*Int. J. Appl. Math. Comput. Sci., 2021, Vol. 31, No. 1, 45–57*
DOI: 10.34768/amcs-2021-0004

# AN OPTIMAL INNOVATION BASED ADAPTIVE ESTIMATION KALMAN FILTER FOR ACCURATE POSITIONING IN A VEHICULAR AD–HOC NETWORK

S. SUMITHRA [a,*], R. VADIVEL [a]

[a]Department of Information Technology
Bharathiar University
Maruthamalai Main Road, Coimbatore, 641046, Tamil Nadu, India
e-mail: sumiphdit@gmail.com, vlr_vadivel@yahoo.co.in

The vehicular ad-hoc network (VANET) is subject to various attacks because of its dynamic nature and ephemeral character. In VANET, vehicles communicate with each other for safety awareness. The positioning of an unknown vehicle is one of the critical factors to determine the vehicle's trustworthiness. Although some positioning techniques have achieved a high accuracy level in VANET, they suffer from dynamic noise in real-world environments. This drawback leads to inaccuracy and unreliability during vehicle positioning. In this paper, an optimal innovation based adaptive estimation Kalman filter (OIAE-KF) is proposed. This algorithm offers an alternative solution for the basic Kalman filter and the innovation based adaptive estimation Kalman filter (IAE-KF). The proposed algorithm makes use of fusion of the global navigation satellite system (GNSS) and the inertial measurement unit (IMU) to improve its performance. The OIAE-KF works based on the innovation sequence and involves three steps such as establishing the innovation sequence, applying the innovation property, checking the optimality of the Kalman filter and, finally, estimating process noise (Q) and measurement noise (R). An optimal swapping method is introduced for optimality check. The efficiency of the proposed OIAE-KF method is proved by comparing the predictions of the existing methods such as the IAE-KF. The results show that the OIAE-KF performs better than the existing techniques. It improves the accuracy and consistency in VANET positioning.

**Keywords:** Kalman filter, innovation, dead reckoning (DR), global positioning system (GPS), simulation of urban mobility (SUMO).

## 1. Introduction

The vehicular ad-hoc network (VANET) aims to support many applications like intelligent transportation systems (ITSs) for traffic efficiency, road safety, and various entertainment services (Lobo *et al.*, 2019). Current research methods on VANET include unmanned vehicles (UVs), cooperative adaptive cross control (CACC) (Milanés *et al.*, 2013), cooperative collision warning (Huang and Lin, 2014), advanced driver-assistance systems (ADASs) (Li *et al.*, 2011), intersection safety (Le *et al.*, 2009) and cooperative driving (Kato *et al.*, 2002). These kinds of VANET applications have great benefits that can save hundreds and thousands of lives every year. A wide range of applications has been discussed (Al-Sultan *et al.*, 2014).

The vehicle's position information is essential for VANET applications to perform better. For example, UV, CCWS, and ADAS require a lane level accuracy of about one meter. Few applications such as traffic efficiency and entertainment applications demand an accuracy of about 5 meters (Skog and Handel, 2009).

Numerous approaches have been proposed for vehicle positioning in VANET, e.g., map matching based, dead reckoning based, GPS based, cellular based, etc. None of the above techniques fulfills the requirements of correct positioning (Amadeo *et al.*, 2012). Some integrated techniques are designed for navigation systems. Nevertheless, those techniques are lacking in the availability and accuracy of positioning requirements.

Most commonly, the Kalman filter can be used for combining positioning algorithms (Lobo *et al.*, 2019). An enormous amount of research has been done to

*Corresponding author

improve the accuracy with several fusion techniques. Bidyuk *et al*. (1993) showed the analytical representation of the fundamental characteristics of the Kalman filter. This representation describes the potential possibility of filtering algorithms for a class of dynamic systems such as moving objects or targets. They also developed an analytical representation for the variance of actual optimal and sub-optimal errors of filtering for the case of correlated measurement noise. Korbicz *et al*. (1994) considered integrating data that come from various navigation sources at different moments. An algorithm of a modified Kalman filter was established to process scalar data separately, even in vector input . This approach allows constructing a unified estimation algorithm for processing variable dimension measurement vectors. The modified Kalman filter algorithm decreases the covariance of estimated errors if a different measurement appears at the filter input from a new source and with different sampling intervals. This method is used to avoid filter divergence resulting from computational errors.

Hide *et al*. (2004) compared the adaptive Kalman filter, process noise scaling, and multiple model adaptive estimation (MMAE). MMAE was developed to improve the estimation in a low-cost inertial navigation system (INS). MMAE is suitable for real-time applications because of its significant time processing. The problem is that the estimated process noise is unstable for dynamic and variant noise environments. Hide *et al.* (2004) compared three positioning algorithms, and the IAE-KF showed low performance. The method of tuning the process noise covariance was developed by Ding *et al*. (2007). Here, the authors fixed the measurement noise covariance as for stationary and known white noise. A drawback of this model is that measurement noise is always more dynamic than process noise due to the dynamic nature of VANET.

ZhiWen *et al*. (2013) introduced a robust IAE-KF algorithm that computes the innovation sequence with the chi-square test. Here, measurement noise covariance is known and fixed. Mehra (1970) introduced an IAE-KF for estimating the process noise covariance and kept the measurement noise as unknown and dynamic. IAE was modified by Loebis *et al*. (2004) for autonomous underwater vehicles (UAVs). Here the author used a fuzzy logic method (Wang *et al*., 2013). The adaptation of the covariance matrix was made based on the sign of the last innovation value. If the sign is positive, then the covariance matrix is decreased, otherwise it is increased.

Havangi *et al*. (2010) introduced an adaptive neuro-fuzzy extended Kalman filtering for robot localization (ANFEKF). The ANFEKF estimates the covariance matrices of process and measurement noise for the extended Kalman filter (EKF). The parameters are trained using the steepest gradient descent to minimize the differences between the actual and the theoretical covariance. Gao *et al*. (2017) developed an improved innovation-based adaptive estimation Kalman filter. A regulatory factor is proposed for the computation of the Kalman gain matrix to solve the singular value problem for matrix inversion. As a result, the measurement noise uncertainties are eliminated during estimation.

Woo *et al*. (2019) developed a fuzzy-innovation based adaptive extended Kalman filter (FI-AKF). Here, the authors used a fuzzy logic system that updates the measurement covariance matrix of the real-time kinematics according to the position dilution of precision (PDOP), the number of satellites involved, and the innovations of the extended Kalman filter. The driving state of the vehicle is also recovered. Alam and Dempster (2013) introduced a positioning technique based on the inertial navigation system (INS) fusion method and GPS measurements. Parker and Valaee (2007) introduced a cooperative positioning concept that integrates GPS and INS. Here the process noise covariance was very small, and the measurement noise covariance was stationary and fixed.

Khattab *et al*. (2015) considered the single RSU based positioning algorithm. The main limitation of this method is that lateral position information was neglected and cost-effective. Chatterjee and Matsuno (2007) projected a neural network-based model for estimating measurement noise to solve simultaneous localization and mapping (SLAM), which is applied in mobile robots. The theory of an adaptive Kalman filter (AKF) is used to resolve environmental noise by prior estimation of process and measurement noise covariances (Bar-Shalom *et al*., 2004). Most of the researchers focus on the innovation based adaptive estimation Kalman filter (IAE-KF). The IAE-KF estimates the process noise covariance and the measurement noise covariance. As a simultaneous process, dynamic state and error covariances are also estimated. The IAE-KF aims to match the actual innovation covariance with the theoretical covariance value. Nevertheless, the IAE-KF is degrading with the property that it cannot reach a steady state until noise estimators like GNSS and IMU come to an agreement, which reduces the rate of convergence and filter accuracy (Hou *et al*., 2013).

Based on the existing methods, most of the techniques operate in a controlled environment for vehicle positioning. In most existing methods, the measurement noise covariance is assumed to be stationary, but the noise may vary based on environmental conditions (Chen *et al*., 2010). This noise variation factor reduces the accuracy in positioning, which results in a mismatch of the vehicle position. The uncertainty of the acquired information about the position is crucial in VANET applications from the above observations. In this paper, the uncertainty of position information is targeted to enhance positioning accuracy and integrity. In many situations, innovations

of the Kalman filter get corrupted. In that situation, the filter is replaced with an alternative prediction technique such as dead reckoning (DR) (Sierociuk and Dzieliński, 2006). This idea is followed in the proposed OIAE-KF positioning algorithm to improve the accuracy and integrity. The contribution of this paper is the following. Initially, the IAE Kalman filter estimates both the process noise covariance and the measurement noise covariance using innovations. This research aims to enhance the IAE-KF as an OIAE-KF. The IAE-KF is modified with enhancements such as obtaining the innovation property based on the innovation sequence and optimality checking. The Pearson correlation coefficient is employed to obtain the autocorrelation coefficient, which helps determine the variance and the covariance. An optimality checking algorithm is introduced to manage the adaptive Kalman filter. Finally, matrices $Q$ and $R$ are estimated based on the OIAE-KF. The accuracy in position measurements is found by comparing the proposed method and the IAE-KF.

## 2. Basic Kalman filter and the innovation based adaptive estimation Kalman filter

The underlying application plays a major role in achieving accurate positioning information of a vehicle (Kalman, 1960). Three essential requirements should be concentrated on to obtain acceptable VANET performance. First, the vehicle should be able to obtain continuous position information in any situation (Mohamed and Schwarz, 1999). The first requirement relates to availability. Second, positioning algorithms must be capable of accommodating vehicle dynamics and the surrounding environment. Finally, the positioning information gathered by the vehicle must be a clear and immediate indication. In simple words, the vehicle should be able to measure the uncertainty of the position information. The second and final requirements are related to accuracy.

Using redundancy and complementary fusion techniques such as DR and GPS, availability issues can be solved (Rezaei and Sengupta, 2007). Nevertheless, solving an accuracy issue is still a challenging task. Position integrity is the level of trust which can be placed on the accuracy of position information. Integrity can be measured by analyzing uncertainty. Increasing the positioning accuracy and improving the integrity level is the basic principle for VANET applications. However, these requirements have not been investigated in existing positioning algorithms (Jiancheng and Sheng, 2011).

VANET nodes (vehicles) travel in harsh environments, so positioning algorithms are prone to environmental noise (ETSI, 2010). White noise is defined as a random signal having spectrum over the frequencies relevant to the circumstances and a Gaussian distribution. The Gaussian distribution is characterized by a bell-shaped curve, and it produces equally likely measurements above and below the mean value (Li *et al*., 2016). The variation in the noise may lead to non-white noise, which affects the Gaussian distribution. The basic Kalman filter is suitable for position estimation if preliminary information about the noise involved in signals is known, and these noise signals are white (Wang *et al*., 2013). Errors in preliminary information lead to inadequate estimates. Sometimes this results in filter divergence, which minimizes the accuracy level of the positioning information.

The Kalman filter depends on two basic models (Li *et al*., 2015). The first is the process model or the system model that describes the state of transition that is used for prediction. The other is the measurement model or the observation model that describes the relationship between the state and observation. The Kalman filter runs based on a discrete linear state model, and a discrete linear measurement model is needed (Sarkka and Nummenmaa, 2009). The Kalman filter is an iterative process. It uses a set of equations and consecutive data inputs to quickly estimate the true value of the objects' position and velocity when it has measured values containing unpredictable errors or uncertainties. The Kalman filter model predicts the true state at time $k$ from the previous state $(k-1)$, according to

$$x_k = Ax_{k-1} + Bu_k + w_k, \tag{1}$$

$$y_k = Hx_k + v_k. \tag{2}$$

The prediction phase has the form

$$x_{k_p} = Ax_{k-1} + Bu_k + w_k, \tag{3}$$

$$P_{k_p} + AP_{k-1}A^T + Q_k. \tag{4}$$

The update phase is

$$\check{e}_k = y_k - Hx_{k_p} = y_k - y_{k_p}, \tag{5}$$

$$S_k = HP_{k_p}H^T + R. \tag{6}$$

The state correction phase is

$$\begin{aligned} K_k &= P_{k_p}H^T(HP_{k_p}H_k^T + R_k)^{-1} \\ &= P_{k_p}H^T S_k^{-1}, \end{aligned} \tag{7}$$

$$\check{x}_k = x_{k_p} + K_k\check{e}_k, \tag{8}$$

$$\check{P}_k = (1 - K_kH)P_{k_p}. \tag{9}$$

**2.1. Innovation based adaptive estimation Kalman filter.** The IAE-KF represents the following equation:

$$\hat{C}_{e_k} = \frac{1}{\mathrm{MW}} \sum_{j=k-\mathrm{MW}+1}^{k} e_j e_j^T, \tag{10}$$

where MW is the size of the moving window. By comparing the actual covariance value with the theoretical covariance value, the measurement noise covariance $\hat{R}_k$ can be found. The theoretical noise covariance is

$$S_k = HP_{k_p}H^T + R_{(k-1)} = E[\hat{e}_j\hat{e}_j^T]. \tag{11}$$

The process noise covariance matrix can be obtained from the actual covariance matrix (Hu, 2017),

$$\hat{R}_k = \hat{C}_{e_k} - HP_{k_p}H^T, \tag{12}$$

$$\hat{Q}_k = K_k\hat{C}_{e_k}K_k^T. \tag{13}$$

In real time, it is difficult to estimate $Q$ and $R$ simultaneously when dynamic noise appears. By using the innovation sequence, the Kalman gain can be calculated directly, i.e.,

$$K_k = P_{k_p}H^T\hat{C}_{e_k}^{-1}. \tag{14}$$

The IAE-KF is mostly used for dynamic noise in linear systems. It uses innovation sequence $\check{e}_k$ to compute the noise covariance to correct the state estimate. The innovation sequence is the difference between the predicted value and the actual measurement. Both the measurement model and the predicted model are in agreement if the value of $\check{e}_k$ is equal to zero. The innovation sequence will first show the difference of statistic data and prior information. Thus, $\check{e}_k$ can produce the actual status of convergence or divergence of the Kalman filter (Mohamed and Schwarz, 1999).

## 3. Optimal innovation based adaptive estimation Kalman filter

The main idea of the OIAE-KF is to remediate the problem of position estimation during vehicle tracking. Defects in error estimation result in incorrectly positioning the vehicle. The proposed algorithm uses velocity and position computed by the sensors of the onboard unit (OBU) in a vehicle such as an accelerometer and a gyroscope (Liu and Xiong, 2011). Measurements of quantities such as the engine RPM, gear ratio, axle ratio, and wheel circumference serve to calculate the velocity. These measurements implement the adaptive noise estimation to directly affect each sensor's performance over the dynamic environment conditions (Yu, 2012). Optimality checking improves the accuracy and minimizes the uncertainties in error estimation. The innovation sequence is the difference between the actual measurement and the predicted one.

An optimal swapping algorithm swaps between the adaptive Kalman filter and the dead reckoning process model based on the uncertainties raised during estimation. To achieve the optimality condition, three necessary steps are to be followed (Wang *et al.*, 2009). Figure 1 shows the diagrammatic representation of the OIAE-KF. The innovation sequence discovers the irregularity between the observed and the measured positioning values. This variation occurs due to various conditions such as weather, the vehicle model, the driver status, and other environmental circumstances (Wang *et al.*, 2013). When vehicles move in harsh environments, the noise projection may not be stationary white noise or Gaussian noise. Therefore the innovation property predicts the deviation from the Gaussian distribution and the unbiased conditions of the noise. The noise is not Gaussian white noise when a variation is encountered, and it is biased. The OIAE-KF utilizes the innovation property to test the filter's optimality and, based on the test result, the estimation swap between the Kalman filter and dead reckoning.

- **Step 1:** Collect the innovation sequence using the actual and predicted measurements.
- **Step 2:** Apply the innovation property to check biasedness (white noise).
- **Step 3:** Check optimality conditions of the Kalman filter.

As the final stage, the process noise covariance and the measurement noise covariance are estimated to ensure the accuracy of the estimation. The accuracy is compared with the online positioning technique.

**3.1. Collecting innovation sequence using actual and predicted measurements.** The variation in the measurements of kinematic sensors and the GNSS produces the innovation sequence $\check{e}_k$. The GNSS and dead reckoning measure the vehicle's position. These measurements are compared with the vehicle's actual position, and the error is collected with time lag $k$. Let $p_{\mathrm{GNSS}}$ be the position obtained by the GNSS and $p_{\mathrm{DR}}$ be the position obtained by dead reckoning. The vehicle's exact or true position is represented as $p_{\mathrm{true}}$. Let $\epsilon_{\mathrm{GNSS}}$ be the error in the GNSS measurement, $\epsilon_{\mathrm{DR}}$ be the error in the DR measurement, and $\Delta T$ be the sampling rate at all instances of time,

$$\check{e}_k = p_{\mathrm{DR_k}} - p_{\mathrm{GNSS_k}}, \tag{15}$$

$$p_{\mathrm{GNSS_k}} = p_{\mathrm{true_k}} + \epsilon_{\mathrm{GNSS_k}}, \tag{16}$$

$$p_{\mathrm{DR_k}} = p_{\mathrm{true_k}} + \epsilon_{\mathrm{DR_k}}. \tag{17}$$

Substitute (16) and (17) in (15) to get

$$\begin{aligned}\check{e}_k &= p_{\mathrm{DR_k}} - p_{\mathrm{GNSS_k}} \\ &= (p_{\mathrm{true_k}} + \epsilon_{\mathrm{DR_k}}) - (p_{\mathrm{true_k}} + \epsilon_{\mathrm{GNSS_k}}) \\ &= \epsilon_{\mathrm{DR_k}} - \epsilon_{\mathrm{GNSS_k}}.\end{aligned} \tag{18}$$

The subtracted value (error) of the GNSS and DR measurements produces the innovation sequence. If the

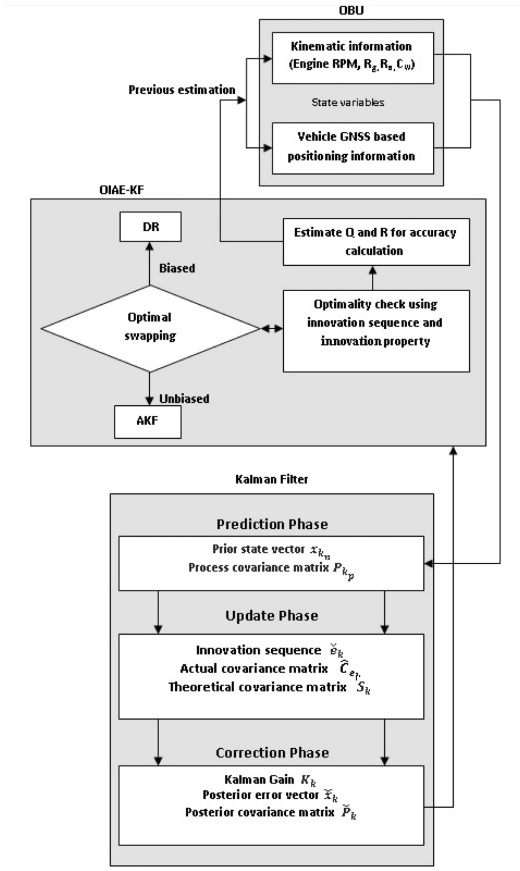


Fig. 1. Diagrammatic representation of the OIAE-KF.

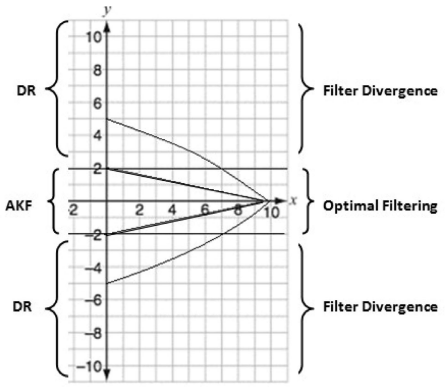


Fig. 2. Optimal swapping.

DR measurement error is minimal, it will be approximated by an error in the GNSS measurement.

**3.2. Applying innovation to check biasedness.** The main aim of applying the innovation property is to find whether the innovation sequence is Gaussian or non-Gaussian white noise. First, the correlation factor $\rho_k$ becomes the innovation property computed for the innovation sequence. In this paper, Pearson's correlation coefficient is used to establish the correlation factor. It shows the strength of the narrow relationship between the errors in the measured values. The innovation property is applied to the innovation sequence to check the optimality of the state estimate of the Kalman filter, $\rho_k$ is the autocorrelation coefficient, which reflects the relationship between the innovation sequence at various time lags in milliseconds. The window size is denoted as $\mathrm{wm}$, and the mean of the innovation sequence is denoted as $\mu_\epsilon$,

$$\rho_k = \frac{\sum_{k=1}^{\mathrm{wm}-1} (\check{e}_k - \mu_\epsilon)(\check{e}_k + 1 - \mu_\epsilon)}{\sqrt{\sum_{k=1}^{\mathrm{wm}} (\check{e}_k - \mu_\epsilon)^2} \sqrt{\sum_{k=1}^{\mathrm{wm}} (\check{e}_{k+1} - \mu_\epsilon)^2}}. \tag{19}$$

**3.3. Checking optimality of the KF.** Algorithm 1 shows the comparison made between the actual error covariance $\hat{C}_{e_k}$ and the theoretical error covariance $S_k$. When $\hat{C}_{e_k}$ is not equivalent to $S_k$, the innovation sequence $\check{e}_k$ will mislead the error deviation. As a result, considerable filter divergence occurs. Divergence happens when the autocorrelation coefficient $\rho_k$ of the innovation sequence is high, or the noise type is non-Gaussian. Now swap the estimation process to DR, and, increase the measurement noise covariance $Q_k$ by time lags until $\check{e}_k$ becomes stationary. When $\hat{C}_{e_k}$ is equal to $S_k$, the correlation coefficient meets the consistency level. That shows that the noise is unbiased, and the error is Gaussian white noise. In this situation, the adaptive Kalman filter (AKF) maintains the estimation process. Thus, the AKF automatically converges to an optimal state.

Minimize the DR process when divergence is minimized the autocorrelation coefficient remains within the range of $\pm 2\sqrt{\mathrm{wm}}$, then conclude that random white noise with no divergence. Mehra (1970) proves that if $\rho_k$ is greater than the bound $-2\sqrt{\mathrm{wm}}$ and smaller than $+2\sqrt{\mathrm{wm}}$, then the filter is unbiased, and the Kalman Filter is optimal. If $\rho_k$ is smaller than the bound $-2\sqrt{\mathrm{wm}}$ and greater than $+2\sqrt{\mathrm{wm}}$, it is said that it is non-Gaussian and biased noise. Figure 2 demonstrates the optimal swapping process.

**3.4. Estimating process noise covariance $Q$ and measurement noise covariance $R$.** The estimation of the process noise covariance is known because it depends on the vehicle's GNSS and kinematic sensor observations. Nevertheless, the measurement noise covariance depends on the environmental condition that results in dynamic noise. Therefore in most cases, $Q$ is kept unknown. The Kalman filter depends on both $Q$ and $R$. If $Q > R$, the Kalman gain $K_k$ relies on the measurement and not on prediction. In contrast, if $Q < R$, the Kalman gain $K_k$ relies on the prediction more than measurement. Because of the known and fixed value of $Q$, it is crucial to analyze

the influence of $R$ to maintain accurate estimation of noise statistics. Fix $Q$ as a small value; then $R$ is approximated according to the correlation between the measurements. Autocorrelation of the innovation sequence is derived as $\rho_k$ based on Pearson's autocorrelation coefficient. The innovation sequence is arranged based on first-order autoregressive representation as $e_k$, where $a_k$ is the Gaussian noise process with zero mean. One of the essential features innovation sequence is that it should always be a positive definite matrix,

$$e_k = \rho(e_{k-1}) + a_k. \tag{20}$$

The variance and the covariance can be computed as follows:

$$\mathrm{Var}(e_k) = \sigma_e^2 = \frac{1}{\mathrm{MW}(1-\rho_k^2)} \sum_{j=k-\mathrm{MW}+1}^{k} e_j e_j^T \tag{21}$$

$$\begin{aligned} \mathrm{Cov}(e_k) &= \sigma_{e_k}\sigma_{e_k+1} \\ &= \frac{\hat{C}_{e_k}}{1-\rho_k^2}(\hat{e}_k - \mu_\epsilon)(\hat{e}_{k+1} - \mu_\epsilon), \end{aligned} \tag{22}$$

where

$$\hat{C}_{e_k} = \frac{1}{\mathrm{MW}} \sum_{j=k-\mathrm{MW}+1}^{k} e_j e_j^T.$$

**Algorithm 1.** Optimality check in the OIAE-KF.

Input: $\hat{C}_{e_k}$, $S_k$, $\rho_k$, $\check{e}_k$, $Q_k$
Output: DR, AKF, biased, unbiased, stationary, non-Gaussian, Gaussian
While $\hat{C}_{e_k} \neq S_k$
  If $\rho_k < -2\sqrt{\mathrm{wm}}$ && $\rho_k > +2\sqrt{\mathrm{wm}}$
    $\check{e}_k \leftarrow$ biased
    error $\leftarrow$ non-Gaussian
    Swap $\leftarrow$ DR (when $\rho_k \leftarrow$ high)
    $Q_k \leftarrow Q_{k+1}$(Until $\check{e}_k \leftarrow$ stationary)
  End If
End While
While $\hat{C}_{e_k} = S_k$
  If $\rho_k > -2\sqrt{\mathrm{wm}}$ && $\rho_k < +2\sqrt{\mathrm{wm}}$
    $\check{e}_k \leftarrow$ unbiased
    error $\leftarrow$ Gaussian
    Swap $\leftarrow$ AKF (when $\rho_k \leftarrow$ low)
  End If
End While

## 4. Performance evaluation

The performance of the OIAE-KF is done based on dynamic error models for measuring noise uncertainty (Rauh *et al*., 2013). The state variables are the position and velocity based on the acceleration. The system state is defined as the 2D coordinates ($x$-axis and $y$-axis) of the position and the velocity. Here $X$ is the state variable (Korbicz *et al*., 1994). The positions at the $x$-axis and the $y$-axis are denoted as $p_x$ and $p_y$, respectively. The velocities at the $x$-axis and $y$-axis are denoted as $v_x$ and $v_y$, respectively. Acceleration is the control force, and it is denoted as $a_x$ and $a_y$ at the $x$-axis and $y$-axis, respectively. $\Delta T$ is the time lag for one second (Thiemann *et al*., 2008),

$$x_k = \begin{bmatrix} p_{x_{(k-1)}} + v_{x_{(k-1)}}\Delta T + a_x \frac{1}{2}\Delta T^2 \\ p_{y_{(k-1)}} + v_{y_{(k-1)}}\Delta T + a_y \frac{1}{2}\Delta T^2 \\ v_{x_{(k-1)}} + a_x \Delta T \\ v_{y_{(k-1)}} + a_y \Delta T \end{bmatrix}. \tag{23}$$

Thus the state variables are preprocessed and submitted into the Kalman filter for estimation.

**4.1. Error model.** In the VANET environment the vehicles are subject to different noise signals. Noise is the error or an undesired random distribution of information (Langbein and Johnson, 1997). In this paper, three different noise types are used, such as stationary white noise, dynamic non-stationary noise, and correlated noise based on the environmental conditions (Kato *et al*., 2002). Noise1 is the combination of stationary white noises with zero mean, Noise2 is non-stationary white noise with time-varying variance, and Noise3 is correlated noise. Using different noise types helps prove the effectiveness of the proposed algorithm.

Stationary white noise occurs when vehicles travel on highways and in rural environments (Montillet *et al*., 2012). In contrast, non-stationary white noise occurs during traveling in harsh environments such as urban areas, cloudy or misty weather, and tunnels (Lee and Lim, 2013). In those situations, signals are affected and damaged. Many researchers in the VANET positioning context analyzed these noise types. Noise1 generally occurs under Line of sight (LoS) measurement conditions such as GNSS-based positioning in a blue sky environment (Lobo *et al*., 1995). Noise1 appears on the highway or in rural environments. Noise2 occurs when vehicles travel in harsh environments such as clouds and trees. Cloud water bodies could absorb the signals (Hubaux *et al*., 2004). Noise3 happens explicitly, such as in tunnels, under bridges, and downtown areas. The noise types are illustrated in Table 1, where MW is the moving window size. The moving window size is 120, and the sampling rate is 100 milliseconds.

**4.2. Simulation model.** This section describes the simulation setup and the simulation process, along with the results obtained from the simulation. The simulation starts with two phases regarding traffic simulation and network simulation. Simulation for Urban Mobility

(SUMO) generates road traffic, and Network Simulator 2 (NS2) runs network simulation.

MATLAB is used for computation (Le *et al.*, 2009). SUMO is a highly portable and open source software for traffic simulation, and it can manage a huge network. SUMO uses a microscopic view and a continuous mobility model. Due to microscopic traffic generation, each vehicle in the simulation is explicitly defined. Every vehicle is defined with a unique path and a unique identifier. In this paper, for effective simulation, each vehicle is fixed with source destination metrics (S/D metrics). This S/D metric holds the source and the destination address. A network file is attached to SUMO simulation for the road network. Network files create a realistic road map on which vehicles run.

In this paper, network files can be created manually by writing route codes, but this is a tough task to create complex networks, and this method is obsolete. For a realistic road map, we have to generate a network file acquired from the real world map in the form of `openstreemap.org`. Figure 3 shows the street map of Coimbatore city's OSM file. The map is downloaded as an OSM file, which is edited using the JOSM Java OpenStreetMap editor (Byrski *et al.*, 2019). The JOSM editor removes all the routes and makes the network file more elegant.

The next stage of editing the OSM file is to create the network file (i.e., convert the OSM file to a net file). For this, the SUMO simulator uses the netconvert command line. This command takes the .osm file as the input and delivers the .net file. Different SUMO scenarios have a .sumo.cfg file representing the corresponding network and the routing files along with the starting point and the end point of the simulation. .sumo.cfg is called the configuration file and is created by joining the network file and the route file. The SUMO traffic simulator is complete now, and the next step is to simulate the network. Figure 4 shows the network file running in SUMO.

The purpose of network simulation is to configure the computer network before it being implemented in the real world. In this paper, Network Simulator 2 is used for network simulation (Milanés *et al.*, 2013). The NS2 simulator compares different networks and makes the user understand the performance problems in the network. NS2 supports a huge simulation scenario where the number of nodes can be up to 20000. This factor makes the simulation results more naturalistic. To simulate the V2V (vehicle to vehicle) communication, the traffic simulator and the network simulator should run simultaneously (Chabir *et al.*, 2018). For this purpose, a TCP connection is used. The protocol used for TCP communication is TraCl (Traffic Control Interface) which provides bidirectional and coupled execution of the road traffic and network traffic. The nodes in NS2 simulation reflect the movement of vehicles in the SUMO

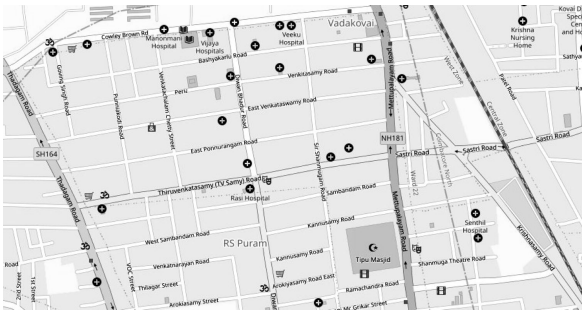


Fig. 3. Street map of the Coimbatore city (OSM file).

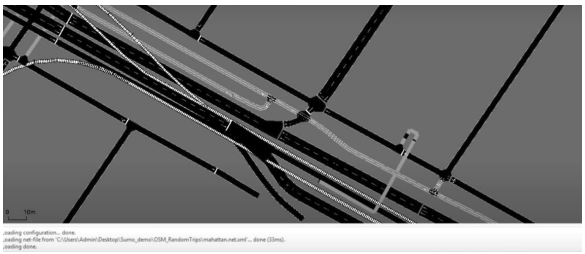

Fig. 4. Network file running in SUMO.

Table 1. Noise types with different errors.

| Noise type | $\mu$ | $\sigma$ |
|---|---|---|
| Noise1 | 0 mw | 10 mw |
| Noise2 | 0 mw | rand() mw |
| Noise3 | f(t) mw | f(t) mw |

traffic simulator (Xu *et al.*, 2004). A new file called launched.xml is created to access the routing file and the network file. NS2 simulation works based on the necessary parameters, which are modified per module. These parameters are used in the simulation in the form of the .ini file, which contains details about the network area size and the physical and application layers' information. The IEEE 802.11p protocol is used in the physical layer for adding wireless access in vehicular environment (WAVE). TraClMobility is the mobility model used in this simulation. Finally, after completing all the steps mentioned above, NS2 simulation is performed. The vehicle movement in the simulation is controlled by both traffic simulation and network simulation (Skog and Handel, 2009). The trace file stores all the information of the communication among the vehicles.

**4.2.1. Synchronization among simulators.** The MATLAB computational simulator executes algorithms which are in mathematical form. MATLAB supplies its results to the NS2 simulator for performance analysis. To ensure synchronization between all the three simulators, blocking characteristics of sockets are utilized. First, the

NS2 code reaches the configuration phase. This NS2 script is blocked at the beginning of the simulation. Now the traffic simulator is initiated. A set of vehicles goes into the communication network during a steady state. NS2 gathers their position information along with the time when the vehicles reached the network (Golestan *et al*., 2015). For every simulation, the sampling time is scheduled for getting the computational results from MATLAB (Xi and Cheng-dong, 2017). Figure 5 shows synchronization among the simulators.

Three-position prediction algorithms such as the basic Kalman filter, the IAE-KF, and the OIAE-KF have been implemented using MATLAB. Next Generation Simulation (NGSIM) trajectory dataset has been used to evaluate the proposed algorithm's effectiveness. NGSIM collects datasets that hold real-world vehicles' trajectories collected to know the driving behavior. NGSIM provides a commonly available and quality dataset that illustrates vehicles' kinematic information. It is an open-source dataset used to model the simulation part of VANET and compute the vehicle trajectory and driving behavior in VANET. The acceleration value is obtained from the speed over time of a vehicle. An exponential weight moving average method regulates velocity measurements. The direction of the vehicle is derived from the position change from the $x$-axis to the $y$-axis.

The NGSIM dataset is divided into two clusters such as lane changing and car flowing based on the driving behavior (Alam and Dempster, 2013). The K-means clustering method clusters driving behavior. Totally 50 vehicles are used for evaluation, and 25 vehicles are selected for each cluster. $K$ is the sample size selected from each cluster. As can be seen in the error models section, various environmental noise signals have been used to evaluate the effectiveness of the proposed OIAE-KF method. Three noise types were injected into vehicles' kinematic information to represent the dynamic noise effect. The noise injection technique is a widely used method to determine the robustness of the proposed technique. For position information and kinematic information, the sampling of 100 milliseconds is used for the positioning update rate. Table 2 shows the simulation of VANET with various parameters. Figure 6 shows network simulation using NS2.

**4.3. Performance metrics.** The effectiveness of the proposed method over the existing methods is evaluated using performance metrics. The divergence ratio, the root mean squared error (RMSE) and accuracy are the common performance metrics used in this paper to compare the proposed and existing Kalman filter algorithms.

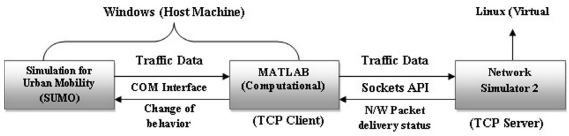


Fig. 5. Synchronization among simulators.

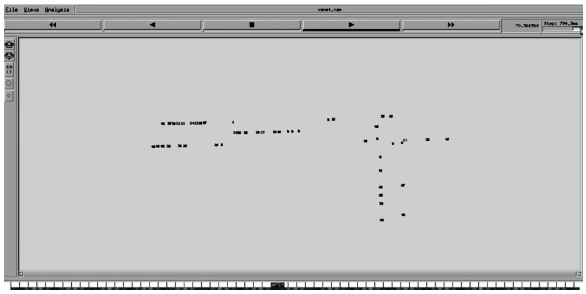

Fig. 6. Network simulation using NS2.

Table 2. Simulation parameters.

| Parameter | Value |
|---|---|
| Traffic simulator | SUMO |
| Network simulator | NS 2.34 |
| Map model | OSM (Coimbatore City) |
| Transport protocol | TCP, UDP |
| Routing protocol | DSR, DSDV, AODV |
| Number of vehicles | 50 |

Table 3. Divergence ratio.

| Type of noise | IAE-KF | OIAE-KF |
|---|---|---|
| Noise1 | 4% | 0% |
| Noise2 | 12% | 0% |
| Noise3 | 20% | 0% |

**4.3.1. Divergence ratio.** When the covariance matrix grows gradually, it indicates the presence of filter divergence. Table 3 shows a comparison of the divergence ratio among the proposed OIAE-KF and IAE-KF algorithms. As shown in the table, the OIAE-KF outperforms other algorithms in terms of the divergence ratio. The swapping decision between a DR and the AKF ensures that the theoretical covariance always remains in a positive definite state. In contrast, the IAE-KF is not stable for a dynamic environment such as VANET. Experiments suggest that the IAE-KF is sensitive to correlated noise signals and changes in the measurement noise distribution (ZhiWen *et al*., 2013). The proposed OIAE-KF method provides 0% of filter divergence.

**4.4. Root mean squared error.** The RMSE evaluates the error rate obtained by the filters. It is mostly used as

Table 4. Estimates and error analysis between the IAE-KF and the OIAE-KF for the position at the $x$-axis.

| Window size | Actual position ($x$-axis) [cm] | Estimated position using IAE-KF [cm] | Error in the IAE-KF [cm] | Estimated position using OIAE-KF [cm] | Error in the OIAE-KF [cm] |
|---|---|---|---|---|---|
| 15 | 117.48 | 91.44 | 26.04 | 114.34 | 3.14 |
| 30 | 567.24 | 438.2 | 129.04 | 559.65 | 7.59 |
| 60 | 1925.3 | 1723.19 | 202.11 | 1890.11 | 35.19 |
| 90 | 3915.44 | 3720.56 | 194.88 | 3895.42 | 20.02 |
| 100 | 5828.31 | 5641.26 | 187.05 | 5781.13 | 47.18 |
| 120 | 7296.05 | 6956.7 | 339.35 | 7191.67 | 104.38 |

Table 5. Estimates and error analysis between the IAE-KF and the OIAE-KF for the position at the $y$-axis.

| Window size | Actual position ($y$-axis) [cm] | Estimated position using IAE-KF [cm] | Error in the IAE-KF [cm] | Estimated position using OIAE-KF [cm] | Error in the OIAE-KF [cm] |
|---|---|---|---|---|---|
| 15 | 271.43 | 150.34 | 121.09 | 253.25 | 18.18 |
| 30 | 628.57 | 529.28 | 99.29 | 610.11 | 18.46 |
| 60 | 2753.45 | 2210.72 | 542.73 | 2739.25 | 14.2 |
| 90 | 4934.81 | 4521.98 | 412.83 | 4928.17 | 6.64 |
| 100 | 6614.56 | 6017.83 | 596.73 | 6598.61 | 15.95 |
| 120 | 8251.99 | 7864.47 | 387.52 | 8210.48 | 41.51 |

a performance metric to determine the effectiveness of an algorithm. It is defined as

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{x}_i - x_i)^2}. \tag{24}$$

Equation (24) illustrates the formulation of the RMSE, where $n$ is the total number of terms for which the RMSE is to be calculated, $x_i$ is the observed value, and $\hat{x}_i$ is the predicted value. Table 8 shows the RMSE for both the IAE-KF and OIAE-KF positioning algorithms used for comparison.

**4.5. Accuracy.** Accuracy is another metric that delivers real performance of the proposed method (Golestan *et al.*, 2015). The technique aims to attain an accurate measurement of the vehicle position. The positioning accuracy is

$$\text{Acc}_p = \text{Pos}_{\text{true}} - \text{Pos}_{\text{est}}. \tag{25}$$

where $\text{Pos}_{\text{true}}$ denotes the actual position and $\text{Pos}_{\text{est}}$ denotes the estimated position. Accurate vehicle position estimates are shown in Tables 4–7. The divergence ratio and the RMSE lead to the measurement of accuracy in prediction. As can be seen earlier, accuracy is the primary factor in vehicle positioning. Figures 7–10 show the graph representing the position accuracy between the proposed methods and the existing one.

## 5. Conclusion

Several approaches have been proposed to vehicle positioning in VANET. None of the techniques accomplish an accurate vehicle position. From the observation, the existing methods work well in a controlled environment (Xi and Cheng-dong, 2017), which is explained in Section 1. In most of the existing methodologies, noise covariance is assumed to be stationary, but it may vary based on the environmental conditions (Mrugalski, 2013). This noise variation factor reduces the accuracy of position estimation.

The Kalman filter is a common method used for predicting the current position of a vehicle. But the filter is suitable only for stationary white noise. Therefore, the problem statement is found in the basic Kalman filter and the innovation based adaptive estimation Kalman filter in Section 2.

In this paper, the optimal innovation based adaptive estimation Kalman filter (OIAE-KF) is introduced. The proposed algorithm offers an alternative solution to positioning problems in the existing IAE-KF. In most known positioning algorithms, measurement noise covariance is kept unknown. Only the process noise covariance has been changed for estimation. Nevertheless, the received signals contain noise other than white noise. In this paper, dynamic noise signals were injected into the VANET nodes to create the dynamic noise environment during simulation (Yim *et al.*, 2011).

The proposed OIAE-KF considers the innovation sequence. By applying the innovation property to the

Table 6. Estimates and error analysis between the IAE-KF and the OIAE-KF for the velocity along the $x$-axis.

| Window size | Actual velocity ($x$-axis) [cm/s] | Estimated velocity using the IAE-KF [cm/s] | Error in the IAE-KF [cm/s] | Estimated velocity using the OIAE-KF [cm/s] | Error in the OIAE-KF [cm/s] |
|---|---|---|---|---|---|
| 15 | 37.5 | 47.19 | – 9.69 | 35.9 | 1.6 |
| 30 | 96.3 | 117.68 | – 21.38 | 95.11 | 1.19 |
| 60 | 174.81 | 269.42 | – 94.61 | 169.54 | 5.27 |
| 90 | 268.83 | 311.87 | – 43.04 | 261.43 | 7.4 |
| 100 | 391.63 | 469.21 | – 77.58 | 389.34 | 2.29 |
| 120 | 504.97 | 619.72 | – 114.75 | 501.49 | 3.48 |

Table 7. Estimates and error analysis between the IAE-KF and the OIAE-KF for the velocity along the $y$-axis.

| Window size | Actual velocity ($y$-axis) [cm/s] | Estimated velocity using the IAE-KF [cm/s] | Error in the IAE-KF [cm/s] | Estimated velocity using the OIAE-KF [cm/s] | Error in the OIAE-KF [cm/s] |
|---|---|---|---|---|---|
| 15 | 41.21 | 35.9 | 5.31 | 42.47 | – 1.26 |
| 30 | 90.33 | 99.72 | – 9.39 | 91.83 | – 1.5 |
| 60 | 197.62 | 235.87 | – 38.25 | 193.45 | 4.17 |
| 90 | 259.98 | 293.81 | – 33.83 | 251.84 | 8.14 |
| 100 | 428.5 | 459.65 | – 31.15 | 419.06 | 9.44 |
| 120 | 683.02 | 702.28 | – 19.26 | 679.12 | 3.9 |

Table 8. RMSE between the IAE-KF and the OIAE-KF.

| Algorithm | Position ($x$-axis) | Position ($y$-axis) | Velocity ($x$-axis) | Velocity ($y$-axis) |
|---|---|---|---|---|
| IAE-KF | 202.6 | 407.38 | 71.34 | 26.02 |
| OIAE-KF | 49.71 | 21.96 | 4.16 | 5.65 |

innovation sequence, the divergence of the error is kept in control. When deviations occur, or a non-Gaussian curve, the estimation swaps to dead reckoning and the measurement noise covariance increases with time until the noise becomes stable white noise (Ghaleb *et al.*, 2016). This idea increases the Kalman filter's accuracy level and maintains an optimal level, which improves the vehicle's reliability. This is explained in Section 3. The performance of the proposed OIAE-KF is compared with the existing IAE-KF. The simulation results prove that OIAE-KF works better in dynamic noise environments, which is described in Section 4.

A performance comparison is made between the IAE-KF and the OIAE-KF with the actual precise position of the vehicle. The results are exposed as tables with the following items: the window size, the position estimated using the IAE-KF, the error in the IAE-KF, the position estimated using the OIAE-KF, and the error in the OIAE-KF. The estimated position at the $x$-axis and the error between the IAE-KF and the OIAE-KF are shown in Table 4. From these results, the IAE-KF could estimate an accurate position with an accuracy of 89.32%. But the proposed OIAE-KF could reach an accuracy of 98.89%.

The positioning difference between the actual position measured along the $x$-axis and the IAE-KF is about 179.74 cm, while in the OIAE-KF, positioning differs only by 36.25 cm. The position at the $y$-axis and the error in positioning are exposed in Table 5. The IAE-KF estimates the real position with an accuracy of 90.79% while the OIAE-KF reaches the real position up to 99.50%. Table 6 illustrates the velocity at the $x$-axis, and that an error occurred. The IAE-KF overestimates the actual velocity of about 60.17 cm/s. The velocity measured by the IAE-KF attains an accuracy of 75.50%. In contrast, the proposed Kalman filter offers an accuracy of about 98.55%. The OIAE-KF makes an error only of 3.53 cm/s. Velocity measurements at the $y$-axis are shown in Table 7. On verifying Table 7, IAE-KF meets the accuracy level of 92.55%, and the OIAE-KF could attain 98.65% accuracy. The IAE-KF produces an error of about 21.09 cm/s. The OIAE-KF causes error only for 3.82 cm/s. From these analytical results, it is confirmed that the proposed method outperforms the existing one. Figures 7–10 shows graphs representing the comparison between the IAE-KF and OIAE-KF positioning with the actual position. The graphs show that the proposed method

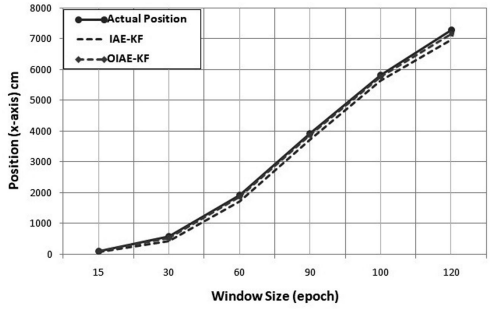


Fig. 7. Estimation analysis between the IAE-KF and the OIAE-KF for the position at the $x$-axis.

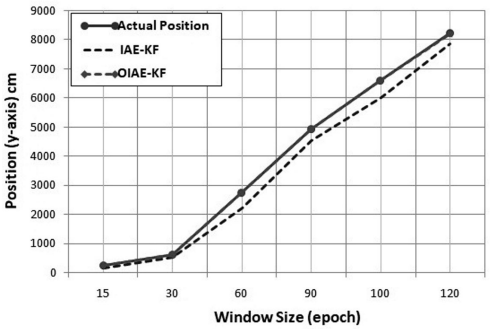


Fig. 8. Estimation analysis between the IAE-KF and the OIAE-KF for the position at the $y$-axis.

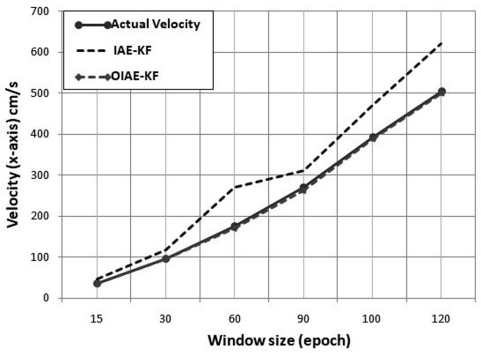


Fig. 9. Estimation analysis between the IAE-KF and the OIAE-KF for the velocity along the $x$-axis.

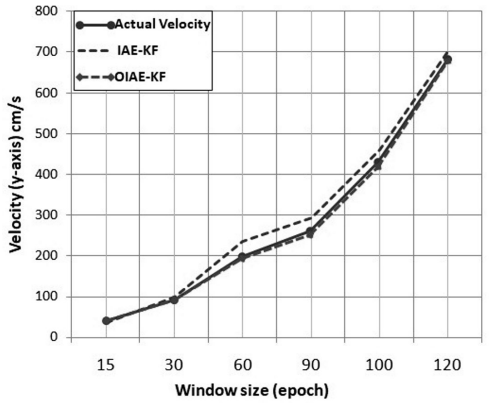


Fig. 10. Estimation analysis between the IAE-KF and the OIAE-KF for the velocity along the $y$-axis.

estimates the position more accurately than the existing one. In conclusion, the proposed OIAE-KF performs better than the existing IAE-KF method.

## References

Al-Sultan, S., Al-Doori, M.M., Al-Bayatti, A.H. and Zedan, H. (2014). A comprehensive survey on vehicular ad hoc network, *Journal of Network and Computer Applications* **37**: 380–392.

Alam, N. and Dempster, A.G. (2013). Cooperative positioning for vehicular networks: Facts and future, *IEEE Transactions on Intelligent Transportation Systems* **14**(4): 1708–1717.

Amadeo, M., Campolo, C. and Molinaro, A. (2012). Enhancing IEEE 802.11 p/wave to provide infotainment applications in VANETs, *Ad Hoc Networks* **10**(2): 253–269.

Bar-Shalom, Y., Li, X.R. and Kirubarajan, T. (2004). *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*, John Wiley, Hoboken.

Bidyuk, P., Podladchikov, V.D. and Podladchikova, I. (1993). Analytical study of the Kalman filter for stationary dynamic systems, *Applied Mathematics and Computer Science* **3**(2): 313–328.

Byrski, W., Drapała, M. and Byrski, J. (2019). An adaptive identification method based on the modulating functions technique and exact state observers for modeling and simulation of a nonlinear MISO glass melting process, *International Journal of Applied Mathematics and Computer Science* **29**(4): 739–757, DOI: 10.2478/amcs-2019-0055.

Chabir, K., Rhouma, T., Keller, J.Y. and Sauter, D. (2018). State filtering for networked control systems subject to switching disturbances, *International Journal of Applied Mathematics and Computer Science* **28**(3): 473–482, DOI: 10.2478/amcs-2018-0036.

Chatterjee, A. and Matsuno, F. (2007). A neuro-fuzzy assisted extended Kalman filter-based approach for simultaneous localization and mapping (SLAM) problems, *IEEE Transactions on Fuzzy Systems* **15**(5): 984–997.

Chen, X., Li, L. and Zhang, Y. (2010). A Markov model for headway/spacing distribution of road traffic, *IEEE Transactions on Intelligent Transportation Systems* **11**(4): 773–785.

Ding, W., Wang, J., Rizos, C. and Kinlyside, D. (2007). Improving adaptive Kalman estimation in GPS/INS integration, *The Journal of Navigation* **60**(3): 517.

ETSI (2010). *TS 102 731 v1. 1.1—Intelligent Transport Systems (ITS); Security; Security Services and Architecture*, European Telecommunications Standards Institute, Sophia-Antipolis.

Gao, W., Li, J., Zhang, Y., Wang, G. and Sun, X. (2017). Improved innovation-based adaptive estimation for measurement noise uncertainty in SINS/GNSS integration system, *2017 Forum on Cooperative Positioning and Service (CPGPS), Harbin, China*, pp. 22–28.

Ghaleb, F.A., Zainal, A. and Rassam, M.A. (2016). Mobility information estimation algorithm using Kalman-filter for vehicular ad hoc networks, *International Journal of Information and Computer Security* **8**(3): 221–240.

Golestan, K., Sattar, F., Karray, F., Kamel, M. and Seifzadeh, S. (2015). Localization in vehicular ad hoc networks using data fusion and V2V communication, *Computer Communications* **71**: 61–72.

Havangi, R., Nekoui, M.A. and Teshnehlab, M. (2010). Adaptive neuro-fuzzy extended Kalman filtering for robot localization, *Proceedings of the 14th International Power Electronics and Motion Control Conference (EPE-PEMC), Ohrid, Macedonia*, pp. T5–130.

Hide, C., Moore, T. and Smith, M. (2004). Adaptive Kalman filtering algorithms for integrating GPS and low cost INS, *PLANS 2004: Position Location and Navigation Symposium, Monterey, USA*, pp. 227–233.

Hou, Y., Edara, P. and Sun, C. (2013). Modeling mandatory lane changing using Bayes classifier and decision trees, *IEEE Transactions on Intelligent Transportation Systems* **15**(2): 647–655.

Hu, I. (2017). Extended Kalman filter approach for reducing traffic congestion in VANET, *International Journal of Engineering Development and Research* **5**(2): 204–210.

Huang, C.-M. and Lin, S.-Y. (2014). Cooperative vehicle collision warning system using the vector-based approach with dedicated short range communication data transmission, *IET Intelligent Transport Systems* **8**(2): 124–134.

Hubaux, J.-P., Capkun, S. and Luo, J. (2004). The security and privacy of smart vehicles, *IEEE Security & Privacy* **2**(3): 49–55.

Jiancheng, F. and Sheng, Y. (2011). Study on innovation adaptive EKF for in-flight alignment of airborne POS, *IEEE Transactions on Instrumentation and Measurement* **60**(4): 1378–1388.

Kalman, R.E. (1960). A new approach to linear filtering and prediction problems, *Transactions of the ASME—Journal of Basic Engineering* **82**(D): 35–45.

Kato, S., Tsugawa, S., Tokuda, K., Matsui, T. and Fujii, H. (2002). Vehicle control algorithms for cooperative driving with automated vehicles and intervehicle communications, *IEEE Transactions on Intelligent Transportation Systems* **3**(3): 155–161.

Khattab, A., Fahmy, Y.A. and Abdel Wahab, A. (2015). High accuracy GPS-FREE vehicle localization framework via an ins-assisted single RSU, *International Journal of Distributed Sensor Networks* **11**(5): 795036.

Korbicz, J., Podladchikov, V. and Bidyuk, P. (1994). Integration of multisensor measurements using modified Kalman filter, *Applied Mathematics and Computer Science* **4**(1): 39–51.

Langbein, J. and Johnson, H. (1997). Correlated errors in geodetic time series: Implications for time-dependent deformation, *Journal of Geophysical Research: Solid Earth* **102**(B1): 591–603.

Le, L., Festag, A., Baldessari, R. and Zhang, W. (2009). Vehicular wireless short-range communication for improving intersection safety, *IEEE Communications Magazine* **47**(11): 104–110.

Lee, S. and Lim, A. (2013). An empirical study on ad hoc performance of DSRC and wi-fi vehicular communications, *International Journal of Distributed Sensor Networks* **9**(11): 482695.

Li, J., Song, N., Yang, G. and Jiang, R. (2016). Fuzzy adaptive strong tracking scaled unscented Kalman filter for initial alignment of large misalignment angles, *Review of Scientific Instruments* **87**(7): 075118.

Li, L., Wen, D., Zheng, N.-N. and Shen, L.-C. (2011). Cognitive cars: A new frontier for ADAS research, *IEEE Transactions on Intelligent Transportation Systems* **13**(1): 395–407.

Li, W., Wang, Z., Wei, G., Ma, L., Hu, J. and Ding, D. (2015). A survey on multisensor fusion and consensus filtering for sensor networks, *Discrete Dynamics in Nature and Society* **2015**, Article ID: 683701, DOI: 10.1155/2015/683701.

Liu, M. and Xiong, F. (2011). A fuzzy adaptive GPS/INS integrated navigation algorithm, *Procedia Engineering* **15**: 660–664.

Loebis, D., Sutton, R., Chudley, J. and Naeem, W. (2004). Adaptive tuning of a Kalman filter via fuzzy logic for an intelligent AUV navigation system, *Control Engineering Practice* **12**(12): 1531–1539.

Lobo, F., Grael, D., Oliveira, H., Villas, L., Almehmadi, A. and El-Khatib, K. (2019). Cooperative localization improvement using distance information in vehicular ad hoc networks, *Sensors* **19**(23): 5231.

Lobo, J., Lucas, P., Dias, J. and De Almeida, A.T. (1995). Inertial navigation system for mobile land vehicles, *Proceedings of the IEEE International Symposium on Industrial Electronics, Athens, Greece,* Vol. 2, pp. 843–848.

Mehra, R. (1970). On the identification of variances and adaptive Kalman filtering, *IEEE Transactions on Automatic Control* **15**(2): 175–184.

Milanés, V., Shladover, S.E., Spring, J., Nowakowski, C., Kawazoe, H. and Nakamura, M. (2013). Cooperative adaptive cruise control in real traffic situations, *IEEE Transactions on Intelligent Transportation Systems* **15**(1): 296–305.

Mohamed, A. and Schwarz, K. (1999). Adaptive Kalman filtering for INS/GPS, *Journal of Geodesy* **73**(4): 193–203.

Montillet, J.-P., Tregoning, P., McClusky, S. and Yu, K. (2012). Extracting white noise statistics in GPS coordinate time series, *IEEE Geoscience and Remote Sensing Letters* **10**(3): 563–567.

Mrugalski, M. (2013). An unscented Kalman filter in designing dynamic GMDH neural networks for robust fault detection, *International Journal of Applied Mathematics and Computer Science* **23**(1): 157–169, DOI: 10.2478/amcs-2013-0013.

Parker, R. and Valaee, S. (2007). Vehicular node localization using received-signal-strength indicator, *IEEE Transactions on Vehicular Technology* **56**(6): 3371–3380.

Rauh, A., Butt, S.S. and Aschemann, H. (2013). Nonlinear state observers and extended kalman filters for battery systems, *International Journal of Applied Mathematics and Computer Science* **23**(3): 539–556, DOI: 10.2478/amcs-2013-0041.

Rezaei, S. and Sengupta, R. (2007). Kalman filter-based integration of DGPS and vehicle sensors for localization, *IEEE Transactions on Control Systems Technology* **15**(6): 1080–1088.

Sarkka, S. and Nummenmaa, A. (2009). Recursive noise adaptive Kalman filtering by variational Bayesian approximations, *IEEE Transactions on Automatic Control* **54**(3): 596–600.

Sierociuk, D. and Dzieliński, A. (2006). Fractional Kalman filter algorithm for the states, parameters and order of fractional system estimation, *International Journal of Applied Mathematics and Computer Science* **16**(1): 129–140.

Skog, I. and Handel, P. (2009). In-car positioning and navigation technologies—A survey, *IEEE Transactions on Intelligent Transportation Systems* **10**(1): 4–21.

Thiemann, C., Treiber, M. and Kesting, A. (2008). Estimating acceleration and lane-changing dynamics from next generation simulation trajectory data, *Transportation Research Record* **2088**(1): 90–101.

Wang, H., Fu, G., Li, J., Yan, Z. and Bian, X. (2013). An adaptive UKF based SLAM method for unmanned underwater vehicle, *Mathematical Problems in Engineering* **2013**, Article ID: 605981.

Wang, Z., Woodward, W.A. and Gray, H.L. (2009). The application of the Kalman filter to nonstationary time series through time deformation, *Journal of Time Series Analysis* **30**(5): 559–574.

Woo, R., Yang, E.-J. and Seo, D.-W. (2019). A fuzzy-innovation-based adaptive Kalman filter for enhanced vehicle positioning in dense urban environments, *Sensors* **19**(5): 1142.

Xi, C. and Cheng-dong, X. (2017). Performance analysis of multi-constellation GNSS in urban canyons based on fuzzy comprehensive evaluation, *29th Chinese Control and Decision Conference (CCDC), Chongqing, China*, pp. 3040–3045.

Xu, Q., Mak, T., Ko, J. and Sengupta, R. (2004). Vehicle-to-vehicle safety messaging in DSRC, *Proceedings of the 1st ACM International Workshop on Vehicular AD HOC Networks, Philadelphia, USA*, pp. 19–28.

Yim, J., Joo, J. and Park, C. (2011). A Kalman filter updating method for the indoor moving object database, *Expert Systems with Applications* **38**(12): 15075–15083.

Yu, M.-J. (2012). INS/GPS integration system using adaptive filter for estimating measurement noise variance, *IEEE Transactions on Aerospace and Electronic Systems* **48**(2): 1786–1792.

ZhiWen, X., XiaoPing, H. and JunXiang, L. (2013). Robust innovation-based adaptive Kalman filter for INS/GPS land navigation, *2013 Chinese Automation Congress, Changsha, China*, pp. 374–379.

**S. Sumithra** is a PhD student at Bharathiar University, Tamil Nadu, India. She received her MSc degree in information technology and her MPhil degree in computer science from Bharathiar University in 2015 and 2017, respectively. She started her research in 2017 in the area of advanced networking. Her current interests include intelligent transport systems, deep learning, artificial intelligence and vehicular ad hoc networks. She is also interested in 5G wireless solutions and optical networking.

**R. Vadivel** is an assistant professor in the Department of Information Technology, Bharathiar University, Tamil Nadu, India. He received his PhD degree in computer science from Manonmaniam Sundaranar University in 2013. He obtained his BE degree in computer science and engineering from Periyar University in 2002, and his ME degree in computer science and engineering from Annamalai University in 2007. He has published over 40 journal papers and over 30 conference papers both at a national and an international level. His areas of interest include computer networks, network security and information security.

Received: 30 May 2020
Revised: 3 September 2020
Re-revised: 20 October 2020
Accepted: 25 November 2020