1|k-1}^{[i]}\right)}{q\left(x_{k|k-1}^{[i]} | x_{k-1|k-1}^{[i]}, y_k\right)} w_{k-1|k-1}^{[i]} \tag{26}$$

$$p\left(x_{k|k-1}^{[i]} | x_{k-1|k-1}^{[i]}\right) = P_{\omega_k}\left(x_{k|k-1}^{[i]} - x_{k-1|k-1} - u_k\right) \tag{27}$$

Here, $x_{k|k-1}^{[i]}$ and $w_{k|k-1}^{[i]}$ are the state vector and the weight of the i-th particle in the prediction step, respectively. $x_{k-1|k-1}^{[i]}$ and $w_{k-1|k-1}^{[i]}$ are the values of the i-th particle in the previous update step. The current value of the state vector is determined by the previous value using a Markov process. y_{k-1} is the terrain height calculated by the difference between the measurements of the IRA and the barometer in the previous step. The transition pdf, $p\left(x_{k|k-1}^{[i]} | x_{k-1|k-1}^{[i]}\right)$ is obtained via the Chapman-Kolmogorov equation [14]. ω_k is the white system process noise that meets $E(\omega_k) = 0$ and $E\left(\omega_k {\omega_k}^T\right) = Q_c \triangle t$. Here, Q_c is the system process noise covariance in the PF, which was set to $[0.04,\ 0.04,\ 0.04]^T$, and $\triangle t$ is the execution period, which was set to 0.02 in this study. $P_{\omega_k}(x_{k|k-1}^{[i]} - x_{k-1|k-1} - u_k)$ is a Gaussian pdf with a standard deviation of ω_k. u_k is an aircraft movement input vector between $k-1$ and k time step. By using transition density as the proposal distribution, $q\left(x_{k|k-1}^{[i]} | x_{k-1|k-1}^{[i]}, y_{k-1}\right) = p\left(x_{k|k-1}^{[i]} | x_{k-1|k-1}^{[i]}\right)$, the sequential weight update equation (26) is simplified as follows:

$$w_{k|k-1}^{[i]} \approx p\left(y_k | x_{k|k-1}^{[i]}\right) w_{k-1|k-1}^{[i]} \tag{28}$$

This is the general SIR algorithm used in the bootstrap particle filter and prevents us from optimizing the proposal density by ignoring the measurement at k time step. In this study, the APF was used for TRN in order to design the proposal distribution optimally. The APF introduces an auxiliary variable $\alpha^{[i]}$, which represents the index of the i-th particle at $k-1$ time step. The auxiliary weight, $q\left(\alpha^{[i]} | y_k\right)$ is updated as follows:

$$q\left(\alpha^{[i]} | y_k\right) = p\left(y_k | \mu_{k|k-1}^{\alpha^{[i]}}\right) w_{k|k-1}^{\alpha^{[i]}} \tag{29}$$

Here, $\mu_{k|k-1}^{\alpha^{[i]}}$ is an expected value conditioned on $x_{k|k-1}^{\alpha^{[i]}}$ in a noise free system model. If the measurement is available, the likelihood of the APF is as follows:

$$p\left(y_k | \mu_{k|k-1}^{\alpha^{[i]}}\right) = p_{v_k}\left(y_k - h_{DB}\left(\lambda_{k|k-1}^{\alpha^{[i]}} + p_{lon,k},\ \phi_{k|k-1}^{\alpha^{[i]}} + p_{lat,k}\right) - \delta x_{u,k|k-1}^{\alpha^{[i]}}\right) \tag{30}$$

Here, v_k is the measurement white noise and $h_{DB}\left(\lambda_{k|k-1}^{\alpha^{[i]}} + p_{lon,k},\ \phi_{k|k-1}^{\alpha^{[i]}} + p_{lat,k}\right)$ denotes the terrain elevation from the DEM at the horizontal position of the nearest point, and $\lambda_{k|k-1}^{\alpha^{[i]}}$ and $\phi_{k|k-1}^{\alpha^{[i]}}$ are the estimated longitude and latitude of the $\alpha^{[i]}$-th particle in the prediction step. When this algorithm

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 March 2020 doi:10.20944/preprints202003.0442.v1

Peer-reviewed version available at Remote Sens. 2020, 12, 1396; doi:10.3390/rs12091396

is run for several steps, a degeneracy phenomenon occurs in which the weights of all particles except one particle are too small. Among the various resampling methods used to solve this degeneracy problem, we used the stratified sampling method [25]. Resampling can solve the degeneracy problem. However, the particles with large weights are replicated when the filter is updated, so a sampling impoverishment problem occurs where the diversity disappears over time. In this study, a roughening was added to the resampled particles. Using the likelihood, the weight of the $\alpha^{[j]}$-th resampled particle is updated as follows:

$$w_{k|k}^{\alpha^{[j]}} \approx \frac{p\left(y_k | x_{k|k-1}^{\alpha^{[j]}}\right)}{q\left(\alpha^{[j]} | y_k\right)} w_{k|k-1}^{\alpha^{[j]}} \tag{31}$$

The state variable estimate and covariance that has the minimum mean square error (MMSE) are as follows [25]:

$$x_{k|k} = \begin{bmatrix} \delta x_e & \delta x_n & \delta x_u \end{bmatrix}^T \approx \sum_{j=1}^{N} w_{k|k}^{\alpha^{[j]}} x_{k|k-1}^{\alpha^{[j]}} \tag{32}$$

$$P_{k|k} \approx \sum_{j=1}^{N} w_{k|k}^{\alpha^{[j]}} \left(x_{k|k}^{\alpha^{[j]}} - x_{k|k}\right)\left(x_{k|k}^{\alpha^{[j]}} - x_{k|k}\right)^T \tag{33}$$

Here, N is the number of particles and was set at 8192. The estimated position by APF-based TRN is as follows:

$$x_k = \left[\hat{\lambda} + \frac{\delta x_e}{R_{ew}\cos\hat{\varphi}}, \quad \hat{\varphi} + \frac{\delta x_n}{R_{ns}}, \quad \hat{h} + \delta x_u\right]^T \tag{34}$$

Here, $\hat{\lambda}$, $\hat{\varphi}$ and $\hat{h}$ are the longitude, latitude and height estimated by the INS, respectively.

As mentioned in the previous section, since the TRN performance degrades on flat or repetitive terrains, the validity check technique of measurements by terrain roughness and uniqueness is an important technique that determines the navigation performance. In areas with flat or repetitive terrain, the likelihood in equation (30) does not have any useful information for the TRN performance. Moreover, although the roughening is applied after the resampling process to alleviate the impoverish problem, it can increase TRN error rapidly on flat or repetitive terrain. Therefore, in this paper, we propose a stochastic terrain roughness as another condition for measurement update.

$$\sigma_{T,k} = \sqrt{\overline{h_{k|k-1}^2} - \overline{h_{k|k-1}}^2} > \sigma_{T1} \tag{35}$$

$$\overline{h_{k|k-1}^2} = \sum_{i=1}^{N} w_{k|k-1}^{[i]} h_{DB}^2\left(\lambda_{k|k-1}^{[i]} + p_{lon,k},\ \phi_{k|k-1}^{[i]} + p_{lat,k}\right) \tag{36}$$

$$\overline{h_{k|k-1}} = \sum_{i=1}^{N} w_{k|k-1}^{[i]} h_{DB}\left(\lambda_{k|k-1}^{[i]} + p_{lon,k},\ \phi_{k|k-1}^{[i]} + p_{lat,k}\right) \tag{37}$$

Details of the residual sequence and the stochastic terrain roughness are given in Ref. [29].

Moreover, we designed the system to perform the batch processing-based acquisition mode again if no measurements were available consecutively for 15 seconds during the tracking mode. As mentioned, we designed a three-dimensional APF that have the 8192 particles. Generally, the larger the number of particles, the better the performance. This causes the computational complexity. In this study, we developed a parallel computing acceleration technology using a GPGPU chipset. The OpenCL-based parallel processing framework uses the i7-4700EQ CPU of a single board computer (SBC) as its host and the AMD E8860 GPGPU as the device for kernel function execution. We used PCIe 3.0 16lane to link the CPU and GPGPU boards for high speed communication of the TRN results.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 March 2020 doi:10.20944/preprints202003.0442.v1

Peer-reviewed version available at Remote Sens. 2020, 12, 1396; doi:10.3390/rs12091396

Parallel computing acceleration is a technology of computing in which the execution of processes are carried out simultaneously [30]-[31]. A GPGPU can perform numerical computations usually handled by a CPU and solve many complex computations in a more efficient way than on one CPU due to the parallel architecture of this device. Complex operations can often be divided into simpler arithmetic operations, which can be computed simultaneously. In other words, performing simple operations on a GPGPU in parallel is more efficient in speed than solving a complex problem sequentially on a single high-performance CPU. In this study, we divided 23 kernel functions to perform the operations corresponding to the blocks in Figure 4 and designed the system to perform the various operations with the 8192 particles divided into 32 work groups consisting of 256 work items on the AMD E8860 GPGPU. We confirmed that the openCL-based parallelized code of the proposed APF-based TRN algorithm shows the same performance as the double precision C-based sequential code of that, and its operation speed is 14 times faster. The one step execution time of the filter was 13.37msec in C-based sequential code and 0.94msec in parallel code based on openCL. Considering the DEM processing, the sequential code may be difficult to perform in real time within the filter update period of 50Hz. In contrast, the proposed parallel code can be implemented in real time on board.

3.4. Real-Time Processing of a High Resolution Terrain Database

A high-resolution DEM is an essential element for improving the performance of TRN. However, large amounts of memory space are needed to store a high-resolution DEM, and additional time is needed to load the DEM in a separate storage space [32]. In the past, the memory capacity of the navigation computer was insufficient, so only part of the DEM around the pre-planned route was loaded into the memory [33]. In this study, we applied a 64GB solid state drive (SSD) to store 3m resolution digital terrain elevation data (DTED) level 4 in the mission area and 10m resolution DTED level 3 in the rest of the flight area. At Korean peninsula area, DTED level 4 requires about 15GB of memory space and DTED level 3 requires about 3.7GB of memory space. Since these high-resolution terrain DBs are large files, they are difficult to process in real time. Typically, since the AMD E8860 GPGPU, where the TRN algorithm performs, has a 2GB synchronous dynamic random access memory (SDRAM), it is impossible to load the entire DEM.

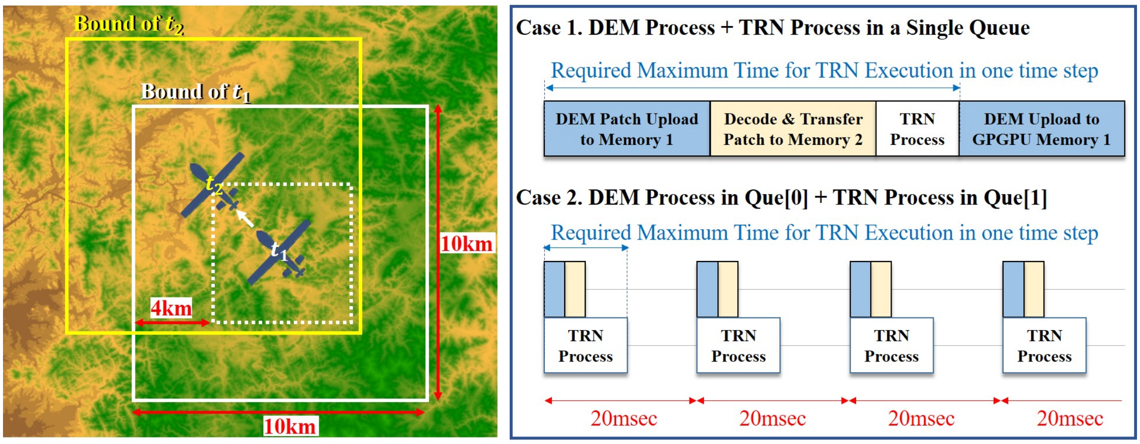


Figure 5. Transmission scheme of the high resolution DEM information segmentation

Therefore, in this study, as shown in Figure 5, the DEM patch corresponding to a $10km \times 10km$ area is generated based on the current estimated position of the aircraft in the CPU, and then transferred to the SDRAM of the GPGPU. For the aircraft flight from t_1 to t_2 in Figure 5, there is a $4km$ difference from the boundary of the DEM patch generated at the point t_1. At this time, another DEM patch with a $10km \times 10km$ area is generated at point t_2. When a DEM patch is generated, the highest resolution DEM among the available DEMs present in the region is selected, and one patch may be made of a combination of several DEMs. When combining DEMs with more than two resolutions, the boundary

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 March 2020 doi:10.20944/preprints202003.0442.v1

Peer-reviewed version available at Remote Sens. 2020, 12, 1396; doi:10.3390/rs12091396

processing uses the linear interpolation method. Two buffers are created in the GPGPU's memory for the purpose of storing DEM patches. One (Memory 2 in Figure 5) stores the DEM patch used in the TRN algorithm, and the other (Memory 1 in Figure 5) loads the newly generated DEM patch. When loading of the new DEM patch in Memory 1 is completed, the DEM patch is decoded. This is because the high-resolution DEM is encoded so that the terrain DB can be completely preserved from enemies even when the aircraft crashes. After the decoding step, the new DEM patch must be moved to Memory 2 for the TRN algorithm. If the DEM and TRN processes sequentially, as in Case 1 of Figure 5, and the maximum time of one TRN loop execution exceeds 20*msec*, then real-time computation is not guaranteed. Therefore, in this study, two queues (Que[0] and Que[1]) were created in GPGPU as in Case 2 of Figure 5, and the DEM and TRN are processed using a task parallel computing method. The DEM processing itself was also divided in order to prevent the moment of accessing Memory 2 at the same time as the TRN processing.

4. Verification of a Three-dimensional Terrain based System

4.1. Experiment Results in a SIL Environment

Before the proposed AP-TAN is applied to UAVs, we investigated the feasibility of the navigation performance on a real aircraft platform. Since the verification on the aircraft platform through captive flight tests are expensive, the pre-verification through Monte Carlo offline simulation or SIL tests is necessary. In particular, this section deals with SIL test results for real-time verification of the proposed navigation software on the PPU prior to flight tests. Among the navigation algorithms, inertial navigation and INS/GNSS aided navigation can generally be verified by vehicle testing. However, because the IRA is inoperable on the ground, we developed the SIL test equipment, shown in Figure 6, to verify the TRN and INS/TRN aided navigation.

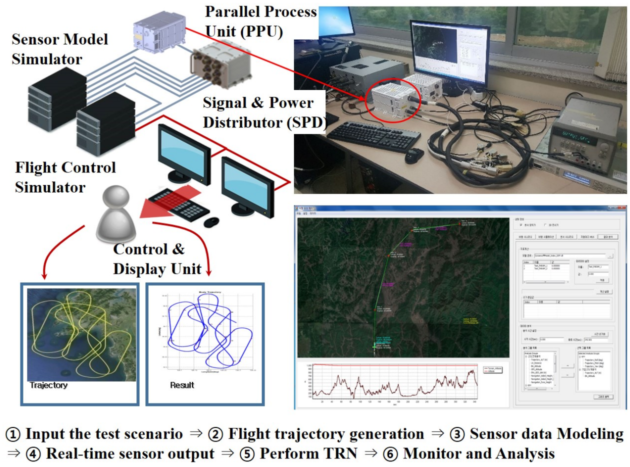


Figure 6. SIL experiment configuration for the proposed AP-TAN

The main functions of the equipment are the creation of flight scenarios (Flight Control Simulator in Figure 6), the generation of simulated sensor data along the scenarios (Sensor Model Simulator in Figure 6), the real-time outputs of the generated signals, and the analysis of navigation results carried out on a PPU (Control and Display Unit in Figure 6). The SPD in Figure 6 delivers the commands from the control and display unit and the sensor data from the sensor model simulator to the PPU, and delivers the navigation results from the PPU to the control and display unit. In addition, real-time simulations using sensor data and flight trajectory data acquired through captive flight tests can be performed, so the proposed system has been continuously validated and modified through the SIL tests. Since the Modeling of the IRA is the most important sensor in the proposed system, we

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 March 2020 doi:10.20944/preprints202003.0442.v1

Peer-reviewed version available at Remote Sens. 2020, 12, 1396; doi:10.3390/rs12091396

developed a sensor model simulator including the IRA signal processing method and verified the simulator through captive flight tests.

Figure 7 and Table 1 show flight trajectories and simulation conditions for three SIL tests. The area enclosed by black lines in Figure 7 is the mission area, and the 3m resolution DTED level 4 was constructed using LiDAR equipment. The 10m resolution DTED Level 3 was used for the remaining. The mission area was chosen close to the runway after excluding urban areas, flat areas such as seas and rivers, and military training areas. In SIL Tests 1 and 2, the GNSS was set to be not available within the mission area, and the GNSS was set to be available again after leaving the mission area. On the other hand, in Test 3, GNSS was set to be unavailable after the aircraft had risen above 1km. As shown in Table 1, the average flight altitudes of SIL Tests 1, 2, and 3 were set to 1.3km, 5.3km, and 2.3km, respectively. When GNSS is not available, the INS/GNSS aided navigation only performs a propagation.

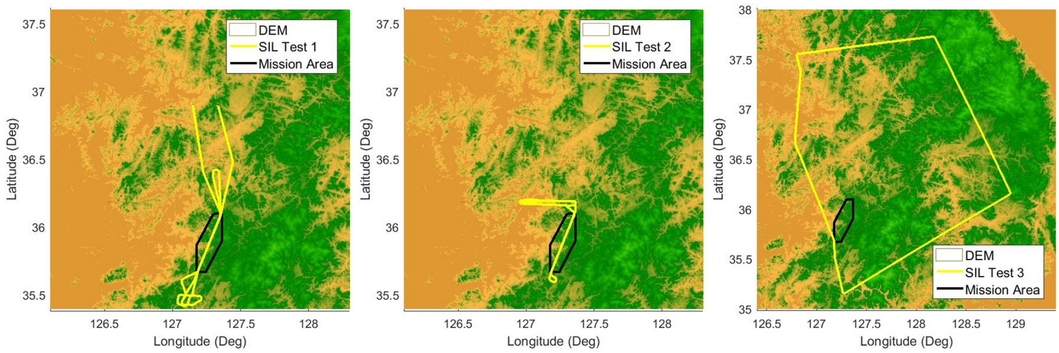


Figure 7. Trajectories for SIL tests

Table 1. Simulation conditions of the SIL tests

No. of Test	Total Flight Time	Mission Time	Flight Altitude	GNSS Unavailable Area
SIL Test 1	$1hr.20min.$	$13min.$	$1.3km$	Mission Area
SIL Test 2	$42min.$	$23min.$	$5.3km$	Mission Area
SIL Test 3	$1hr.50min.$	$1hr.40min.$	$2.3km$	Whole Area

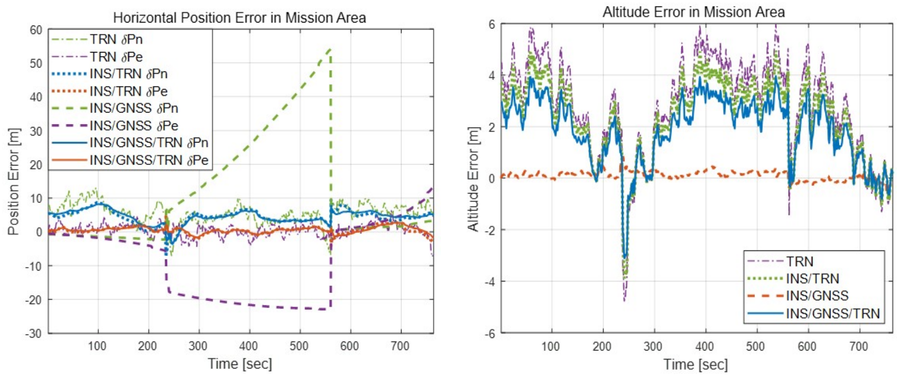
Since INS/GNSS aided navigation has been validated in many previous studies, we dealt with the results of the performance analysis on a mission area on which TRN was performed and GNSS was unavailable. For SIL Test 3, we performed TRN simulation without GNSS for all flight times except for take-off time. The simulation results are shown in Figures 8, 9, 10, and Table 2. For SIL Test 1, three tests were conducted in the mission area, and the test resumed after a long flight without GNSS before entering the second mission area. This was done to analyze the effect of the absence of GNSS for a long time on TRN. For this reason, the slope of the INS/GNSS aided navigation error increased rapidly in the second test section of Figure 8(a), but this did not affect the TRN. Because GNSS was temporarily available before flying into the mission area, there was a moment when the INS/GNSS aided navigation error decreased between the three missions, as shown in Figure 8(a). In SIL Test 1, the inertial navigation altitude through the 3rd damping loop was stable because no error is applied to the barometer model, and the altitude performance of the INS/GNSS aided navigation is stable as shown in Figure 8(b). On the other hand, the altitude error of TRN was relatively large, but it was due to the discrepancy between the altitude bias model predicted by the APF and the ideal barometer model without error.

As shown in Figure 9, SIL Test 2 was the simulation result using real flight test data with an average flight altitude of 5.3km. As the trajectory data used in the simulation, the carrier differential global positioning system (CDGPS) data obtained from the real flight test was used. For SIL Test 2,

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 March 2020 doi:10.20944/preprints202003.0442.v1

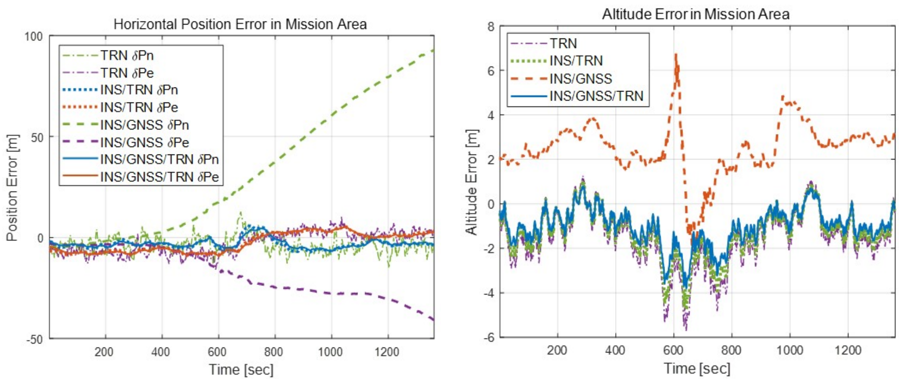
Peer-reviewed version available at *Remote Sens.* **2020**, *12*, 1396; doi:10.3390/rs12091396

one test was conducted in the mission area. Comparing Figure 8(a) and Figure 9(a) shows that the TRN performance was somewhat degraded as the altitude increased, but the navigation results were stable even in the GNSS unavailable environment. In addition, unlike in Figure 8(b), Figure 9(b) show that when we used real flight data with a barometer error, the altitude error of TRN was properly estimated. As shown in Figures 8(a) and 9(a), the INS/GNSS/TRN integrated navigation performed slightly better than the INS/TRN aided navigation when the GNSS error was small, even in the GNSS available environment. However, as the GNSS error increased the performance was almost equivalent to that of the INS/TRN aided navigation. In SIL Test 3, a TRN simulation was performed in over the entire Korean peninsula using 10m resolution DTED Level 3.



(a) (b)

Figure 8. Results of SIL Test 1 when GNSS is unavailable in the mission area (a) Horizontal position error (b) Altitude error



(a) (b)

Figure 9. Results of SIL Test 2 when GNSS is unavailable in the mission area (a) Horizontal position error (b) Altitude error

As shown in Figure 10(a), because GNSS was unavailable during one hour and 40 minutes of flight time, the INS/GNSS aided navigation error increased significantly to the inertial navigation level. In contrast, both the INS/TRN aided navigation and INS/GNSS/TRN integrated navigation maintained stable performance. As shown in Figure 10(b), the INS/TRN performance was degraded because the enough terrain roughness for the reliable TRN performance was not guaranteed between 100 and 1400 seconds through urban areas. However, it can be seen that the performance of the INS/TRN aided navigation was restored soon after the roughness was secured. As shown in Figures 10(c) and 10(d), the INS/TRN aided navigation and INS/GNSS/TRN integrated navigation altitude errors were generally stable, although they were somewhat larger than the INS/GNSS aided navigation altitude error, which was almost the same as the barometer's third damping loop output.

Peer-reviewed version available at Remote Sens. 2020, 12, 1396; doi:10.3390/rs12091396

Table 2 reveals that the SIL Test 1 with 3m resolution DTED Level 4 had a 3.54m circular error probability (CEP) for the INS/TRN aided navigation and a 3.50m CEP for the INS/GNSS/TRN integrated navigation at a 1.3km flight altitude. Moreover, the SIL Test 2 with 3m resolution DTED Level 4 had a 5.62m CEP for the INS/TRN aided navigation and a 5.43m CEP for the INS/GNSS/TRN integrated navigation at a 5.3km flight altitude. Even under conditions where the GNSS was unavailable for one hour and 40 minutes, the SIL Test 3 with 10m resolution DTED Level 3 had a 7.23m CEP for the INS/TRN aided navigation and a 6.73m CEP for the INS/GNSS/TRN integrated navigation at a 2.3km flight altitude. In addition, in all SIL tests, the TRN acquisition mode by batch processing delivered a coarse position with errors less than a 30m CEP within 40 seconds in the APF-based TRN tracking mode.

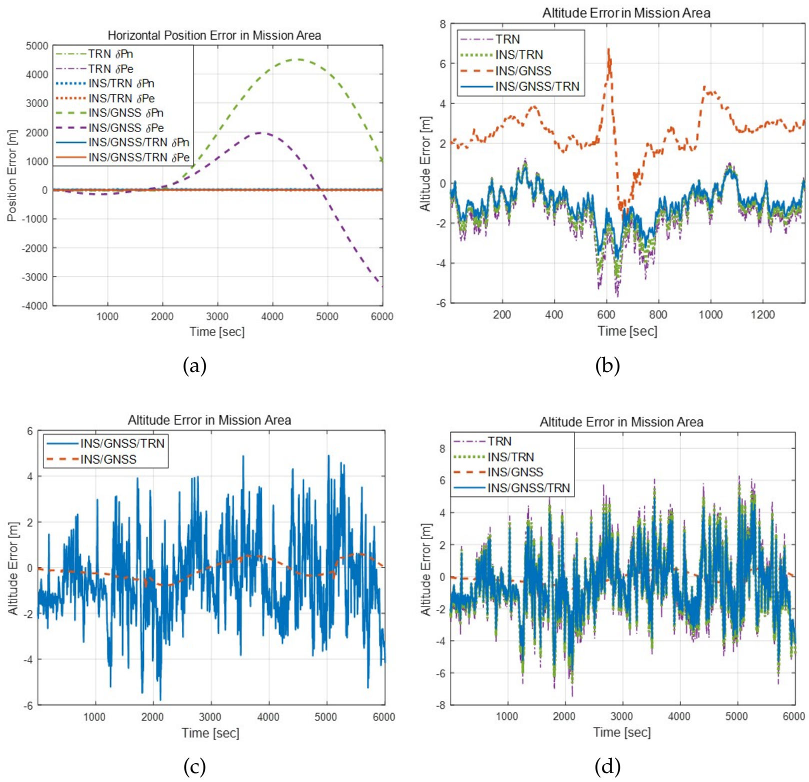


Figure 10. Results of SIL Test 3 when GNSS is unavailable in the whole area (a) Horizontal position error (b) Horizontal position error without INS/GNSS aided navigation error (c) Altitude error without TRN and INS/TRN aided navigation error (d) Altitude error

Table 2. Simulation results of the SIL tests in a GNSS-denied environment

Hor. Pos. Error [m CEP]	SIL Test 1	SIL Test 2	SIL Test 3
TRN	4.89m	6.74m	9.13m
INS/TRN	3.54m	5.62m	7.23m
INS/GNSS	20.06m	39.52m	2242.68m
INS/GNSS/TRN	3.50m	5.43m	6.73m
Altitude Error [m PE]	**SIL Test 1**	**SIL Test 2**	**SIL Test 3**
TRN	2.29m	1.35m	1.49m
INS/TRN	1.91m	1.13m	1.35m
INS/GNSS	0.14m	1.86m	0.26m
INS/GNSS/TRN	1.53m	0.90m	1.18m

4.2. Verification using Captive Flight Tests

Captive flight test techniques include a full range of techniques for verifying real-time inertial navigation, integrated navigation and TRN performance by equipping a real aircraft with the proposed AP-TAN. In this study, we designed and manufactured a testbed including an IRA, an INS, a GNSS receiver, a barometer, a PPU and a SPD, as shown in Figure 11. Prior to the test evaluation, digital surface model (DSM) and DEM with a 3m resolution were constructed using LiDAR. The captive flight tests used a CESSNA 172 Skyhawk with a hole to mount the IRA antenna on the back of the aircraft as shown in Figure 11. To minimize the mounting errors between the IRA antenna assembly and the INS, the testbed was manufactured to be mounted on the same jig, and all other equipment were mounted inside the aircraft using a separate rack. The PPU developed in this study was equipped with an x86 series i7-4799EQ SBC with a 8GB DDR3 memory, AMD E8860 embedded GPGPU, and 64GB SSD for high resolution DEMs. In the SBC, inertial navigation and integrated navigation were performed, while TRN and DEM real-time processing were performed in the GPGPU board.

Figure 11. Testbed and proposed AP-TAN for captive flight tests

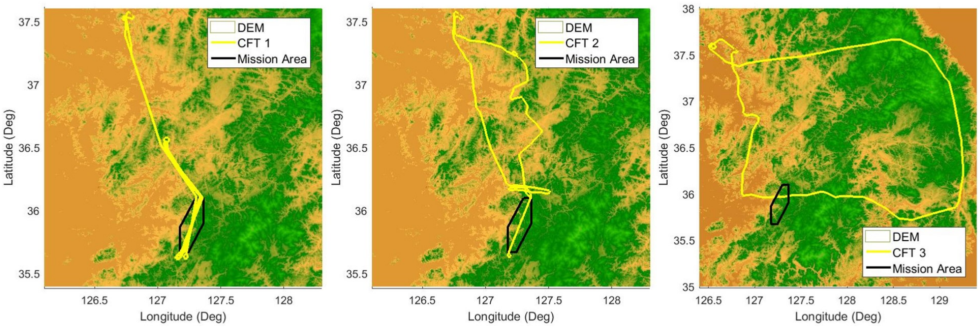


Figure 12. Trajectories for captive flight tests

Table 3. Conditions of captive flight tests

No. of Test	Total Flight Time	Mission Time	Flight Altitude	GNSS Unavailable Area
CFT 1	$3hr. 33min.$	$1hr. 58min.$	$1.5km$	Mission Area
CFT 2	$2hr. 33min.$	$23min.$	$5.1km$	Mission Area
CFT 3	$3hr. 40min.$	$1hr. 42min.$	$1.3km \sim 3.3km$	Whole Area

In this section, we analyze the results of three system performance evaluation tests when a GNSS was not available, as shown in Table 3 and Figure 12. The three flight tests had the same purpose as the SIL tests mentioned in section 4.1 above. However, in the case of CFT 3, unlike in the SIL Test 3, the flight altitude was freely determined by the operator within a range of 1.3km to 3.3km. The shorter mission time than flight time in CFT 2 was due to the long waiting time for entry permission into

 doi:10.20944/preprints202003.0442.v1

Peer-reviewed version available at Remote Sens. 2020, 12, 1396; doi:10.3390/rs12091396

the mission area. On the other hand, in the case of CFT 3, the waiting time was necessary to confirm whether the proposed system operates normally even when the GNSS is unavailable for a long time.

Figures 13, 14, and 15 below show the results of the three CFTs. In Figures 13(a), 14(a), and 15(a), the INS altitude represents the barometer's third damping loop output. As shown in Figure 13(a), the INS altitude error was about 15.99m probable error (PE) compared to CDGPS data at a 1.4km flight altitude. On the other hand, the remaining TRN and integrated navigation show similar performance within 5m PE. Based on these results, we can confirm that the proposed barometer error compensation scheme works in a real environment. As shown in Figure 13(b), the CFT 1 performed three tests in the mission area. Before each test set, the GNSS was made available temporarily and then re-entered into the mission area and was again unavailable. Accordingly, the INS/GNSS integrated navigation error decreased momentarily at 1500 seconds and 3250 seconds and then increased continuously after entering the mission area as shown in Figure 13(b). On the other hand, as shown in Figure 13(c) and 13(d), the TRN, INS/TRN aided navigation and INS/GNSS/TRN integrated navigation provided stable performances at a flight altitude of 1.5km when GNSS was unavailable. However, even if the terrain was rough enough, the IRA availability decreased at 1500 seconds and 3250 seconds when the aircraft was turning, leading to poor performance of the TRN, as shown in Figure 13(a).

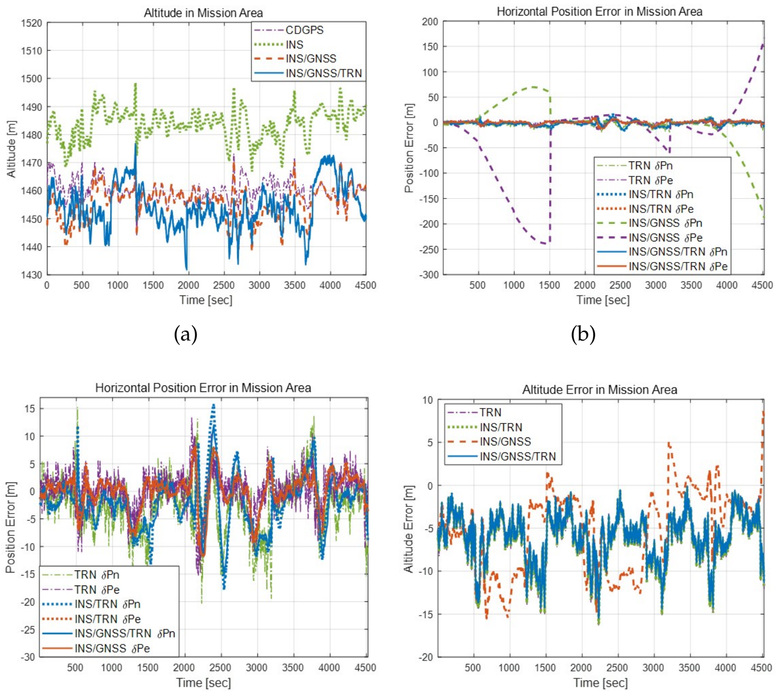


Figure 13. Results of CFT 1 when GNSS is unavailable in the mission area (a) Attitude profile (b) Horizontal position error (c) Horizontal position error without INS/GNSS aided navigation error (d) Altitude error

Figure 14(a) shows that the INS altitude error was about 92.74m PE compared to the CDGPS data at a 5.1km flight altitude. Comparing Figure 13(a) and Figure 14(a) reveals that the INS altitude error of CFT 2 was larger than that of CFT 1. This was because there was a barometer scale factor error, so the altitude error for flying at 5.1km was larger than for flying at 1.5km. In addition, the scale factor error of the barometer was affected not only by altitude but also by temperature. In the case of CFT 2, as the waiting time increased before the mission was performed, the temperature deviation between the takeoff and the mission was large, which resulted in a larger INS altitude error. On the other hand, the remaining TRN and integrated navigation had similar performances within 4m PE. Based on these results, we can confirm that the proposed barometer error compensation scheme works

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 March 2020 doi:10.20944/preprints202003.0442.v1

Peer-reviewed version available at Remote Sens. 2020, 12, 1396; doi:10.3390/rs12091396

in a real environment. As shown in Figure 14(b), the CFT 2 performed one test in the mission area in consideration of flight safety under the restrictions of operating with an oxygen mask at altitudes above 5km. The INS/GNSS integrated navigation error increased continuously after entering the mission area. On the other hand, as shown in Figures 14(c) and 14(d), the TRN, INS/TRN aided navigation and INS/GNSS/TRN integrated navigation provided stable performances at a flight altitude of 5.1km when GNSS was unavailable.

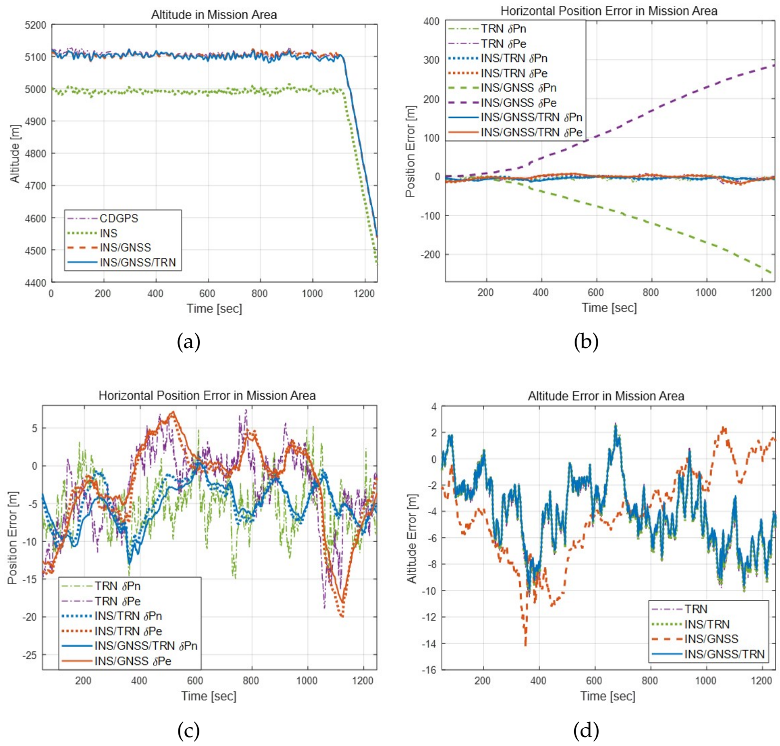


Figure 14. Results of CFT 2 when GNSS is unavailable in the mission area (a) Attitude profile (b) Horizontal position error (c) Horizontal position error without INS/GNSS aided navigation error (d) Altitude error

As shown in Figure 15(a), the INS altitude error was about 12.47m PE compared to CDGPS data. On the other hand, the remaining TRN and integrated navigation had similar performances within 7m PE. The CFT 1 and 2 were performed during a level flight, but in the case of the CFT 3, the altitude error was slightly increased because it was freely flown between 1.3km and 3.3km altitudes as shown in Figure 15(d), but the results were improved compared to the third damping loop output. In Figure 15(b), the INS/GNSS integrated navigation error increases continuously after entering the mission area. On the other hand, as shown in Figures 15(c) and 15(d), the TRN, INS/TRN aided navigation and INS/GNSS/TRN integrated navigation provide stable performances at any flight altitude when GNSS is unavailable. As shown in Figure 15(d), when the altitude changes, the longer the GNSS is unavailable, the larger the altitude error of INS/GNSS aided navigation. On the other hand, the altitude of INS/TRN aided navigation showed stable performance even with altitude change.

Table 4 below summarizes the TRN acquisition results based on batch processing. In CFT 1, the conditions for converting from acquisition mode to tracking mode were so severe that the coarse position error was small but the acquisition time was slightly longer. Therefore, the conversion conditions were designed to be alleviated as shown in equations (23) and (24) of section 3.2 so that the acquisition time would be within 30 seconds. The position root mean square (RMS) errors of acquisition mode were within 20m. Table 5 summarizes the navigation results of CFTs in a GNSS unavailable environment. Applying DTED Level 4 in a GNSS unavailable environment, Table 5 shows a 3.07m CEP and a 5.90m CEP for INS/TRN aided navigation errors at 1.4km and 5.1km, respectively.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 March 2020 doi:10.20944/preprints202003.0442.v1

Peer-reviewed version available at Remote Sens. 2020, 12, 1396; doi:10.3390/rs12091396

On the other hand, when DTED Level 3 was applied, the error of the INS/TRN aided navigation was 8.87m CEP. In addition, even if there were many barometer errors, the altitude of the proposed system had a stable performance.

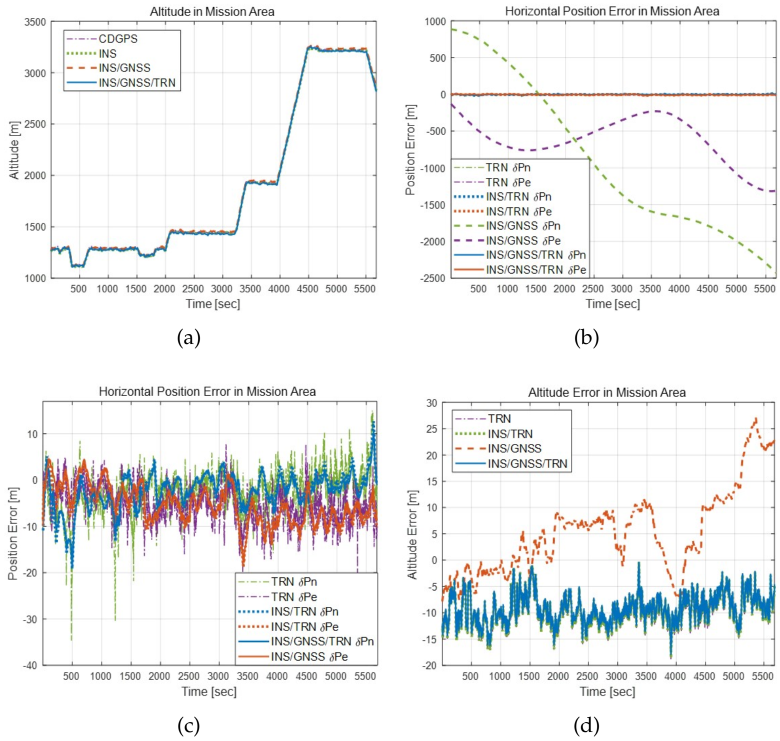


Figure 15. Results of CFT 3 when GNSS is unavailable in the whole area (a) Attitude profile (b) Horizontal position error (c) Horizontal position error without INS/GNSS aided navigation error (d) Altitude error

Table 4. Batch processing-based TRN acquisition position error of the captive flight tests

No. of Test	Acquisition Time	Latitude Error	Longitude Error
CFT 1	$47sec$	$5m$	$3.5m$
CFT 2	$30sec$	$-10m$	$-10m$
CFT 3	$24sec$	$-16m$	$-1m$

Table 5. Experiment results of the captive flight tests in a GNSS-denied environment

Hor. Pos. Error [m CEP]	CFT 1	CFT 2	CFT 3
TRN	$3.97m$	$6.45m$	$9.85m$
INS/TRN	$3.07m$	$5.90m$	$8.87m$
INS/GNSS	$78.92m$	$164.85m$	$1212.85m$
INS/GNSS/TRN	$3.75m$	$6.28m$	$8.39m$
Altitude Error [m PE]	**CFT 1**	**CFT 2**	**CFT 3**
TRN	$4.62m$	$3.33m$	$6.79m$
INS/TRN	$4.60m$	$3.29m$	$6.74m$
INS/GNSS	$4.76m$	$3.62m$	$6.47m$
INS/GNSS/TRN	$4.46m$	$3.16m$	$6.48m$

5. Conclusions

In this study, we introduced the AP-TAN, that can be used even in environments where a GNSS is unavailable or terrain roughness cannot be determined. First, an IRA was used as the altimeter of the proposed system, and a suitable error compensation method was introduced. Next, to secure UAV survivability and reliability in an environment where a GNSS is unavailable for a long time, we developed an optimal combination TRN algorithm consisting of an acquisition mode based on batch processing and a tracking mode based on an APF. In order to improve the performance of TRN, a high resolution DEM was constructed. A GPGPU-based parallel computing technology was applied to deal with a large amount of computational APF-based TRN and high resolution DEM real-time processing. We also developed the INS/GNSS/TRN integrated navigation based on the federated filter to guarantee the navigation performance even in environments where either GNSS or TRN is unavailable. Finally, SIL tests and CFTs were performed to verify the performance of the proposed system when GNSS was denied. As a result, an INS/TRN aided navigation performance of about a 3.1m CEP at 1.5km flight altitude and about a 5.9m CEP at 5.1km altitude was obtained using 3m resolution DTED. Also, the INS/TRN/GNSS integrated navigation performance of about 8.4m CEP was obtained using 10m resolution DTED between 1.3km to 3.3km altitude when there is no GNSS for about two hours.

Author Contributions: conceptualization, C.-K.S. and M.-J.Y.; methodology, C.-K.S. and J.L.; software, J.L., C.-K.S., J.O. and K.H; validation, J.L. and J.O.; formal analysis, J.L. and C.-K.S.; data curation, J.L.; writing–original draft preparation, J.L.; writing–review and editing, J.L.; supervision, S.L. and M.-J.Y.;

Funding: This research received no external funding

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

1. Grewal, M.S.; Weill, L.R.; Andrews, A.P. *Global positioning systems, inertial navigation, and integration*; John Wiley & Sons, 2007.
2. Zhang, G.; Hsu, L.T. Intelligent GNSS/INS integrated navigation system for a commercial UAV flight control system. *Aerospace Science and Technology* **2018**, *80*, 368–380.
3. Courbon, J.; Mezouar, Y.; Guénard, N.; Martinet, P. Vision-based navigation of unmanned aerial vehicles. *Control Engineering Practice* **2010**, *18*, 789–799.
4. Belmonte, L.M.; Morales, R.; Fernández-Caballero, A. Computer Vision in Autonomous Unmanned Aerial Vehicles—A Systematic Mapping Study. *Applied Sciences* **2019**, *9*, 3196.
5. Konovalenko, I.; Kuznetsova, E.; Miller, A.; Miller, B.; Popov, A.; Shepelev, D.; Stepanyan, K. New approaches to the integration of navigation systems for autonomous unmanned vehicles (UAV). *Sensors* **2018**, *18*, 3010.
6. Popp, M.; Scholz, G.; Prophet, S.; Trommer, G. A laser and image based navigation and guidance system for autonomous outdoor-indoor transition flights of MAVs. 2015 DGON Inertial Sensors and Systems Symposium (ISS). IEEE, 2015, pp. 1–18.
7. Thoma, J.; Paudel, D.P.; Chhatkuli, A.; Probst, T.; Gool, L.V. Mapping, Localization and Path Planning for Image-Based Navigation Using Visual Features and Map. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 7383–7391.
8. Melo, J.; Matos, A. Survey on advances on terrain based navigation for autonomous underwater vehicles. *Ocean Engineering* **2017**, *139*, 250–264.
9. Jeon, H.C.; Park, W.J.; Park, C.G. Grid design for efficient and accurate point mass filter-based terrain referenced navigation. *IEEE Sensors Journal* **2017**, *18*, 1731–1738.
10. ZELENKA, R. Integration of radar altimeter, precision navigation, and digital terrain data for low-altitude flight. Guidance, Navigation and Control Conference, 1992, p. 4420.
11. Oh, J.; Sung, C.K.; Lee, J.; Lee, S.W.; Lee, S.J.; Yu, M.J. Accurate Measurement Calculation Method for Interferometric Radar Altimeter-Based Terrain Referenced Navigation. *Sensors* **2019**, *19*, 1688.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 March 2020 doi:10.20944/preprints202003.0442.v1

Peer-reviewed version available at *Remote Sens.* **2020**, *12*, 1396; doi:10.3390/rs12091396

12. Hager, J.R.; Almsted, L.D. Methods and systems for enhancing accuracy of terrain aided navigation systems, 2008. US Patent 7,409,293.
13. Turan, B.; Kutay, A.T. Particle filter studies on terrain referenced navigation. 2016 IEEE/ION Position, Location and Navigation Symposium (PLANS). IEEE, 2016, pp. 949–954.
14. Lee, J.; Bang, H. A robust terrain aided navigation using the Rao-Blackwellized particle filter trained by long short-term memory networks. *Sensors* **2018**, *18*, 2886.
15. Groves, P.D.; Handley, R.J.; Runnalls, A.R. Optimising the integration of terrain referenced navigation with INS and GPS. *The Journal of Navigation* **2006**, *59*, 71–89.
16. Peters, M.A. Development of a TRN/INS/GPS integrated navigation system. IEEE/AIAA 10th Digital Avionics Systems Conference. IEEE, 1991, pp. 6–11.
17. Luyang, T.; Xiaomei, T.; Huaming, C.; Xiaohui, L. An Adaptive Federated Filter in Multi-source Fusion Information Navigation System. IOP Conference Series: Materials Science and Engineering. IOP Publishing, 2018, p. 062195.
18. Carlson, N.A. Federated square root filter for decentralized parallel processors. *IEEE Transactions on Aerospace and Electronic Systems* **1990**, *26*, 517–525.
19. Carlson, N.A.; Berarducci, M.P. Federated Kalman filter simulation results. *Navigation* **1994**, *41*, 297–322.
20. Seo, J.; Lee, J.G.; Park, C.G. A New error compensation scheme for INS vertical channel. *IFAC Proceedings Volumes* **2004**, *37*, 1119–1124.
21. Lee, J.; Sung, C.; Park, B.; Lee, H. Design of INS/GNSS/TRN Integrated Navigation Considering Compensation of Barometer Error. *Journal of the Korea Institute of Military Science and Technology* **2019**, *22*, 197–206.
22. Jeong, S.H.; Yoon, J.H.; Park, M.G.; Kim, D.Y.; Sung, C.K.; Kim, H.S.; Kim, Y.H.; Kwak, H.J.; Sun, W.; Yoon, K.J. A performance analysis of terrain-aided navigation (TAN) algorithms using interferometric radar altimeter. *Journal of the Korean Society for Aeronautical & Space Sciences* **2012**, *40*, 285–291.
23. Lee, J.; Bang, H. Radial basis function network-based available measurement classification of interferometric radar altimeter for terrain-aided navigation. *IET Radar, Sonar & Navigation* **2018**, *12*, 920–930.
24. Yoo, Y.M.; Lee, S.M.; Kwon, J.H.; Yu, M.J.; Park, C.G. Profile-based TRN/INS integration algorithm considering terrain roughness. *Journal of Institute of Control, Robotics and Systems* **2013**, *19*, 131–139.
25. Candy, J.V. *Bayesian signal processing: classical, modern, and particle filtering methods*; Vol. 54, John Wiley & Sons, 2016.
26. Pitt, M.K.; Shephard, N. Filtering via simulation: Auxiliary particle filters. *Journal of the American statistical association* **1999**, *94*, 590–599.
27. Teixeira, F.C.; Quintas, J.; Pascoal, A. AUV terrain-aided navigation using a Doppler velocity logger. *Annual Reviews in Control* **2016**, *42*, 166–176.
28. Teixeira, F.C.; Quintas, J.; Maurya, P.; Pascoal, A. Robust particle filter formulations with application to terrain-aided navigation. *International Journal of Adaptive Control and Signal Processing* **2017**, *31*, 608–651.
29. Han, K.J.; Sung, C.K.; Yu, M.J.; Park, C.G. Performance Improvement of Laser TRN using Particle Filter Incorporating Gaussian Mixture Noise Model and Novel Fault Detection Algorithm. ION/GNSS+ 2018 the 31st International Technical Meeting of the Satellite Division of The Institute of Navigation. ION, 2018, pp. 3317–3326.
30. Munshi, A.; Gaster, B.; Mattson, T.G.; Ginsburg, D. *OpenCL programming guide*; Pearson Education, 2011.
31. Yigit, H.; Yilmaz, G. Development of a GPU accelerated terrain referenced UAV localization and navigation algorithm. *Journal of Intelligent & Robotic Systems* **2013**, *70*, 477–489.
32. Lee, J.; Sung, C.; Oh, J. Terrain Referenced Navigation Using a Multilayer Radial Basis Function-Based Extreme Learning Machine. *International Journal of Aerospace Engineering* **2019**, *2019*.
33. Chen, X.; Wang, C.; Wang, X.; Ming, M.; Wang, D. Research of Terrain Aided navigation system. 2015 International Conference on Automation, Mechanical Control and Computational Engineering. Atlantis Press, 2015.