

RESEARCH

Open Access

ANN-assisted robust GPS/INS information fusion to bridge GPS outage



Mehdi Aslinezhad, Alireza Malekijavan* and Pouya Abbasi

* Correspondence: amalekijavan1@gmail.com; amalekijavan@yahoo.com

Department of Electrical Engineering, Shahid Sattari Aeronautical University of Science and Technology, Tehran 1384663113, Iran

Abstract

Inertial navigation is an edge computing-based method for determining the position and orientation of a moving vehicle that operates according to Newton's laws of motion on which all the computations are performed at the edge level without need to other far resources. One of the most crucial struggles in Global Positioning System (GPS) and Inertial Navigation System (INS) fusion algorithms is that the accuracy of the algorithm is reduced during GPS interruptions. In this paper, a low-cost method for GPS/INS fusion and error compensation of the GPS/INS fusion algorithm during GPS interruption is proposed. To further enhance the reliability and performance of the GPS/INS fusion algorithm, a Robust Kalman Filter (RKF) is used to compensate the influence of gross error from INS observations. When GPS data is interrupted, Kalman filter observations will not be updated, and the accuracy of the position will continuously decrease over time. To bridge GPS data interruption, an artificial neural network-based fusion method is proposed to provide missing position information. A well-trained neural network is used to predict and compensate the interrupted position signal error. Finally, to evaluate the effectiveness of the proposed method, an outdoor test using a custom-designed hardware, GPS, and INS sensors is employed. The results indicate that the accuracy of the positioning has improved by 67% in each axis during an interruption. The proposed algorithm can enhance the accuracy of the GPS/INS integrated system in the required navigation performance.

Keywords: Inertial navigation, Data fusion, Robust Kalman Filter, NAR Neural Network

1 Introduction

In recent years, accurate vehicle positioning through edge computing concepts has significantly affected many applications of the transportation industry, such as intelligent driver assistance technology, routing, and auto-drive systems. Generally, the overall performance of low-cost inertial sensors degrades over a certain amount of time, and by using a high-grade sensor, the production cost of the vehicle will be significantly affected. Consequently, using low-cost sensors with an accurate positioning algorithm is the best solution to maintain accuracy and productivity [1]. Global Positioning System (GPS) and other similar systems such as Global Navigation Satellite System (GNSS), Global Navigation Satellite System (GLONASS), and Beidou are now widely accessible around the world and stand as

standard navigation systems in transportation applications. Inertial Navigation System (INS) consists of 3-axis sensors measuring linear acceleration and angular velocity in order to calculate position, velocity, and Euler angles. With enhancements in processors, integration of two or more low-cost sensors with a more complex but accurate fusion algorithm can have a significant influence on the extension of flight, shipping, and car industries [2, 3]. For GPS/INS integrated system, the navigation solution is traditionally achieved by using Kalman Filter (KF). However, the major inadequacy related to KF is that the system states are usually nonlinear. Extended Kalman Filter (EKF) is the nonlinear version of the KF, which linearizes about an estimate of the current mean and covariance. Since there are two integrators in calculating position from acceleration data, a second-order EKF is the most suitable approach in data fusion algorithms [4, 5]. In order to improve the accuracy of the position and compensate the INS error in the long-term absence of GPS data, the neural network is an excellent approach to reduce the cumulative error of INS [6]. In this paper, the ultimate objective is to train an artificial neural network from GPS data to learn the latest available position signals and then use this trained network to predict the time series in the absence of the GPS signal. Considering extensive capability of neural networks to solve nonlinear problems and overcome the drawbacks of KF and different types of Artificial neural networks (ANN) such as Multilayer Perceptron (MLP), Radial Basis Function (RBF), and Adaptive Neuro-Fuzzy Inference System (ANFIS) approaches are used in data fusion applications [7, 8]. Also, a variety of training methods are used in this field, but the most common methods are based on error Back-Propagation (BP) methods [9, 10].

Recently, there has been a significant focus on GPS and INS sensor data fusion algorithms, especially in absolute position estimation during a failure in communication with sensors or an outage in GPS. The primary focus in most of the articles is to maintain position accuracy improvement. For example, in [11], Design of an INS/GPS algorithm using parallel Kalman filters is investigated. In [12], an Adaptive IIR/FIR fusion filter approach is implemented. In [6, 13], a GPS/INS data fusion is applied using Unscented Kalman Filter (UKF) and RKF. In [14], the Fusion algorithm using UKF and BP neural networks is investigated. Also, in [15, 16], hybrid integration methods using different types of neural networks are investigated.

Kalman Filter is the most remarkable real-time optimal estimator, which has found a vast field of application [17]. KF has become an indispensable data fusion approach for GPS/INS integrated navigation. Compared to other alternative methods of GPS/INS fusion like Particle Filter [18] or IIR/FIR Filters [19], Kalman Filter proposes a better adaptivity and flexibility of implementation in low-cost systems along with strong optimality and accuracy of estimation. However, the standard KF algorithm can hardly deal with nonlinearity in model and robust issues such as uncertainty. In this research, EKF has been utilized to solve the issue of dealing with nonlinearity. Recently, researchers have proposed and suggested robust filter algorithm [20–22] which stands for high stability and flexibility and better capability of controlling the negative effect of both the dynamic model uncertainty and the long-term measurements errors. The performance of integrated positioning will be degraded due to the gross error from observations in challenging

environments. A robust Kalman filter could be employed to reduce the effect of gross error in observation.

In this work, EKF, along with a robust method, has been proposed to ensure dealing with the mentioned issues, and, improve the accuracy of the GPS/INS system. The robust algorithm evaluates the innovation vector with chi-square test and modifies the faulty innovation vector. Kalman filter robustness is the ability of the filter to deal with adverse environments and input conditions.

The primary purpose of this research is to design a high precision fusion algorithm along with a custom standalone hardware for civil and small-industry applications. Since increasing cost would affect negatively on the final product, in this research, low-cost MEMS sensors are used. Although expensive sensors provide high accuracy data, MEMS sensors provide a lower but acceptable range of accuracy along with a much lower price. Furthermore, all other parts of the hardware, i.e., processor, are chosen carefully from a low to medium range of price to ensure production capability of the system. Compared to other works in the same field as [11, 15, 16, 23], two primary purposes in this work have been improved. First, the design of the hardware from both aspects of cost-effectiveness and multi-platform capability of the product (i.e., robots and autonomous drones) has been considered. Second, the accuracy improvement of the algorithm in most probable scenarios, standard scenario (using RKF), and GPS outage scenario (using Neural Network), has been investigated.

Choosing between ANN or machine learning methods in GPS/INS fusion algorithms depends on the type of the vehicle and main scenario, in which the algorithm is being used. Neural Networks implicitly integrates temporal data and recognizes temporal data patterns with arbitrary time intervals. Therefore, Neural Networks is suitable for applications where input-output data has temporal patterns. This has a beneficial effect on prediction accuracy [9]. The most crucial difference between ANN and machine learning methods is its performance as the scale of data increases. ANN algorithms need a large amount of data in order to be capable of accurate prediction, whereas machine learning methods perform well with less amount of data. On the other hand, when there is a large amount of data, the performance of ANN methods increases over time. In this work, GPS and INS sensors produce a tremendous amount of data over time, as the INS sensor works at 100 Hz. Thus, after a certain amount of time, when ANN is trained well, its performance would overcome the machine learning algorithm.

In the second section, GPS, INS, and the proposed scheme and formulations of the data fusion are presented. Results of an outdoor test with designed hardware are presented in Section 3. Finally, Section 4 makes the concluding remarks.

2 Materials and methods

In this section, the formulation of the work is briefly described. Primarily, position calculation of GPS and also output variables of INS are described. Also, GPS/INS fusion algorithm using EKF as a primary step for integrated navigation is investigated. Then, a robust methodology to increase the performance of EKF is proposed. Finally, integration schemes based on loosely-coupled integration method along with the proposed algorithm are demonstrated.

2.1 Formulation

GPS is a collection of 32 satellites orbiting the earth’s surface and is continuously providing position data to receivers on earth [5]. The position is determined through measuring a range between receiver and a minimum of four satellite positions by measuring the bias error between receiver and GPS clock. This range is called pseudo-range and is determined as per Eqs. (1)–(3), as below.

$$\rho_i = r_i + b \tag{1}$$

$$r_i = \sqrt{(x-x_j)^2 + (y-y_j)^2 + (z-z_j)^2} \tag{2}$$

$$b = ct_j \tag{3}$$

where (x_j, y_j, z_j) denotes the satellite position in three-dimensional space, (x, y, z) is the receiver position, ρ_i is the pseudo-range in meters, b is the clock bias of the receiver, c is the speed of light, and t_j is the satellite clock [24]. The final coordinate frame in this paper is Earth Centered Earth Fixed (ECEF), which is denoted by superscript e . Other coordinate frames used in this paper are Inertia and Body frames denoted by superscripts i and b , respectively. The z-axis in ECEF frame is towards the rotation axis of earth from the center to the North Pole, and the x-axis is towards the equinox (in Greenwich) (Fig. 1).

Accelerometer vector in and gyroscope angular velocity in body frame are denoted by $[a_x^b \ a_y^b \ a_z^b]^T$ and $[\omega_x^b \ \omega_y^b \ \omega_z^b]^T$, respectively. Also, the transformation matrix for converting body-frame angular rates to Euler angular rates is given by Eq. (4), as below.

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi \tan\theta & \cos\phi \tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi \sec\theta & \cos\phi \sec\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{4}$$

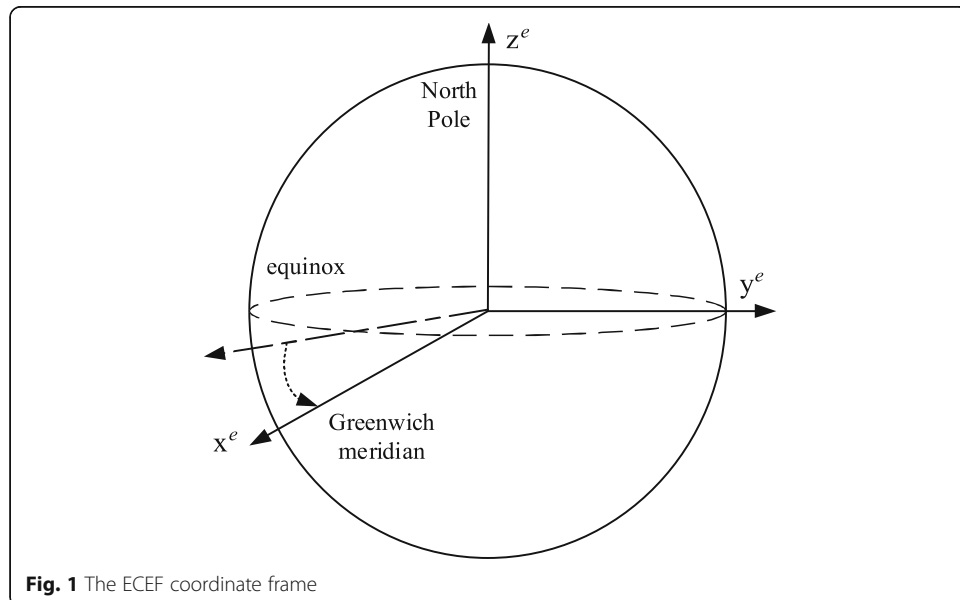


Fig. 1 The ECEF coordinate frame

where $[p \ q \ r]^T$ is the angular velocity vector and by integrating $[\dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T$, Euler angles are obtained. C_b^e is the transformation matrix from body frame to earth frame defined as per Eq. (5). Also, state space representation of the vehicle is defined as Eq. (6), as below.

$$C_b^e = \begin{bmatrix} \cos\theta \cos\psi & \sin\phi \sin\theta \cos\psi + \cos\phi \sin\psi & -\cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi \\ -\cos\theta \sin\psi & -\sin\phi \sin\theta \sin\psi + \cos\phi \cos\psi & \cos\phi \sin\theta \sin\psi + \sin\phi \cos\psi \\ \sin\theta & -\sin\phi \cos\theta & \cos\phi \cos\theta \end{bmatrix} \quad (5)$$

$$\{ \delta\dot{x}(t) = F(t)\delta x(t) + G(t)u(t)y(t) = H \delta x(t) + w(t) \quad (6)$$

where $F(t)$, $G(t)$, and $H(t)$ are 15×15 , 15×6 , and 15×3 matrices, and u_{acc} and u_{gyro} are the accelerometer and gyroscope noises and are described in Eqs. (7)–(9), as below.

$$F(t) = \begin{bmatrix} 0 & I & 0 & 0 & 0 \\ 0 & -2\Omega_{ie}^e & \tilde{a}^e & 0 & C_b^e \\ 0 & 0 & -\Omega_{ie}^e & -C_b^e & 0 \\ 0 & 0 & 0 & -I & 0 \\ 0 & 0 & 0 & 0 & -I \end{bmatrix}, G(t) = \begin{bmatrix} 0 & 0 \\ 0 & C_b^e \\ -C_b^e & 0 \\ I & 0 \\ 0 & I \end{bmatrix} \quad (7)$$

$$H(t) = [I_{3 \times 3} \quad I_{3 \times 3} \quad 0_{3 \times 3} \quad 0_{3 \times 3} \quad 0_{3 \times 3}] \quad (8)$$

$$u(t) = \begin{bmatrix} u_{acc} \\ u_{gyro} \end{bmatrix} \quad (9)$$

where \tilde{a}^e and Ω_{ie}^e , respectively represent the skew-symmetric forms of the acceleration and angular velocity vectors.

H sets for different conditions as follows. Due to the type of GPS receivers, velocity data may also be received. The first array of the matrix H represents observations of Position, and the second array is observations of velocity. When GPS is jammed, or the signal is interrupted, no data is available, and therefore, this matrix becomes zero, and another approach should be utilized for estimation. The state vector is defined as Eq. (10), as below.

$$\delta x(t) = [\delta p^e \quad \delta v^e \quad \zeta \quad \delta \omega_{ib}^b \quad \delta a^b]^T \quad (10)$$

where δp^e , δv^e , and ζ are position n , velocity, and attitude errors, respectively. Also, $\delta \omega_{ib}^b$ and δa^b are angular velocity and acceleration error. The error values are determined by integrating the differential equations in (6) using appropriate initial conditions. The error between output position and velocity of GPS/INS algorithm and the reference position and velocity, e , is defined as Eq. (11), as below.

$$e(t) = [e_p \quad e_v]^T \quad (11)$$

The state space model is nonlinear because matrices $F(t)$ and $G(t)$ contain nonlinear components. As a result, a linearized model needs to apply KF since the linear dynamics of the sensors are unknown [12]. To implement KF in a microprocessor-based system, a discrete form of the state space model is required as described in Eq. (12).

$$\delta x_{k+1} = \Phi_k \delta x_k + G_{dk} u_k \quad (12)$$

where $u(k)$ is the process noise. The relations between continuous and discrete forms of converted matrices are defined in Eqs. (11) and (12), as follows.

$$\Phi_k \approx e^{FT_s} = I + F(kT_s)T_s \quad (13)$$

$$G_{dk} = F^{-1}(kT_s) \left(e^{F(kT_s)T_s} - I \right) G(kT_s) \approx G(kT_s)T_s \quad (14)$$

where Φ_k is the state transition matrix. T_s is the sampling time. The measurement equation would be defined as in (15), as below.

$$\delta y_k = H_k \delta x_k + w_k \quad (15)$$

where δy_k is the numerical difference between GPS and INS vectors, and w_k is the measurement noise.

2.2 Kalman Filter

In this paper, an EKF is implemented. At the first step of signal filtering, initial values for estimation of the state variables X_0^- and the covariance matrix of state estimation error P_0^- are selected. Next, the value of the state variables and the covariance matrix of state estimation error are predicted for the current moment in Eqs. (16) and (17) as:

$$\delta \hat{x}_{k+1}^- = \Phi_k \delta \hat{x}_k^- \quad (16)$$

$$P_{k+1}^- = \Phi_k P_k^- \Phi_k^T + G_d Q_{d,k} G_d^T \quad (17)$$

To improve the priori estimation, first, the Kalman gain is calculated and then, using information vector $y(k)$, the posteriori estimations are updated in Eqs. (18)–(20), as follows.

$$K_k = P_k^- H_k^T (H_k P_{k+1}^- H_k^T + R_k)^{-1} \quad (18)$$

$$\delta \hat{x}_{k+1}^- = \delta \hat{x}_{k+1}^- + K_k (y_k - H_k \delta \hat{x}_k^-) \quad (19)$$

$$P_{k+1}^- = P_{k+1}^- - K_k H_k P_{k+1}^- \quad (20)$$

These calculations repeat every time a new data is received.

2.3 Robust Kalman Filter

Generally, the performance of the GPS/INS system entering a tunnel, area with buildings, or in a forest may frequently be degraded, causing gross observation errors in filtering and estimation. Moreover, INS outputs are always exposed to high noise and different types of uncertainties, such as bias, scale factor, and misalignment. Therefore, inconsistency between vehicle and sensors lead to INS gross errors [25, 26]. Availability of correct and persistent GPS data as observation is essential to continue correct estimation in integrated navigation systems. Theoretically, GPS observations must be Gaussian distributed with mean $H_k \delta \hat{x}_k^-$ and covariance $H_k P_{k+1}^- H_k^T + R_k$. Therefore, according to Eqs. (16)–(20), the standard EKF is carried out [27]. The square of the Mahalanobis distance of the observations y_k to its mean is considered as a test statistic described in Eq. (21) as:

$$\gamma_k = M_k^2 = (y_k - H_k \delta \hat{x}_k^-)^T (H_k P_{k+1}^- H_k^T + R_k)^{-1} (y_k - H_k \delta \hat{x}_k^-) \sim \chi_m^2 \quad (21)$$

where M_k is the Mahalanobis distance, γ_k is a static variable used as an index for probability test, m is the degree of freedom, which is the same dimension as the observation vector, and $\chi_a^2(m)$ is the probability threshold at significance level α , in which the null

hypothesis test will be rejected. Usually, α is a small value, e.g., below 5%. In this contribution, 1% for α_1 and 0.01% for α_2 are adopted. According to the law of the chi-square test, if the statistical index γ_k is between two thresholds, the observations are subject to gross error.

A scalar factor β is introduced in Eq. (22) to adjust the covariance of the observation prediction in (17) and (18) to ensure robustness as follows:

$$\hat{R}_k = \beta_k R_k \tag{22}$$

Equation (22) can be satisfied as stated in Eq. (23) as follows.

$$f(\beta_k) = (y_k - H_k \delta \hat{x}_k^-)^T (H_k P_{k+1}^- H^T + R_k)^{-1} (y_k - H_k \delta \hat{x}_k^-) - \chi_m^2 = 0 \tag{23}$$

Since equation (23) is nonlinear in β_k , by solving it iteratively using newton's method, Eq. 24 would be resulted as:

$$\beta_k(i+1) = \beta_k(i) - \frac{f(\beta_k(i))}{f'(\beta_k(i))} \tag{24}$$

Therefore, β_k would be summarized as per Eq. (24) as follows.

$$\begin{cases} \beta_k(i) = 1 & , i = 0 \\ \beta_k(i+1) = \beta_k(i) - \frac{M_k^2(i) - \chi_m^2}{(y_k - H_k \delta \hat{x}_k^-)^T (H_k P_{k+1}^- H^T + R_k)^{-1} R_k (H_k P_{k+1}^- H^T + R_k)^{-1} (y_k - H_k \delta \hat{x}_k^-)} & , i \geq 0 \end{cases} \tag{25}$$

If the gross error is small, GPS observations would significantly affect the update part of the filter by magnifying the covariance matrix of the measurement noise. However, if the error is considerably large, the observation is not beneficial. So, the covariance matrix is set to infinity to remove the negative effect of observation with the gross error. Therefore, the robust factor would be as Eq. (26).

$$\beta_k = \begin{cases} 1 & \text{if } \gamma_k < \chi_{a_1}^2(m) \\ \frac{M_k^2}{\chi_{a_1}^2(m)} & \text{if } \chi_{a_1}^2(m) < \gamma_k < \chi_{a_2}^2(m) \\ \infty & \text{if } \chi_{a_1}^2(m) < \gamma_k \end{cases} \tag{26}$$

Also, the Kalman gain in Eq. (15) is changed to the gain stated in Eq. (27), as follows.

$$K_k = \begin{cases} P_k^- H_k^T (H_k P_{k+1}^- H_k^T + R_k)^{-1} & \text{if } \gamma_k < \chi_{a_1}^2(m) \\ \beta_k P_k^- H_k^T (\beta_k H_k P_{k+1}^- H_k^T + R_k)^{-1} & \text{if } \chi_{a_1}^2(m) < \gamma_k < \chi_{a_2}^2(m) \end{cases} \tag{27}$$

This way, the information from the dynamic model would dominate the effect of actual observation, and therefore, the gross error will be overcome. This method is tested with the designed hardware and is presented in Section 3.

2.4 Integration scheme

There are two major types of integration schemes in sensor data fusion problems. In loosely-coupled integration, GPS output data and inertial sensors are integrated using a KF while in tightly-coupled integration, raw GPS data and INS measurements are used to calculate position, velocity, and orientation [28]. The integrated navigation system adopts the loosely-coupled integration scheme, and the fusion

method incorporating the data preprocessing algorithm and Nonlinear Auto-Regressive (NAR) Neural Network is proposed in the system, as shown in Fig. 2a. The input of the KF is the position error of the INS calculated by subtracting the GPS position from INS. Then, estimated system states are applied to the position, velocity, and orientation of INS and consequently, the final navigation output is resulted.

GPS/INS data fusion using KF, due to its high performance and simplicity, is the first choice in meeting primary navigation demands, and in usual conditions, it calculates the position of a vehicle with acceptable precision. However, in many occasional scenarios like in urban and military areas in which GPS is interrupted for a long period, due to reasons like entering a tunnel or intentional signal jamming, GPS accuracy is dramatically reduced because KF cannot estimate current position in the absence of GPS signal. In order to correct this problem, the structure depicted in Fig. 2a is used together with a neural network.

Using the neural network in real-world applications is highly dependent on the proper training of the network before using it. In this work, the neural network should provide data for a limited period but after a specified minimum amount of time. This time is dedicated to the training of the neural networks and is necessary for accurate prediction of the network and the entire algorithm during GPS out-ages. In this structure, the neural network is used in two modes of training and prediction. When the GPS data is available, the neural network trains from the KF output data and, when the GPS signal interrupts, as in Fig. 2b, the neural network predicts the position, velocity, and orientation error in the absence of the KF.

Feedforward neural network consists of three layers: input, hidden, and output layer. Each layer is composed of several neurons and different layers and can have different quantities of neurons (Fig. 3).

In this network, neurons are connected to their adjacent layers. Information transfers using these connections, and the corresponding transfer function is selected separately [14]. One of the conventional methods of training a neural network is the backpropagation approach. This approach consists of two paths: forward and backward. In forward path, the input vector is applied to a network, and its outputs are transferred through hidden layers into output layers so that the final output of MLP is achieved. Also, weight matrixes and bias vectors are considered constant and invariant. This path is represented using Eq. (28), as follows.

$$\hat{y}^{i+1}(K) = f^{i+1}(W^{i+1}(K)\hat{y}^i + b^{i+1}(K)) \tag{28}$$

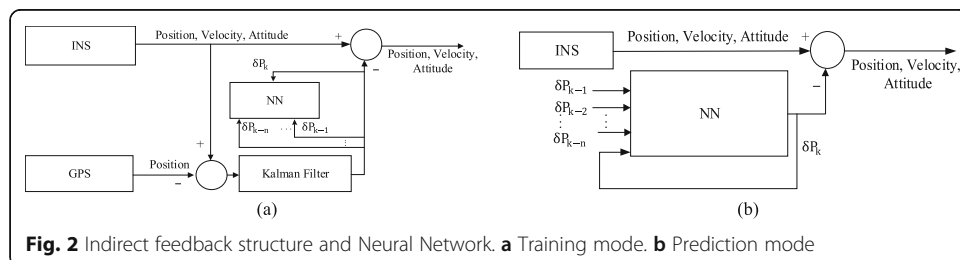
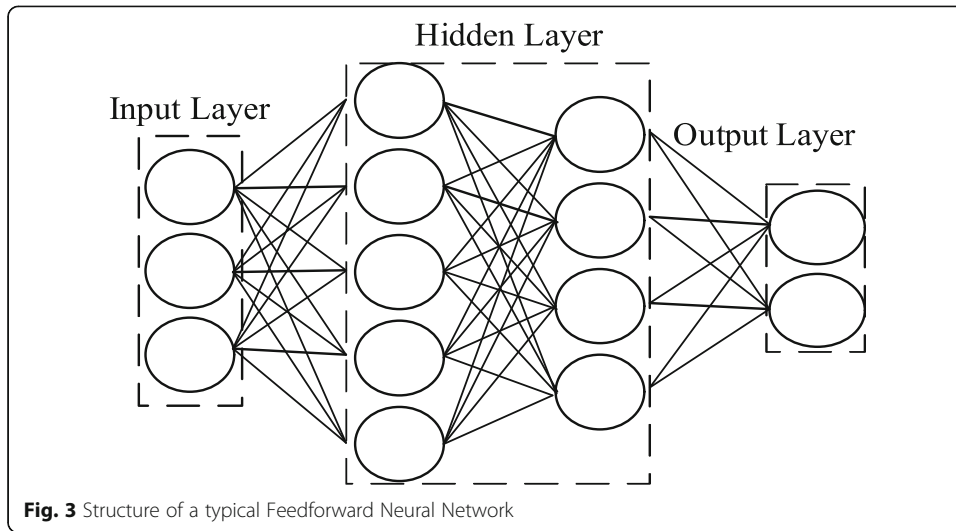


Fig. 2 Indirect feedback structure and Neural Network. **a** Training mode. **b** Prediction mode



where i is the layer number and \hat{y}^i is the previous layer output used to calculate layer $i + 1$ as its input.

In the backward path, the network parameters are adjusted based on the error back-propagation algorithm. The error vector is the difference between the desired response and the network response. The error value in the training step is sent in the opposite direction, from the output layer to the input layer, to adjust the weights and compensate the error to reach the desired response. When the output vector is compared with the desired vector, the error value in the n th output of the last layer is defined for the k th pattern as per Eq. (29).

$$e_n(K) = y(K) - \hat{y}(K) \quad (29)$$

The main goal is to estimate the network weights in order to minimize the Root Mean Square Error (RMSE) of the system and model output. The error value in each iteration for the n th neuron of the output layer is defined as $e_n^2(K)$. As a result, the network cost function is described in Eq. (30).

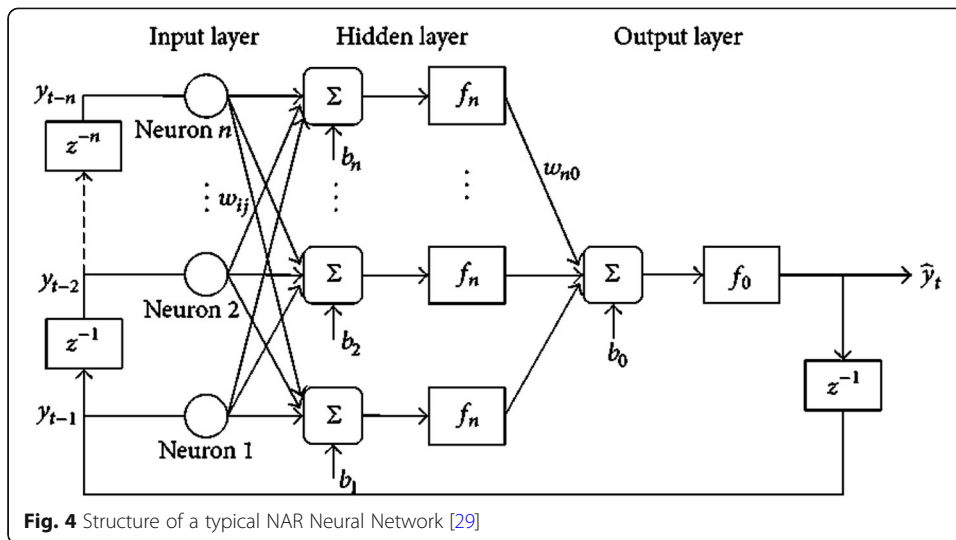
$$J(K) = \sum_{j=1}^n e_j^2(K) \quad (30)$$

With the capabilities of the NAR model to adapt to the complex and nonlinear dynamics of the time series, this model is one of the most appropriate tools for modeling and predicting time series of GPS/INS integration problems and is widely used in various applications [29] (Fig. 4).

NAR model input is a nonlinear function of previous outputs which is described as Eq. (31).

$$y(t) = f[y(t-1), y(t-2), \dots, y(t-q)] + e(t) \quad (31)$$

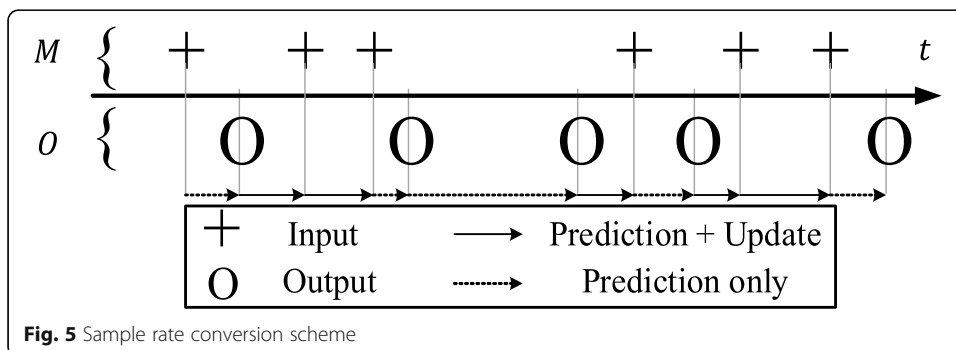
where $y(t-1), y(t-2), \dots, y(t-q)$ is a model output feedback, q is the order of the system, $e(t)$ is the noise, and $f(\cdot)$ is the nonlinear function, which should be estimated.



In control and navigation applications, an estimation of the system states is required at fixed intervals or a predetermined rate. Measurements of different sensors with different sampling characteristics should be synchronized, and standards of data transfer in the processor must be met. Therefore, an algorithm for converting sample rates should be used to ensure the availability of the data for signal processing at each interval [30]. The scheme of converting the sample rate is shown in Fig. 5.

At a specified time t , if an observation is available, the KF applies prediction and update steps. However, when the output is required, and measurement is not available, only the prediction step will be applied, and the update step will be deactivated. In Fig. 5, M contains measured samples, and O contains samples at the resampled rate. Because GPS receiver has a lower sample rate than IMU, and for some other system-based reasons, the data are usually not synchronous and should be resampled using different hardware or software methods.

Software resampling methods are generally based on interpolation, considering different interpolation methods such as linear, cubic, and spline. Because of the small differences in the final values, and the fact that the computation of linear interpolation is much easier than other methods, the linear method is used for resampling.



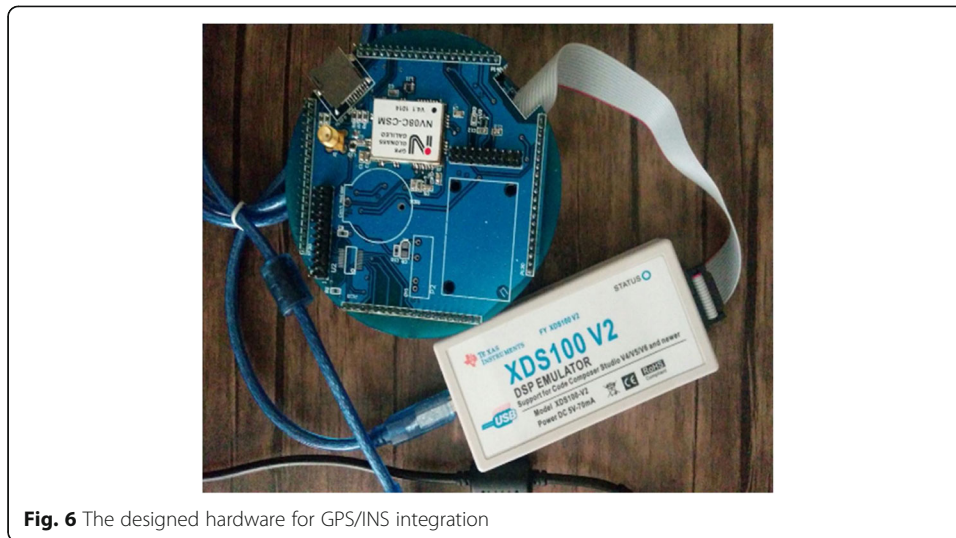


Fig. 6 The designed hardware for GPS/INS integration

3 Results and discussion

Moving an object toward an area with a signal jammer or similarly, during GPS signal interruption like the time that the vehicle is moving fast or passing through a tunnel, are important scenarios in aeronautical and automotive applications. In this scenario, it is presumed that at least four satellites are available, and at a particular moment, this number decreases to less than three satellites and stays in this state for a long period. In this section, the behavior of the system based on the defined scenario is studied, and additionally, the effect of the neural network on the system performance over a limited time is investigated.

An outdoor test has been performed to investigate the accuracy of the fusion algorithm in the defined scenario. The test hardware is installed on a car, and a closed path in an urban area has been travelled. The whole test is performed in 700 s, and after 600 s, GPS data has been interrupted for 14 s. The position results of the test are shown in Figs. 12 and 13 separately.

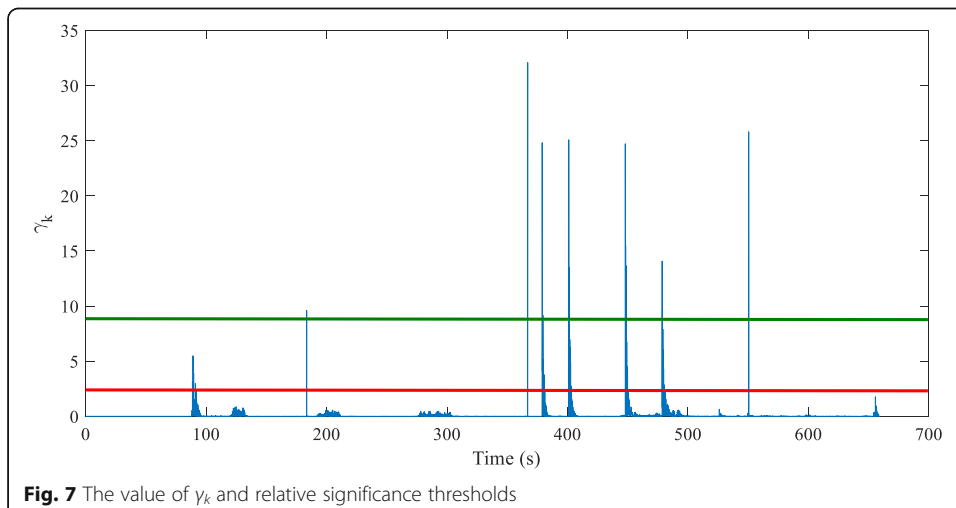


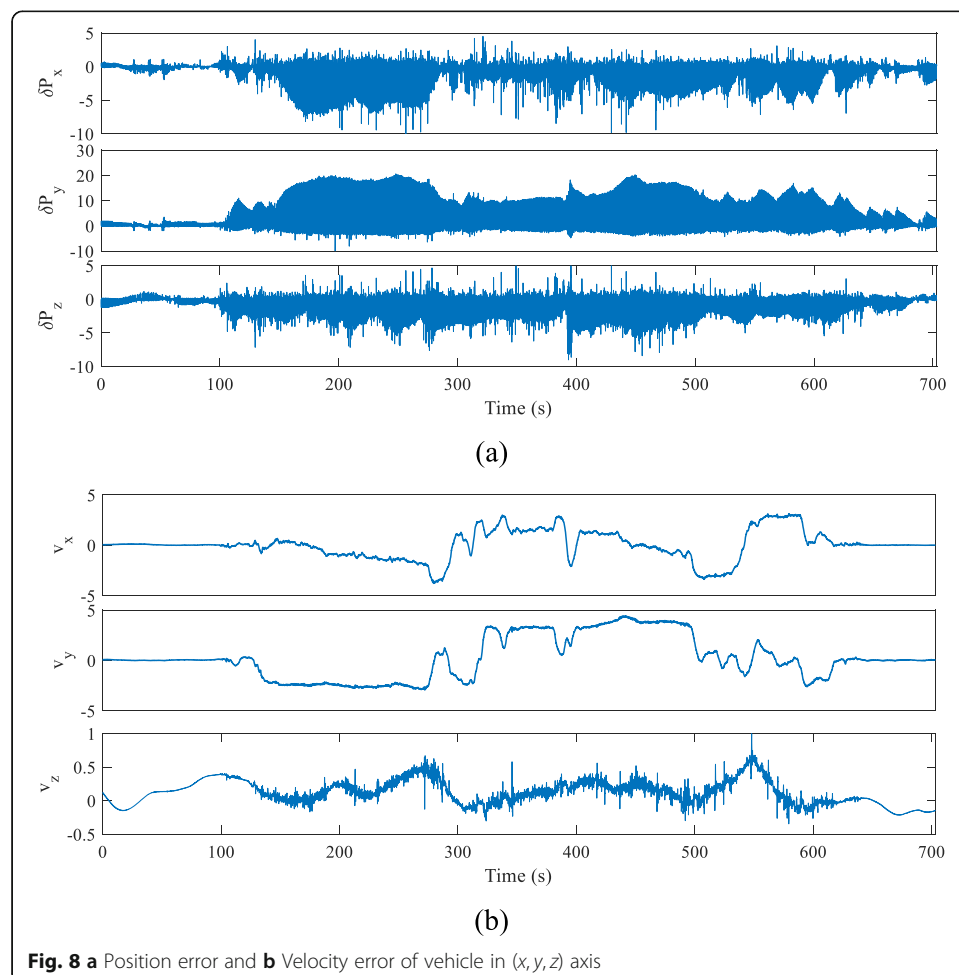
Fig. 7 The value of γ_k and relative significance thresholds

Table 1 RMS comparison of position error for the proposed algorithm

	Latitude	Longitude	Altitude
Standard KF	1.022	0.769	0.981
Robust KF	0.968	0.634	0.805

The test hardware is a custom-designed circuit based on Digital Signal Processor (DSP) in circular shape with a 100-mm diameter in order to fit in every moving vehicle and work standalone (Fig. 6). The design based on a modular approach in three distinct modular layers and the IMU sensor are also removable and replaceable.

The first layer includes power supply; the second layer includes the processor and communication modules. The processor is TMS320F28335 made by Texas Instruments with 150 MHz and single-precision floating-point unit (FPU), which makes it suitable for data acquisition and motor control. The third layer includes sensors, GPIO, and PWM outputs with IMU model ADIS16448, which is connected through SPI port and GPS model NV08C-CSM, which is connected through SCI port. Also, the programming and online plot of results are processed through an XDS100 V2 emulator.

**Fig. 8** a Position error and b Velocity error of vehicle in (x, y, z) axis

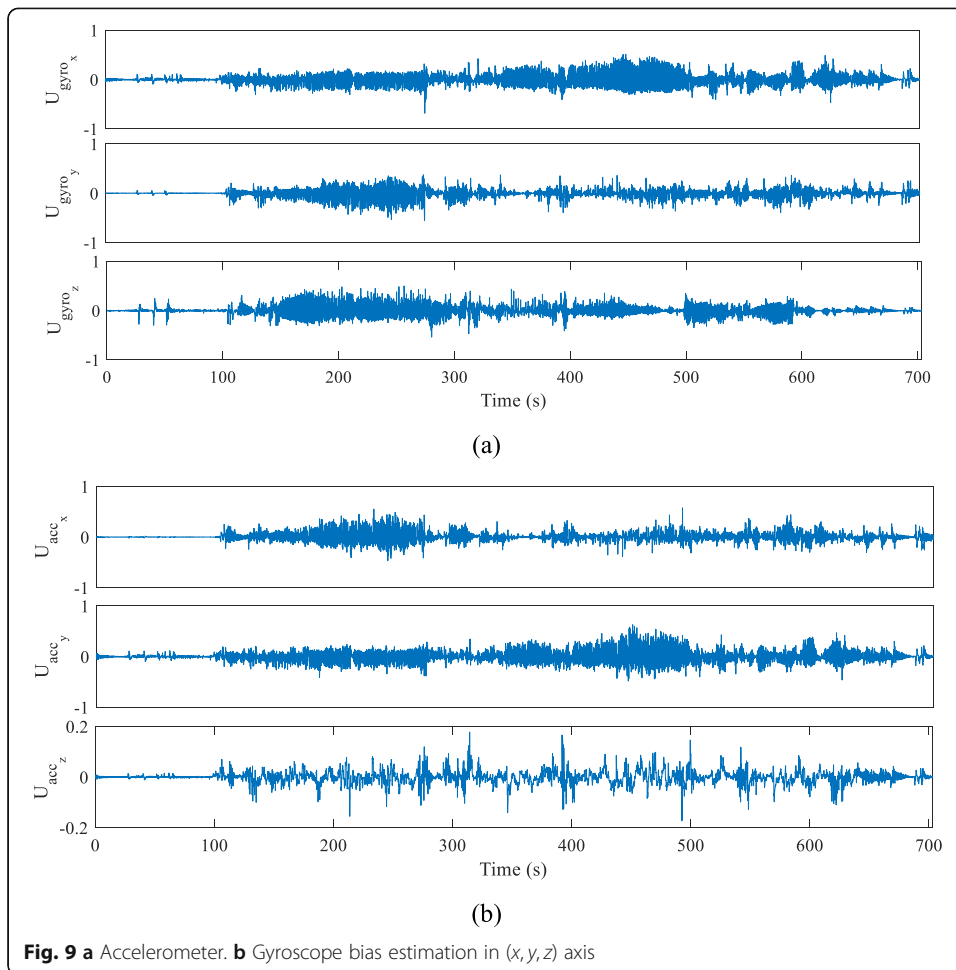


Fig. 9 a Accelerometer. b Gyroscope bias estimation in (x, y, z) axis

Figure 7 shows the values of γ_k , and the red and green lines are the threshold values of significance levels α_1 and α_2 , respectively. There are 71 values of γ_k which are between 8.9 and 3.2; also, there are 34 values of γ_k which are higher than 8.9. The proposed algorithm for Robust Kalman Filter (RKF) is employed to process the various levels of gross error in the system.

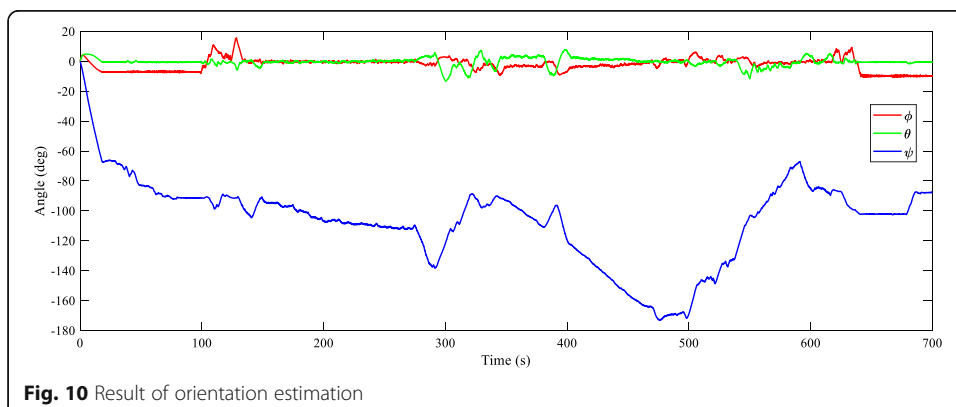


Fig. 10 Result of orientation estimation

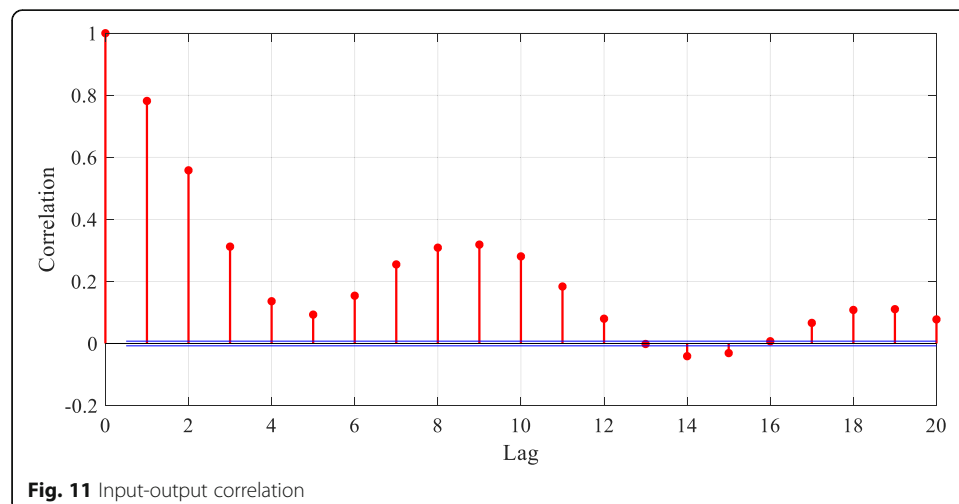
Table 1 illustrates the Root Mean Square (RMS) of the position error of two utilized Kalman filters. Figures 8 and 9 represent the errors of position and velocity as well as estimation of accelerometer and gyroscope biases in (x, y, z) .

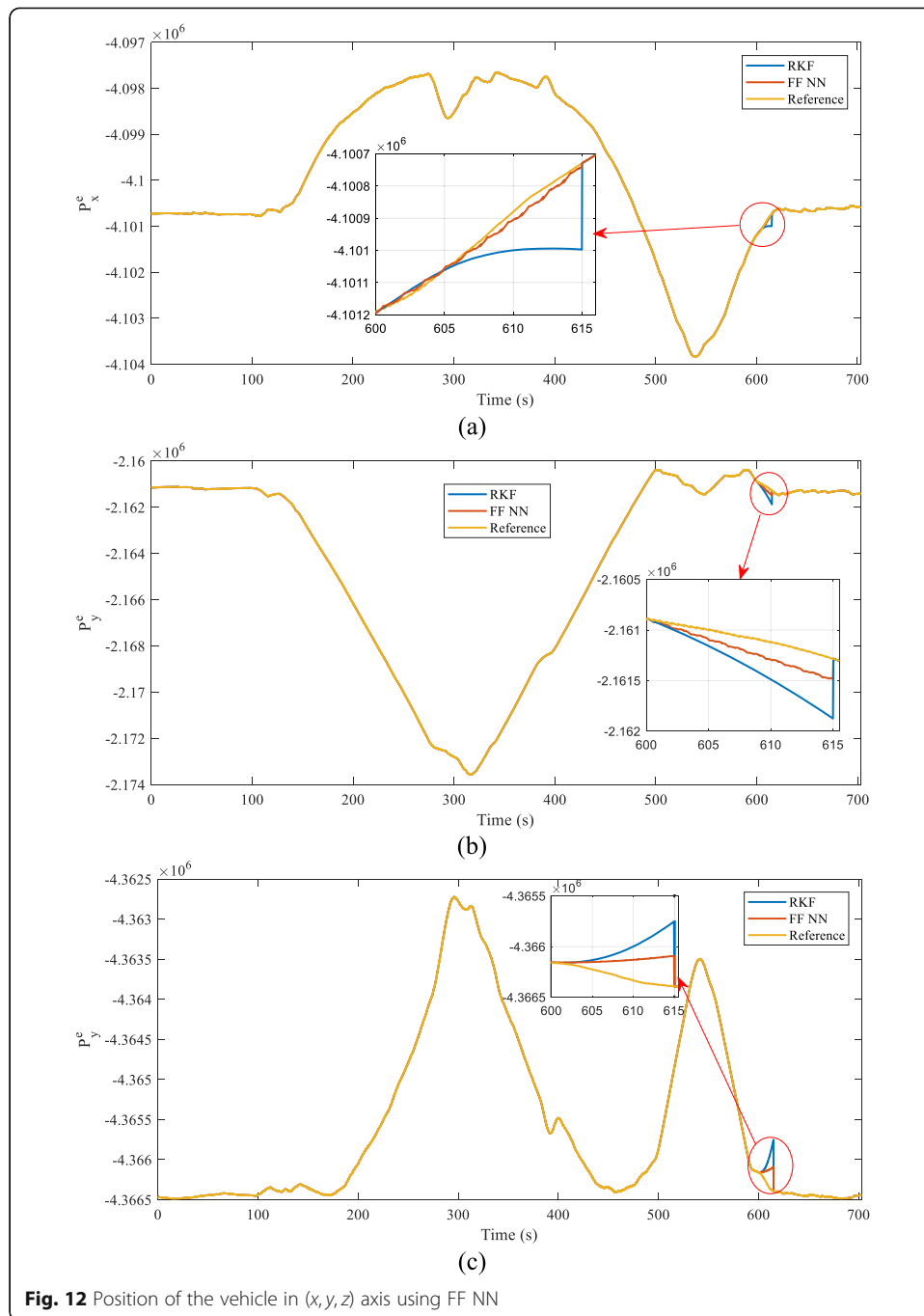
Results of orientation estimation are represented in Fig. 10. Since the test has been performed by a car in an approximately closed path and low slope road, it is evident that the ϕ and θ angles are close to zero. Also, the ψ angle clearly shows the closed path. There is also a saturated data recorded at the first 10 s of the test due to the immediate start acceleration of the test vehicle.

In order to improve the accuracy of the system, and also, identify the most accurate neural network model, the order of the model must be appropriately defined. Therefore, the best order of the model is calculated by observing the correlation of the inputs and the outputs of the network. Comparing the values of the correlations, and by the fact that the minimum absolute value is the best order to be chosen, best order should be 13 or 16. Nevertheless, considering the correlation results and limitations on processor computation capacity, the order of the system is chosen as 5. therefore, five previous inputs must be used simultaneously as the input of the Neural network. The result of the input and output data correlation is shown in Fig. 11.

Figure 12 represents a comparison between the position of the test path as reference, proposed RKF, and position estimation of the proposed neural network while GPS data is interrupted. GPS data has been interrupted for about 17 s, and the results show improvement of the estimation using the proposed neural network. Figure 13 also shows the same results of Fig. 12, but instead of Feedforward Neural Network, results of the NAR Neural Network are compared. The improvements by different methods are compared in Table 2.

In the proposed neural networks (Fig. 2), 75% of the data is dedicated to training, and the remaining 25% is for test. To simulate and test the described scenario, GPS data is presumed disconnected for 1500 samples starting from the sample number 60000 to 61500, and the last available GPS data is considered for these samples.





In Table 2, position error improvement by both feedforward and NAR Neural Networks is shown. The results indicate that both neural networks have a good effect on the reduction of position error during GPS data interruption.

4 Conclusion

This study developed an implementation of ANN aided GPS/INS integration for moving vehicle applications towards an edge-enable and cloud-based access to information [31–34]. In this paper, information fusion of GPS and INS sensors using Kalman Filter

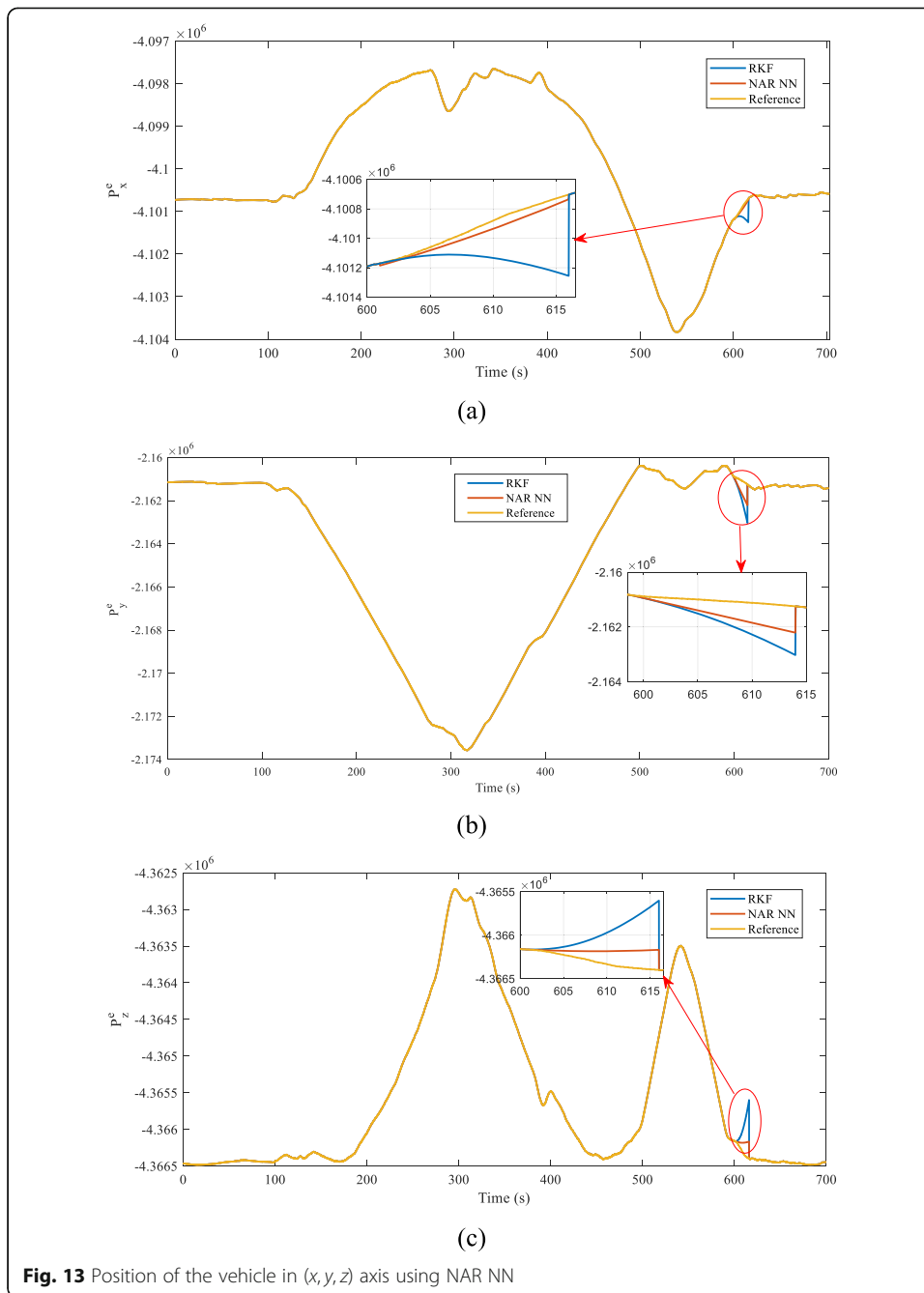


Table 2 Comparison of position improvement by applying the Neural Network

	Latitude	Longitude	Altitude
Feedforward	82.3%	67.4%	69.1%
NAR	86.3%	73.2%	71.6%

has been investigated. To improve the performance of the Kalman Filter, a robust method using Mahalanobis distance has been utilized. Also, in the previously defined scenario, two different types of Neural Networks are used to bridge GPS data interruption. A comparison between the proposed neural networks has been presented. Also, hardware was designed to test the proposed algorithm under the defined scenario. According to the presented figures, the proposed algorithm of GPS/INS data fusion shows a positive effect on the compensation of navigation error, reducing RMS of the position error by about 5%, and by using the neural network, the output position estimation error is reduced by at least 67%. The plot diagram shows that by using the mentioned algorithm, navigation system errors are well estimated in the absence of the GPS navigation system, and errors are limited to a specified range. The final accuracy of the state estimation highly depends on the covariance of the process and measurement noise, quality, and grade of the sensors used in hardware; therefore, navigation error compensation is generally limited.

Abbreviations

GPS: Global Positioning System; INS: Inertial Navigation System; KF: Kalman Filter; EKF: Extended Kalman Filter; UKF: Unscented Kalman Filter; RKF: Robust Kalman Filter; ANN: Artificial neural network; MLP: Multilayer Perceptron; RBF: Radial Basis Function; ANFIS: Adaptive Neuro-Fuzzy Inference Network; ECEF: Earth Centered Earth Fixed; NAR: Nonlinear Auto-Regressive; RMS: Root Mean Square; FPU: Floating Point Unit

Authors' contributions

MA, AMJ, and PA participated in the design of the proposed method and its implementation. MA and AMJ coordinated industrial application. PA and AMJ have completed the first draft of this paper. All authors have read and approved the manuscript.

Funding

Not applicable.

Availability of data and materials

All the data and computer programs are available.

Competing interests

The authors declare that they have no competing interests.

Received: 3 March 2020 Accepted: 3 June 2020

Published online: 19 June 2020

References

1. Maybeck P.S. "Stochastic models, estimation, and control," Mathematics in Science and Engineering. 3, Ohio, 1982.
2. Titterton D. "Strapdown Inertial Navigation Technology," IET Radar, Sonar & Navigation. 17, London, 1997.
3. Magnusson N. and Odenman T. "Improving absolute position of an automotive vehicle using GPS in sensor fusion," Chalmers University of Technology, 2012.
4. Gelb A. "Applied Optimal Estimation," The M.I.T. Press, 1974.
5. W. Wang, Z. Liu, R. Xie, Quadratic extended Kalman filter approach for GPS/INS integration. *Aerosp. Sci. Technol.* **10**, 709–713 (2006)
6. Zhao L., Qiu H., Feng Y. "Analysis of a robust Kalman filter in loosely coupled GPS/INS navigation system. *Measurement*," **80**, pp. 138–47, 2016.
7. A. Noureldin, R. Sharaf, A. Osman, N. El-Sheimy, INS/GPS data fusion technique utilizing radial basis functions neural networks. *Position Location and Navigation Symposium*. Monterey, USA, 280–284 (2004)
8. Sharaf R., Tarbouchi M., El-Shafie A., and A. Noureldin. "Real-time implementation of INS/GPS data fusion utilizing adaptive neuro-fuzzy inference system," *Institute of Navigation, National Technical Meeting, San Diego, CA*, pp. 235–42. 2005.
9. A. Noureldin, A. El-Shafie, M. Bayoumi, GPS/INS integration utilizing dynamic neural networks for vehicular navigation. *Information Fusion* **12**, 48–57 (2011)
10. R. Hecht-Nielsen, Theory of the backpropagation neural network. *Proceedings of the International Joint Conference on Neural Networks* **1**, 593–605 (1989)
11. T. Duc-Tan, P. Fortier, H.T. Huynh, Design, simulation, and performance analysis of an INS/GPS system using parallel Kalman filters structure. *REV Journal on Electronics and Communications.* **1**, 88–96 (2011)
12. Salytcheva A.O. "Medium accuracy INS/GPS integration in various GPS environments," University of Calgary, 2004.
13. G. Hu, S. Gao, Y. Zhong, A derivative UKF for tightly coupled INS/GPS integrated navigation. *ISA Trans.* **56**, 135–144 (2015)
14. Zhang Q. and Li B. "A low-cost GPS/INS integration based on UKF and BP neural network," *Fifth International Conference on Intelligent Control and Information Processing*. Dalian, China, pp. 100–107, 2014.

15. Y. Yao, X. Xu, C. Zhu, C.Y. Chan, A hybrid fusion algorithm for GPS/INS integration during GPS outages. *Measurement* **103**, 42–51 (2015)
16. X. Chen, C. Shen, W. Zhang, M. Tomizuka, Y. Xu, K. Chiu, Novel hybrid of strong tracking Kalman filter and wavelet neural network for GPS/INS during GPS outages. *Measurement*. **46**, 3847–3854 (2013)
17. R.E. Kalman, A new approach to linear filtering and prediction problems. *J. Basic Eng.* **82**(1), 35–45 (1960)
18. Ren Y., and Ke X. "Particle filter data fusion enhancements for MEMS-IMU/GPS," *Intelligent Information Management*, **2**(7), (2010).
19. S.Y. Cho, B.D. Kim, Adaptive IIR/FIR fusion filter and its application to the INS/GPS integrated system. *Automatica* **44**(8), 2040–2047 (2008)
20. Y. Yang, H. He, G. Xu, Adaptively robust filtering for kinematic geodetic positioning. *J. Geod.* **75**(2-3), 109–116 (2001)
21. Y. Yang, W. Gao, Comparison of two fading filters and adaptively robust filter. *Geo-Spatial Information Science* **10**(3), 200–203 (2007)
22. Y. Yang, X. Cui, Adaptively robust filter with multi adaptive factors. *Surv. Rev.* **40**(309), 260–270 (2008)
23. J. Li, N. Song, G. Yang, M. Li, Q. Cai, Improving positioning accuracy of vehicular navigation system during GPS outages utilizing ensemble learning algorithm. *Information Fusion* **35**, 1–10 (2017)
24. J.Z. Sasiadek, P. Hartana, GPS/INS Sensor fusion for accurate positioning and navigation based on Kalman Filtering. *IFAC Proceedings*. **37**, 115–120 (2004)
25. A.G. Quinchia, G. Falco, E. Falletti, F. Dovis, C. Ferrer, A comparison between different error modelling of MEMS applied to GPS/INS integrated systems. *Sensors* **13**, 9549–9588 (2013)
26. J. Wang, H. Han, X. Meng, Z. Li, Robust wavelet-based inertial sensor error mitigation for tightly coupled GPS/BDS/INS integration during signal outages. *Surv. Rev.* **49**(357), 419–427 (2017)
27. Z. Li, G. Chang, J. Gao, J. Wang, A. Hernandez, GPS/UWB/MEMS-IMU tightly coupled navigation with improved robust Kalman filter. *Adv. Space Res.* **58**, 2424–2434 (2016)
28. V.T. Pham, V.T. Nguyen, D.T. Chu, D.T. Tran, 15-state extended Kalman filter design for INS/GPS navigation system. *Journal of Automation and Control Engineering*. **3**, 109–114 (2015)
29. H.T. Pham, B.S. Yang, A hybrid nonlinear autoregressive model with exogenous input and autoregressive moving average model for long-term machine state forecasting. *Expert Syst. Appl.* **37**, 3310–3317 (2010)
30. T. Kautz, B.M. Eskofier, A robust Kalman framework with resampling and optimal smoothing. *Sensors* **15**(3), 4975–4995 (2015)
31. C. Zhang, M. Yang, J. Lv, W. Yang, An improved hybrid collaborative filtering algorithm based on tags and time factor. *Big Data Mining and Analytics* **1**(2), 128–136 (2018)
32. D. Shen, J. Luo, F. Dong, J. Zhang, Joint Coflow Scheduling and Virtual Machine Placement in Cloud Data Centers. *Tsinghua Sci. Technol.* **24**(5), 630–644 (2019)
33. A. Ramlatchan, M. Yang, Q. Liu, M. Li, J. Wang, Y. Li, A survey of matrix completion methods for recommendation systems. *Big Data Mining and Analytics* **1**(4), 308–323 (2018)
34. G. Li, S. Peng, C. Wang, J. Niu, Y. Yuan, An Energy-Efficient Data Collection Scheme Using Denoising Autoencoder in Wireless Sensor Networks. *Tsinghua Sci. Technol.* **24**(1), 86–96 (2019)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)
