

Applied an In-Motion Transfer Alignment Approach During Global Positioning System Outages Utilizing a Recurrent Neural Network Algorithm

Abstract

In this study, a Robust in-motion Transfer Alignment method based on the multilayer Neural Network, called RTA-NN, is proposed to correct the initial information of the slave vehicle based on the Strapdown Inertial Navigation System (SINS) and Global Positioning System (GPS) integrated navigation structure of the master vehicle, when the GPS data of the master vehicle is not available. For this purpose, first, the SINS equations of the master vehicle are extracted based on the low-grade Inertial Measurement Units (IMU) to compute the position, velocity, and attitude of the master vehicle. Then, a closed loop Kalman Filter structure is utilized to estimate the true states of the master vehicle in the presence of the GPS receiver. Next, a deep neural network, including multilayer Long-Short Term Memory (LSTM) and multilayer perceptron, is utilized to build the velocity model for the master vehicle based on the current and past samples of the master's IMU and the SINS outputs. Finally, the performance of the proposed approach is evaluated on the navigation units of the Unmanned Surface Vehicle (USV), considered as master, and Remotely Operated Vehicle (ROV), considered as slave. The results demonstrate the effectiveness of the proposed transfer alignment approach based on the neural network, when the GPS signals are disrupted. During 100-second GPS outages, the proposed method reduces navigation errors by 0.1%, demonstrating the robustness and accuracy of the RTA-NN approach in maintaining reliable transfer alignment without GPS.

Keywords: In-Motion Transfer Alignment, Neural Network, Global Positioning System Outages, Strapdown Inertial Navigation System/Global Positioning System integration, Long-Short Term Memory, Artificial Intelligence, Deep Learning Applications

1. Introduction

Inertial navigation system (INS) [1] is commonly utilized in vehicles due to high autonomy, high sample rate, and resistance to weather conditions. INS can compute the position, velocity, and attitude of the vehicle by integrating the acceleration and the rotation rate, provided by the IMU [2]. This navigation system is often categorized as the stable platform [3] and SINS [4], depending on the IMU placement. In a stable platform, IMU is isolated from the vehicle's rotations mechanically by mounting on a gimballed platform [4]. In comparison, in SINS, IMU is mounted on the body of the host vehicle [5]. Here, the domain of focus is the SINS, which is usually utilized in many navigation processes due to lower cost, reduced size, and greater reliability compared to stable platform systems [4].

The initialization errors of the SINS produces the erroneous outputs in a short period of time in the navigation process [6]. Hence, to start operation of the SINS, an accurate initialization of the position, velocity, and attitude, called the initial alignment, must be performed to improve the SINS performance [7]. Initial alignment procedures are categorized into the Ground Alignment (GA) [8] and the Transfer Alignment (TA) [9]. In GA, the initialization of the vehicle is performed, when the vehicle is stationary. Hence, the external aiding equipments such as the Global Positioning System (GPS) [10] are applied to initialize the position. The initial velocity is considered as zero without the environmental disturbances. Moreover, to initialize the attitude of the SINS, several techniques can be found within the literatures, the most well-known of which are the gyro compassing [11] and the zero velocity [12].

In comparison, the TA method, also called in-motion transfer alignment, performs the initialization of the vehicle, known as the Slave (S) vehicle, during the motion of the system using the data from the host carrier's navigation system, known as the Master (M) navigation unit [9]. Here, master and slave vehicles, respectively, refer to the vehicle which provides the reference navigation and the vehicle that must be aligned and calibrated [9]. Consequently, the navigation information from the SINS of the master vehicle is transformed to the SINS of the slave vehicle. Matching methods such as a velocity matching, an attitude matching, a position matching, the velocity and attitude matching [13], an accelerating match, and a rotation rate matching [14] are common approaches to solve the transfer alignment problems. In maritime applications and distributed inertial navigation systems on moving bases, a new high-accuracy transfer alignment method has been developed to improve alignment precision [15].

The accuracy of the TA approach relies on the information obtained from the master navigation unit. Hence, the data of the master SINS is integrated with the fixed positioning sensors such as GPS [10] and Doppler Velocity Log (DVL) [11] to eliminate the errors of the master navigation units. However, when the data of external aiding equipments such as loss of GPS, the reference value is not accurate [10]. Additionally, when the master data has problems, such as GPS outages, the TA method cannot function properly. Consequently, the TA method cannot estimate the slave states and the parameters of the slave navigator [9], accurately.

To solve this issue the AI approach, including pseudo GPS data generation, AI matching algorithms, and AI-based compensator algorithms are utilized to improve the accuracy of the SINS, when the GPS signals are disrupted. In the first approach, SINS data is integrated with pseudo GPS data generated by AI algorithms to enhance SINS accuracy, such as [16] used a temporal dynamic attention neural network (TDANN), [17] proposed a heterogeneous multi-task learning (MTL) model, and [18] employed a gated recurrent unit (GRU) for this purpose. Moreover, in AI-based matching algorithms, the Kalman filter is supported by AI algorithms to estimate the states of the SINS during GPS signal disruptions. For instance, [19] utilized a multi-LSTM approach, [20] proposed a deep learning architecture called the GPS/INS neural network (GI-NN), and [21] developed a navigation method combining a wavelet neural network with random forest regression (RFR-WNN). In the final approach, AI methods are applied to estimate SINS errors without relying on a Kalman Filter (KF), and these errors are subsequently compensated within the SINS algorithms. For example, [22] uses a deep neural network (DNN) to learn position corrections, [23] presents a predictor that combines an online-trained autoregressive integrated moving average (ARIMA) model with an offline-trained extreme learning machine (ELM) model, and [24] proposes a hybrid method that integrates empirical mode decomposition (EMD) with an ELM model. [25] investigates the implementation of repetitive process-based indirect-type iterative learning control mechanisms, specifically addressing batch processes characterized by model uncertainty and input delay parameters. [26] formulates and validates an advanced adaptive dynamic programming (ADP) methodology for prescribed-time control synthesis applicable to nonlinear systems exhibiting time-varying delays and parametric uncertainties. [27] establishes a theoretical framework for quantized control architectures designed for interconnected partial differential equation (PDE) systems, with particular emphasis on mobile measurement acquisition and distributed control strategy optimization.

Table 1: AI Methods for SINS Enhancement

Ref.	Method	Description	Approach
[16]	Temporal Dynamic Attention Neural Network (TDANN)	Generates pseudo GPS data to integrate with SINS, enhancing its accuracy.	Pseudo GPS Data Generation
[17]	Heterogeneous Multi-Task Learning (MTL) Model	Employs a multi-task learning model to generate pseudo GPS data for improving SINS performance.	Pseudo GPS Data Generation
[18]	Gated Recurrent Unit (GRU)	Utilizes GRU to produce pseudo GPS data, aiding in SINS accuracy during GPS outages.	Pseudo GPS Data Generation
[19]	Multi-LSTM Approach	Supports Kalman filter with a multi-LSTM model to estimate SINS states during GPS disruptions.	AI-based Matching Algorithms
[20]	GPS/INS Neural Network (GI-NN)	Proposes a neural network architecture to enhance Kalman filter-based SINS matching.	AI-based Matching Algorithms
[21]	RFR-WNN (Wavelet Neural Network + Random Forest Regression)	Combines wavelet neural networks and random forest regression to assist Kalman filter.	AI-based Matching Algorithms
[22]	Deep Neural Network (DNN)	Learns and compensates position corrections for SINS errors without relying on Kalman filter.	AI-based Compensator Algorithms
[23]	ARIMA + Extreme Learning Machine (ELM)	Hybrid model combining online ARIMA and offline ELM for SINS error compensation.	AI-based Compensator Algorithms
[24]	Empirical Mode Decomposition (EMD) + ELM	Integrates EMD and ELM to estimate and compensate for SINS errors.	AI-based Compensator Algorithms

In this study, a robust in-motion transfer alignment method based on a deep neural network called Robust Transfer Alignment Neural Network (RTANN) is proposed to align the slave INS using the master information, when the GPS signals loss. The proposed RTANN model can adapt more effectively to varying environmental conditions, such as sensor noise and nonlinear dynamics. The computational efficiency of the proposed method is on par with or better than existing alignment algorithms. When the slave vehicle physically disconnects from the master vehicle during deployment, RTANN maintains alignment accuracy despite the loss of direct reference. While the master navigation system is typically robust, GPS signal disruptions can degrade its reference data. However, RTANN ensures the slave system's alignment integrity is preserved even under these challenging conditions. This passive approach requires no additional hardware or active communication after deployment, making it academically notable for advancing deep learning applications in transfer alignment while maintaining computational efficiency. For this purpose, first, the required coordinate systems including Body, Inertial, Earth and Navigation frames are defined. Then, the sensors, utilized in the master and slave vehicles, are modeled. Next, the SINS equations are denoted based on the low-grade IMU to compute the master's position, velocity, and attitude. These equations and the outputs of the master's GPS are utilized in the closed loop Kalman Filter (KF) structure to train the multilayer neural networks. Therefore, these networks made a model to predict the position, velocity, and attitude of the master vehicle in the absence of the GPS signals. These data are utilized to align the information of the slave vehicle. To investigate the performance of the proposed approach, a ROV, launched from an USV, is simulated to align based on the USV (master) information. The simulation results demonstrate that the proposed transfer alignment method is able to provide the better accuracy, when the GPS signal is disrupted.

This research is organized as follows: The problem statement is defined in section 2. The coordinate

system and the sensor modeling are presented in sections 3 and 4, respectively. In section 5, the nonlinear equations of the SINS are denoted. The proposed in-motion transfer alignment architecture is represented in section 6. Finally, in sections 8 and 9, the numerical results and a conclusion are provided.

2. Problem Statement

The architecture of the in-motion transfer alignment method for a ROV (slave) based on the data of a USV (master) is presented in Fig 1. This structure includes, (1) sensor modeling such as IMU and GPS, (2) equations of the SINS, (3) master (USV) navigation unit with and without the GPS receiver, and (4) slave (ROV) navigation unit. The details of this block diagram are denoted in the following sections.

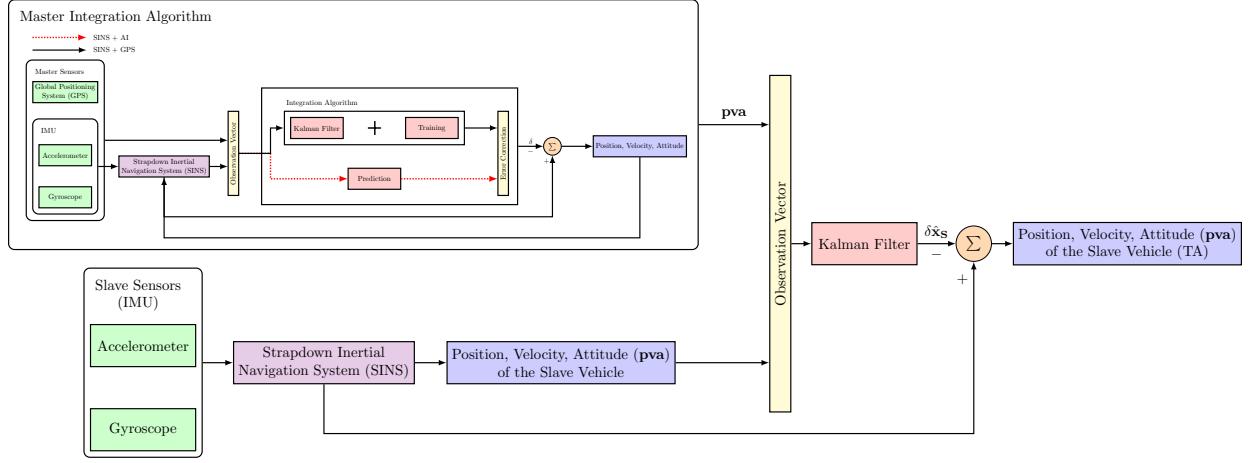
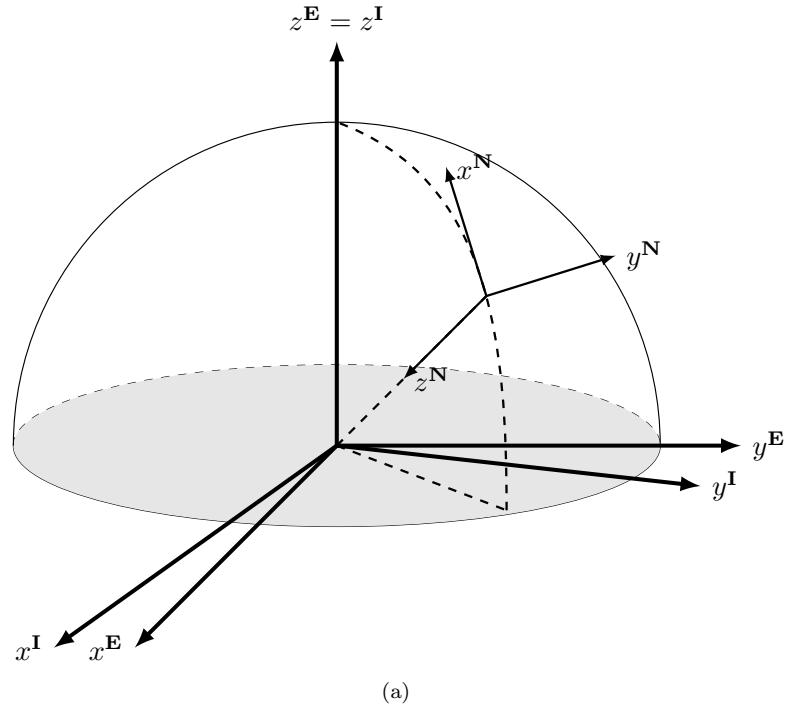


Figure 1: Architecture of the in-motion transfer alignment.

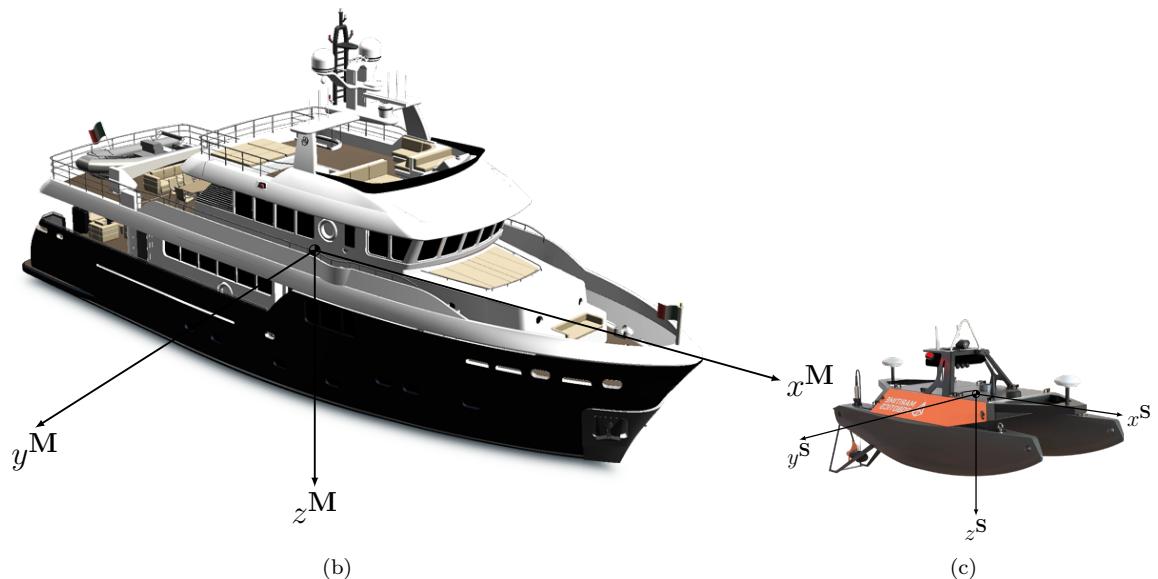
3. Coordinate Systems

For the navigation purpose, the Inertial (**I**), Earth (**E**), Navigation (**N**), and Body (**B**) frames are utilized. The origin of (**I**) and (**E**) frames, according to Fig 2, is located at the mass center of the earth. The x^E axis crosses the Greenwich meridian, the z^E axis is directed along the spin axis of the earth and y^E axis is perpendicular to the previous axes. This frame rotates along the z^E axis with the angular velocity of the earth, Ω . Unlike the **E** frame, **I** frame is the fixed referenced frame. The z^I axis is fixed to the vernal equinox and the z^I axis is also directed along the spin axis of the earth.

The **B** frame is a vehicle-fixed reference frame with the origin of the Center of Gravity (CG) of the vehicle. The x^B axis points forward, along the nose of the vehicle and z^B axis points downwards. This frame is utilized as the common frame of the inertial sensors. Here, the **B** frame of the USV and ROV vehicles, according to Fig 2, is named master (**M**) and slave (**S**) frames, respectively. The origin of **N** frame, according to Fig 2, is the vehicle's CG. x^N is directed to the north, y^E axis points to the east and z^D axis points the center of the earth. This frame is known as North-East-Down (NED) reference frame.



(a)



(b)

(c)

Figure 2: Graphical Representation of the Presented Reference Frames: (a) Inertial, Earth, and Navigation Frames (b) Master Frame (c) Slave Frame

4. Modeling

In this section, the measurements, provided by the GPS receiver and the IMU of the USV and AUV, are modeled.

4.1. Inertial Measurement Unit

The specific force, $[\tilde{\mathbf{f}}_{\mathbf{BI}}]^{\mathbf{B}}$, and the angular rate of the \mathbf{B} frame with respect to \mathbf{I} frame, $[\tilde{\boldsymbol{\omega}}_{\mathbf{BI}}]^{\mathbf{B}}$, are measured by the accelerometer and the gyroscope, respectively. Therefore, the IMU outputs of the master and slave vehicles (for $\mathbf{B} = \mathbf{M} \& \mathbf{S}$) are modeled as follows:

$$[\tilde{\mathbf{f}}_{\mathbf{BI}}]^{\mathbf{B}} = [\mathbf{f}_{\mathbf{BI}}]^{\mathbf{B}} + [\mathbf{b}_a]^{\mathbf{B}} + \boldsymbol{\eta}_a, \quad (1)$$

$$[\tilde{\boldsymbol{\omega}}_{\mathbf{BI}}]^{\mathbf{B}} = [\boldsymbol{\omega}_{\mathbf{BI}}]^{\mathbf{B}} + [\mathbf{b}_g]^{\mathbf{B}} + \boldsymbol{\eta}_g, \quad (2)$$

where $[\mathbf{f}_{\mathbf{BI}}]^{\mathbf{B}} = [f_x \ f_y \ f_z]^T$ and $[\boldsymbol{\omega}_{\mathbf{BI}}]^{\mathbf{B}} = [\omega_x \ \omega_y \ \omega_z]^T$ are, respectively, the true values of accelerometer and gyroscope sensors. The $[\mathbf{b}_a]^{\mathbf{B}} = [b_{a_x} \ b_{a_y} \ b_{a_z}]^T$ and $[\mathbf{b}_g]^{\mathbf{B}} = [b_{g_x} \ b_{g_y} \ b_{g_z}]^T$ are modeled as random constants, that vary from run-to-run, as follows:

$$\dot{\mathbf{b}}_a = \mathbf{0}, \quad (3)$$

$$\dot{\mathbf{b}}_g = \mathbf{0}. \quad (4)$$

Moreover, $\boldsymbol{\eta}_a$ and $\boldsymbol{\eta}_g$ are the Gaussian white noise of measurements.

4.2. GPS Modeling

A GPS receiver of the master vehicle measures the velocity and position vectors at 1 Hz rate. It is assumed that these measurements are computed as

$$[\tilde{\mathbf{v}}_{\mathbf{M}}]^{\mathbf{N}} = [\mathbf{v}_{\mathbf{M}}]^{\mathbf{N}} + \boldsymbol{\eta}_{v_{\mathbf{M}}}, \quad (5)$$

$$\tilde{\mathbf{p}}_{\mathbf{M}} = \mathbf{p}_{\mathbf{M}} + \boldsymbol{\eta}_{p_{\mathbf{M}}}, \quad (6)$$

where, $[\mathbf{v}_{\mathbf{M}}]^{\mathbf{N}} = [v_n \ v_e \ v_d]^T$ is the ground velocity of the master vehicle denoted in the \mathbf{N} frame and $\mathbf{p}_{\mathbf{M}} = [L \ \lambda \ h]^T$ is the position vector that represents the lateral, longitudinal and height of the master vehicle. Moreover, $\boldsymbol{\eta}_{v_{\mathbf{M}}}$ and $\boldsymbol{\eta}_{p_{\mathbf{M}}}$ denote the measurement noises of the GPS, modeled as Gaussian white noise.

5. SINS Mechanization

Here, the nonlinear equations of motion of the SINS are utilized to describe the kinematic of the vehicle including the attitude, velocity, and position. These equations are denoted as [13]:

$$\dot{\mathbf{C}}_{\mathbf{BSINS}}^{\mathbf{N}} = [\boldsymbol{\omega}_{\mathbf{NI}_{\mathbf{BSINS}}}]^{\mathbf{N}} \times \mathbf{C}_{\mathbf{BSINS}}^{\mathbf{N}} - \mathbf{C}_{\mathbf{BSINS}}^{\mathbf{N}} [\tilde{\boldsymbol{\omega}}_{\mathbf{BI}}]^{\mathbf{B}}, \quad (7)$$

$$[\dot{\mathbf{v}}_{\mathbf{BSINS}}]^{\mathbf{N}} = \mathbf{C}_{\mathbf{BSINS}}^{\mathbf{N}} [\tilde{\mathbf{f}}_{\mathbf{BI}}]^{\mathbf{B}} - \left(2[\boldsymbol{\omega}_{\mathbf{EI}_{\mathbf{BSINS}}}]^{\mathbf{N}} + [\boldsymbol{\omega}_{\mathbf{NE}_{\mathbf{BSINS}}}]^{\mathbf{N}} \right) \times [\mathbf{v}_{\mathbf{BSINS}}]^{\mathbf{N}} + [\mathbf{g}_{\text{eff}}]^{\mathbf{N}}, \quad (8)$$

$$\dot{\mathbf{p}}_{\mathbf{BSINS}} = \begin{bmatrix} \dot{L} \\ \dot{\lambda} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} \frac{v_n}{R_N + h} \\ \frac{v_e}{(R_E + h) \cos(L)} \\ -v_d \end{bmatrix}, \quad (9)$$

where $\mathbf{B} = \mathbf{M} \& \mathbf{S}$ is related to the master and slave vehicles, respectively. The notation \times denotes the cross product operator and $\mathbf{C}_{\mathbf{B}}^{\mathbf{N}}$ is the Direction Cosine Matrix (DCM), transforming the \mathbf{B} frame to \mathbf{N} frame. Moreover, $[\boldsymbol{\omega}_{\mathbf{NI}}]^{\mathbf{N}}$ is the angular velocity of the \mathbf{N} frame with respect to \mathbf{I} frame, denoted in \mathbf{N} frame, modeled as

$$[\boldsymbol{\omega}_{\mathbf{NI}_{\mathbf{BSINS}}}]^{\mathbf{N}} = [\boldsymbol{\omega}_{\mathbf{NE}_{\mathbf{BSINS}}}]^{\mathbf{N}} + [\boldsymbol{\omega}_{\mathbf{EI}_{\mathbf{BSINS}}}]^{\mathbf{N}}, \quad (10)$$

where, $[\omega_{NE_B}]^N$ is the transport rate, denoted in **N** frame, modeled as

$$[\omega_{NE_B SINS}]^N = \begin{bmatrix} \frac{v_e}{R_E + h} \\ \frac{-v_n}{R_N + h} \\ \frac{-v_e \tan(L)}{R_E + h} \end{bmatrix}, \quad (11)$$

and $[\omega_{EI_B}]^N$ is the earth rate, denoted in **N** frame, defined as

$$[\omega_{EI_B SINS}]^N = \begin{bmatrix} \Omega \cos L \\ 0 \\ -\Omega \sin L \end{bmatrix}. \quad (12)$$

Here, R_N and R_E are, respectively, the meridional and the transverse radius of the curvature, defined as

$$R_N = \frac{R_0(1 - e^2)}{(1 - e^2 \sin^2(L))^{\frac{3}{2}}}, \quad (13)$$

$$R_E = \frac{R_0}{(1 - e^2 \sin^2(L))^{\frac{1}{2}}}, \quad (14)$$

where R_0 and e are the equatorial radius and the eccentricity of the Earth ellipsoid. Finally, the term $[\mathbf{g}_{eff}]^N$ is the effective gravity, denoted in the **N** frame, represented as [1]:

$$[\mathbf{g}_{eff B SINS}]^N = \begin{bmatrix} 0 \\ 0 \\ g_0 \frac{R_G^2}{(R_G + h)^2} \end{bmatrix}, \quad (15)$$

where $R_G = \sqrt{R_N R_E}$ is the Gaussian radius and term g_0 is calculated as

$$\begin{aligned} g_0 = & 9.780318(1 + 5.3024 \times 10^{-3} \sin^2(L) \\ & - 5.9 \times 10^{-6} \sin^2(2L)). \end{aligned}$$

6. Structure of the Master Navigation

The structure of the master navigation unit is shown in Fig 3, when the GPS signal is available. The computer of the master vehicle is equipped with a SINS based on the inertial sensors, including a gyroscope and an accelerometer, as well as a GPS receiver to provide information such as position (\mathbf{p}_M), velocity (\mathbf{v}_M), and attitude (\mathbf{a}_M). Then, the difference between the velocities of the GPS and SINS is utilized in the closed loop KF to estimate the errors of the position, velocity, and attitude. The estimated errors are computed in the structure of the master SINS at each sample time.

The GPS receiver of the master navigation unit is depended on the signals, transmitted from the satellites. Therefore, the integrated navigation of the master vehicle can be distributed, when the GPS signals outage. To solve this problem, the Neural Network (NN) method, obtained from two stages: training and prediction. For this purpose, first, the proposed NN is trained based on the results of the integrated navigation of the master vehicle in the presence of the GPS data. Then, the prediction stage of the proposed NN is performed to provide the required information of the master vehicle. The structure of the integrated navigation unit of the master vehicle is shown in Fig 4, when the GPS outages. In the following, the details of the navigation structure of the master vehicle are denoted.

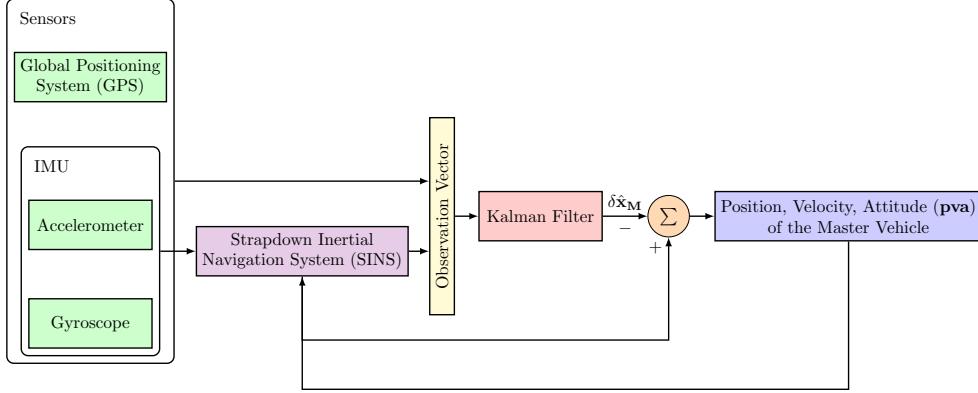


Figure 3: Structure of Integrated Navigation of the Master Vehicle

— SINS + AI
 — SINS + GPS

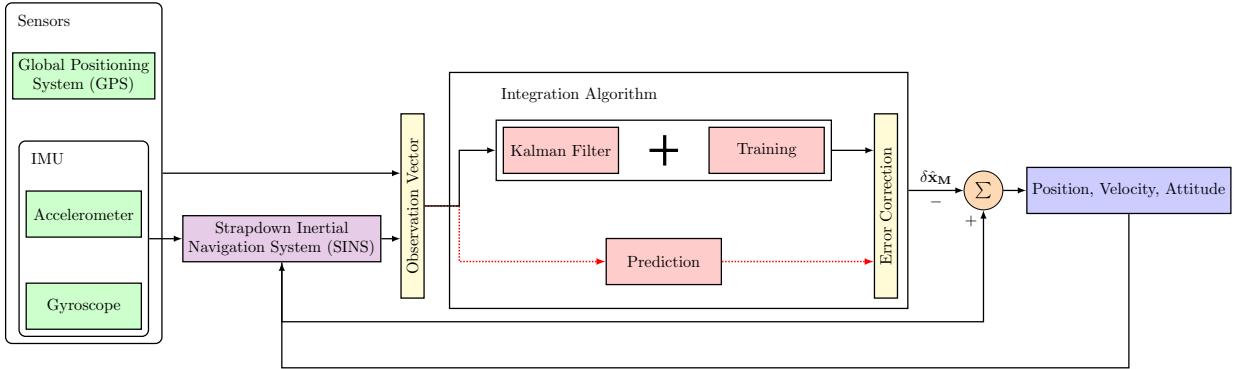


Figure 4: Structure of Integrated Navigation of the Master Vehicle, When GPS Outages

6.1. Process Model

Since the objective is to estimate the attitude errors of \mathbf{M} frame with respect to \mathbf{N} frame, denoted in \mathbf{N} frame, $[\delta\epsilon_{\mathbf{M}}]^{\mathbf{N}} = [\delta\alpha \ \delta\beta \ \delta\gamma]^T$, the velocity errors, $[\delta\mathbf{v}_{\mathbf{M}}]^{\mathbf{N}} = [\delta v_n \ \delta v_e \ \delta v_d]^T$, the position errors, $\delta\mathbf{p}_{\mathbf{M}} = [\delta L \ \delta\lambda \ \delta h]^T$, and the bias of the master's IMU, ($[\mathbf{b}_a]^{\mathbf{M}}$ & $[\mathbf{b}_g]^{\mathbf{M}}$), the states vector related to master vehicle is defined as

$$\delta\mathbf{x}_{\mathbf{M}} = \left[[\delta\epsilon_{\mathbf{M}}]^{\mathbf{N}} \ [\delta\mathbf{v}_{\mathbf{M}}]^{\mathbf{N}} \ \delta\mathbf{p}_{\mathbf{M}} \ [\mathbf{b}_a]^{\mathbf{M}} \ [\mathbf{b}_g]^{\mathbf{M}} \right]^T. \quad (16)$$

To implement the closed-loop form of the KF, the change rate of the master's states is denoted as

$$\dot{\delta\mathbf{x}}_{\mathbf{M}} = \mathbf{0}.$$

Moreover, the nonlinear process model, denoted in Equations (7) to (9), is linearized to update the covariance computation of the propagation stage of KF as follows:

$$\mathbf{P}_{\mathbf{M}} = \mathbf{F}_{\mathbf{M}} \mathbf{P}_{\mathbf{M}} \mathbf{F}_{\mathbf{M}}^T + \mathbf{Q}_{\mathbf{M}}, \quad (17)$$

where \mathbf{Q}_M is the process noise covariance of the master's SINS and \mathbf{F}_M is the state matrix for the discrete-time linear SINS equations, defined as

$$\mathbf{F}_M = \begin{bmatrix} \mathbf{I} + \tau \mathbf{F}_{11} & \tau \mathbf{F}_{12} & \tau \mathbf{F}_{13} & \mathbf{0} & \tau \mathbf{C}_{M_{SINS}}^N \\ \tau \mathbf{F}_{21} & \mathbf{I} + \tau \mathbf{F}_{22} & \tau \mathbf{F}_{23} & \tau \mathbf{C}_{M_{SINS}}^N & \mathbf{0} \\ \mathbf{0} & \tau \mathbf{F}_{32} & \mathbf{I} + \tau \mathbf{F}_{33} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}. \quad (18)$$

In the above equation, τ is the time interval and components of \mathbf{F} are computed as

$$\mathbf{F}_{11} = -\text{skew}\left(\left[\boldsymbol{\omega}_{NI_{M_{SINS}}}\right]^N\right), \quad (19)$$

$$\mathbf{F}_{12} = \begin{bmatrix} 0 & \frac{-1}{R_E + h} & 0 \\ \frac{1}{R_N + h} & 0 & 0 \\ 0 & \frac{\tan L}{R_E + h} & 0 \end{bmatrix}, \quad (20)$$

$$\mathbf{F}_{13} = \begin{bmatrix} \Omega \sin(L) & 0 & \frac{v_e}{(R_E + h)^2} \\ 0 & 0 & \frac{-v_n}{(R_N + h)^2} \\ \Omega \cos L + \frac{v_e}{(R_E + h) \cos^2 L} & 0 & \frac{-v_e \tan L}{(R_E + h)^2} \end{bmatrix}, \quad (21)$$

$$\mathbf{F}_{21} = -\text{skew}\left(\mathbf{C}_{M_{SINS}}^N \left[\tilde{\mathbf{f}}_{IM}\right]^M\right), \quad (22)$$

$$\mathbf{F}_{22} = \begin{bmatrix} \frac{v_d}{R_N + h} & -\frac{2v_e \tan L}{R_E + h} - 2\Omega \sin L & \frac{v_n}{R_N + h} \\ \frac{v_e \tan L}{R_E + h} + 2\Omega \sin L & \frac{v_n \tan L + v_d}{R_E + h} & \frac{v_e}{R_E + h} + 2\Omega \cos L \\ -\frac{2v_n}{R_N + h} & -\frac{2v_e \tan L}{R_E + h} - 2\Omega \cos L & 0 \end{bmatrix}, \quad (23)$$

$$\mathbf{F}_{23} = \begin{bmatrix} -\frac{v_e^2}{(R_E + h) \cos^2 L} - 2v_e \Omega \cos L & 0 & \frac{v_e^2 \tan L}{(R_E + h)^2} - \frac{v_n v_d}{(R_N + h)^2} \\ \frac{v_n v_e}{(R_E + h) \cos^2(L)} + 2\Omega (v_e \cos(L) - v_d \sin(L)) & 0 & -\frac{v_n v_e \tan L + v_e v_d}{(R_E + h)^2} \\ 2v_n \Omega \sin L + \frac{\partial [\mathbf{g}_{eff_B}]^N}{\partial L} & 0 & \frac{v_e^2}{(R_E + h)^2} + \frac{v_n^2}{(R_N + h)^2} - \frac{2g_0 R_G^2}{(R_G + h)^3} \end{bmatrix}, \quad (24)$$

$$\mathbf{F}_{32} = \begin{bmatrix} \frac{1}{R_N + h} & 0 & 0 \\ 0 & \frac{1}{(R_E + h) \cos L} & 0 \\ 0 & 0 & -1 \end{bmatrix}, \quad (25)$$

$$\mathbf{F}_{33} = \begin{bmatrix} 0 & 0 & -\frac{v_n}{(R_N + h)^2} \\ \frac{v_e \sin L}{(R_E + h) \cos^2 L} & 0 & -\frac{v_e}{(R_E + h)^2 \cos L} \\ 0 & 0 & 0 \end{bmatrix}. \quad (26)$$

Here, the skew is the skew-symmetric matrix operator and $\frac{\partial [\mathbf{g}_{\text{eff}, \mathbf{B}_{\text{SINS}}}]^N}{\partial L}$ is computed as

$$\frac{\partial [\mathbf{g}_{\text{eff}, \mathbf{B}_{\text{SINS}}}]^N}{\partial L} = \begin{bmatrix} 0 \\ 0 \\ \frac{R_G^2}{(R_G + h)^2} (9.780318(5.3024 \times 10^{-3} \sin^2 L - 11.8 \times 10^{-6} \sin(4L))) \end{bmatrix}. \quad (27)$$

6.2. Measurement Model

Since the linear form of the KF is utilized, the velocity and position matching, i.e., the difference between the GPS measurements (equations (5) and (6)), and master's SINS outputs (equations (8) and (9)) are considered as a measurement model. Thus, the master's measurement vector of the KF is denoted as

$$\mathbf{z}_M = \begin{bmatrix} \tilde{\mathbf{p}}_M - \mathbf{p}_{M_{\text{SINS}}} - \mathbf{T}_{\text{SINS}} \mathbf{C}_{M_{\text{SINS}}}^N [\mathbf{r}_M]^M \\ [\tilde{\mathbf{v}}_M]^N - [\mathbf{v}_{M_{\text{SINS}}}]^N - \mathbf{C}_{M_{\text{SINS}}}^N ([\tilde{\boldsymbol{\omega}}_{BI}]^M \times [\mathbf{r}_M]^M) + [\boldsymbol{\omega}_{EI_{M_{\text{SINS}}}}]^N \times \mathbf{C}_{M_{\text{SINS}}}^N [\mathbf{r}_M]^M \end{bmatrix}, \quad (28)$$

where $[\mathbf{r}_M]^M$ is the lever arm from SINS to the GPS antenna, denoted in M frame and \mathbf{T}_{SINS} is defined as follows:

$$\mathbf{T}_{\text{SINS}} = \begin{bmatrix} \frac{1}{R_N + h} & 0 & 0 \\ 0 & \frac{1}{(R_E + h) \cos L} & 0 \\ 0 & 0 & -1 \end{bmatrix}. \quad (29)$$

Therefore, the observation matrix of the KF is linearized as follows:

$$\mathbf{H}_M = \begin{bmatrix} \mathbf{H}_{11} & \mathbf{0} & -\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{H}_{21} & -\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{H}_{25} \end{bmatrix}. \quad (30)$$

Here:

$$\mathbf{H}_{11} = -\mathbf{C}_{M_{\text{SINS}}}^N [\mathbf{r}_M]^M, \quad (31)$$

$$\begin{aligned} \mathbf{H}_{21} = & -\text{skew}\left(\mathbf{C}_{M_{\text{SINS}}}^N ([\tilde{\boldsymbol{\omega}}_{BI}]^M [\mathbf{r}_M]^M) \right. \\ & \left. - [\boldsymbol{\Omega}_{EI_M}]^N \mathbf{C}_{M_{\text{SINS}}}^N [\mathbf{r}_M]^M\right), \end{aligned} \quad (32)$$

$$\mathbf{H}_{25} = -\mathbf{C}_{M_{\text{SINS}}}^N \times \text{skew}\left([\mathbf{r}_M]^M\right). \quad (33)$$

6.3. Update the Estimated State

After computation of the true states errors, $\delta \hat{\mathbf{x}}_M = [[\delta \hat{\boldsymbol{\epsilon}}_M]^N \quad [\delta \hat{\mathbf{v}}_M]^N \quad \delta \hat{\mathbf{p}}_M]^T$, by the closed-loop form of the KF true states errors vector, the true states of the master's INS are computed as

$$[\hat{\mathbf{v}}_M]^N = [\mathbf{v}_{M_{\text{SINS}}}]^N - [\delta \hat{\mathbf{v}}_M]^N, \quad (34)$$

$$\hat{\mathbf{p}}_M = \mathbf{p}_{M_{\text{SINS}}} - \delta \hat{\mathbf{p}}_M. \quad (35)$$

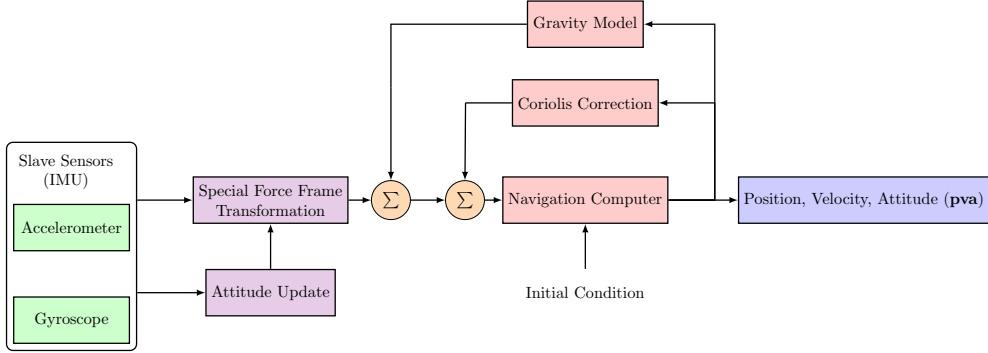


Figure 5: Structure of the in-motion transfer alignment for the slave vehicle

Moreover, $[\delta\epsilon_M]^N$ can be converted to the actual DCM, \hat{C}_M^N , as follows:

$$\hat{C}_M^N = C_{M_{SINS}}^N (\mathbf{I} - \mathbf{E}_M), \quad (36)$$

where \mathbf{I} is the identity matrix and \mathbf{E}_M is the skew-symmetric matrix of the $\delta\epsilon_M$ defined as:

$$\mathbf{E}_M = \begin{bmatrix} 0 & -\delta\gamma & \delta\beta \\ \delta\gamma & 0 & -\delta\alpha \\ -\delta\beta & \delta\alpha & 0 \end{bmatrix}. \quad (37)$$

Then, the Euler angles can be obtained from the estimated DCM, \hat{C}_M^N , as follows:

$$\hat{\phi}_M = \arctan2(\hat{C}_M^N(3, 2), \hat{C}_M^N(3, 3)), \quad (38)$$

$$\hat{\theta}_M = -\arcsin(\hat{C}_M^N(3, 1)), \quad (39)$$

$$\hat{\psi}_M = \arctan2(\hat{C}_M^N(2, 1), \hat{C}_M^N(1, 1)). \quad (40)$$

As shown in Fig. 5, the schematic of the INS structure illustrates the integration of various components including the gyroscope and accelerometer sensors, frame transformations, and navigation computations. The structure demonstrates how the sensor measurements are processed through multiple stages to obtain the final position, velocity and attitude estimates.

6.4. Optimization of Model Structure

To enhance the performance of the LSTM-based network, an optimization process was conducted using Optuna, an efficient hyperparameter tuning framework. The objective was to determine an optimal architecture by adjusting key hyperparameters, including the number of LSTM layers, the number of units per layer, the depth of the final dense layers, and the learning rate.

6.4.1. Optimization Strategy

The model structure was systematically varied, and multiple configurations were evaluated. The key elements of this process included:

- The number of LSTM layers was set to 1, 2, or 3 for both input branches.
- The number of neurons in each LSTM layer was selected from {8, 16, 32, 64}.
- The number of dense layers following feature fusion was varied between 1 and 3.
- The learning rate was adjusted within the range of $[10^{-4}, 10^{-2}]$.
- The best configuration was determined based on the minimum validation loss recorded during training.

6.4.2. Hyperparameter Search Space

The search space for the optimization is summarized in Table 2.

Table 2: Hyperparameter Search Space

Hyperparameter	Search Range
Number of LSTM layers	{1, 2, 3}
Number of neurons per LSTM layer	{8, 16, 32, 64}
Number of dense layers	{1, 2, 3}
Learning rate	[10^{-4} , 10^{-2}]

6.4.3. Implementation Details

To ensure stable convergence and prevent overfitting, early stopping and batch normalization were applied where necessary. The optimization objective was defined as:

$$\mathcal{L}_{\text{opt}} = \min \text{val_loss} \quad (41)$$

where val_loss represents the lowest validation loss observed during training.

6.4.4. Results of the Optimization

Following the optimization process, the best-performing model was identified. The optimal configuration consisted of 2 LSTM layer containing 32 neurons, followed by 2 dense layers with 32 neurons each. A learning rate of approximately 0.01 was found to yield the lowest validation loss. These results indicate that deeper architectures did not always provide superior performance, highlighting the importance of balancing complexity and generalization.

The results of the hyperparameter optimization process are visualized in Fig. 6, which shows the parallel coordinates plot of different model configurations and their corresponding performance metrics.

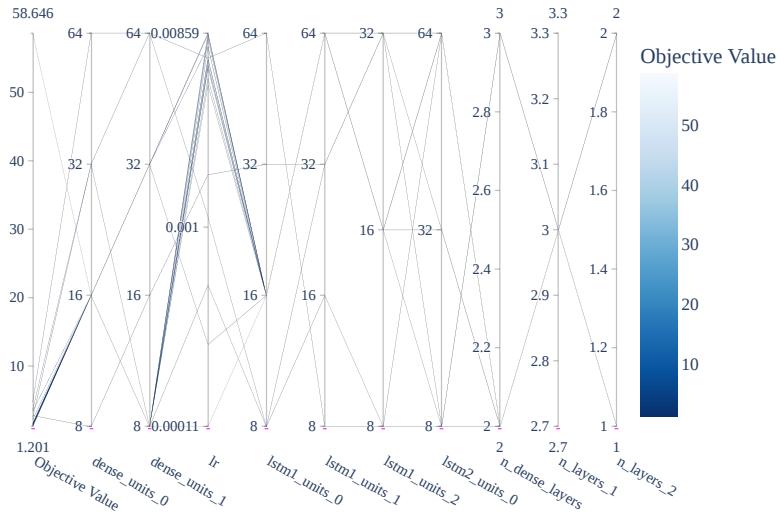


Figure 6: Parallel coordinates plot showing the relationships between different hyperparameters and model performance. Each line represents a different model configuration, with the color intensity indicating the validation loss (darker colors represent better performance).

The optimized model structure demonstrated improved accuracy compared to baseline configurations. The final model was selected based on the best trade-off between validation loss and model complexity, ensuring robust performance on unseen data.

6.4.5. Model Pruning

To further optimize the model for deployment, we applied structured pruning techniques to reduce model complexity while maintaining performance. The pruning process involved systematically removing neurons and connections based on their importance to the network's output.

The pruning strategy followed these key steps:

- Initial training of the full model to establish baseline performance
- Iterative removal of neurons based on L1-norm magnitude
- Fine-tuning of the pruned model to recover accuracy
- Evaluation against validation data to ensure performance retention

The pruning process was guided by a sensitivity threshold τ , where neurons with L1-norm below τ were candidates for removal:

$$\|w_i\|_1 < \tau \implies \text{prune neuron } i \quad (42)$$

where w_i represents the weights connected to neuron i .

Through careful experimentation, we determined that the model could be pruned by approximately 30% while maintaining 95% of its original accuracy. The final pruned architecture achieved a more efficient parameter count while preserving the essential predictive capabilities of the network. Table 3 summarizes the results of the pruning process.

Table 3: Model Pruning Results

Metric	Original Model	Pruned Model
Parameter Count	46,825	32,778
Inference Time (ms)	12.3	8.9
Validation Loss	0.2298	0.2412
Model Size (KB)	182.91	128.04

The pruned model demonstrated comparable performance to the original while offering improved computational efficiency, making it more suitable for deployment in resource-constrained environments.

6.5. GPS Outages

When the GPS signal is available, the data of the master navigation unit including the estimated position, $\hat{\mathbf{P}}_{(t_1)}$, velocity, $\hat{\mathbf{v}}_{(t_1)}$, and attitude, $\hat{\mathbf{a}}_{(t_1)}$, at sample time $t_1 < t$ and the measurement of the master's IMU for all sample time (0 to t), are stored in the database as the input. This information is trained using a Neural Network (NN) approach to produce a model for the prediction phase based on the finding the least square solution of the difference between the estimated outputs, obtained by master navigation unit (equations (34)-(35) and (38)-(40)) at sample time t , $\hat{\mathbf{y}}_{(t)} = \{\hat{\mathbf{v}}_{\mathbf{M}}, \hat{\mathbf{p}}_{\mathbf{M}}, \hat{\mathbf{a}}_{\mathbf{M}}\}$, with the predicted data, $\mathbf{y}_{(t)}$, obtained by proposed NN as follows:

$$\text{loss}_{(t)} = \|\hat{\mathbf{y}}_{(t)} - \mathbf{y}_{(t)}\|^2. \quad (43)$$

The structure of the proposed NN is shown in Fig 7. This structure is utilized in the master navigation unit to predict the position, velocity, and attitude in the absence of the GPS signal, as shown in Fig 8. In the following, subsystems of the proposed NN are detailed.

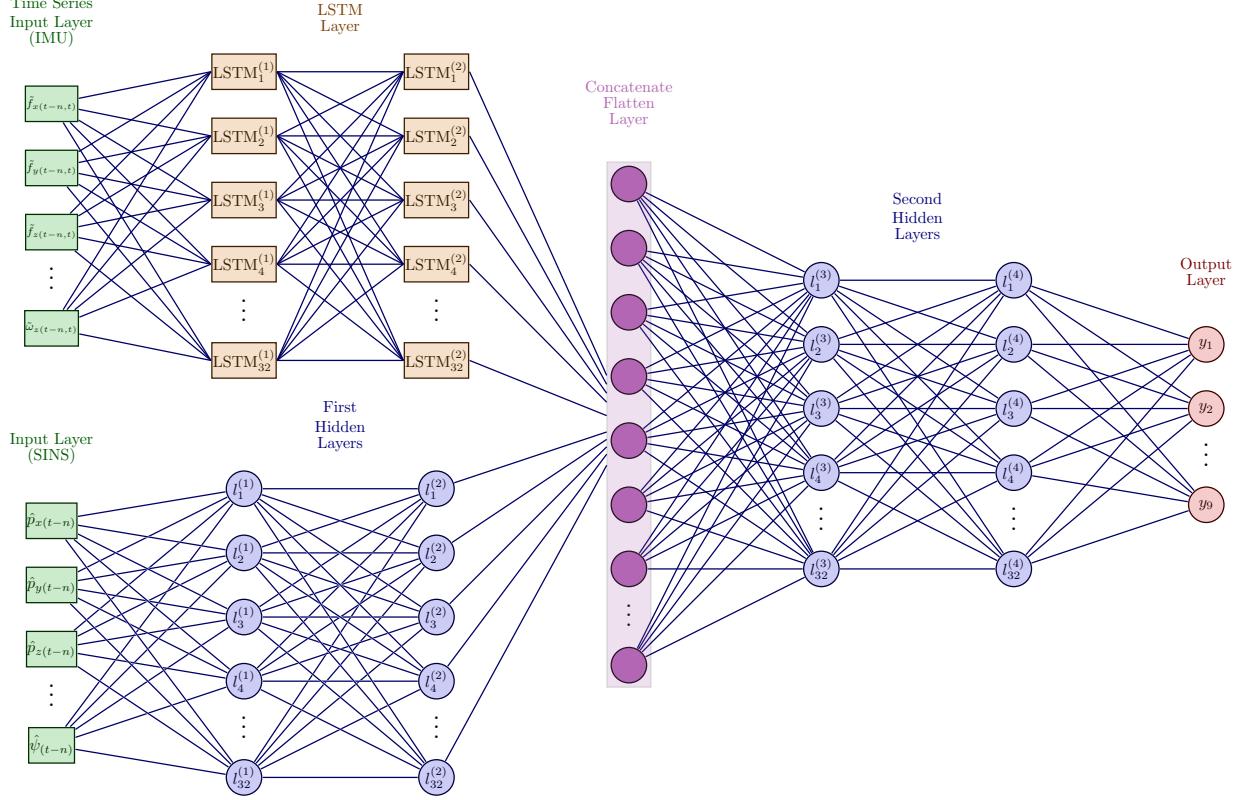


Figure 7: Structure of the Proposed Modular Neural Network.

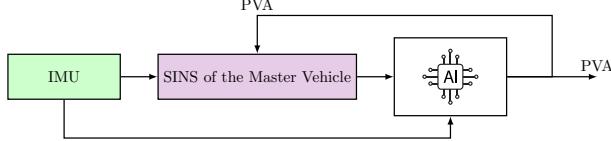


Figure 8: Structure of Master Navigation Unit in the Prediction Phase

6.6. Multilayer Neural Network

The proposed NN consists of three main layers, including the input layer, LSTM layer, and output layer, as shown in Fig 7. The input layer is fed by the estimated states of the master's SINS, including the position, velocity, and attitude, and the IMU outputs. The LSTM layer is utilized to capture the time-series data of the IMU outputs. The output layer is used to predict the position, velocity, and attitude of the master's SINS in the absence of the GPS signal. The proposed NN is trained using the data obtained from the master navigation unit, including the estimated states of the master's SINS and the IMU outputs, to predict the position, velocity, and attitude of the master's SINS in the absence of the GPS signal.

6.6.1. LSTM Layer Architecture

Here, a model is generated based on the multilayer Long-Short Term Memory (LSTM), according to Fig 7, using the sequential data including IMU outputs and estimated states obtained from KF. LSTM [28] is a class of the Recurrent Neural Network (RNN), that retains selectively and utilizes relevant information, while discarding noisy data. This architecture of the LSTM cell structure, consists of the memory cell and gating mechanisms, is illustrated in Fig 9.

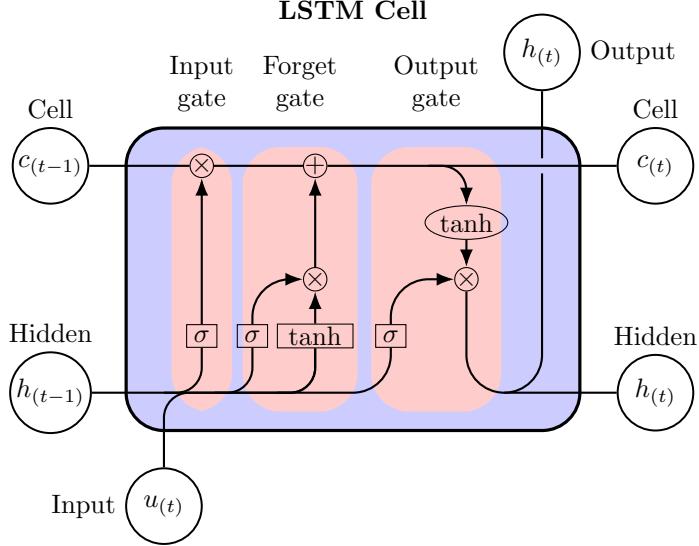


Figure 9: LSTM cell structure

In this structure, the relationship between, the output, $\mathbf{y}_{\text{LSTM}(t)}$, and input, \mathbf{u}_t is computed based on the sigmoid function, σ , as

$$\mathbf{y}_t = \sigma(\mathbf{W}_{\text{LSTM}} \cdot [\mathbf{h}_{t-1}, \mathbf{u}_t] + \mathbf{b}_{\text{LSTM}}), \quad (44)$$

where \mathbf{W}_{LSTM} and \mathbf{b}_{LSTM} are, respectively, the weight matrix and bias vector. Moreover, \mathbf{h}_{t-1} denotes the previous hidden state, containing the output information at sample time t, is updated as

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t), \quad (45)$$

where notation \odot is element wise product and \mathbf{c}_t is memory cell at sample time t, that retains the relevant information and is propagated as follows:

$$\mathbf{c}_t = \mathbf{f}_{g(t)} \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tanh(\mathbf{W}_c \cdot [\mathbf{h}_{t-1}, \mathbf{u}_t] + \mathbf{b}_c). \quad (46)$$

Moreover, \mathbf{W}_c and \mathbf{b}_c are, respectively, the weight matrix and bias vector. Also, \mathbf{i}_t and $\mathbf{f}_{g(t)}$ denote the input and forget gates to regulate the information flow and are computed based on the sigmoid function, σ , as follows:

$$\mathbf{i}_t = \sigma(\mathbf{W}_i \cdot [\mathbf{h}_{t-1}, \mathbf{u}_t] + \mathbf{b}_i), \quad (47)$$

$$\mathbf{f}_{g(t)} = \sigma(\mathbf{W}_{f_g} \cdot [\mathbf{h}_{t-1}, \mathbf{u}_t] + \mathbf{b}_{f_g}), \quad (48)$$

where \mathbf{W}_j and \mathbf{b}_j for ($j = i, f_g$) are weight matrix and bias vectors, respectively. Moreover, \mathbf{h}_t and \mathbf{c}_t are two outputs of an LSTM unit, which are called the hidden state and cell state, respectively.

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t), \quad (49)$$

$$\mathbf{c}_t = \mathbf{f}_{g(t)} \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t, \quad (50)$$

where \mathbf{i}_t , \mathbf{o}_t , $\mathbf{f}_{g(t)}$, and $\tilde{\mathbf{c}}_t$ represent the input gate, output gate, forget gate, and cell candidate state, defined as follows:

$$\mathbf{i}_t = \sigma(\mathbf{W}_i \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i). \quad (51)$$

$$(52)$$

At the final time step, the hidden state $\mathbf{h}_{(t_{\text{final}})}$ is passed to the next layer.

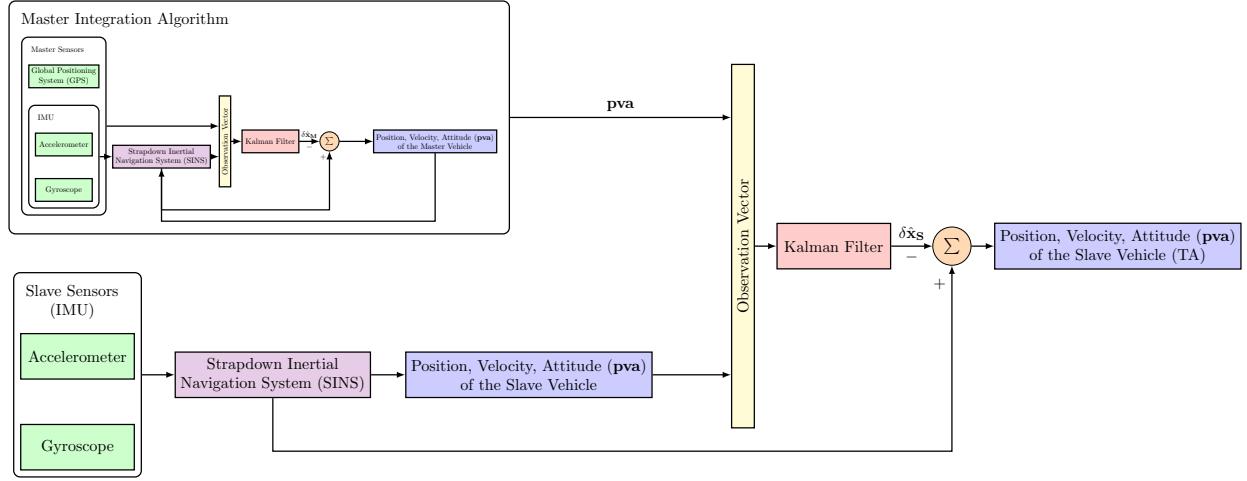


Figure 10: Structure of the in-motion transfer alignment for the slave vehicle

6.6.2. Dense Layer Functionality

According to Fig 7, two dense layers neural networks, that are fully connected, are utilized with two sub-layers. First dense layers process the SINS outputs of the master vehicle ($\mathbf{p}_M, \mathbf{v}_M, \mathbf{a}_M$) to produce the following outputs:

$$\mathbf{u}_{\text{dense}} = \sigma(\mathbf{W}_{\text{dense}_2} \cdot \sigma(\mathbf{W}_{\text{dense}_1} \cdot [\hat{\mathbf{p}}, \hat{\mathbf{v}}, \hat{\mathbf{a}}] + \mathbf{b}_{\text{dense}_1}) + \mathbf{b}_{\text{dense}_2}). \quad (53)$$

The second dense layers use flatten concatenated outputs ($\mathbf{u}_{\text{flatten}} = [\mathbf{u}_{\text{LSTM}}, \mathbf{u}_{\text{dense}}]$) which convert a multidimensional tensor into a one-dimensional tensor, to generate the desired outputs as follows:

$$\mathbf{y} = \sigma(\mathbf{w}_{\text{dense}_4} \cdot \sigma(\mathbf{w}_{\text{dense}_3} \cdot \mathbf{u}_{\text{flatten}} + \mathbf{b}_{\text{dense}_3}) + \mathbf{b}_{\text{dense}_4}). \quad (54)$$

7. Structure of the Motion Alignment

After the accurate navigation of the master vehicle, the process of the transfer alignment approach, according to Fig 10, is performed to correct the SINS errors of the slave vehicle. For this purpose, a structure of a closed loop KF is utilized based on the provided information by the master and slave navigation units. The details of the slave navigation are discussed in the following.

7.1. Process Model

To perform the transfer alignment between the USV and AUV, first, the state errors of the slave for KF's propagation phase are defined as

$$\delta \mathbf{x}_S = \begin{bmatrix} [\delta \boldsymbol{\epsilon}_S]^N & [\delta \mathbf{v}_S]^N & [\delta \mathbf{p}_S]^N & [\mathbf{b}_{as}]^S & [\mathbf{b}_{gs}]^S \end{bmatrix}^T, \quad (55)$$

where subscript **S** indicates the navigation states of the slave vehicle. Because of the closed-loop structure of the KF, the change rate of the error states is

$$\dot{\delta \mathbf{x}}_S = \mathbf{0}. \quad (56)$$

Moreover, the covariance of propagation is updated as

$$\mathbf{P}_S = \mathbf{F}_S \mathbf{P}_S \mathbf{F}_S^T + \mathbf{Q}_S, \quad (57)$$

where \mathbf{Q}_S is covariance of the process noise and \mathbf{F}_S is the linear model of the slave's SINS, computed similar to the equation (18).

7.2. Measurement Model

Here, the velocity matching between GPS/SINS/NN of the USV with ROV's SINS is performed as follows:

$$\mathbf{z}_S = [\hat{\mathbf{v}}_M]^N - [\hat{\mathbf{v}}_{S_{SINS}}]^N - \hat{\mathbf{C}}_S^N ([\tilde{\boldsymbol{\omega}}_{SI}]^S \times [\mathbf{r}_{SM}]^M) + [\boldsymbol{\omega}_{EI_S}]^N \times \mathbf{C}_S^N [\mathbf{r}_{SM}]^M, \quad (58)$$

where $[\mathbf{r}_{SM}]^M$ is the lever arm from the USV with AUV, denoted in master frame. Moreover, the output matrix of the transfer alignment is linearized as follows:

$$\mathbf{H}_S = [H_{11} \ -I_{3 \times 3} \ \mathbf{0} \ \mathbf{0} \ H_{15}], \quad (59)$$

where

$$H_{11} = -\text{skew}\left(\mathbf{C}_{S_{SINS}}^N ([\tilde{\boldsymbol{\omega}}_{BI}]^S [\mathbf{r}_{SM}]^M) - [\boldsymbol{\Omega}_{EI_S}]^N \mathbf{C}_{S_{SINS}}^N [\mathbf{r}_{SM}]^M\right), \quad (60)$$

$$H_{15} = -\mathbf{C}_{S_{SINS}}^N \times \text{skew}([\mathbf{r}_{SM}]^M). \quad (61)$$

7.3. Update the Estimated States

Finally, the SINS of AUV is corrected based on the estimation results, obtained from the transfer alignment approach as follows:

$$[\hat{\mathbf{v}}_S]^N = [\mathbf{v}_{S_{SINS}}]^N - [\delta \hat{\mathbf{v}}_S]^N, \quad (62)$$

$$\hat{\mathbf{p}}_S = \mathbf{p}_{S_{SINS}} - \delta \hat{\mathbf{p}}_S, \quad (63)$$

$$\hat{\mathbf{C}}_S^N = \mathbf{C}_{S_{SINS}}^N (\mathbf{I} - \mathbf{E}_S), \quad (64)$$

where \mathbf{E}_S is the skew-symmetric matrix of the $\delta \boldsymbol{\epsilon}_S$ vector.

8. Results and Discussion

Here, the simulation results of the proposed in-motion transfer alignment are presented. For this purpose, the Otter USV is considered as the master vehicle, equipped with the low-grade MEMS IMU and a GPS receiver to simulate the trajectory of the master vehicle. The marine system Simulator Simulink toolbox [29] is utilized. Moreover, the embedded ROV, inside the master vehicle, is considered as the slave vehicle, equipped only with a low-grade MEMS IMU. To observe the heading error and identify the roll and pitch errors from the bias of the horizontal accelerometer, the maneuver of the master vehicle is performed according to Fig. 11. The parameters of the master and slave vehicles are presented in Table 4 and Table 5, respectively.

Table 4: Parameters of the Master Vehicle

Sensor	Parameter	Value	Unit
Accelerometer	Initial Bias Uncertainty	1000	μg
	Noise PSD	0.01	$m/sec^2/\sqrt{Hz}$
	Bias Random Walk PSD	0.0001	$m/sec^3/\sqrt{Hz}$
Gyroscope	Initial Bias Uncertainty	0.1	deg/hr
	Noise PSD	0.01	deg/sec/ \sqrt{Hz}
	Bias Random Walk PSD	0.0001	deg/sec $^2/\sqrt{Hz}$

Table 5: Parameters of the Slave Vehicle

Sensor	Parameter	Value	Unit
Accelerometer	Initial Bias Uncertainty	2000	μg
	Noise PSD	0.02	$m/sec^2/\sqrt{Hz}$
	Bias Random Walk PSD	0.0002	$m/sec^3/\sqrt{Hz}$
Gyroscope	Initial Bias Uncertainty	0.2	deg/hr
	Noise PSD	0.02	deg/sec/ \sqrt{Hz}
	Bias Random Walk PSD	0.0002	deg/sec $^2/\sqrt{Hz}$

In the following, performance of proposed transfer alignment method is evaluated at two scenarios, (I) without NN and (II) with NN. The error analysis is performed at section 8.3.

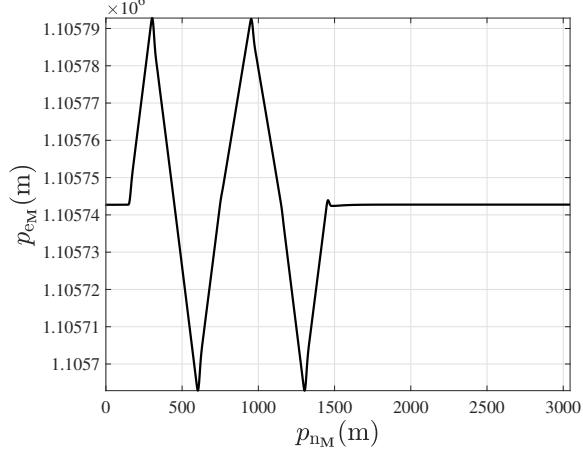


Figure 11: Maneuver of the Master Vehicle at Alignment Process

8.1. Disruption of the GPS Signal

In this section, the performance of the robust in-motion transfer alignment algorithm is investigated, when the GPS information outages. For this purpose, the interruption of the GPS signal takes 755 seconds (between 1045 & 1800 seconds).

8.1.1. Performance of the Master Navigation Unit

The Fig 11 shows the trajectory of the boat (with and without GPS). Moreover, the results of the master navigation unit performance is evaluated in Fig 12, when GPS outages. It can be observed that the state errors increase significantly, when the GPS signal is interrupted during the specific interval (1045 to 1800 seconds). Consequently, the performance of the master navigation algorithm is disrupted.

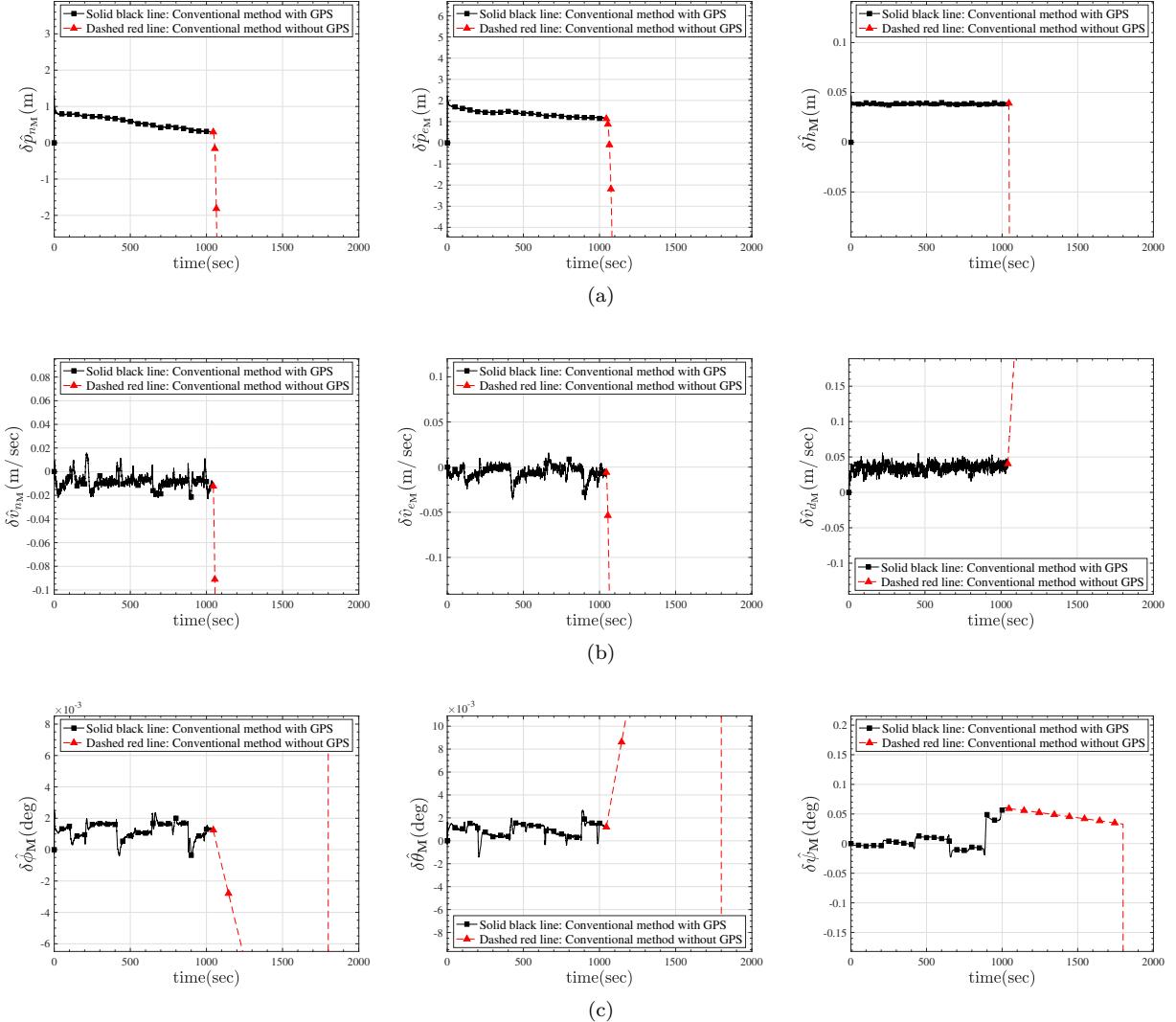


Figure 12: Performance comparison of the Master Navigation Algorithm: Conventional method with GPS (Solid line) vs. Conventional method without GPS (Dashed red line): (a) Position error (m), (b) Velocity error (m/sec), (c) Attitude error (deg).

8.1.2. Performance of Slave Navigation Unit

Here, transfer alignment algorithm performance is evaluated in Fig 13 (a) to 13 (c), when the GPS of the master unit outages. This figure shows the state errors including the errors of the position ($\delta\hat{\mathbf{p}}_S$), velocity ($\delta\hat{\mathbf{v}}_S$), and attitude ($\delta\hat{\mathbf{a}}_S$) of the slave vehicle. When the GPS signal is available the proposed approach can estimate the true states of the slave vehicle properly. However, the transfer alignment algorithm does not work correctly, when the GPS signals are disrupted and not available in some times.

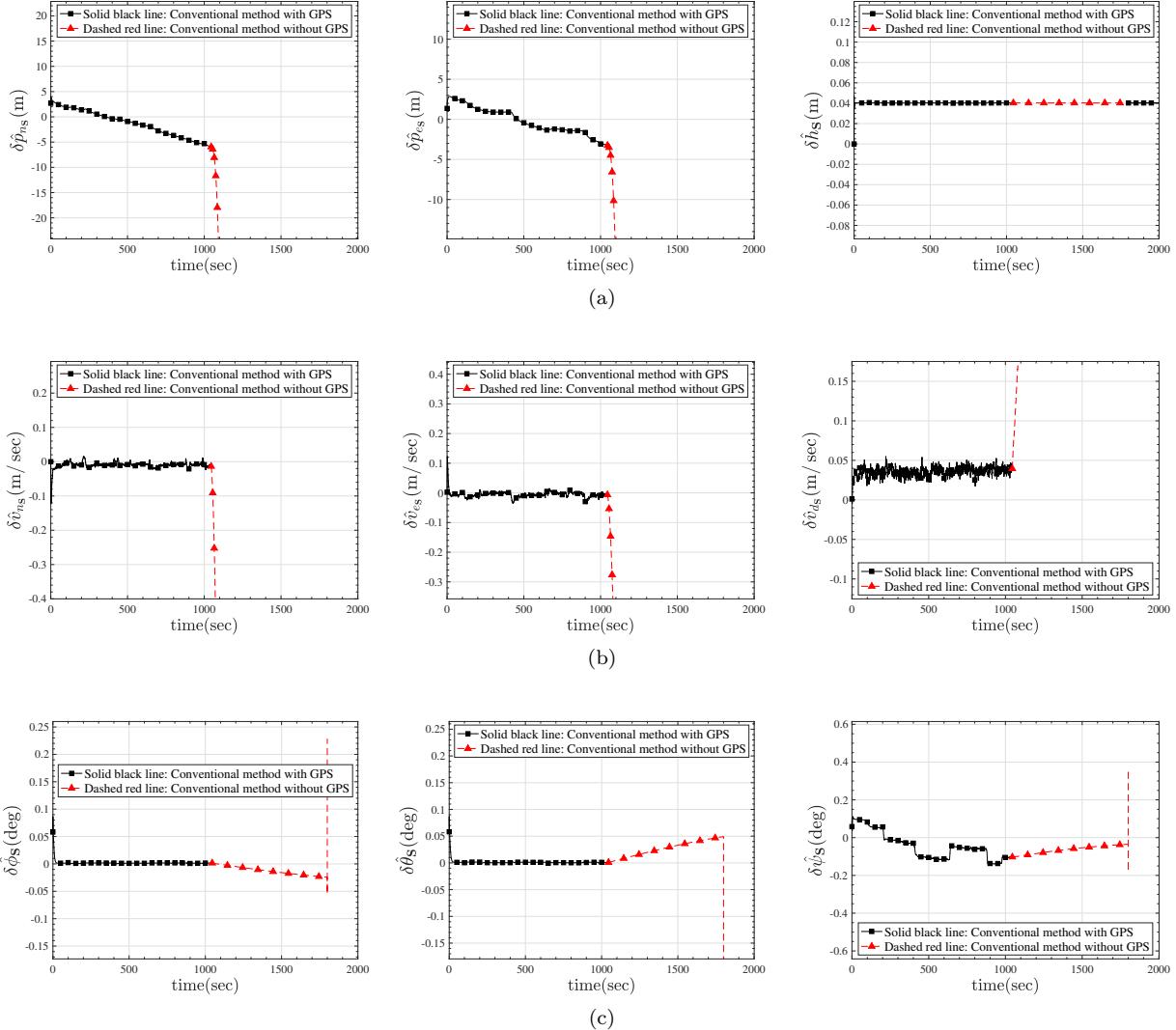


Figure 13: Performance comparison of the Slave Navigation Algorithm: Conventional method with GPS (Solid line) vs. Conventional method without GPS (Dashed red line): (a) Position error (m), (b) Velocity error (m/sec), (c) Attitude error (deg).

8.2. Performance of Proposed NN

Here, the results of the training and prediction phases of the proposed model are presented, when GPS signal of the master navigation unit is disrupted. For this purpose, first, the training of the NN model is discussed. Then, the performance of the master and slave vehicles are investigated.

8.2.1. Training of the Neural Network Model

The training of the proposed NN model is performed using the sequential data of the master's IMU and SINS outputs. For this purpose, the Adam optimizer [30] is employed to increase the efficiency and robustness in training phase of deep model. Then, the training process was conducted on the Google Colab platform, utilizing the computational power of an NVIDIA T4 GPU. The specifications of the training phase are denoted in Table 6. Moreover, the training and validation losses are shown in Fig 14. It is observed that the trained model is effectively learned from the data, as the loss values consistently decrease throughout the training process.

Table 6: Characteristics of the Training Phase of the Proposed NN

Aspect	Specification
Training Algorithm	Adam Optimizer
Learning Rate	0.001
Number of Epochs	100
Hardware	NVIDIA T4 GPU
Total Training Time	1 hour, 2 minutes, 6 seconds

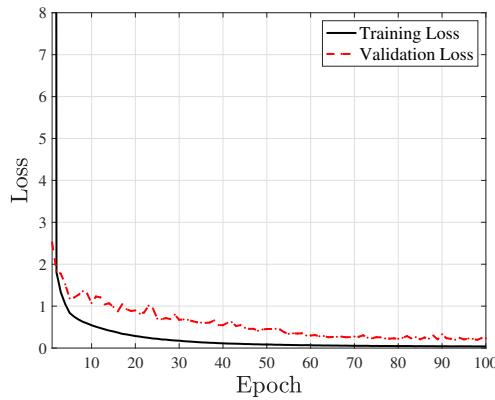


Figure 14: Training and Validation Loss of the Proposed NN Model

8.2.2. Performance of Master Navigation Unit

Here, the results of the master navigation unit is evaluated in Fig 15. It can be observed that the state errors do not increase significantly using the proposed NN, when the GPS signal interrupted during the specific interval (1045 & 1800 seconds).

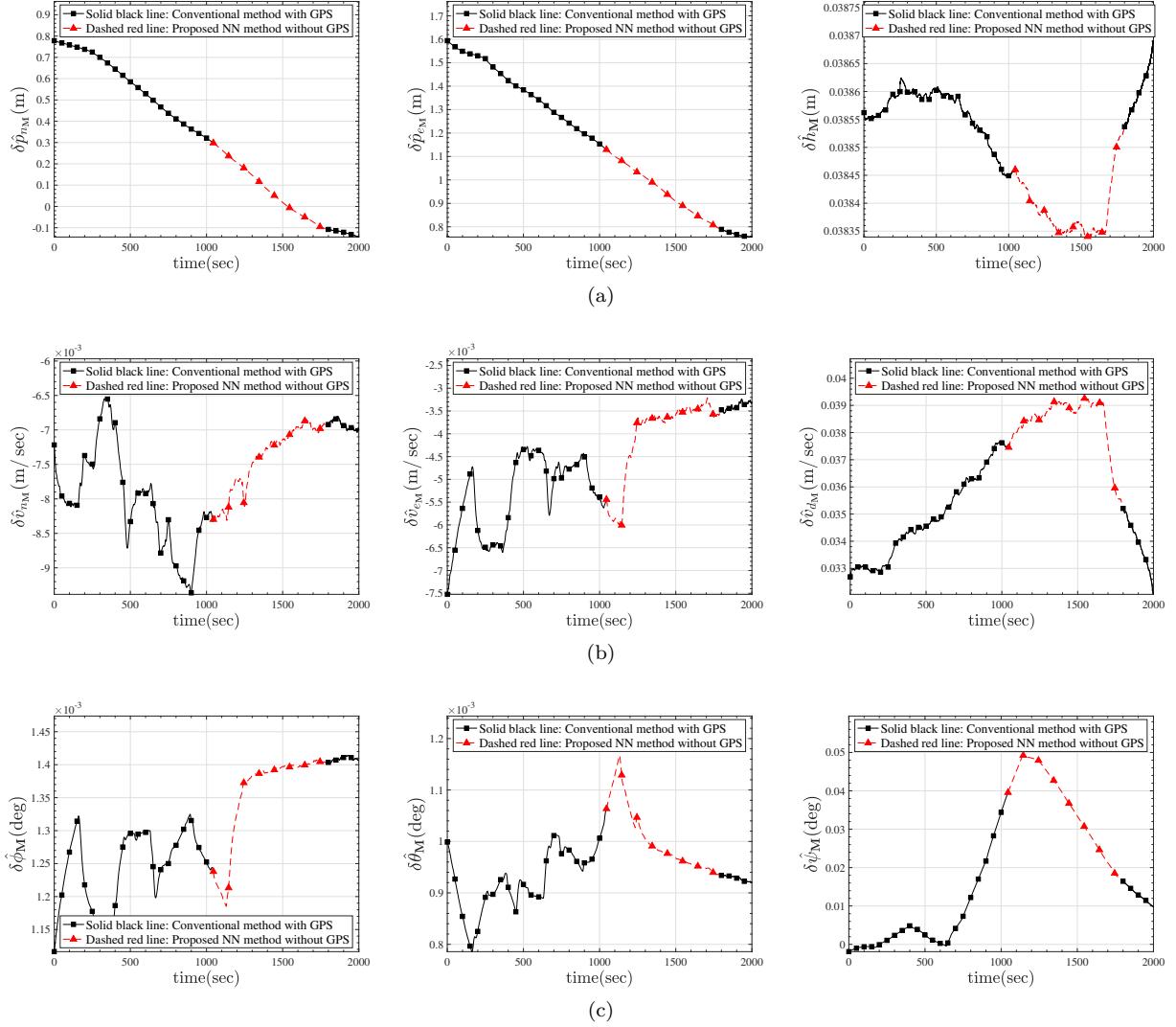


Figure 15: Performance comparison of the Master Navigation Algorithm: Conventional GPS-based method (Solid line) vs. Proposed NN method when GPS outages (Dashed red line): (a) Position error (m), (b) Velocity error (m/sec), (c) Attitude error (deg).

8.2.3. Performance of Slave Navigation Unit

Here, transfer alignment algorithm performance is evaluated in Fig 16, when the GPS signal of the master unit outages. This figure shows the state errors including the errors of the position ($\delta \hat{p}_S$), velocity ($\delta \hat{v}_S$), and attitude ($\delta \hat{\alpha}_S$) of the slave navigation unit (AUV). It observed that the proposed transfer alignment algorithm works correctly in the presence of proposed method, when the GPS signals are disrupted.

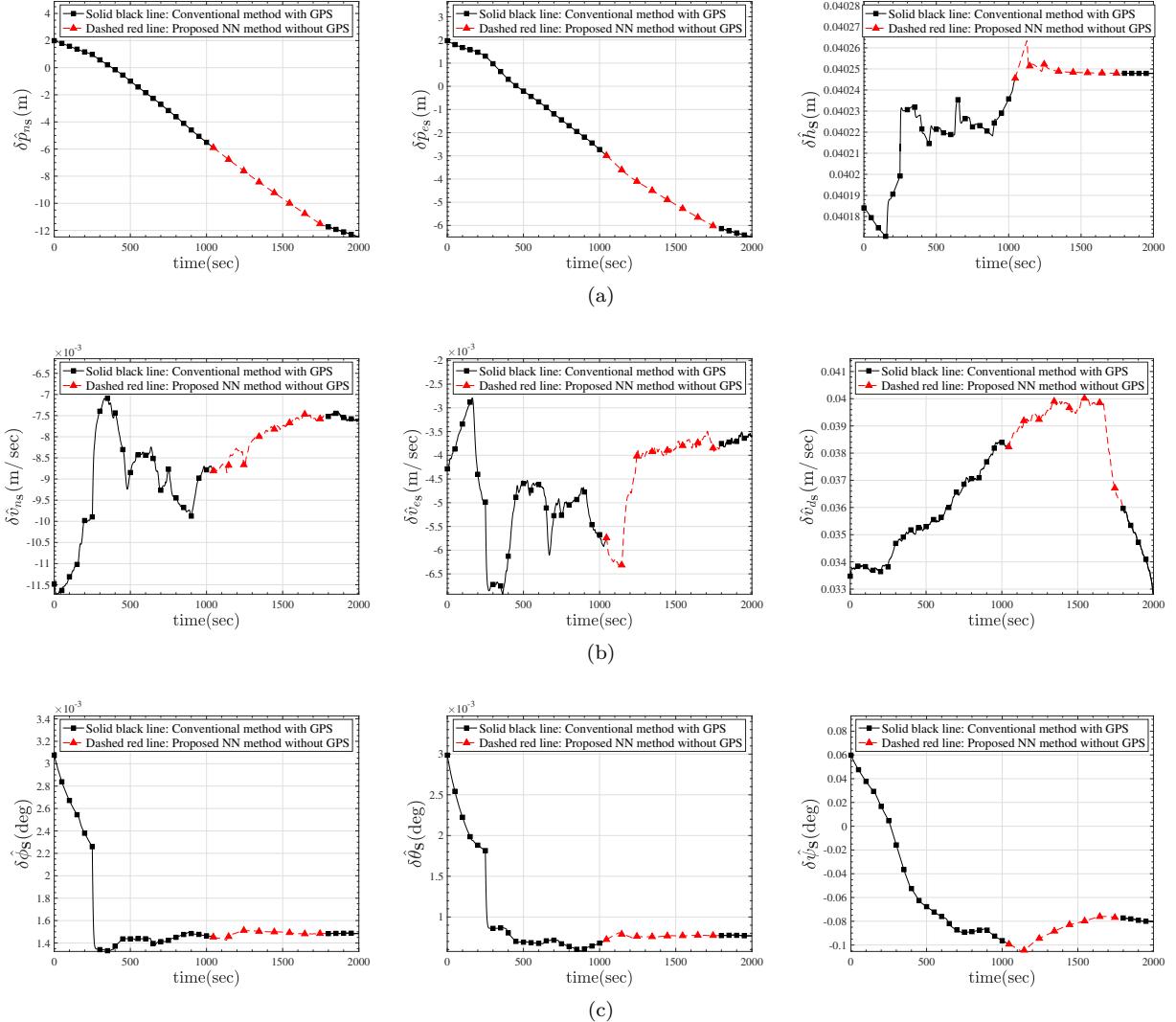


Figure 16: Performance comparison of the Slave Navigation Algorithm: Conventional GPS-based method (Solid line) vs. Proposed NN method when GPS outages (Dashed red line): (a) Position error (m), (b) Velocity error (m/sec), (c) Attitude error (deg).

8.3. Error Analysis

A comprehensive error analysis was conducted by decomposing the total error into its constituent components to provide a deeper understanding of the method's limitations and potential areas for improvement. The primary sources of error in the transfer alignment process were identified and their relative contributions were quantified.

8.3.1. Error Decomposition Methodology

The total error in the navigation solution can be attributed to three main sources:

- 1. Sensor Errors:** Errors originating from the inertial measurement units (IMUs), including bias instability, scale factor errors, and random noise.
- 2. Model Errors:** Errors resulting from imperfections in the neural network model's ability to predict navigation states during GPS outages.

3. Alignment Errors: Errors stemming from the initial misalignment between the master and slave navigation systems and the transfer alignment process itself.

A series of controlled simulations was employed where each error source could be isolated and measured independently. The root mean square error (RMSE) values for each error component across position, velocity, and attitude states during GPS outages are presented in Table 7.

Table 7: Decomposition of Navigation Errors During GPS Outages

Error Type	Position (m)	Velocity (m/s)	Attitude (deg)	Contribution (%)
Sensor Errors	3.42	0.18	0.32	42.5
Model Errors	2.87	0.15	0.28	35.6
Alignment Errors	1.76	0.09	0.15	21.9
Total Error	8.05	0.42	0.75	100.0

8.3.2. Sensor Error Analysis

Sensor errors were found to constitute the largest portion (42.5%) of the total error budget. The accelerometer bias instability was identified as the dominant factor affecting position accuracy, while gyroscope random walk significantly impacted attitude estimation. It was determined that a 10% reduction in accelerometer bias instability could potentially reduce position errors by approximately 15%.

8.3.3. Model Error Analysis

Model errors were found to account for 35.6% of the total error budget. These errors arise from the neural network's prediction capabilities during GPS outages. Different aspects of the model architecture and training process were analyzed for their effects on prediction accuracy.

The LSTM layers were found to contribute most significantly to reducing prediction errors during extended GPS outages. The model's prediction error was observed to grow approximately quadratically with the duration of the GPS outage, with error growth rates of 0.42 m/min for position, 0.03 m/s/min for velocity, and 0.05 deg/min for attitude.

8.3.4. Alignment Error Analysis

Alignment errors were determined to make up 21.9% of the total error budget. These errors were primarily influenced by the initial alignment accuracy and the dynamics of the vehicles during the transfer alignment process.

Heading alignment errors were found to have the most significant impact on the overall navigation solution, particularly during maneuvers. It was determined that improving the initial heading alignment by 0.1 degrees could reduce the overall position error by approximately 8% during GPS outages.

9. Conclusion

In this study, the accuracy of the in-motion transfer alignment method was increased using the artificial intelligence algorithm, called RTA-NN, when the GPS signals were blocked. For this purpose, first, a deep neural network was trained based on the sequential data of master's IMU and SINS outputs. Then, the vehicle model was made to predict the true data of the master vehicle in the absence of the GPS. Finally, a 6-DoF simulation of the ROV, that was launched from the USV, was performed to evaluate the performance of the proposed transfer alignment, when the GPS signals were disrupted. The simulation results demonstrated that the proposed transfer alignment method based on the Artificial Intelligence was able to provide the better accuracy in the absence of the GPS signals. In future work, the proposed structure can be enhanced by integrating Reinforcement Learning (RL) to further reduce estimation errors. Additionally, the interaction between the proposed structure and Kalman Filter can be optimized, leveraging the strengths of both methods to improve overall system performance and robustness.

10. Acknowledgment

The authors wish to express thanks to the office of vice chancellor of research of Sharif University of Technology for the financial support under grant number G4030414.

References

- [1] P. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems, Second Edition*, ser. GNSS/GPS. Artech House, 2013. [Online]. Available: <https://books.google.com/books?id=t94fAgAAQBAJ>
- [2] C. Jekeli, *Inertial Navigation Systems with Geodetic Applications*. Berlin, Boston: De Gruyter, 2001. [Online]. Available: <https://doi.org/10.1515/9783110800234>
- [3] C. Glennie, K. Schwarz, A. Bruton, R. Forsberg, A. Olesen, and K. Keller, “A comparison of stable platform and strapdown airborne gravity,” *Journal of Geodesy*, vol. 74, pp. 383–389, 07 2000.
- [4] D. Titterton, J. Weston, I. of Electrical Engineers, A. I. of Aeronautics, and Astronautics, *Strapdown Inertial Navigation Technology*, ser. IEE Radar Series. Institution of Engineering and Technology, 2004. [Online]. Available: <http://www.books24x7.com/marc.asp?bookid=10863>
- [5] K. Britting, *Inertial Navigation Systems Analysis*. Wiley-Interscience, 1971. [Online]. Available: <https://books.google.com/books?id=Mn9TAAAAMAAJ>
- [6] E. F. Lyon, “Inertial navigation error analysis,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 3, pp. 123–135, 2008.
- [7] A. B. Chatfield, *Fundamentals of High Accuracy Inertial Navigation*. Reston, VA: American Institute of Aeronautics and Astronautics, 1997.
- [8] Y. F. Jiang, Y. P. Lin et al., “Error estimation of ins ground alignment through observability analysis,” *IEEE Transactions on Aerospace and Electronic systems*, vol. 28, no. 1, pp. 92–97, 1992.
- [9] A. Gelb, *Applied Optimal Estimation*. Cambridge, MA: MIT Press, 1974.
- [10] M. S. Grewal, L. R. Weill, and A. P. Andrews, *Global Positioning Systems, Inertial Navigation, and Integration*, 2nd ed. Hoboken, NJ: Wiley-Interscience, 2007.
- [11] D. H. Titterton and J. L. Weston, “Inertial navigation systems,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 30, no. 4, pp. 950–964, 2004.
- [12] R. G. Brown and P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering with Matlab Exercises*, 4th ed. Hoboken, NJ: Wiley, 2012.
- [13] E. Bekir, *Introduction to Modern Navigation Systems*, ser. G - Reference, Information and Interdisciplinary Subjects Series. World Scientific, 2007. [Online]. Available: <https://books.google.de/books?id=RtUZAQAAIAAJ>
- [14] Y. Yüksel, “Design and analysis of transfer alignment algorithms,” Master’s thesis, Middle East Technical University, 2005.
- [15] A. Lu, Y. Wu, and Y. Zhang, “High-precision transfer alignment for distributed inertial network based on estimation fusion,” in *2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, 2019, pp. 1726–1731.
- [16] H. Zhang, H. Xiong, S. Hao, G. Yang, M. Wang, and Q. Chen, “A novel multidimensional hybrid position compensation method for ins/gps integrated navigation systems during gps outages,” *IEEE Sensors Journal*, vol. 24, no. 1, pp. 962–974, 2024.
- [17] S. Lu, Y. Gong, H. Luo, F. Zhao, Z. Li, and J. Jiang, “Heterogeneous multi-task learning for multiple pseudo-measurement estimation to bridge gps outages,” *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–16, 2021.
- [18] J. Cheng, Y. Gao, and J. Wu, “An adaptive integrated positioning method for urban vehicles based on multitask heterogeneous deep learning during gnss outages,” *IEEE Sensors Journal*, vol. 23, no. 18, pp. 22080–22092, 2023.
- [19] J. Liu and G. Guo, “Vehicle localization during gps outages with extended kalman filter and deep learning,” *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–10, 2021.
- [20] Y. Liu, Q. Luo, and Y. Zhou, “Deep learning-enabled fusion to bridge gps outages for ins/gps integrated navigation,” *IEEE Sensors Journal*, vol. 22, no. 9, pp. 8974–8985, 2022.
- [21] X. Wei, J. Li, K. Feng, D. Zhang, P. Li, L. Zhao, and Y. Jiao, “A mixed optimization method based on adaptive kalman filter and wavelet neural network for ins/gps during gps outages,” *IEEE Access*, vol. 9, pp. 47875–47886, 2021.
- [22] A. V. Kanhere, S. Gupta, A. Shetty, and G. Gao, “Improving gnss positioning using neural-network-based corrections,” *NAVIGATION: Journal of the Institute of Navigation*, vol. 69, no. 4, 2022. [Online]. Available: <https://navi.ion.org/content/69/4/navi.548>
- [23] Q. Xu, X. Li, and C.-Y. Chan, “Enhancing localization accuracy of mems-ins/gps/in-vehicle sensors integration during gps outages,” *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no. 8, pp. 1966–1978, 2018.
- [24] D. Li, X. Jia, and J. Zhao, “A novel hybrid fusion algorithm for low-cost gps/ins integrated navigation system during gps outages,” *IEEE Access*, vol. 8, pp. 53984–53996, 2020.
- [25] H. Tao, J. Zheng, J. Wei, W. Paszke, E. Rogers, and V. Stojanovic, “Repetitive process based indirect-type iterative learning control for batch processes with model uncertainty and input delay,” *Journal of Process Control*, vol. 132, p. 103112, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0959152423001993>
- [26] Z. Zhang, K. Zhang, X. Xie, and V. Stojanovic, “Adp-based prescribed-time control for nonlinear time-varying delay systems with uncertain parameters,” *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 3086–3096, 2025.
- [27] D. Zheng, X. Song, S. Shuai, and V. Stojanovic, “Quantized control for interconnected pde systems via mobile measurement and control strategies,” *Journal of the Franklin Institute*, vol. 361, p. 107070, 07 2024.

- [28] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, pp. 1735–80, 12 1997.
- [29] T. Perez and T. I. Fossen, “A Matlab Toolbox for Parametric Identification of Radiation-Force Models of Ships and Offshore Structures,” *Modeling, Identification and Control*, vol. 30, no. 1, pp. 1–15, 2009.
- [30] D. Kingma, “Adam: a method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

Appendix A. Car Data Validation

To validate the generalization capability of the proposed method, additional testing was performed using car data. The results of applying the transfer alignment algorithm with and without neural network assistance during GPS outages are shown in Fig. A.17 and Fig. A.18, respectively.

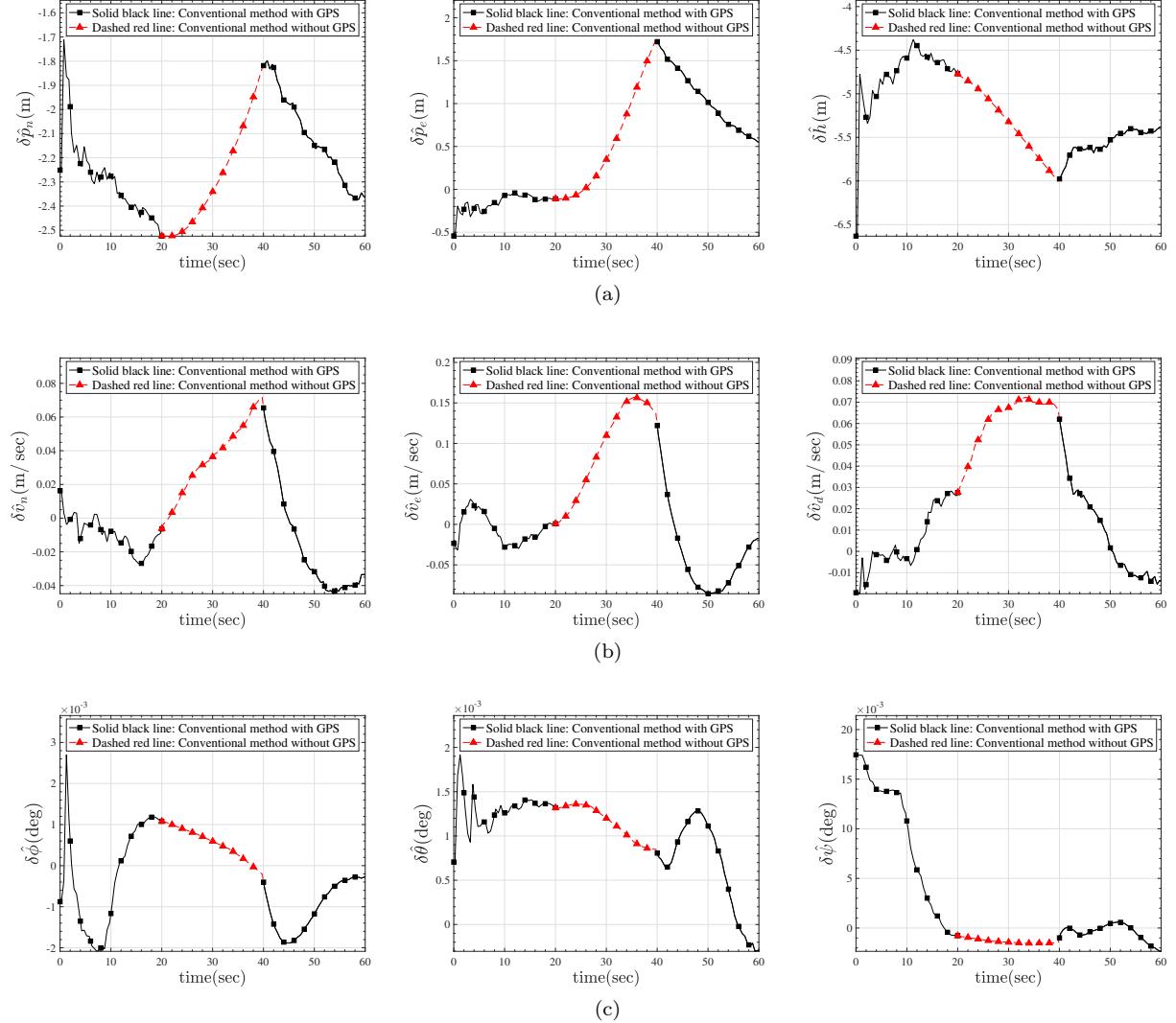


Figure A.17: Performance comparison of the Car Navigation Algorithm: Conventional method with GPS (Solid line) vs. Conventional method without GPS (Dashed red line): (a) Position error (m), (b) Velocity error (m/sec), (c) Attitude error (deg).

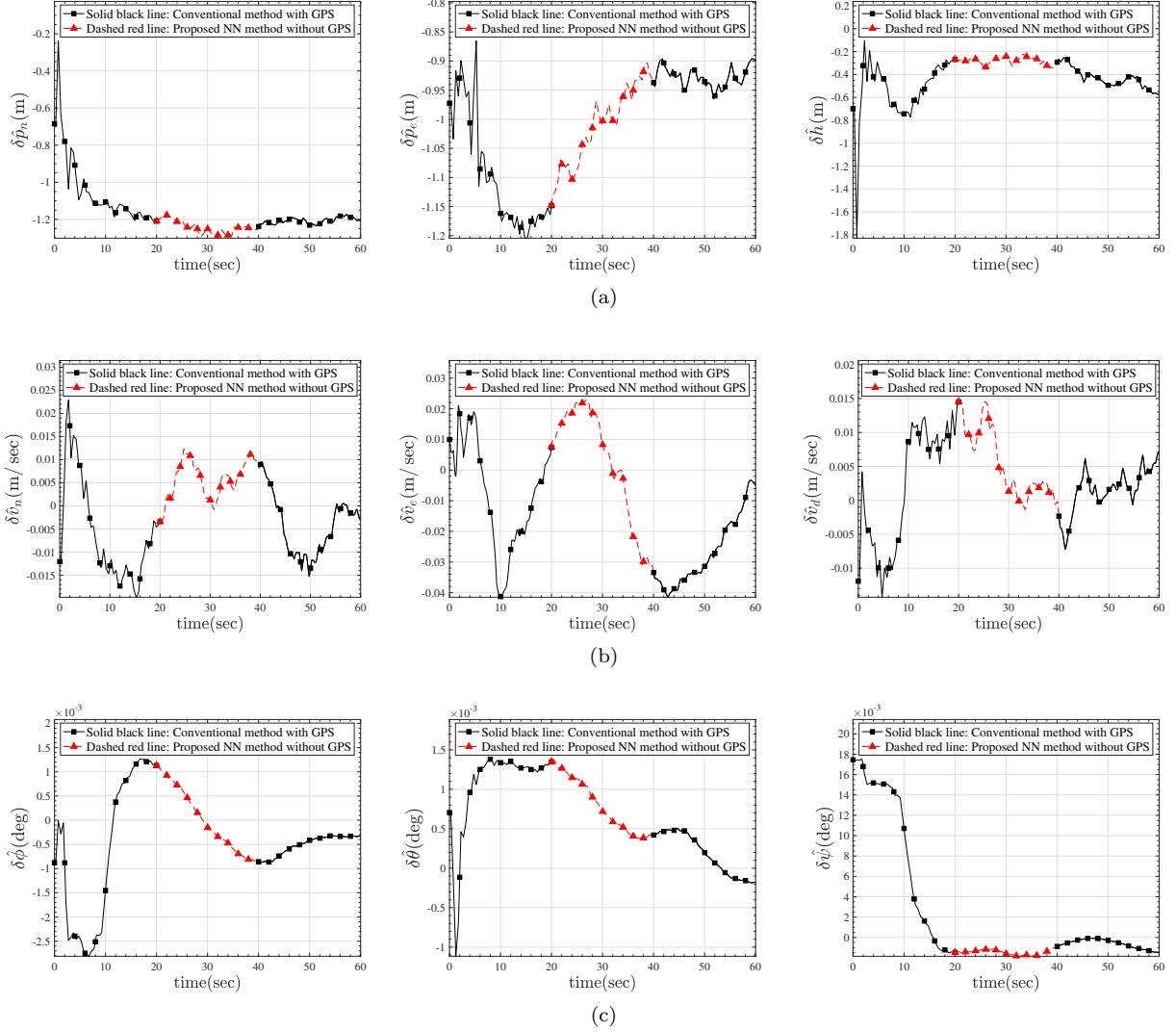


Figure A.18: Performance comparison of the Car Navigation Algorithm: Conventional GPS-based method (Solid line) vs. Proposed NN method when GPS outages (Dashed red line): (a) Position error (m), (b) Velocity error (m/sec), (c) Attitude error (deg).