

Article

Not peer-reviewed version

A Practical Approach for Real-Time DOV Compensation in an Embedded computer of an INS using Multi-Layer Perceptron

[Hyunseok Kim](#)^{*}, Hyungsoo Kim, Yunhyuk Choi, Yunchul Cho, [Chansik Park](#)^{*}

Posted Date: 25 September 2023

doi: 10.20944/preprints202309.1667.v1

Keywords: inertial navigation system (INS); earth gravity model (EGM); deflection of vertical (DOV); multi-layer perceptron (MLP)



Preprints.org is a free multidiscipline 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 is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Article

A Practical Approach for Real-Time DOV Compensation in an Embedded Computer of an INS Using Multi-Layer Perceptron

Hyunseok Kim ^{1,*}, Hyungsoo Kim ¹, Yunhyuk Choi ¹, Yunchul Cho ¹ and Chansik Park ²

¹ Agency for Defense Development, Yuseong P.O. Box 35, Daejeon 34186, Korea; hs_kim@add.re.kr(H.K.); yh_choi@add.re.kr(Y.C.); ycheol@add.re.kr(Y.C.)

² Dept. of Intelligent Systems & Robotics, Research Institute for Computer and Information Communication, Chungbuk National University, Korea; chansp@cbnu.ac.kr

* Correspondence: hyskim77@add.re.kr

Abstract: In this paper presents analytical findings regarding the influence of gravity disturbance on INS. A data from the high-precision gravity model, EGM2008, we introduce a novel real-time gravity disturbance compensation approach for INS. This method estimates gravity disturbances along paths for platforms on both land and water, utilizing the Multi-Layer Perceptron (MLP) technique. To estimate these gravity disturbances, we developed four distinct MLP models and compared their training results using Half-Mean Square Error (half-MSE) and Root Mean Square Error (RMSE). This comparative analysis allowed us to identify the MLP model that exhibited the best performance. In order to validate the proposed gravity compensation method, field test was conducted using a test vehicle. The results of this field test unequivocally demonstrated the effectiveness of our approach in enhancing positioning accuracy. Over the course of a 2-hour field experiment, it was evident that employing the proposed gravity compensation method improved positioning accuracy by a notable 24%.

Keywords: inertial navigation system (INS); earth gravity model (EGM); deflection of vertical (DOV); multi-layer perceptron (MLP)

1. Introduction

The Inertial Navigation System (INS) is an autonomous system capable of continuously determining a platform's attitude, speed, and position using its built-in gyroscope and accelerometer [1], [2]. This self-contained feature makes the INS impervious to external interference, rendering it invaluable in both military and civilian applications such as missile navigation and civil aircraft. However, the INS, particularly in the case of the dead reckoning approach, is susceptible to accumulating errors stemming from its own inertial sensor inaccuracies. Among these error sources, gyroscope and accelerometer drift bias emerge as significant contributors to INS inaccuracies. Substantial efforts have been invested in reducing the magnitude of these biases. On one hand, advanced technologies like cold atom interferometry have been employed to enhance the precision of inertial sensors [3]. On the other hand, system-level compensation methods, such as rotational modulation, have significantly improved INS performance [4]. As INS performance has improved, previously overlooked error sources have come to the forefront, with gravity disturbance being one of the key factors limiting the further enhancement of INS accuracy [5].

Traditionally, INS has relied on a value known as normal gravity or theoretical gravity for gravity compensation. This normal gravity is always perpendicular to the Earth's ellipsoid, resulting in a horizontal component of '0' and only a vertical component. The value can be computed using the Taylor series expansion of the Somigliana formula. The difference between normal gravity and actual Earth gravity is termed gravity disturbance, encompassing the gravity anomaly (the magnitude difference between normal and actual gravity) and the horizontal gravity disturbance known as the

deflection of the vertical (DOV), which represents the angular deviation from normal gravity to Earth gravity. In error analysis of INS, DOV primarily impacts initial alignment and speed calculation accuracy. Several studies, including those by Kwon and Jekeli [6], Jekeli [7], and Jekeli et al. [8], have explored the performance enhancement achieved through gravity compensation in error-free INS.

Hanson [8] investigated why DOV compensation is not universally applied in initial alignment and found a correlation between uncompensated accelerometer drift bias and DOV. However, the conclusions in [8] were qualitative, and the compensation procedure presented was specific to certain cases. George [9] introduced the LN-93E, a military-standard ring laser inertial navigation device that improved performance through DOV compensation during alignment. It's worth noting that in prior studies, DOV compensation values were preloaded before alignment, relying on precalculated or measured values along the route, rather than real-time calculations. Determining high-quality DOV compensation values over a large area requires significant computational resources, limiting DOV compensation to initial alignment in practice.

In [10, 11], researchers explored gravity disturbance compensation for INS by estimating gravity disturbance based on measured gravity data and compensating for it within the INS equations to minimize positional error propagation. Recently, there has been increasing interest in employing high-resolution global gravity field models to compensate for gravity disturbance in INS. Many such models are based on spherical harmonic models (SHM), with the Earth Gravity Model 2008 (EGM2008) being a prominent example, featuring a maximum spherical degree and order of 2160. Studies [13-16] have investigated how SHM, particularly EGM2008, can enhance INS performance. However, much of this work focused on simplifying SHM for real-time applications due to limited computing power, especially in onboard INS computers. For instance, Wang et al. [13] assessed the feasibility and accuracy of a modified SHM for real-time INS applications, while Wang et al. [14] proposed a simplified 2D second-order polynomial model derived from SHM. Wu et al. [15] explored the effective minimum update rate of gravity using SHM for marine and airborne INS. [16] investigated DOV compensation using EGM2008 for both initial alignment and navigation solution calculation, concluding that compensating DOV only during navigation solution calculation is most desirable. Extensive research into gravity disturbance compensation, particularly for initial alignment and INS navigation solutions, has been ongoing.

In [11], an artificial neural network called ELM was applied to learn precalculated gravity drift bias in real time with high precision, providing gravity and DOV values. However, this study did not present the computation time results for the artificial neural networks, which can vary based on the network size and computing environment.

Despite existing research confirming the potential for DOV compensation to improve INS navigation accuracy, there is a notable gap in the literature regarding methods for real-time DOV compensation within the embedded computer of an INS. Consequently, this paper explores the use of a Multilayer Perceptron (MLP) artificial neural network for real-time DOV compensation and assesses its impact on position accuracy.

To accomplish this, the research structure is organized as follows: Section II introduces the reference frame related to INS and defines and calculates the gravity disturbance vector using the SHM method. Section III provides a detailed analysis of the coupling relationship between horizontal gravity disturbance and INS attitude during initial alignment. In Section IV, we introduce and train an MLP for real-time DOV compensation during INS navigation and present the results of field tests to demonstrate the effectiveness of DOV compensation. Finally, Section V offers concluding remarks.

2. Reference coordinate frame and definition of gravity disturbance

2.1. Reference coordinate frame

In the context of INS, it's essential to define parameters within specific reference frames. Additionally, vectors initially expressed in one frame often need to be transformed into another. Hence, this section explores the coordinate frames employed in deriving INS [5].

- *Earth-Centered Inertial Frame 'e'*: The Earth-Centered inertial (ECI) frame is at the center of the Earth. Its z-axis aligns with the direction of the North Pole, the x-axis extends toward the mean vernal equinox, and the y-axis forms a right-handed orthogonal frame to complete the orientation. Importantly, the ECI frame remains nonrotating in relation to distant galaxies. Consequently, the outputs of both the gyroscope and accelerometer are referenced relative to this fixed ECI frame.
- *Earth-Centered Earth-Fixed Frame 'e'*: The origin of this coordinate frame is located at the center of the Earth. The z-axis extends in the direction of the North Pole, the x-axis aligns with the Greenwich meridian, and the y-axis completes a right-handed orthogonal frame. This coordinate frame rotates along with the Earth at a constant rate denoted $\omega_{ie}^e = [0 \ 0 \ \Omega]$. The Ω represents the angular velocity of the Earth's rotation.
- *Body coordinate frame with Right-Forward-Up definition 'b'*: This frame is defined based on the input axis of the inertial sensor, mounted on the platform, and points to the right-forward-up of the platform respectively.
- *Navigation coordinate frame with East-North Up definition 'n'*: This frame is the local geodetic coordinate frame with its origin at the vehicle's position. Its xyz-axes point towards east, north, and up, respectively. The INS calculation is conducted in the navigation frame. Therefore, all vectors should be transformed into this frame, represented by geodetic latitude (L), longitude (λ), and height (h). The position vector in Earth-Centered Earth-Fixed (ECEF) frame can be obtained according to the geodetic values as:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} (R_N + h)\cos L \cos \lambda \\ (R_N + h)\cos L \sin \lambda \\ (R_N(1 - e^2) + h)\sin L \end{bmatrix} \quad (1)$$

where R_N is the radius of curvature in the prime vertical and e is the first eccentricity of the reference ellipsoid. The direction cosine matrix (DCM) between the ECEF frame and the navigation frame is a function of the geodetic position. This relationship will be employed in the subsequent gravity calculation.

$$C_e^n = \begin{bmatrix} -\sin \lambda & \cos \lambda & 0 \\ -\sin L \cos \lambda & -\sin L \sin \lambda & \cos L \\ \cos L \cos \lambda & \cos L \sin \lambda & \sin L \end{bmatrix} \quad (2)$$

The previously mentioned reference frames are commonly utilized in INS. However, it's essential to note that gravity potential is conventionally expressed in spherical coordinates. To use high-precision gravity data effectively, we need to establish the connection between Cartesian coordinates and spherical coordinates. In the spherical coordinate system, the position vector is represented as (r, θ, λ) with θ representing the spherical polar angle, as illustrated in Figure 1. As Figure 1 suggests, we can define the relationship between Cartesian coordinates and spherical coordinates using Equation (3). By combining Equation (1) and (3), we can determine the position (r, θ, λ) in the spherical coordinate system based on the geodetic position (L, λ, h) .

$$\begin{cases} x = r \sin \theta \cos \lambda \\ y = r \sin \theta \sin \lambda \\ z = r \cos \theta \end{cases} \quad (3)$$

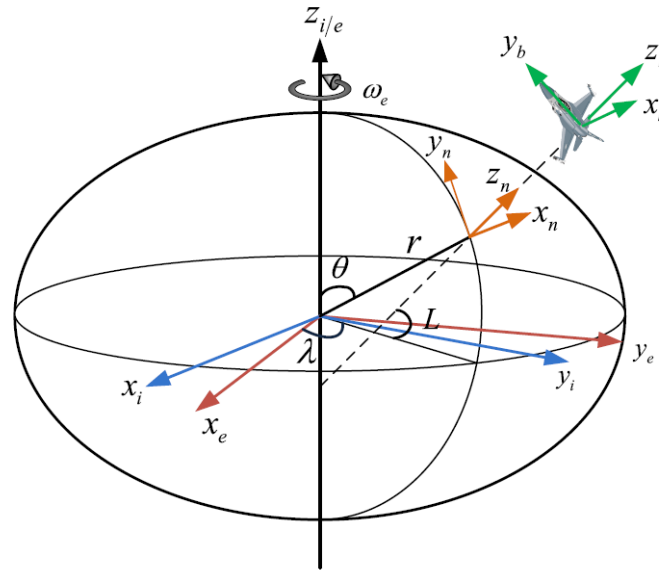


Figure 1. The definition of the coordinate system.

2.2. Gravity disturbance and deflection of vertical

Figure 2 illustrates the concept of a gravity disturbance. According to potential theory, the gravity vector corresponds to the perpendicular line of an equipotential surface of gravity. The Earth's equipotential surface of gravity is highly complicated, and for practical purpose, we often approximate it using a reference ellipsoid model like WGS-84. As depicted in Figure 2, g represents the Earth's gravity vector at point P and γ denotes the normal gravity vector at the same point. The gravity disturbance vector, in turn, is the difference between the Earth's gravity vector and the normal gravity vector. The disparity in their magnitudes define the gravity disturbance, while the difference in their directions gives rise to the DOV. Owing to the DOV, there exist certain projection components of the Earth's gravity vector within the horizontal plan, which are referred to as horizontal gravity disturbance.

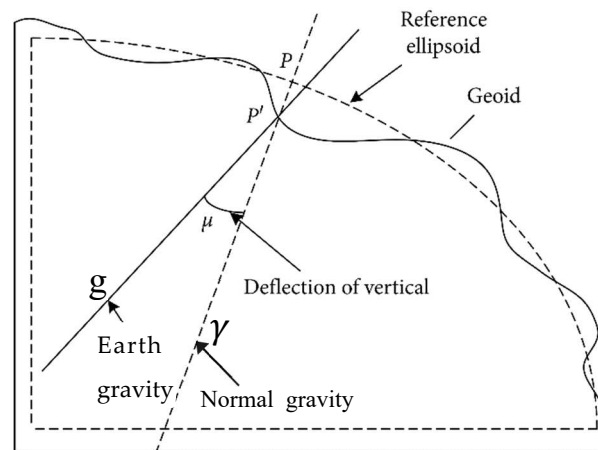


Figure 2. Description of gravity disturbance vector and normal gravity.

The gravity disturbance, Δg^n , is defined as the difference between the Earth true gravity, g^n , and the normal gravity, γ^n , and can be expressed as:

$$\Delta g^n = g^n - \gamma^n = [\Delta g_E \quad \Delta g_N \quad g_U]^n - [0 \quad 0 \quad \gamma]^n = [\Delta g_E \quad \Delta g_N \quad \Delta g_U]^n \quad (4)$$

where the eastern and northern components of the gravity disturbance are denoted by Δg_E and Δg_N . Δg_U is called the normal gravity perturbation or gravity anomaly. And γ is the norm of the normal gravity vector. The superscript n means that these vectors are projected onto n -coordinate frames.

For DOV, the northern and eastern angular components are represented by ξ and η , respectively, as shown in Figure 3. The relationship between horizontal gravity disturbance and DOV is as follows [7].

$$\xi \approx \tan \xi = -\frac{\Delta g_N}{g_U}, \eta \approx \tan \eta = -\frac{\Delta g_E}{g_U} \quad (5)$$

where g_U is the magnitude of vertical gravity and can be calculated directly using the EGM2008 model.

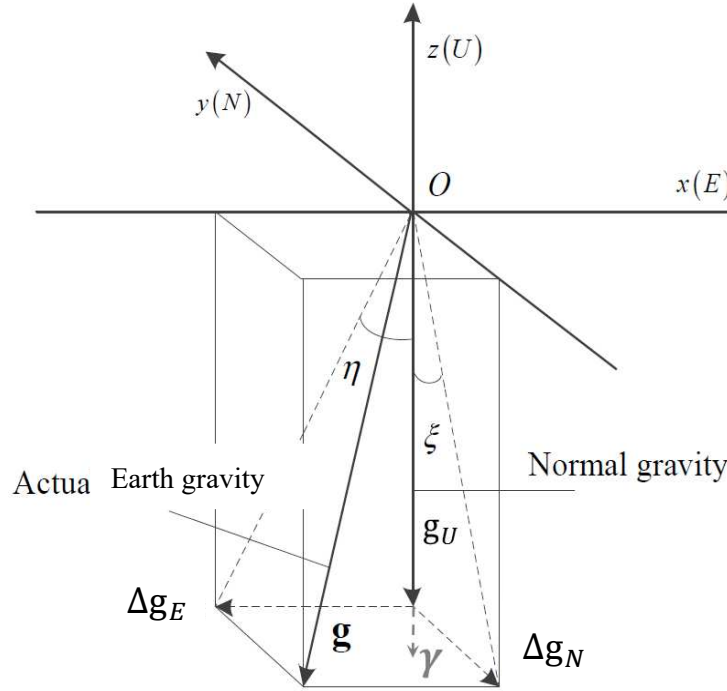


Figure 3. The definition of deflection of vertical, normal gravity and Earth gravity.

For a global target, the magnitude of the DOV component can reach up to 100 arcsec (1 arcsec = $1^\circ/3600$), corresponding to 500 mGal (1 mGal = 10^{-5} m/s²) in terms of accelerometer bias. This value is significantly larger than the bias value of 10 mGal, typical for high-precision accelerometer [6]. By incorporating Equation (5) into Equation (4), the gravity disturbance can be expressed as:

$$\Delta g^n = [-\eta g \quad -\xi g \quad \Delta g_U]^n \quad (6)$$

where horizontal gravity disturbance can be calculated through a spherical harmonic model (SHM) of the Earth's gravity field as follows [3].

$$\Delta g_N = -\frac{GM}{r^2} \sum_{n=2}^{n_{max}} \sum_{m=0}^n \left(\frac{a}{r}\right)^n [\bar{C}_{nm}^* \cdot \cos m\lambda + \bar{S}_{nm} \cdot \sin m\lambda] \frac{d\bar{P}_{nm}(\cos\vartheta)}{d\vartheta} \quad (7)$$

$$\Delta g_E = \frac{GM}{\sin\vartheta \cdot r^2} \sum_{n=2}^{n_{max}} \sum_{m=0}^n \left(\frac{a}{r}\right)^n m [\bar{C}_{nm}^* (-\sin m\lambda) + \bar{S}_{nm} \cdot \cos m\lambda] \bar{P}_{nm}(\cos\vartheta) \quad (8)$$

where G is the gravitational constant, M is the mass of the Earth, a is the length of the principal axis of the reference ellipsoid, r is the radial distance from the calculated point to the center of the reference ellipsoid, and ϑ is the latitude defined in the spherical coordinate system. λ is the longitude of the calculated point, n and m are the degree and order of the SHM, \bar{C}_{nm}^* and \bar{S}_{nm} are the coefficients of the SHM, n_{max} is the highest degree used in the SHM calculation, and $\bar{P}_{nm}(\cos\vartheta)$ is the fully normalized Legendre function of degree n and order m .

3. Impact of DOV on INS

In this chapter, we will analyze the impact of DOV on the INS initial alignment procedure.

3.1. Kinematic equations

The attitude kinematics equation using DCM parameterization is defined as:

$$\dot{\mathbf{C}}_b^n = \mathbf{C}_b^n (\boldsymbol{\omega}_{nb}^b \times) \quad (9)$$

where $\boldsymbol{\omega}_{nb}^b$ is the angular velocity of the body relative to the navigation frame and is defined as:

$$\boldsymbol{\omega}_{nb}^b = \boldsymbol{\omega}_{ib}^b - \mathbf{C}_n^b (\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n) \quad (10)$$

where $\boldsymbol{\omega}_{ib}^b$ is the angular velocity of the body relative to the inertial frame and is measured by the gyroscope, $\boldsymbol{\omega}_{ie}^n$ is the angular velocity of the Earth's rotation and is defined as:

$$\boldsymbol{\omega}_{ie}^n = [0 \quad \omega_{ie} \cos L \quad \omega_{ie} \sin L] \quad (11)$$

Also, $\boldsymbol{\omega}_{en}^n$ is the angular rate of the navigation frame relative to the ECI frame caused by the linear motion of the object on the ellipsoidal surface. $\boldsymbol{\omega}_{en}^n$ is given as:

$$\boldsymbol{\omega}_{en}^n = \left[-\frac{V_N^n}{R_M + h} \quad \frac{V_E^n}{R_N + h} \quad \frac{V_E^n}{R_N + h} \tan L \right] \quad (12)$$

where R_M and R_N are the meridian and transverse radius of the curvature of the ellipsoid, respectively. V_E^n and V_N^n are the eastern and northern components of velocity, respectively. The velocity kinematics equation of the navigation frame is defined as

$$\dot{\mathbf{V}}^n = \mathbf{C}_b^n \mathbf{f}^b - (2\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n) \times \mathbf{V}^n + \tilde{\mathbf{g}}^n \quad (13)$$

where \mathbf{f}^b is the specific force measured by the accelerometer. $\tilde{\mathbf{g}}^n$ is the gravity vector which can be the high-precision gravity calculated through EGM2008 gravity model or the normal gravity model.

The position kinematics equations are as follows:

$$\dot{L} = \frac{1}{R_M + h} V_N^n, \quad \dot{\lambda} = \frac{\sec L}{R_N + h} V_E^n, \quad \dot{h} = V_U^n \quad (14)$$

where V_U^n is the vertical component of velocity of INS represented in n -frame.

3.2. Error equations

Typically, the initial alignment is performed in a stationary state, so the actual velocity of INS is nearly zero. In this case, the non-zero velocity output of the INS represents the velocity error, and the Kalman filter (KF) can be used to estimate the corresponding error of the INS. The KF for initial alignment is built based on the error equations of the INS.

If the attitude error and velocity error are expressed as, $\boldsymbol{\Phi}$ and $\delta \mathbf{V}^n$, respectively, the corresponding linear error equation is given in [5] as follows.

$$\dot{\boldsymbol{\Phi}} = \boldsymbol{\Phi} \times (\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n) + \delta \boldsymbol{\omega}_{ie}^n + \delta \boldsymbol{\omega}_{en}^n - \boldsymbol{\varepsilon}^n \quad (15)$$

$$\delta \dot{\mathbf{V}}^n = \mathbf{f}^n \times \boldsymbol{\Phi}^n - (2\delta \boldsymbol{\omega}_{ie}^n + \delta \boldsymbol{\omega}_{en}^n) \times \mathbf{V}^n - (2\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n) \times \delta \mathbf{V}^n + \mathbf{C}_b^n \nabla^b + \delta \mathbf{g}^n \quad (16)$$

where $\boldsymbol{\Phi}$ is the attitude error vector, ϕ_E is the east component of the attitude error vector, ϕ_N is the north component, and ϕ_U is the vertical component. $\delta \mathbf{V}^n$ is the velocity error vector, ∇^b is the noise of the accelerometer, \mathbf{C}_b^n is the direction cosine matrix (DCM) from the b -frame to the n -frame, and $[\mathbf{f}^n \times]$ is the skew symmetric matrix of the specific force of the n -frame, as shown in Equation (17).

$$[\mathbf{f}^n \times] = \begin{bmatrix} 0 & -f_U & f_N \\ f_U & 0 & -f_E \\ -f_N & f_E & 0 \end{bmatrix} \quad (17)$$

where f_E is the eastern component of the specific force, f_N is the north component, f_U is the upper component.

Next, we will focus on the influence of δg^n on the INS error equation. Gravity disturbance is defined as the difference between the true gravity and the normal gravity in the INS calculation. In practice, the true value of gravity or absolutely exact gravity cannot be obtained, so the normal gravity is traditionally used in INS. We can assume that the gravity obtained by the ultra-precision gravity model like EGM2008 can be regarded as a true gravity, and the disturbance is denoted as:

$$\delta g^n = g^n - \hat{g}^n = \Delta g^n \quad (18)$$

It is evident that the gravity disturbance directly affects the velocity error propagation and has an impact on the attitude and position errors through error coupling. When a high-accuracy gravity model can be obtained, the gravity disturbance should be considered as one of the sources of INS error and compensated for. However, the effect of δg^n is influenced by the error source $C_b^n \nabla^b$. If the magnitudes of these two error sources are similar and the signs are opposite, their effects on the velocity error can cancel each other out. In this case, using a high-precision gravity model for INS may result in higher performance compared to using a normal gravity model. However, it's important to note that the sign of $C_b^n \nabla^b$, especially in dynamic cases, cannot be determined in advance, as it is time-varying under the influence of C_b^n . Therefore, in such cases, it is still advisable to use a normal gravity model conservatively.

If the accelerometer drift bias is pre-calibrated or modulated through rotation, the gravity disturbance becomes the dominant error source in the velocity error equation. In this scenario, it must be compensated for. Conversely, if the accelerometer drift bias is much larger than the gravity disturbance, the compensation performance for the gravity disturbance is expected to be reduced.

3.2. Initial alignment attitude error analysis by DOV

Assuming an initial alignment process in a stationary state, we can derive the velocity errors of the INS. The errors $\delta \dot{V}^n$, δV^n , $\delta \omega_{ie}^n$ and $\delta \omega_{en}^n$ can be determined through calculations and treated as parameters. Therefore, by rearranging Equation (16), we obtain the following Equation.

$$\alpha = f^n \times \phi^n + \nabla^n + \delta g^n = \delta V^n + (2\delta \omega_{ie}^n + \delta \omega_{en}^n) \times V^n + (2\omega_{ie}^n + \omega_{en}^n) \times \delta V^n \quad (19)$$

where α is a linear combination of known parameters and converges to '0' since we assume a stationary state. Expanding Equation (19), we can express the east and north components as follows:

$$\alpha_E = f_U \phi_N + f_N \phi_U + \nabla_E + \Delta g_E \approx 0 \quad (20)$$

$$\alpha_N = f_U \phi_E - f_E \phi_U + \nabla_N + \Delta g_N \approx 0 \quad (21)$$

where α_E and α_N represent the east and north components of α , respectively. ∇_E and ∇_N are the east and north components of the accelerometer drift bias in n -frame, respectively.

In this paper, we focus on analyzing only the east and north components as we separate the vertical and horizontal channels. We assume that the platform is in a stationary state with a negligible inclination angle. Therefore, we establish $f_E \approx 0$, $f_N \approx 0$ and $f_U \approx g_U$. Substituting these into Equations (20) and (21), we can express the east and north components of the attitude error as follows:

$$\phi_E = \frac{\alpha_N - \nabla_N - \Delta g_N}{g_U} \approx \frac{-\nabla_N - \Delta g_N}{g_U} \quad (22)$$

$$\phi_N = \frac{-\alpha_E + \nabla_E + \Delta g_E}{g_U} \approx \frac{\nabla_E + \Delta g_E}{g_U} \quad (23)$$

by combining Equation (5), Equations (22) and (23) can be rewritten as:

$$\phi_E = \hat{\phi}_E + \delta \phi_E = \frac{\alpha_N - \nabla_N}{g_U} + \xi \approx -\frac{\nabla_N}{g_U} + \xi \quad (24)$$

$$\phi_N = \hat{\phi}_N + \delta \phi_N = \frac{-\alpha_E + \nabla_E}{g_U} - \eta \approx \frac{\nabla_E}{g_U} - \eta \quad (25)$$

where ϕ_E can be divided into two parts. $\hat{\phi}_E$ is independent of the DOV and the random bias of the northern accelerometer component acts as a major error factor. Residual $\delta\phi_E$ means the eastern component of the attitude estimation error due to DOV equal to ξ . It can be divided into $\hat{\phi}_N$ and $\delta\phi_N$ in the same way for ϕ_N . Similarly, $\hat{\phi}_N$ is independent of the DOV, and the random bias of the accelerometer to the eastern accelerometer component is a major error factor, and $\delta\phi_N$ is the north component of the attitude estimation error due to the DOV, and has a negative relationship with η .

The attitude error transition function of INS in the n -frame is expressed in Equation (15). Since (15) $\delta\omega_{ie}^n$ and $\delta\omega_{en}^n$ are relatively small compared to the other error terms, we can reasonably ignore their effects in the following analysis.

Expanding Equation (15), we can express the corresponding east, north and vertical components as follows:

$$\dot{\phi}_E = (\Omega_U + \omega_U)\phi_N - (\Omega_N + \omega_N)\phi_U - \varepsilon_E \quad (26)$$

$$\dot{\phi}_N = -(\Omega_U + \omega_U)\phi_E + \omega_E\phi_U - \varepsilon_N \quad (27)$$

$$\dot{\phi}_U = (\Omega_N + \omega_N)\phi_E - \omega_E\phi_N - \varepsilon_U \quad (28)$$

where ε_E , ε_N and ε_U denote the east, north and vertical components of gyro random bias represented in the n -frame, respectively. Ω_N and Ω_U denote the north and vertical components of ω_{ie}^n ($\Omega_E = 0$).

From Equations (26) – (28), it is evident that ϕ_E , ϕ_N and ϕ_U are coupled to each other. Therefore, if the DOV causes an error in the inclination angle of the platform, it can also affect the estimation of the azimuth angle. By rearranging Equation (26) for ϕ_U , we get:

$$\phi_U = \beta - \frac{\varepsilon_E}{\Omega_N + \omega_N} \quad (29)$$

$$\beta = \frac{(\Omega_U + \omega_U)\phi_N - \dot{\phi}_E}{\Omega_N + \omega_N} \quad (30)$$

The azimuth error can be calculated from Equation (29). Since the platform's velocity is close to '0' during initial alignment, we can assume $\omega_E \approx \omega_N \approx \omega_U \approx 0$ and β can be expressed as follows:

$$\beta \cong \phi_N - \frac{\dot{\phi}_E}{\Omega_N} \quad (31)$$

Therefore, equation (29) can be expressed as:

$$\phi_U = \phi_N - \frac{(\dot{\phi}_E + \varepsilon_E)}{\Omega_N} \quad (32)$$

Combining Equations (24), (31) and (26), we can rewrite Equation (32) as:

$$\phi_U = \phi_N - \frac{(\dot{\phi}_E + \varepsilon_E)}{\Omega_N} \approx \left(\frac{\nabla_E}{g_U} - \eta \right) - \frac{1}{\Omega_N} \left(\frac{\dot{\nabla}_N}{g_U} + \xi \right) \quad (33)$$

From Equation (33), we can see that ϕ_U has a negative correlation with η and $\frac{\xi}{\Omega_N}$. Additionally, it can also be influenced by the magnitude of η rather than by ξ .

4. Real-time DOV prediction based on multi-layer perceptron

With the current increase in computing power and the widespread adoption of GPUs, neural network technology has demonstrated its value across a wide range of fields, consistently delivering exceptional results. It has become one of the most utilized tools for tackling complex problems that traditionally required intricate mathematical models. Notably, it has shown promise in addressing the real-time prediction of Degree of Vertical (DOV) in a high-order precise gravity model.

Artificial neural networks, often referred to simply as neural networks, draw loose inspiration from biological neural networks found in animal brains. A fundamental characteristic shared by both

artificial and biological neural networks is their ability to learn tasks from examples rather than relying on explicit programming. Neural networks consist of interconnected nodes known as artificial neurons, which model the neurons present in biological systems. These nodes, connected via synapses, transmit or inhibit information in response to specific activation functions [22].

The inception of artificial neural networks can be traced back to 1943 when McCulloch and Pitts introduced them [23]. Earlier implementations, known as perceptrons, comprised a series of artificial neurons connected to an output layer. It was not until 1975 that Werbos developed the backpropagation algorithm, making training of multi-layer networks (referred to as Multi-Layer Perceptrons or MLP) feasible and efficient [24]. As computing power continues to advance, thanks to GPUs and distributed systems, the depth of neural network models can increase. These deep learning networks excel at solving image and visual recognition challenges.

4.1. Basics of MLP

A multilayer neural network, known as a multilayer perceptron (MLP), consists of multiple hidden layers sandwiched between an input layer and an output layer. MLP networks excel at modeling complex nonlinear relationships. For instance, in an object identification neural network, each object can be represented as a hierarchical composition of basic image elements. Additional layers can then amalgamate features from lower layers gradually. MLP networks can be trained using standard backpropagation algorithms [25]. The weights are updated through the gradient descent method using the equation below:

$$\Delta w_{ij}(t+1) = \Delta w_{ij}(t) + \epsilon \frac{\partial C}{\partial \Delta w_{ij}} \quad (34)$$

where Δw_{ij} represents weights, ϵ is the learning rate and C denotes the cost function.

The choice of the cost function depends on factors like the learning type and activation function. For supervised learning in multi-class classification problems, a softmax function and a cross-entropy function are commonly employed as the activation and cost functions, respectively.

Despite their efficacy, deep neural networks, including MLPs, can encounter challenges such as overfitting and high time complexity. To mitigate overfitting, regularization methods like weight decay (l_2 -regularization) or sparsity (l_1 -regularization and dropout regularization have emerged. Dropout involves randomly excluding some units from hidden layers during training. Error back-propagation and gradient descent are favored for their ease of implementation and local optimization capabilities. However, training deep neural networks with these methods can be computationally intensive.

To address time complexity and overfitting concerns, techniques such as mini-batch training and dropout have been developed, offering partial solutions. Additionally, specialized GPUs optimized for matrix and vector computations have significantly enhanced learning speed.

In this study, we construct a multi-layer perceptron network using a hidden layer composed of two or more fully connected layers. We analyze accuracy by varying the number of hidden layers and nodes in the neural network model.

4.2. Training dataset

To train the MLP and assess its accuracy, a database of gravity disturbance corresponding to latitude and longitude coordinates is essential. Achieving accurate gravity disturbance calculations requires a precise gravity model, with the 2160th EGM2008 model developed by the National Geospatial-Intelligence Agency (NGA) of the United States being a representative choice for the entire Earth. In this study, we constructed an EGM2008 gravity model for the Korean Peninsula area by calculating the 2160th order spherical harmonic model on a PC. The gravity disturbance data used for network training was computed at 3 arcsec($= \frac{1^\circ}{3600}$) intervals for areas near the Korean Peninsula, covering latitudes from 33° to 39° and longitudes from 124° to 132°. This resulted in a dataset of 61,924,800 points used for MLP learning. To validate prediction accuracy during MLP training, a

verification dataset of 6,192,480 random locations (10% of the training data) was created. The results are displayed in Figure 4.

The EGM2008 gravity model is primarily a function of latitude, longitude, and altitude, represented in spherical coordinates. Given the assumption that the platform, which is the focus of this study, moves exclusively on the Earth's surface, altitude can be consistently specified based on the platform's horizontal latitude and longitude coordinates. However, defining altitude poses challenges, as various definitions exist. This will be discussed further in the next section.

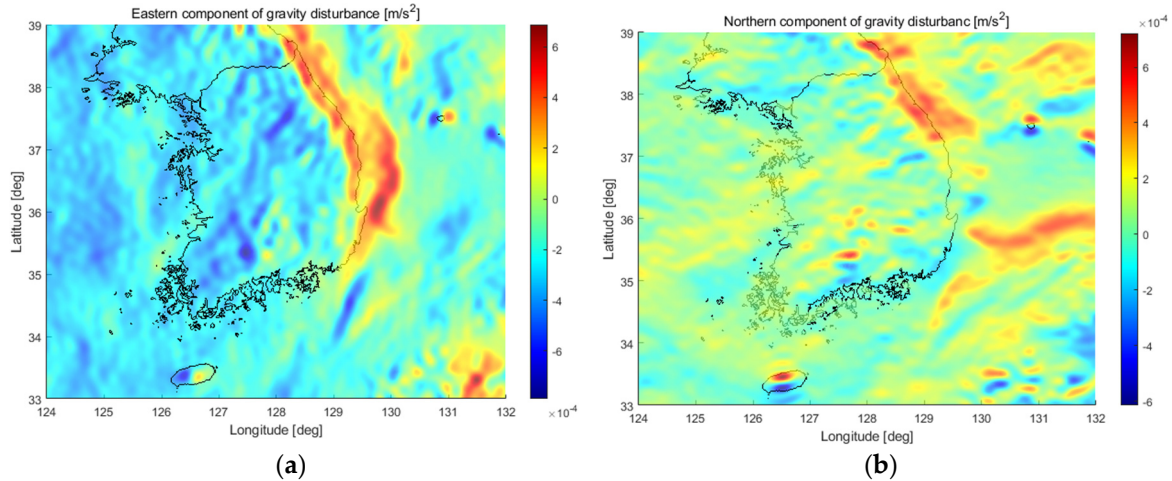


Figure 4. Horizontal gravity disturbance map of the Korean Peninsula used for MLP neural network training: (a) represents the eastern component of horizontal gravity disturbance, and (b) depicts the northern component of horizontal gravity disturbance.

4.2.1. Geoid undulation

Common definitions of altitude include GPS altitude, representing the height of an ellipsoid defined by a reference ellipsoid, and mean sea level altitude, employed in geodesy and surveying. The relationship between ellipsoidal height (h), mean sea level height (H), and geoid undulation (N) can be defined as follows:

$$h = H + N \quad (35)$$

where h is the ellipsoidal height, H is the mean-sea level height, N is called geoid undulation or geoid height, can be calculated using the EGM2008 geoid model or interpolated with pre-calculated geoid undulation values, as described in [26].

The geoid represents the shape of the sea level influenced by Earth's gravity, accounting for factors like gravity pull and rotation while excluding external influences such as wind and tides. It is a smooth yet irregular surface, shaped by the uneven distribution of mass within and on Earth. Accurate knowledge of the geoid necessitates extensive gravity measurements and calculations. Importantly, every point on the geoid surface shares the same Earth potential, the sum of gravity potential energy and centrifugal potential energy. Gravity acts perpendicular to the geoid everywhere. In the absence of other influences, the plumb line points perpendicularly, and sea level aligns parallel to the geoid. The geoid's surface rises above the reference ellipsoid in areas with positive gravity anomalies, indicating an excess of mass, and falls below the ellipsoid where negative gravity anomalies signify a deficit of mass. Geoid undulation signifies the height of the geoid relative to the reference ellipsoid. Figure 5 illustrates the distribution of geoid undulation around the Korean Peninsula, displaying a gradual gradient from southeast to northwest, ranging from approximately 15m to 31m.

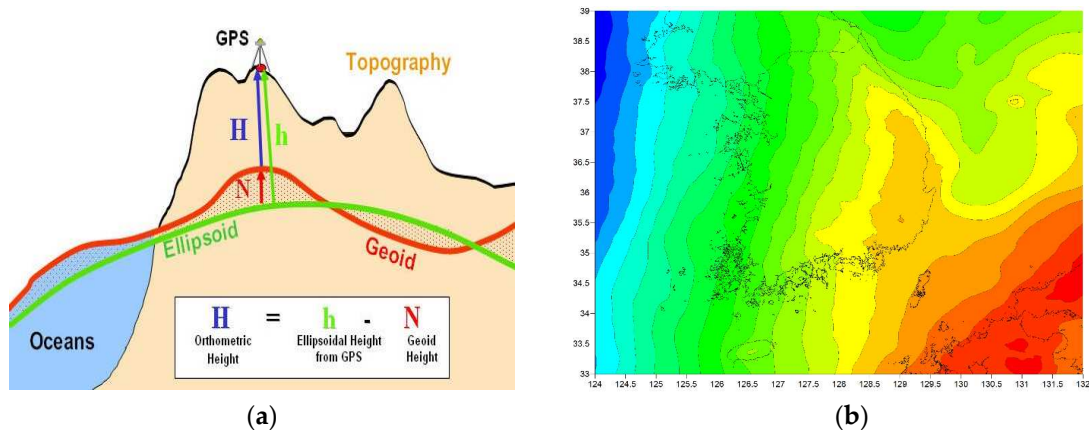


Figure 5. The definition of height: (a) illustrates the relationship between geoid height, ellipsoidal height, and orthometric height; (b) represents geoid undulation (geoid height) around the Korean Peninsula.

4.2.2. Digital elevation model(DEM)

Calculating altitude at sea is relatively straightforward, assuming mean sea level altitude as '0'. However, for land areas, the process is more complex. It involves determining the elevation of the ground surface at each grid point, accounting for terrain irregularities and the shape of the ground or road. A digital elevation model (DEM) is used for this purpose. A DEM is a grid raster data representation of terrain, excluding terrain vector features (e.g., streams, ridges) and artificial structures (wires, buildings, towers), as well as natural features like trees and vegetation.

Various definitions are associated with digital elevation models, with two prominent ones being Digital Terrain Model (DTM) and Digital Surface Model (DSM). In most cases where distinguishing between bare ground and surface objects is not critical, the generic term DEM suffices. DEMs are created from remote sensing data collected via satellites, drones, airplanes, typically employing technologies like laser scanning (LiDAR), geodetic surveying, or radar interferometry (InSAR).

- *Digital terrain model(DTM)*: DTM represents a three-dimensional portrayal of terrain or surface topography, featuring points with defined heights. DTM includes feature points such as rivers, ridges, and breaklines but excludes natural or man-made objects found on the Earth's surface, like vegetation and buildings. It approximates the altitude of bare land without accounting for surface features. DTM encompasses a set of 2D points with height values approximating the vertical distance between a feature point and a datum plane or geodetic data. In certain research fields, DTMs are considered vector datasets augmented with linear features of bare-ground terrain (e.g., breaklines, ridges). Photogrammetric processing of aerial and space stereo images is often used to create DTMs. It's noteworthy that DTMs can also be derived from DSMs by calculating the difference between the height values of surface objects (e.g., trees, buildings) and their surroundings.
- *Digital surface model(DSM)*: DSM represents a three-dimensional portrayal of the Earth's surface height, encompassing both natural and man-made objects on the Earth's surface. It characterizes the mean sea level elevation of reflective surfaces, including vegetation, buildings, and other objects above the bare ground. DSM typically overlays a canopy model onto the bare ground surface. Notable examples of DSMs include the Shuttle Radar Topography Mission (SRTM), which employs NASA's satellite-based InSAR DEM technique to cover around 80% of the Earth's landmass. SRTM data can be freely downloaded from the internet, offering global coverage with 1 arcsec resolution [27].
- *DEM vs DTM vs DSM*: It's important to distinguish between these three models. A DEM is the overarching term, including both DTM and DSM. DTM is a DEM that focuses on bare-earth elevation, while DSM incorporates all surface objects. Figure 6 illustrates the difference between DTM and DSM, with DTM following the ground's contours and DSM capturing the surface structures such as the tops of buildings and trees.

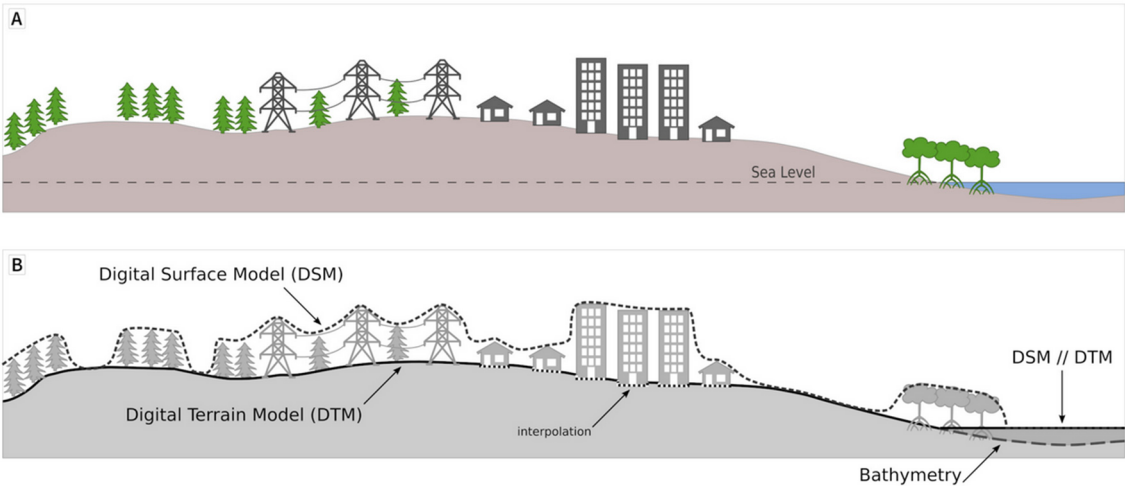


Figure 6. A schematic representation of surfaces represented by a DSM and DTM.

DEM is extensively used in digital cartography, geographic information systems, and serves as the primary basis for terrain representation. DTMs find applications in flood modeling, land use studies, geological research, and planetary science, while DSMs are preferred for tasks like telecommunications, landscape modeling, urban modeling, and visualization.

Strictly speaking, when using SRTM data, there's an error equivalent to the height of surface features when converting to ellipsoidal altitude using Equation (35) and geoid undulation. However, this study focuses on ground vehicles or water vessels, assuming that they operate solely on land with no obstacles. Therefore, it's reasonable to assume that the numerical altitude provided by SRTM coincides with DTM within the scope of this research. It's worth noting that SRTM's numerical altitude at sea is consistently '0'.

4.3. MLP network design and training

In this study, we designed and trained a neural network using the MATLAB Deep Learning Toolbox [28]. The target of our work is an inertial navigation system that calculates navigation solutions in real-time on an embedded computer powered by the P2020 dual-core 1GHz processor [29]. To meet the stringent real-time requirements, all calculations must be completed within a maximum of 1 millisecond (ms) [30].

The computational complexity, denoted as T_c , can be calculated as follows:

$$T_c = (N \times M)MUL + (N + M)ADD \tag{36}$$

Based on Equation (36), a preliminary experiment confirmed that an execution time of approximately 1 ms is required when the computational complexity (T_c) is 13,250. This is analogous to the runtime of a neural network model with 5 hidden layers, each containing 50 neurons when the input and output dimensions are 2 and 1, respectively. Therefore, we designed and trained a network with parameters falling within this computational complexity range. Four MLP networks were created and are summarized in Table 1.

Table 1. MLP Networks: Number of Hidden Layers, Neurons, and time complexity.

	Input	Number of hidden layer	Number of neuron per hidden layer	Output	T_c
MLP1	2	31	20	1	13740
MLP2	2	20	25	1	13625
MLP3	2	14	30	1	13590
MLP4	2	5	50	1	13250

In this network design, the input layer comprises two neurons to host the input vector elements [latitude, longitude]. Layers from the 2nd to the Nth serve as internal hidden layers, and each layer is fully connected. The output layer yields the eastern and northern components of the DOV. To train the neural network, the hidden layers (excluding the output layer) consist of a 'fullyConnectedLayer' for full connectivity, 'batchNormalizationLayer' to expedite learning and resolve local optima, and the 'swishLayer' as an activation function. The 'swish' activation function, given by Equation (37), is a Google-developed function that is shown to outperform ReLU in deep layer training.

$$f(x) = \frac{1}{(1 + e^{-x})}x, \quad f(\dot{x}) = f(x) + \text{sigmoid}(x)(1 - f(x)) \tag{37}$$

The MLP network learning took place in MATLAB R2021a. The network sizes for each case are shown in Table 1, and the learning conditions are detailed in Table 2. We utilized the 'adam' optimization function, suitable for regression analysis, with an initial learning rate of 0.1, decreasing by a factor of 0.4 every 5th epoch. This dynamic learning rate initially increases the verification error deviation but ultimately enhances learning speed. As epochs progress, the learning rate decreases, leading to reduced verification error deviation.

To evaluate the training outcome, we calculated the loss using Half Mean Squared Error and validated MLP network performance with Root Mean Squared Error, as defined in Equations (38) and (39).

$$\text{Half-MSE} = \frac{1}{2n} \sum (\hat{y}_i - y_i)^2 \tag{38}$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum (\hat{y}_i - y_i)^2} \tag{39}$$

Table 2. Training options for DOV prediction MLP networks.

Optimizer	Adam	Learning rate schedule	Piece wise
Mini-batch size	1,238,496	Shuffle	Once
Activation function	Swish function	Validation frequency	Every 25 th epoch
Initial learn rate	0.1	Epoch	100
Learn rate drop factor	0.4	Iteration	5000
Learn rate drop period	5	Iteration per epoch	50

The neural network training process was executed on a computer based on an Intel Xeon Gold 2.5GHz dual processor. Figure 8 illustrates the evolution of the training process, with the upper part representing RMSE prediction error and the lower part showing RMS loss. A mini-batch size of 1,238,496 was used, meaning the network received 500 database items, and neuron weight adjustments were made using the backpropagation algorithm. This resulted in 5,000 training iterations, given a total of 61,924,800 data sets in the full training set. To enhance learning rate stability and prevent abrupt fluctuations toward the end of training, we initialized the learning rate at 0.1 and reduced it by a factor of 0.4 every 5 epochs.

Every 50 calibration iterations, we assessed network accuracy against 6,192,480 validation set items, represented in black in the plot. This evaluation determined if the network was learning effectively. The training process was configured to conclude after 5,000 training iterations. Input and output data were pre-normalized to eliminate training bias.

Figure 7 displays the training progress results for the eastern and northern components of MLP1 ~ MLP4, with training times exceeding 2,000 minutes on average, except for Figure 7-(b). However, due to the similar MLP sizes in each case, we couldn't establish a meaningful correlation between neural network size and training time in this study.

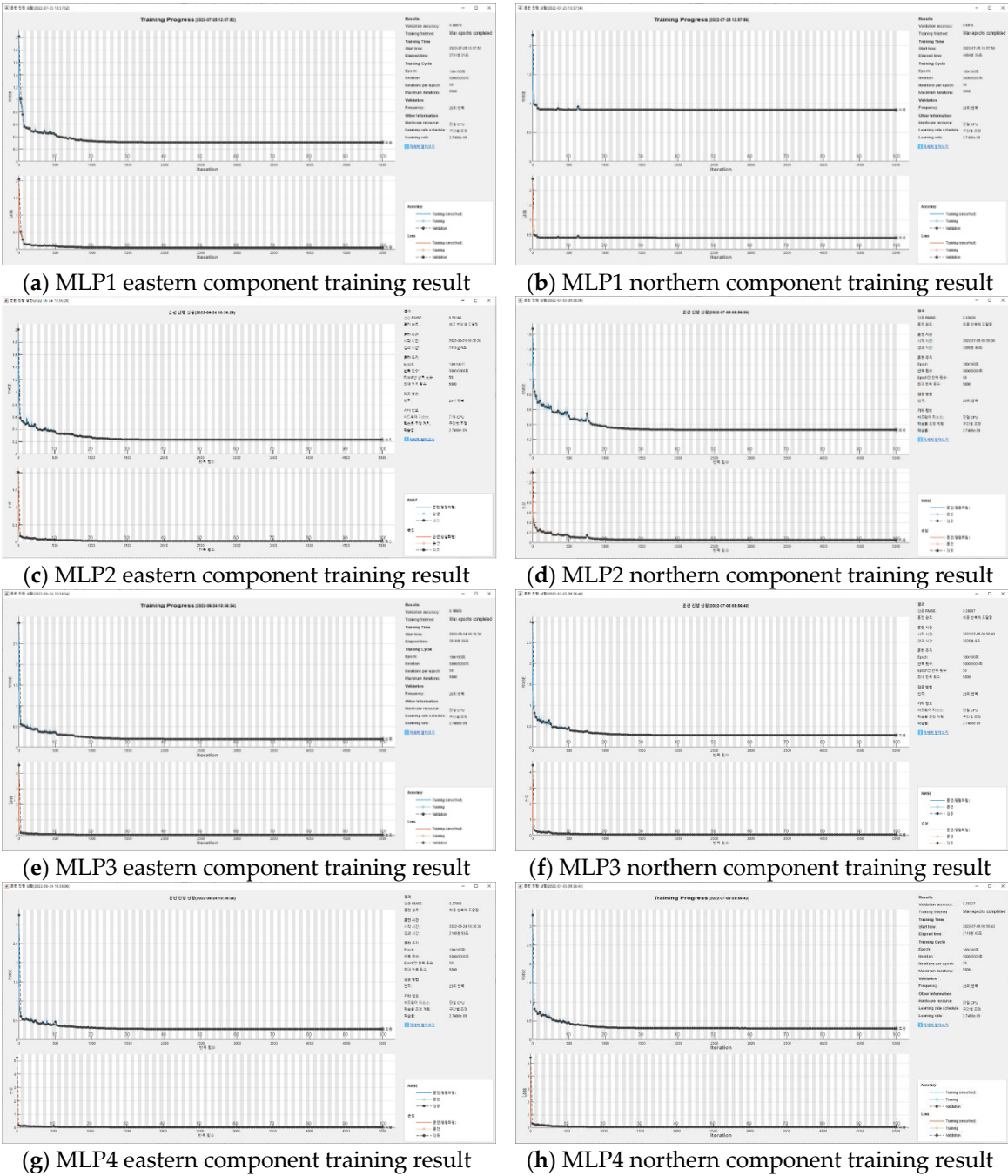


Figure 7. Training progress graphs showing results of MLP1~MLP4 networks.

In Table 3, we present the final training loss value and final validation error for each case. From the table, it's evident that MLP3 exhibits the best training performance. In the following section, we will implement MLP3 on the INS's built-in embedded computer and present the field test results.

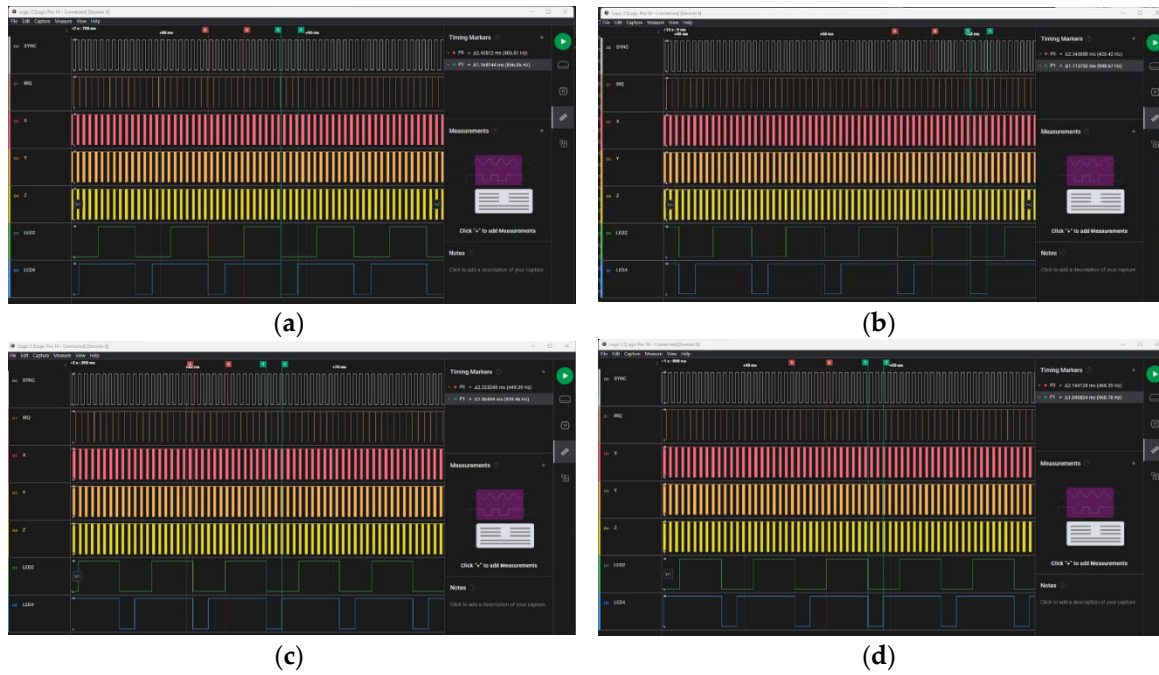
Upon completing the neural network training, we obtained the bias and weight values for each layer of every MLP neural network. These weights and biases were implemented on the actual INS embedded computer to measure the execution time of each neural network accurately. A logic analyzer was used to record the time difference before and after entering the neural network model, and the results are depicted in Figure 8. Table 4 provides the specifications of the embedded computer used in the INS and for real-time DOV calculations. To ensure real-time processing, we adopted VxWorks 6.9, a commercial real-time operating system [31]. As anticipated, the execution time of MLP1, with the highest computational complexity, was the longest, with execution times decreasing as the computational complexity reduced.

Table 3. Final validation loss and RMSE of MLP1~MLP4 training results.

	MLP1		MLP2		MLP3		MLP4	
	Δg_E	Δg_N	Δg_E	Δg_N	Δg_E	Δg_N	Δg_E	Δg_N
HMSE	0.0477	0.3974	0.0269	0.0539	0.0199	0.0420	0.0391	0.0460
RMSE	0.3087	0.8915	0.2320	0.3283	0.1993	0.2899	0.2796	0.3033

Table 4. Specifications of INS's built-in embedded computer used in field test.

CPU	Flash Rom	RAM	Real-time OS
P2020, dual core 1GHz, 512k L2 Cache	Nor Flash 16M byte	DDR2 400MHz 256M byte	VxWorks 6.9

**Figure 8.** Execution time of MLP neural network measured on embedded computer using logic analyzer. (a) Execution time of MLP1(1.168 ms); (b) Execution time of MLP2(1.113 ms); (c) Execution time of MLP3(1.064 ms); (d) Execution time of MLP4(1.041 ms).

4.4. Experimental Study on Real-Time DOV Compensation

To assess the real-time DOV compensation performance, we conducted a vehicle-mounted experiment. The experimental setup included a navigation-grade Inertial Navigation System (INS) and a single-antenna GPS receiver. The INS provided raw inertial unit measurements and position/velocity/attitude at 200 Hz, while the GPS offered reference position information at 1 Hz. The GPS provided position accuracy of 3 meters horizontally and 5 meters vertically, serving as a sufficient benchmark for evaluating INS position errors. The primary inertial sensor errors for the INS were gyro drift bias of 0.003 deg/h and accelerometer drift bias of 30 μ g. We implemented the MLP3 neural network training results on an embedded computer for the field test.

The experiment involved a journey from Daejeon Metropolitan City to the Mt. Jirisan area. During the experiment, the vehicle initially underwent alignment while stationary for approximately 900 seconds before driving for about 2 hours. Figure 9 illustrates the corresponding trajectory. Most of the experiment took place on highways with low attitude and speed. The vehicle's movement trajectory began in the lowlands to the north and progressed southward, with a noticeable rapid increase in altitude near Mt. Jirisan.

Notably, the deviation from the gravity anomaly is relatively small near the starting point but increases as the destination is approached. Additionally, the deviation of the northern component of the gravity disturbance is greater than that of the eastern component.

As an evaluation criterion for comparing navigation performance with and without DOV compensation, we used position error. The DOV was compensated in real-time using the MLP network implemented on the embedded computer board.

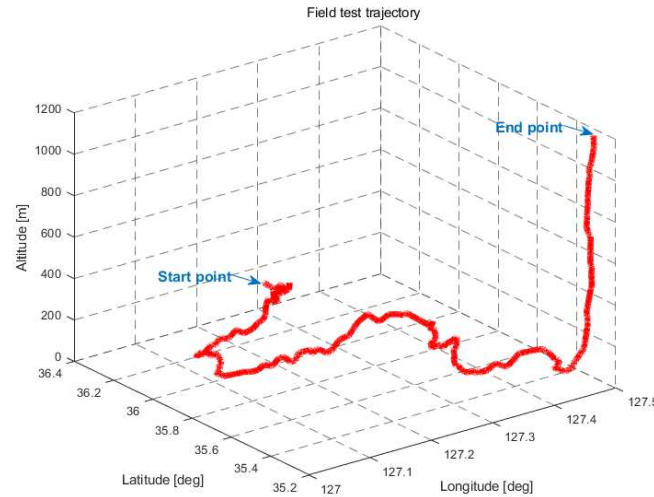


Figure 9. The field testing trajectory. The vehicle followed a route from Daejeon Metropolitan City to the summit of Mt. Jirisan.

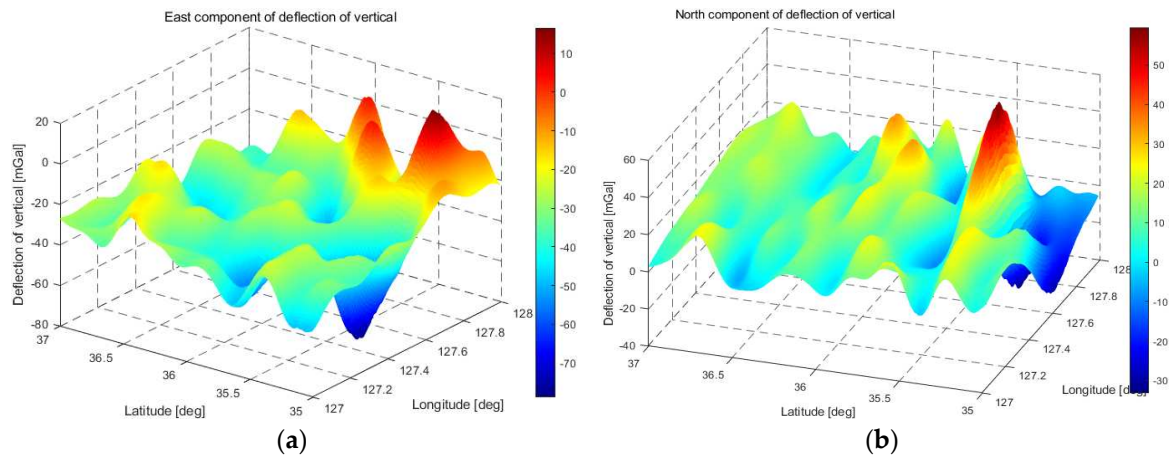


Figure 10. The graph illustrates the distribution of horizontal gravity disturbance in the field test area: (a) is the east component of the horizontal gravity disturbance; (b) is the north component of the horizontal gravity disturbance.

And in order to compare the results with and without compensation effect of DOV, the raw data of the inertial sensors were stored. In Figure 11 and Table 5, we can see that the effect of DOV compensation during the INS alignment and navigation calculation. In Figure 11, The black solid line is the result of no DOV compensation and the solid purple line is the result of calculating navigation with real-time compensating for DOV.

In particular, the compensation of DOV during INS navigation calculation resulted in a smaller position error, with a maximum horizontal position error showing approximately 27% improved performance compared to no DOV compensation.

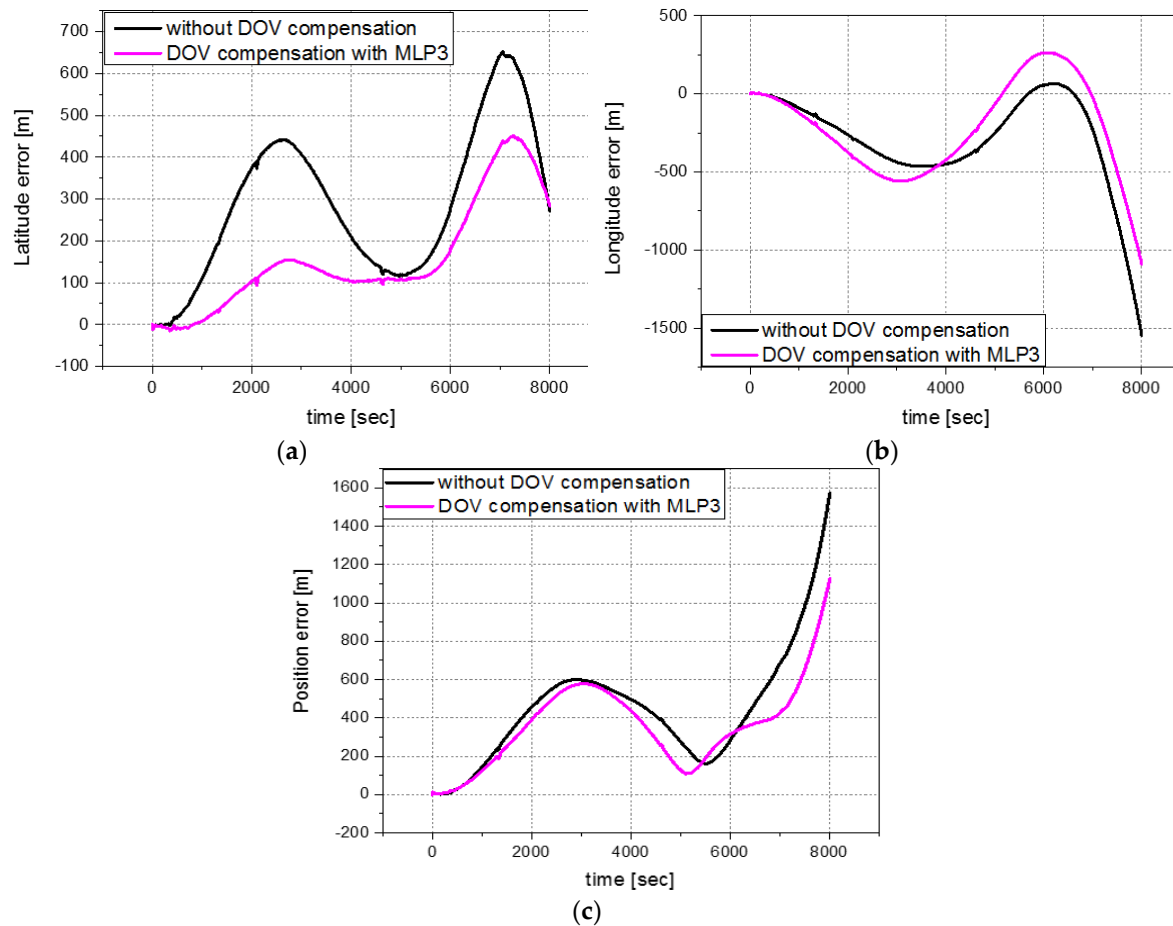


Figure 11. Position errors during the field test are compared with GPS results. In each figure, the black line represents the result of calculating navigation solution without compensation for DOV. The purple line represents the result of compensating for DOV during alignment and navigation solution calculation with MLP3 network: (a) latitude error of field test results for both cases; (b) longitude error of the field test results for both cases; (c) root sum square(RSS) error of the field test results for both case.

Table 5. Maximum and mean value of latitude, longitude, altitude and 2D position error are compared with GPS results.

	Latitude error [m]		Longitude error [m]		Altitude error [m]		Horizontal position error [m]	
	w/o DOV compensation	with DOV compensation	w/o DOV compensation	with DOV compensation	w/o DOV compensation	with DOV compensation	w/o DOV compensation	with DOV compensation
Mean [m]	290.0	140.0	-288.4	-234.1	3550.9	1126.3	446.9	361.4
Max [m]	651.6	425.1	-1547.1	-1115.2	14060.2	2257.3	1570.8	1146.6

5. Conclusions and Future Work

In this paper, we introduce a real-time method for estimating gravity disturbances using a Multilayer Perceptron (MLP) neural network trained on ground surface gravity data derived from the EGM2008 gravity model. The scope of our study pertains to platforms exclusively navigating flat land or roads on the ground surface, and it assumes that altitude for vehicles or ships is predetermined based on latitude and longitude coordinates. This limitation allows us to treat the learning data as two-dimensional points, enabling the use of MLP-based learning.

To optimize computational efficiency, we formulated a complexity formula to determine the neural network's size, ensuring that the calculation of eastern and northern components of gravity anomalies can be completed in approximately 1.2 milliseconds. As a result, we designed four distinct neural network sizes, denoted as MLP1 through MLP4. These trained neural networks were subsequently implemented on the onboard computer of an Inertial Navigation System (INS), and their execution times were measured, revealing a sequential decrease in execution time from MLP1 to MLP4.

To assess learning accuracy in the final verification process, we measured Half Mean Squared Error (MSE) and Root Mean Squared Error (RMSE), with MLP3 exhibiting the smallest error. Building on these findings, we conducted vehicle-mounted tests during journeys to the summits of Daejeon and Jirisan Mountains. The results demonstrated a reduction in position error when applying our method for Disturbance Overestimation Value (DOV) compensation, compared to results obtained without DOV compensation.

In conclusion, this paper demonstrates the feasibility of real-time compensation for gravity disturbances through the estimation of a high-precision gravity model on an INS's onboard computer. Implementing this method can enhance the accuracy of the INS, making it a valuable tool for improved navigation.

Author Contributions: In this paper, Hyunseok Kim proposed the real-time gravity disturbance compensation method for navigation grade INS based on multi-layer perceptron. Hyungsoo Kim modified the English and performed the field test. Yunhyuk Choi implemented the source code of INS algorithm. Yunchul Cho participated in the field test and analyzed the results. Chansik Park analyzed the feasibility of the algorithm.

Funding: This work supported by Agency for Defense Development Grant funded by the Korean Government (2023).

Data Availability Statement: Data sharing is not applicable to this article.

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

References

1. D. H. Titterton; J. L. Weston. *Strapdown Inertial Navigation Technology*, 2nd ed.; IEE: London, U.K., 2004.
2. X. Xu; T. Zhang; Z. Wang. In motion filter-QUEST alignment for strapdown inertial navigation systems. *IEEE Trans. Instrum. Meas.* **2018**, *67*, 1979–1993.
3. M. de Angelis *et al.*; Precision gravimetry with atomic sensors. *Meas. Sci. Technol.* **2009**, *20*, 022001, doi: 10.1088/0957-0233/20/2/022001.
4. J. H. Kwon; C. Jekeli. Gravity requirements for compensation of ultra-precise inertial navigation. *J. Navigation.* **2005**, *58*, 479–492.
5. Chang, L. B.; Qin, F.J.; Wu, M.P. Gravity Disturbance Compensation for Inertial Navigation System. *IEEE Trans. Instrum. Meas.* **2019**, *68*, 3751–3765.
6. C. Jekeli. Precision free-inertial navigation with gravity compensation by an onboard gradiometer. *J. Guid. Cont. Dyn.* **2006**, *29*, 704–713.
7. C. Jekeli; J. K. Lee; J. H. Kwon. On the computation and approximation of ultra-high-degree spherical harmonic series. *J. Geodesy* **2007**, *81*, 603–615.
8. P. O. Hanson. Correction for deflection of the vertical at the runup site. In Proceedings of the 1988 Record Navigation into the 21st Century Position Location and Navigation Symposium (IEEE PLANS '88), Orlando, FL, USA, 29 November–2 December 1988.
9. G. George. High accuracy performance capabilities of the military standard ring laser gyro inertial navigation unit. In Proceedings of the Position Location and Navigation Symposium, Las Vegas, NV, USA, 11–15 April 1994.
10. Zhou, X.; Yang, G.; Wang, J.; Li, J; An improved gravity compensation method for high-precision free-INS based on MEC-BP-adaboost. *Meas. Sci. Technol.* **2016**, *70*, 125007.
11. Zhou, X.; Yang, G.; Cai, Q.; Wang, J. A novel gravity compensation method for high precision free-INS based on extreme learning machine. *Sensors* **2016**, *16*, doi:10.3390/s16122019.

12. Förste, C.; Abrykosov, O.; Lemoine, J.-M.; Flechter, F.; Balmino, G.; Barthemes, F. EIGEN-6C4 The latest combined global gravity field model including GOCE data up to degree and order 2190 of GFZ Potsdam and GRGS Toulouse. 5th GOCE user workshop, Paris, France, 25–28 November 2014.
13. Wang, J.; Yang, G.; Li, X.; Zhou, X. Application of the spherical harmonic gravity model in high precision inertial navigation systems. *Meas. Sci. Technol.* **2016**, *27*, 095103.
14. Wang, J.; Yang, G.; Li, J.; Zhou, X. An Online Gravity Modeling Method Applied for High Precision Free-INS. *Sensors* **2016**, *16*, 1541.
15. Wu, R.; Wu, Q.; Han, F.; Liu, T.; Hu, P.; Li, H. Gravity compensation using egm2008 for high-precision long-term inertial navigation systems. *Sensors* **2016**, *16*, 2177.
16. Tie, J.; Wu, M.; Cao, J.; Lian, J.; Cai, S. The impact of initial alignment on compensation for deflection of vertical in inertial navigation. In Proceedings of the IEEE 8th International Conference on Cybernetics and Intelligent Systems, Robotics Automation and Mechatronics, Ningbo, China, 19–21 November 2017.
17. Zaki, A.; Mansi, A.H.; Selim, M.; Rabah, M.; Fiky, G.E. Comparison of Satellite Altimetric Gravity and Global Geopotential Models with Shipborne Gravity in the Red Sea. *Marine Geodesy* **2018**, *41*, 258–269.
18. Chang, L.; Li, J.; Chen, S. Initial Alignment by Attitude Estimation for Strapdown Inertial Navigation Systems. *IEEE Trans. Instrum. Meas.* **2015**, *64*, 784–794.
19. Tie, J.; Cao, J.; Chang, L.; Cai, S.; Wu, M.; Lian, J. A Model of Gravity Vector Measurement Noise for Estimating Accelerometer Bias in Gravity Disturbance Compensation. *Sensors* **2018**, *18*, 883.
20. Levinson, E.; Ter Horst, J.; Willcocks, M. The Next Generation Marine Inertial Navigator Is Here Now. In Proceedings of the 1994 IEEE Position, Location and Navigation Symposium (IEEE PLANS '94), Las Vegas, NV, USA, 11–15 April 1994.
21. Sun, X.; Chen, P.; Macabiau, C.; Han, C. Autonomous orbit determination via kalman filtering of gravity gradients. *IEEE Trans. Aerosp. Electron. Syst.* **2016**, *52*, 2436–2451.
22. Casado, R.; Bermúdez, A. Neural Network-Based Aircraft Conflict Prediction in Final Approach Maneuvers. *Sensors* **2020**, *9*, 1708, doi:10.3390/electronics9101708.
23. McCulloch, W.S.; Pitts, W. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biol.* **1943**, *5*, 115–133.
24. Werbos, P.J. Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. Ph.D. Thesis, Harvard University, Cambridge, MA, USA, 1974.
25. K. T. Lee, M. S. Kim, H. J. Kim, J. H. Kim. A Model to Predict Occupational Safety and Health Management Expenses in Construction Applying Multi-variate Regression Analysis and Deep Neural Network. *Journal of the Architectural Institute of Korea* **2021**, *37*, 217–226.
26. Hyunseok, K.; Chansik, P. A Study on Real-time Calculation of Geoid applicable to Embedded Systems. *Jour. of Advanced Navigation Tech.* **2020**, *24*, 374–381.
27. USGS.gov. Available online: <https://earthexplorer.usgs.gov> (accessed on 1 July 2023).
28. The MathWorks Inc. Deep Learning Toolbox. Available online: <https://www.mathworks.com/products/deep-learning.html> (accessed on 1 September 2023).
29. NXP semiconductors. QorIQ P2020. Available online: <https://www.nxp.com/products/processors-and-microcontrollers/power-architecture/qorIQ-communication-processors/p-series/qorIQ-p2020-and-p2010-dual-and-single-core-communications-processors:P2020> (accessed on 1 September 2023).
30. Hyunseok, K.; Chansik, P. DNN Based Geoid Undulation Prediction Accuracy Evaluation Using EGM08 Gravity Model. *Jour. of Advanced Navigation Tech.* **2020**, *24*, 374–381.
31. The Wind River Systems, Inc. VxWorks. Available online: <https://www.windriver.com/products/vxworks> (accessed on 1 September 2023).

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