Performance Enhancement of MEMS-Based INS/GPS Integration for Low-Cost Navigation Applications

Aboelmagd Noureldin, *Senior Member, IEEE*, Tashfeen B. Karamat, Mark D. Eberts, and Ahmed El-Shafie

***Abstract*—The relatively high cost of inertial navigation systems (INSs) has been preventing their integration with global positioning systems (GPSs) for land-vehicle applications. Inertial sensors based on microelectromechanical system (MEMS) technology have recently become commercially available at lower costs. These relatively lower cost inertial sensors have the potential to allow the development of an affordable GPS-aided INS (INS/GPS) vehicular navigation system. While MEMS-based INS is inherently immune to signal jamming, spoofing, and blockage vulnerabilities (as opposed to GPS), the performance of MEMS-based gyroscopes and accelerometers is significantly affected by complex error characteristics that are stochastic in nature. To improve the overall performance of MEMS-based INS/GPS, this paper proposes the following two-tier approach at different levels: 1) improving the stochastic modeling of MEMS-based inertial sensor errors using autoregressive processes at the raw measurement level and 2) enhancing the positioning accuracy during GPS outages by nonlinear modeling of INS position errors at the information fusion level using neuro-fuzzy (NF) modules, which are augmented in the Kalman filtering INS/GPS integration. Experimental road tests involving a MEMS-based INS were performed, which validated the efficacy of the proposed methods on several trajectories.**

***Index Terms*—Global positioning system (GPS), inertial navigation system (INS), Kalman filter (KF), microelectromechanical system (MEMS), neuro-fuzzy (NF) systems, wavelet.**

I. Introduction

GLOBAL positioning systems (GPSs) provide positioning information with a consistent and acceptable accuracy when there is a direct line of sight to four or more satellites [1], [2]. However, it may suffer from outages, jamming, and multipath effects in urban canyons. Inertial navigation systems (INSs), on the other hand, are autonomous systems that are immune to external interference, but their accuracy deteriorates in the long term due to sensors' bias error drift, scale factor instability, and misalignment [3]–[5]. Some of the INS errors are random in nature and cannot be removed by calibration. Integrating both INS and GPS provides superior performance than any of them operating alone. For instance, GPS-derived positions have approximately white noise characteristics with bounded errors and can therefore be used to update INS and improve its long-term accuracy [3], [4]. On the other hand, INS provides positioning information during GPS outages, assists GPS signal reacquisition after an outage, and reduces the search domain required for detecting and correcting GPS cycle slips [1], [5]. INS is also capable of providing positioning, velocity, and attitude information at higher data rates than GPS. Kalman filtering is traditionally used to optimally fuse the position and velocity information from both INS and GPS [6]–[12].

A. Research Objectives

The ultimate objective of this paper is to provide a reliable MEMS-based GPS-aided INS (INS/GPS) positioning module that enhances the overall system accuracy and enables robust and accurate positioning information during GPS outages. The typical applications of our research are low-cost navigation solutions pertaining mainly to the commercial land vehicles, unmanned autonomous vehicles, personal location, and navigation. This paper will focus on two major issues, namely 1) improving the stochastic modeling of MEMS inertial measurement unit (IMU) sensor's bias drift errors using autoregressive (AR) models and 2) augmenting the KF-based integration algorithm with neuro-fuzzy (NF) modules for nonlinear modeling of INS position errors to achieve a reliable positioning accuracy.

B. MEMS Inertial Sensors

MEMS accelerometers have been manufactured for many years, serving the need of consumer applications. The physical mechanisms underlying MEMS accelerometers include capacitive, piezoresistive, electromagnetic, piezoelectric, ferroelectric, optical, and tunneling [13]. The most successful types are based on capacitive transduction due to the simplicity of the sensor element, small size, low power consumption, and stability over a wide temperature range. All MEMS gyroscopes

Manuscript received August 9, 2007; revised February 29, 2008 and April 16, 2008. First published May 23, 2008; current version published March 17, 2009. This work was supported in part by the Geomatics for Informed Decision (GEOIDE) Network Centers of Excellence, Natural Sciences and Engineering Research Council, through research grants, and in part by the Canada Foundation for Innovation, the Ontario Innovation Trust, and the Royal Military College of Canada. The review of this paper was coordinated by Dr. K. T. Wong.

A. Noureldin is with the Department of Electrical and Computer Engineering, Queen's University, Kingston, ON K7L 3N6, Canada, and the Department of Electrical and Computer Engineering, Royal Military College of Canada, Kingston, ON K7K 7B4, Canada.

T. B. Karamat is with the Royal Military College of Canada, Kingston, ON K7K 7B4, Canada.

M. D. Eberts is with the Aerospace and Telecommunications Engineering Support Squadron, Trenton, ON K0K 3W0, Canada.

A. El-Shafie is with the Smart Engineering System, Department of Civil and Structural Engineering, University Kebangsaan Malaysia, Bangi 43600, Malaysia.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2008.926076

take advantage of the Coriolis effect. Most MEMS gyroscopes fall into the categories of tuning fork gyros, oscillating wheels, Foucault pendulums, and wine glass resonators [14]. When rotated, the Coriolis force creates an orthogonal vibration that can be sensed by a variety of mechanisms.

C. Inertial Sensor Errors

Errors in INS can result from the following two sources: 1) the sensor itself and 2) the numerical integration process inherent in the INS mechanization [14], [15]. Inertial sensors have both deterministic and random errors. The deterministic errors can be obtained using calibration procedures and then removed from the raw measurements, but the random errors must be modeled stochastically to extenuate their deleterious effect on the positioning accuracy. The random inertial sensor errors are usually modeled as a first-order Gauss–Markov (GM) process [16], which may not always be suitable for MEMS-based inertial sensors.

D. INS/GPS Integration

To overcome the disadvantages associated with the stand-alone operation of GPS and INS, the two systems are often paired together in a complimentary fashion so that their drawbacks are minimized or eliminated. The INS/GPS data fusion is commonly performed using KF [7], [8], [10], [17]. This method requires a dynamic model of both INS and GPS errors, a stochastic model of the inertial sensor errors, and *a priori* information about the covariances of the data provided by both systems [8]. Since GPS has a relatively consistent long-term accuracy, it is used to update both INS position and velocity components and, thus, prevents the long-term growth of their errors. On the other hand, the accurate short-term information provided by the INS is used to overcome GPS cycle slips and clock biases. Should a GPS outage occur, KF operates in prediction mode, correcting the INS information based on the system error model. KF has widely been used for data fusion and is considered as the benchmark for INS/GPS integration [5], [18]–[21]. There are, however, some inadequacies, including the following: 1) the necessity of accurate stochastic modeling, which may not be accurate enough in case of low-cost MEMS sensors; 2) the requirement for *a priori* information of the system and measurement covariance matrices for each new sensor; 3) the relatively poor accuracy during long GPS outages; 4) the weak observability of some of the error states that may lead to unstable estimates of other error states [22]; 5) the necessity to tune the parameters of the stochastic model and the *a priori* information for each new sensor system; and 6) the divergence that results from approximations during any linearization process and system mismodeling.

E. Artificial Intelligence (AI)-Based INS/GPS Integration

The limitations of KF have motivated researchers to investigate alternative methods of INS/GPS integration, which are predominantly based on AI. Recently, algorithms based on multilayer perception (MLP) neural networks have been suggested and applied to different types and grades of INS [23]. It was shown that a position and velocity update architecture (PVUA) that utilizes two MLP networks could process the INS azimuth and velocity to provide the position components along both the east and north directions [24]. The parameters of the MLP networks were adapted using GPS position and velocity updates. However, the PVUA system did not provide an update scheme for the vehicle altitude. In addition, the neural network model of PVUA dealt with the INS position components instead of their errors. Thus, no information about the accuracy of this system could be delivered during the navigation mission. Furthermore, in addition to the fact that no sensitivity analysis was provided for the effect of changing the internal structure (number of hidden layers and number of neurons in each layer) of the MLP networks on the system performance, the real-time implementation and the accuracy of the system during this mode of operations were not addressed.

In an attempt to design a model-free module that operates similarly to KF but without the need for dynamic or stochastic models of the INS, the P–δP model was proposed [24]. It was initially suggested to use an MLP network for each position component that processes the INS position (P) at the input and provides the corresponding INS position error (δP) at the output. The suggested method updated the three P–δP networks that utilize GPS position information. Although information about the accuracy achieved during the navigation mission became available, the internal structure of each of the MLP networks had to change until the best performance was realized. In addition, the issue of real-time implementation was not considered.

The P–δP architecture was improved by using radial basis function (RBF) neural networks instead of MLP networks [25]. RBF networks can be utilized without identifying the number of neurons in its hidden layer, as they are dynamically generated during the training procedure to achieve the desired performance [26]. One major limitation of this method was the use of all INS and GPS data prior to a GPS outage to train the RBF network. In addition, the real-time implementation and the factors that affect the performance of the system during this mode of operation have not been addressed in the online RBF-based P–δP module.

Recently, the fuzzy system was employed, utilizing an adaptive NF inference system (ANFIS) to provide an ANFIS-based P–δP module for mobile multisensor system integration [27], [28]. The ANFIS-based module was designed to work in real time to fuse INS and GPS position data [29]. However, this method showed a very limited success when applied to a MEMS-based INS/GPS navigation system due to the high noise level and bias instability of MEMS inertial sensors. Lately, Semeniuk and Noureldin [30] has suggested the AI-based segmented forward predictor (ASFP) that processed segments of INS and GPS position and velocity data using the RBF neural network to provide prediction of the INS errors. Although effective for both tactical and navigational grade INS, one limitation of the ASFP technique was the virtual extension of GPS outages due to the nature of GPS and INS data segmentation. In addition, this technique assumes that the INS error pattern is the same for two consecutive segments.

Moreover, the ASFP method exhibited inadequate performance when applied to MEMS-based INS/GPS.

Therefore, in this paper, we propose to augment KF with AI to develop an INS/GPS integration module that is particularly suitable for MEMS-based inertial sensors. The suggested system operates at two different levels. The first level capitalizes on improved stochastic modeling of inertial sensor errors at the raw measurement level in a loosely coupled INS/GPS integration through KF. At the second level, an AI module is employed to predict INS position error during GPS outages.

Owing to its effectiveness in dealing with uncertainty, imprecision, and vagueness in the input data in dynamic environments, the fuzzy system was used to provide an ANFIS-based P–δP module for mobile multisensor system integration. The ANFIS-based module designed in this study works in real time to fuse MEMS INS (after being processed by KF) and GPS position data, estimates the residual INS position errors, and enhances the INS positioning accuracy. Moreover, the same module predicts the vehicle position during GPS outages based on processing INS position components.

II. Methodology

A. Inertial Sensor Prefiltering

Inertial sensors always contain a certain amount of measurement noise. On the sensor level, this noise is usually separated into long-term (low-frequency) and short-term (high-frequency) noise [31]. The high-frequency component consists mainly of white noise and short-term sensor errors, whereas the low-frequency components consist of correlated long-term noise, white noise, and vehicle motion dynamics. Wavelet denoising is frequently used to process inertial sensors measurements to improve the signal-to-noise ratio (SNR) and is considered to be the common method for denoising MEMS-based inertial sensors [32].

In this study, wavelet decomposition is performed for a number of decomposition levels that are chosen based on the sampling frequency. Since most of the motion dynamics of land vehicles exist between 0 and 5 Hz [33]–[35], we have chosen the number of decomposition levels so that the decomposition process stops when this frequency band is reached. For example, when acquiring the inertial sensor measurements at a 20-Hz data rate, only one level of decomposition was used, whereas at a 200-Hz data rate, we continued the decomposition process up to the fourth level. To provide a near-real-time solution, each of the inertial sensor measurements is divided into windows of 400 samples, and each window is processed by wavelet denoising procedure separately. This denoising process is performed using the MATLAB Wavelet Toolbox using the "db5" wavelet function, the "rigrsure" thresholding criterion, and the soft thresholding method. Details about wavelet denoising, decomposition, and reconstruction procedures, thresholding criteria, and wavelet functions can be found in [23] and [36]–[40].

B. Stochastic Modeling of Inertial Sensor Errors

1) GM Model: In most KF implementations for INS/GPS, the inertial sensor error model is based on a first-order GM model with a decaying exponential autocorrelation sequence, as shown in Fig. 1. This model is usually proposed and adopted successfully for navigational and tactical grade inertial sensors [7]. For MEMS inertial sensors, the assumption that the sensor random errors follow the stochastic nature of a first-order GM model is not always valid. Using a stationary run of more than 4 h, the autocorrelation sequences of two of the MEMS-grade Crossbow IMU sensors were calculated as shown in Figs. 2 and Fig. 3. By comparing these figures, it can be determined that the random errors associated with these MEMS inertial sensors are different from that of a first-order GM process.

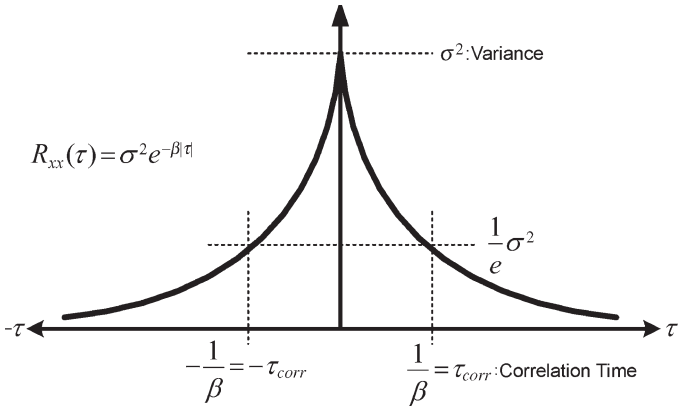

Fig. 1. Autocorrelation sequence of a first-order GM process.

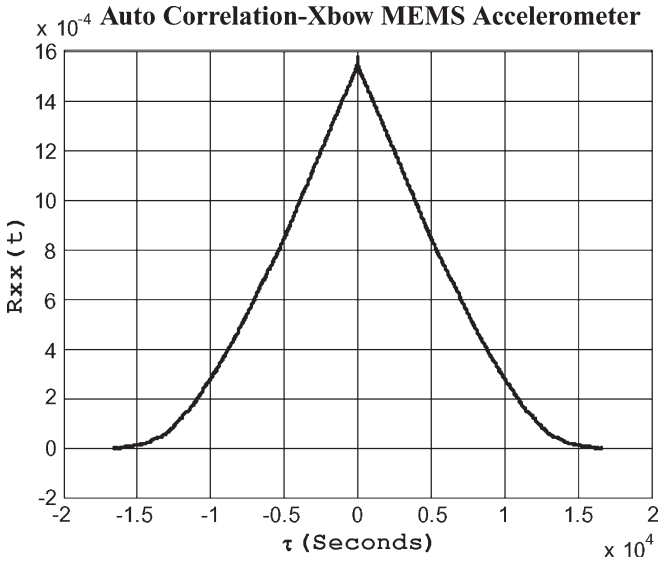

Fig. 2. Autocorrelation sequence of a Crossbow MEMS accelerometer.

2) GM Model Parameters: First-order GM model for an inertial sensor error is given as [16], [41]

$$\dot{x}(t) = -\beta x(t) + \sqrt{2\beta\sigma^2}w(t) \tag{1}$$

where β is the reciprocal of correlation time, and σ^2 is the variance of the system noise $w(t)$. In the discrete-time domain, (1) is written as follows [7], [41]:

$$x_k = (I - \beta\Delta t)x_{k-1} + \sqrt{2\beta\sigma^2}w_k\Delta t \tag{2}$$

where Δt is the sampling interval.

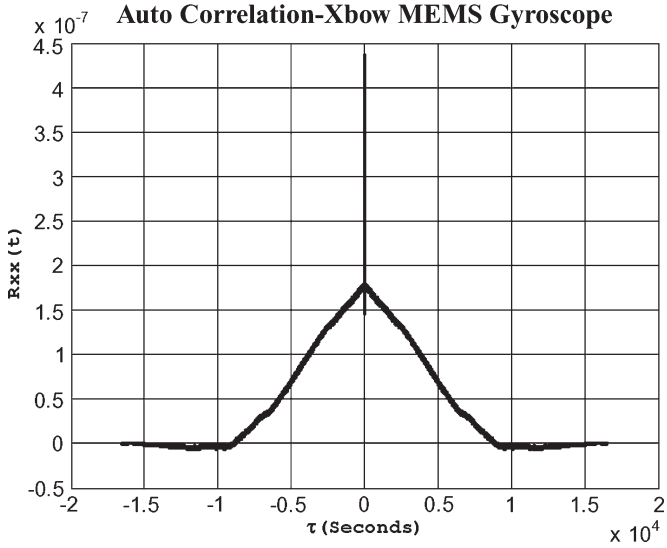

Fig. 3. Autocorrelation sequence of a Crossbow MEMS-based gyroscope.

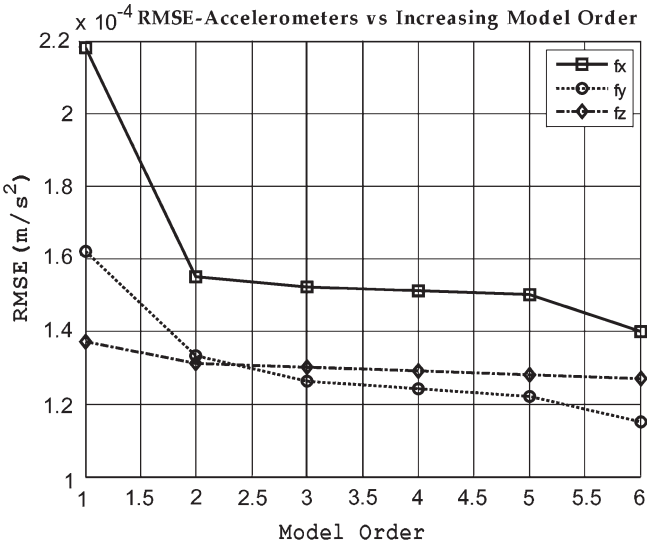

Fig. 4. Crossbow MEMS IMU accelerometer prediction RMSE using Burg AR modeling.

3) AR Model: Better stochastic modeling of inertial sensor errors can be achieved by modeling these errors as higher order AR processes and obtaining the AR model parameters adaptively. This requires collecting long-term measurements from each inertial sensor while it is stationary and its output contains only its inherent sensor errors (both short term and long term). These stationary sensor measurements are used for computing the coefficients of a higher order AR model [42]. Fitting stationary experimental data to an AR model can result in a better stochastic model than assuming the sensor conforms to a first-order GM model. The Burg estimation method [43]–[45] is used in this study to adaptively determine the parameters of the stochastic AR model for each inertial sensor. The pth-order AR model for a discrete-time domain sequence can be described by the following difference equation [43]:

$$y(n) = -\sum_{k=1}^{p} \alpha_k y(n-k) + \beta_0 w(n). \tag{3}$$

This can be expanded as

$$y(n) = -\alpha_1 y(n-1) - \alpha_2 y(n-2) - \cdots - \alpha_p y(n-p) + \beta_0 w(n) \tag{4}$$

where $\alpha_1, \alpha_2, \ldots, \alpha_p$ are the model parameters, and β_0 is the standard deviation of the sensor white noise. This pth-order difference equation has to be reduced to p first-order difference equations before they can be implemented using KF.

The Burg method is used to fit a pth-order AR model to the input signal by minimizing the forward and backward prediction errors while constraining the AR parameters to satisfy the Levinson–Durbin recursion [43]. The raw sensor measurements of the stationary experiment are assumed to be the output of an AR model driven by white noise, and the optimal values of the coefficients are obtained in a way that minimizes the mean square error (MSE) [21]. The Burg method was introduced to overcome some of the drawbacks with other methods (like the Yule–Walker and covariance methods) by providing more stable models and improved estimates with shorter data records [5].

Since the AR model is going to be applied to all six inertial sensors, each increase in the model order will lead to six more states added to the KF error state vector. For this reason, the lowest possible order must be chosen, where the root MSE (RMSE) of the model converges to avoid the complexity of the KF algorithm. Fig. 4 shows the RMSE versus different AR model orders when the Burg method was used to obtain the AR model parameters for the Crossbow MEMS-based accelerometers. It can be observed in this figure that the Burg method converges to approximately the same RMSE for AR model orders of 2 or above. It is also evident that the first-order model gives a higher RMSE than the higher order models. This means that better stochastic modeling of each inertial sensor error can be achieved by an AR model of order 2 or higher.

4) AR Model Parameters: To keep the INS error model simple and to facilitate the KF implementation, a second-order AR model was used for each inertial sensor. For a second-order model, (4) takes the following form:

$$y(n) = -\alpha_1 y(n-1) - \alpha_2 y(n-2) + \beta_0 w(n). \tag{5}$$

By defining two state variables, i.e., $x_1(n) = y(n-1)$ and $x_2(n) = y(n)$, (5) can be reduced to a set of first-order difference equations as follows:

$$x_1(n) = x_2(n-1) \tag{6}$$

$$x_2(n) = -\alpha_1 x_2(n-1) - \alpha_2 x_1(n-1) + \beta_0 w(n). \tag{7}$$

Therefore, in terms of state space phase variables, we have

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}_n = \begin{pmatrix} 0 & 1 \\ -\alpha_2 & -\alpha_1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}_{n-1} + \begin{pmatrix} 0 \\ \beta_0 \end{pmatrix} w(n). \tag{8}$$

It can be seen in (8) that for each inertial sensor, a second-order AR model produces two state variables x_1 and x_2 and two coefficients α_1 and α_2 that describe the model. Using this

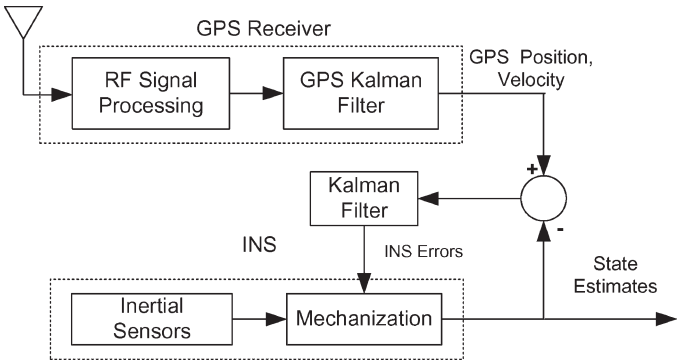

Fig. 5. Decentralized closed-loop implementation of INS/GPS integration using KF.

second-order error model, the KF state space would increase from 15 (for the GM model) to 21 variables, and the state transition matrix would include 12 coefficients (instead of six for the GM model).

C. Extended KF (EKF)

Since we are dealing with nonlinear systems and the low-cost MEMS inertial sensors used are of relatively poor performance, it is therefore decided to use EKF, where nonlinear systems are linearized about the estimated trajectory to get better performance. This is done through the closed-loop (feedback) implementation of INS/GPS integration, as shown in Fig. 5. The theory of EKF is well established, and details can be found in [7]–[11], [46], and [47].

The discrete-time nonlinear dynamic model state equation is given as follows [9]:

$$x_k = f(x_{k-1}, k-1) + g(x_{k-1}, k-1)w_{k-1} w_k \sim N(0, Q_k). \tag{9}$$

The nonlinear measurement equation is

$$z_k = h(x_k, k) + v_k \qquad v_k \sim N(0, R_k). \tag{10}$$

Now, based on the measurements $z_1, z_2, \ldots, z_{k-1}$, if we have $\hat{x}_{k-1}$ as optimal state estimate of the state x_{k-1} at epoch k, we can linearize (9) around $\hat{x}_{k-1}$ to obtain the approximate linear state equation [9], [10] as

$$x_k = \Phi_{k-1} x_{k-1} + G_{k-1} w_{k-1} \tag{11}$$

where $\Phi_{k-1} \triangleq (\partial f(\hat{x}_{k-1}, k-1)/\partial x_{k-1})$, and $G_{k-1} \triangleq g(\hat{x}_{k-1}, k-1)$.

Similarly, based on the above measurements, if we have $\hat{x}_k^-$ as the estimate of the systems state x_k, (10) can also be linearized around $\hat{x}_k^-$ to give its linear approximation [9], [47] as

$$z_k \approx H_k x_k + v_k \tag{12}$$

where $H_k \triangleq (\partial h(\hat{x}_k^-, k)/\partial x_k)$.

The well-known KF equations [7]–[11], [46] can now be applied to these linearized system models to obtain an optimal filter for state estimation x_k based on the measurements up to z_k. We can, therefore, obtain the linear minimum variance estimate of x_k as follows:

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - \hat{z}_k^-) \tag{13}$$

where

$$\hat{x}_k^- = \Phi_{k-1}\hat{x}_{k-1} \tag{14}$$

$$\hat{z}_k^- = H_k \hat{x}_{k-1}^- \tag{15}$$

$$K_k = \left(P_k^-\right) H_k^T \left(H_k P_k^- H_k^T + R_k\right)^{-1} \tag{16}$$

$$P_k^- = \Phi_{k-1} P_{k-1} \Phi_k^T + G_{k-1} Q_{k-1} G_{k-1}^T \tag{17}$$

$$P_k = (I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k R_k K_k^T. \tag{18}$$

1) Divergence Control: Modeling errors and linearization can result in the divergence of KF, and various methods have been suggested to avoid it [7]–[9], [48]. It is worth mentioning that we are using the Joseph's form of *a posteriori* error covariance P_k that helps avoid divergence due to assurance of positive semidefiniteness of P_k. However, there are still other factors that may lead to divergence problems that have been mitigated while implementing KF in this study. The mitigation procedure includes the following: 1) thresholding the covariance matrix of the estimation error so that it does not go lower than a certain value and 2) validating the GPS position and velocity data used by KF in updating INS.

Thresholding the covariance matrix: During the KF operation, the covariance matrix P_k keeps getting smaller, resulting in a smaller filter gain K_k. When P_k becomes very small, the gain K_k approaches zero, and new measurements are not given enough weight, as can be observed in (16). KF starts relying on its estimates alone and rejects new measurements. This is called "KF incest" and can be cured by keeping the covariance above a minimum threshold [9], [48], [49]. In our implementation, we also used a limiting technique where a lower bound is placed on the error covariance matrix P by an appropriate amount ε. This value is determined empirically, and the modified Kalman gain equation can be rewritten as

$$K_k = \left(P_k^- + \varepsilon I\right) H_K^T \left(H_k P_k^- H_k^T + R_k\right)^{-1}. \tag{19}$$

Data validity criteria: Another problem that sometimes plagues the KF accuracy is related to the integrity of GPS data. When the GPS data are corrupted, the KF tries to follow them, and the resulting output either is not smooth or diverges. This becomes a particularly sensitive issue when the covariance has already been limited to avoid the "KF incest" problem discussed above. To contain this situation, a data rejection method was employed, which compares the innovation sequence with a predetermined value and, based on the outcome, decides to accept or reject the data [8]. This technique is used to detect the anomaly in the absolute amplitude of the data as well as relative amplitude. The data are rejected if any of the following two conditions exists:

$$\begin{aligned} |(z_k - H_k\hat{x}_k)_i| &> V_{\max} \\ |(z_k - H_k\hat{x}_k)_{i+1} - (z_k - H_k\hat{x}_k)_i| &> \delta V_{\max}. \end{aligned} \tag{20}$$

When invalid data are encountered, the corresponding measurement state and covariance update is skipped, and repeated violation of data integrity check should be considered to be a sensor malfunction. The value of $V_{\max}$ can be determined based on the fact that the denominator of the Kalman gain K_k is the root mean square (rms) value of the innovation sequence $(z_k - H_k\hat{x}_k)$. Therefore, a 3σ (or higher) value could be used to reject the measurement outlier [48].

D. Implementation Details

To compare the two error models for inertial sensors, state equations for both GM and AR models were implemented in EKF. The details of various parameters, in relation to (11) and (12), will be treated in the following sections. The common elements of both implementations are the first nine elements of the state vector and the INS dynamic error model that is part of the state transition matrix Φ. The first nine elements of the state vector are given as follows:

$$x = (\,\delta\varphi \quad \delta\lambda \quad \delta h \quad \delta V_e \quad \delta V_n \quad \delta V_u \quad \delta p \quad \delta r \quad \delta y\,). \tag{21}$$

where $\delta\varphi$, $\delta\lambda$, and δh are the latitude, longitude, and altitude errors, δV_e, δV_n, and δV_u are the east, north, and up velocity errors, and δp, δr, and δy are the pitch, roll, and yaw errors. The dynamic error part of state matrix is a 9×9 matrix that contains the linearized INS error model given as

$$\Phi_{9\times 9} = \begin{pmatrix} 0 & 0 & \Phi_{13} & 0 & \Phi_{15} & 0 & 0 & 0 & 0 \\ \Phi_{23} & 0 & \Phi_{23} & \Phi_{24} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \Phi_{36} & 0 & 0 & 0 \\ \Phi_{41} & 0 & 0 & \Phi_{44} & \Phi_{45} & \Phi_{46} & 0 & \Phi_{48} & \Phi_{49} \\ \Phi_{51} & 0 & 0 & \Phi_{54} & \Phi_{55} & \Phi_{56} & \Phi_{57} & 0 & \Phi_{59} \\ \Phi_{61} & 0 & \Phi_{63} & \Phi_{64} & \Phi_{65} & 0 & \Phi_{67} & \Phi_{68} & 0 \\ 0 & 0 & \Phi_{73} & 0 & \Phi_{75} & 0 & 0 & \Phi_{78} & \Phi_{79} \\ \Phi_{81} & 0 & \Phi_{83} & \Phi_{84} & 0 & 0 & \Phi_{87} & 0 & \Phi_{89} \\ \Phi_{91} & 0 & \Phi_{92} & \Phi_{94} & 0 & 0 & \Phi_{97} & \Phi_{98} & 0 \end{pmatrix}. \tag{22}$$

The definitions of the individual elements of the state transition matrix can be found in Appendix A. For the first nine states, the process noise coupling vector G contains the associated standard deviation of the noise vector w_k. This vector is defined as

$$G_{9\times 1} = (\,\sigma_\varphi \quad \sigma_\lambda \quad \sigma_h \quad \sigma_{ve} \quad \sigma_{vn} \quad \sigma_{vu} \quad \sigma_p \quad \sigma_r \quad \sigma_y\,)^T \tag{23}$$

where $\sigma_{(\cdot)}$ are the standard deviations of associated states.

Vector z is a measurement vector that contains the difference of GPS and INS for position and velocity components, as given in (24), shown at the bottom of the page.

H is a measurement design matrix, which is basically a 6×6 identity matrix for six measurable states and contains zeros for the rest of the states as follows:

$$H_{6\times 15} = (\,I_{6\times 6} \quad 0_{6\times 9}\,), \qquad \text{for GM model} \tag{25}$$
$$H_{6\times 21} = (\,I_{6\times 6} \quad 0_{6\times 15}\,), \qquad \text{for AR model.} \tag{26}$$

1) KF for the GM Model: For the GM model, our KF consists of the 15 error states given in (27), shown at the bottom of the page.

The additional states, namely, $\delta\omega_x$, $\delta\omega_y$, and $\delta\omega_z$ and δf_x, δf_y, and δf_z are gyroscope and accelerometer bias errors, respectively. From the definition of the GM process in (2), we have the state space form of the six sensor errors as given in (28), shown at the bottom of the page, where β is the reciprocal of correlation time, and σ^2 is the variance of sensors white noise w_k.

In this case, the state transition matrix Φ is a 15×15 matrix that contains the abovementioned $\Phi_{9\times 9}$ matrix, and the sensor error model equation is given in the following form:

$$\Phi_{15\times 15} = \begin{pmatrix} {}^1\Phi_{3\times 9} & 0_{3\times 3} & 0_{3\times 3} \\ {}^2\Phi_{3\times 9} & 0_{3\times 3} & R^{bl}_{3\times 3} \\ {}^3\Phi_{3\times 9} & R^{bl}_{3\times 3} & 0_{3\times 3} \\ 0_{6\times 9} & {}^1M_{6\times 3} & {}^2M_{6\times 3} \end{pmatrix} \tag{29}$$

$$z_k = [\,(\varphi_{\mathrm{INS}} - \varphi_{\mathrm{GPS}}) \quad (\lambda_{\mathrm{INS}} - \lambda_{\mathrm{GPS}}) \quad (h_{\mathrm{INS}} - h_{\mathrm{GPS}}) \quad (V_{e_{\mathrm{INS}}} - V_{e_{\mathrm{GPS}}}) \quad (V_{n_{\mathrm{INS}}} - V_{n_{\mathrm{GPS}}}) \quad (V_{u_{\mathrm{INS}}} - V_{u_{\mathrm{GPS}}})\,]^T \tag{24}$$

$$x = (\delta\varphi \quad \delta\lambda \quad \delta h \quad \delta V_e \quad \delta V_n \quad \delta V_u \quad \delta p \quad \delta r \quad \delta y \quad \delta\omega_x \quad \delta\omega_y \quad \delta\omega_z \quad \delta f_x \quad \delta f_y \quad \delta f_z) \tag{27}$$

$$\begin{pmatrix} \delta\dot{\omega}_x \\ \delta\dot{\omega}_y \\ \delta\dot{\omega}_z \\ \delta\dot{f}_x \\ \delta\dot{f}_y \\ \delta\dot{f}_z \end{pmatrix}_k = \begin{pmatrix} 1-\beta_{\omega x}\Delta t & 0 & 0 & 0 & 0 & 0 \\ 0 & 1-\beta_{\omega y}\Delta t & 0 & 0 & 0 & 0 \\ 0 & 0 & 1-\beta_{\omega z}\Delta t & 0 & 0 & 0 \\ 0 & 0 & 0 & 1-\beta_{fx}\Delta t & 0 & 0 \\ 0 & 0 & 0 & 0 & 1-\beta_{fy}\Delta t & 0 \\ 0 & 0 & 0 & 0 & 0 & 1-\beta_{fz}\Delta t \end{pmatrix} \begin{pmatrix} \delta\omega_x \\ \delta\omega_y \\ \delta\omega_z \\ \delta f_x \\ \delta f_y \\ \delta f_z \end{pmatrix}_{k-1} + \begin{pmatrix} \sqrt{2\beta_{\omega x}\sigma^2_{\omega x}} \\ \sqrt{2\beta_{\omega y}\sigma^2_{\omega y}} \\ \sqrt{2\beta_{\omega z}\sigma^2_{\omega z}} \\ \sqrt{2\beta_{fx}\sigma^2_{fx}} \\ \sqrt{2\beta_{fy}\sigma^2_{fy}} \\ \sqrt{2\beta_{fz}\sigma^2_{fz}} \end{pmatrix} w_k \Delta t \tag{28}$$

where R^{bl} (which is defined as R_b^l in Appendix B) is the rotation matrix to transform the states from body frame to local level frame, and ${}^1\Phi_{3\times9}$, ${}^2\Phi_{3\times9}$, and ${}^3\Phi_{3\times9}$ are subdivisions (which are defined in Appendix B) of matrix $\Phi_{9\times9}$. ${}^1M_{6\times3}$ and ${}^2M_{6\times3}$ are matrices that contain the state transition part of the GM model and are defined as follows:

$$ {}^1M_{6\times3} = \begin{pmatrix} 1-\beta_{\omega x}\Delta t & 0 & 0 \\ 0 & 1-\beta_{\omega y}\Delta t & 0 \\ 0 & 0 & 1-\beta_{\omega z}\Delta t \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{30} $$

$$ {}^2M_{6\times3} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1-\beta_{fx}\Delta t & 0 & 0 \\ 0 & 1-\beta_{fy}\Delta t & 0 \\ 0 & 0 & 1-\beta_{fz}\Delta t \end{pmatrix}. \tag{31} $$

Six additional bias error states, shown in (32) at the bottom of the page, are also appended to vector G.

2) KF for the AR Model: For AR modeling of inertial sensor errors, the first nine states of the KF are the same; however, there are two more states for each sensor that represent its bias error. The complete error state vector, therefore, is given in (33), shown at the bottom of the page.

From the definition of (8), the additional six states for gyroscopes can be written in state space form as in (34), shown at the bottom of the page.

Similarly, the six states for accelerometers are modeled as in (35), shown at the bottom of the page.

In this case, the state transition matrix Φ is a 21×21 matrix that contains the subdivisions of the abovementioned $\Phi_{9\times9}$ matrix, and the equations for the AR sensor error models are arranged in a state transition matrix as follows:

$$ \Phi_{21\times21} = \begin{pmatrix} {}^1\Phi_{3\times9} & 0_{3\times6} & 0_{3\times3} & 0_{3\times3} \\ {}^2\Phi_{3\times9} & 0_{3\times6} & 0_{3\times3} & R^{bl}_{3\times3} \\ {}^3\Phi_{3\times9} & 0_{3\times6} & R^{bl}_{3\times3} & 0_{3\times3} \\ 0_{3\times9} & 0_{3\times6} & I_{3\times3} & 0_{3\times3} \\ 0_{3\times9} & 0_{3\times6} & 0_{3\times3} & I_{3\times3} \\ 0_{6\times9} & {}^1A_{6\times6} & {}^2A_{6\times3} & {}^3A_{6\times3} \end{pmatrix} \tag{36} $$

where ${}^1A_{6\times6}$, ${}^2A_{6\times3}$ and ${}^3A_{6\times3}$ are the matrices consisting of AR parameters and are given as follows:

$$ {}^1A_{6\times6} = \begin{pmatrix} -\alpha_2^{\omega x} & 0 & 0 & 0 & 0 & 0 \\ 0 & -\alpha_2^{\omega y} & 0 & 0 & 0 & 0 \\ 0 & 0 & -\alpha_2^{\omega z} & 0 & 0 & 0 \\ 0 & 0 & 0 & -\alpha_2^{fx} & 0 & 0 \\ 0 & 0 & 0 & 0 & -\alpha_2^{fy} & 0 \\ 0 & 0 & 0 & 0 & 0 & -\alpha_2^{fz} \end{pmatrix} \tag{37} $$

$$ G = \left(\sqrt{2\beta_{\omega x}\sigma^2_{\omega x}} \quad \sqrt{2\beta_{\omega y}\sigma^2_{\omega y}} \quad \sqrt{2\beta_{\omega z}\sigma^2_{\omega z}} \quad \sqrt{2\beta_{fx}\sigma^2_{fx}} \quad \sqrt{2\beta_{fy}\sigma^2_{fy}} \quad \sqrt{2\beta_{fx}\sigma^2_{fz}} \right)^T \tag{32} $$

$$ x = (\delta\varphi \;\; \delta\lambda \;\; \delta h \;\; \delta V_e \;\; \delta V_n \;\; \delta V_u \;\; \delta r \;\; \delta p \;\; \delta y \;\; \delta\omega_{x1} \;\; \delta\omega_{y1} \;\; \delta\omega_{z1} \;\; \delta f_{x1} \;\; \delta f_{y1} \;\; \delta f_{z1} \;\; \delta\omega_{x2} \;\; \delta\omega_{y2} \;\; \delta\omega_{z2} \;\; \delta f_{x2} \;\; \delta f_{y2} \;\; \delta f_{z2})^T \tag{33} $$

$$ \begin{pmatrix} \delta\omega_{x1} \\ \delta\omega_{y1} \\ \delta\omega_{z1} \\ \delta\omega_{x2} \\ \delta\omega_{y2} \\ \delta\omega_{z2} \end{pmatrix}_k = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -\alpha_2^{\omega x} & 0 & 0 & -\alpha_1^{\omega x} & 0 & 0 \\ 0 & -\alpha_2^{\omega y} & 0 & 0 & -\alpha_1^{\omega y} & 0 \\ 0 & 0 & -\alpha_2^{\omega z} & 0 & 0 & -\alpha_1^{\omega z} \end{pmatrix} \begin{pmatrix} \delta\omega_{x1} \\ \delta\omega_{y1} \\ \delta\omega_{z1} \\ \delta\omega_{x2} \\ \delta\omega_{y2} \\ \delta\omega_{z2} \end{pmatrix}_{k-1} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ \sigma_{\omega x} \\ \sigma_{\omega y} \\ \sigma_{\omega z} \end{pmatrix} w(k) \tag{34} $$

$$ \begin{pmatrix} \delta f_{x1} \\ \delta f_{y1} \\ \delta f_{z1} \\ \delta f_{x2} \\ \delta f_{y2} \\ \delta f_{z2} \end{pmatrix}_k = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -\alpha_2^{fx} & 0 & 0 & -\alpha_1^{fx} & 0 & 0 \\ 0 & -\alpha_2^{fy} & 0 & 0 & -\alpha_1^{fy} & 0 \\ 0 & 0 & -\alpha_2^{fz} & 0 & 0 & -\alpha_1^{fz} \end{pmatrix} \begin{pmatrix} \delta f_{x1} \\ \delta f_{y1} \\ \delta f_{z1} \\ \delta f_{x2} \\ \delta f_{y2} \\ \delta f_{z2} \end{pmatrix}_{k-1} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ \sigma_{fx} \\ \sigma_{fy} \\ \sigma_{fz} \end{pmatrix} w(k) \tag{35} $$

$$ {}^{2}A_{6\times 3} = \begin{pmatrix} -\alpha_1^{\omega x} & 0 & 0 \\ 0 & -\alpha_1^{\omega y} & 0 \\ 0 & 0 & -\alpha_1^{\omega z} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{38} $$

$$ {}^{3}A_{6\times 3} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ -\alpha_1^{fx} & 0 & 0 \\ 0 & -\alpha_1^{fy} & 0 \\ 0 & 0 & -\alpha_1^{fz} \end{pmatrix}. \tag{39} $$

In the above matrices, $\alpha_1^{\omega(\cdot)}$ and $\alpha_2^{f(\cdot)}$ are the AR parameters for three gyroscopes and three accelerometers, respectively.

For the AR model, the additional 12 states of noise coupling matrix are given as

$$ G = (0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad \sigma_{\omega x} \quad \sigma_{\omega y} \quad \sigma_{\omega z} \quad \sigma_{fx} \quad \sigma_{fy} \quad \sigma_{fz})^T \tag{40} $$

where σ is the standard deviation of the sensors noise.

E. Nonlinear Position Error Modeling Using the NF Technique

Since KF cannot totally remove the nonlinear and stochastic parts of the INS errors, it is beneficial to augment KF with a system that can provide nonlinear modeling of these errors such as a NF module. This further decreases the position error when using MEMS-based inertial sensors. In this study, the NF module is used to provide a reliable prediction of the vehicle position by using ANFIS to accurately predict the vehicle position during GPS outages. In the early stages of this paper, it was determined that AI techniques like ANFIS cannot be used alone for MEMS-based INS/GPS integration. The main reason is the fast growth and accumulation of INS errors over time due to serious random bias and drift errors that exist at the output of MEMS-based inertial sensors. Therefore, adequate modeling of these errors (using AR stochastic modeling techniques) and optimal estimation procedures like KF are essential for the proper integration between INS and GPS. Some AI-based techniques developed for integrating navigational grade or tactical grade INS with GPS can be adopted for MEMS-based inertial systems but only when they are augmented to KF to limit the growth of INS errors.

In this study, both INS and GPS are integrated together in such a way that 1) the INS position errors are prevented from growing in the long term using GPS position updates, and 2) the corrected INS position is used during GPS outages. The ANFIS-based module was designed to be augmented to KF to mimic the INS position errors, to create the corresponding nonlinear error model, and to predict these errors during a GPS outage, thus enhancing the overall system accuracy.

1) ANFIS-Based System Architecture: In this paper, we utilize the P–δP INS/GPS integration scheme, which is based on the estimation of INS position error δP_{INS}, by processing the INS position P_{INS}. The ANFIS is employed to provide an optimal temporal estimation of the INS errors $(\delta P_{\mathrm{INS}})$. The proposed system architecture compromises the following three modes of operation: 1) the initialization mode; 2) the update mode [Fig. 6(a)]; and 3) the prediction mode [Fig. 6(b)].

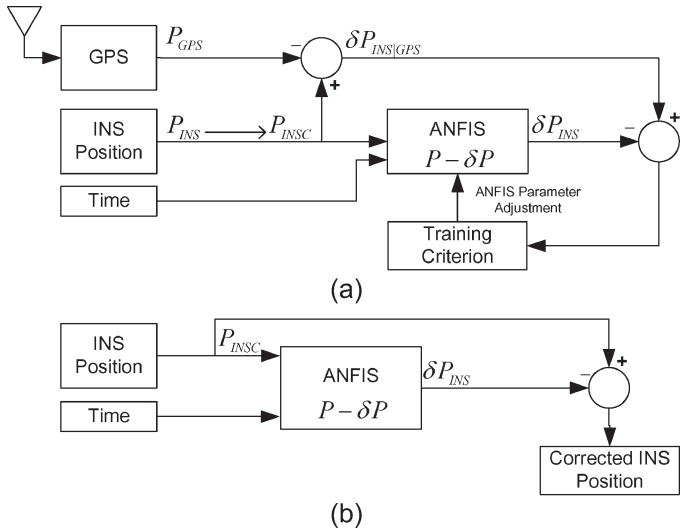

Fig. 6. Proposed P–δP system architecture.

The initialization and update modes are used, as long as the GPS signal is available, to initialize the first learning rule-base and to limit INS error growth. The INS position input shown in Fig. 6 is the output of a suboptimal KF module. The prediction mode is used to correct the INS position when the GPS signal is lost. Thus, the P–δP module is trained during the availability of the GPS signal to recognize patterns of the position error embedded in the input position components. In case of satellite signal blockage, the P–δP module mimics the latest vehicle dynamics and delivers a prediction of the vehicle position error.

The INS position (P_{INS}) and time (T) are the inputs to the module, while the error in the INS position $(\delta P_{\mathrm{INS}})$ is the module output. A schematic representation of the P–δP module for establishing the rule-base, relating the INS position (P_{INS}) and time (T) to the error in the INS position $(\delta P_{\mathrm{INS}})$, is shown in Fig. 7.

The estimated INS position error provided by the P–δP module is then compared with the error between the INS original position and the corresponding GPS position $(\delta P_{\mathrm{INS|GPS}})$. The number and shape of membership functions shall be predefined (Gaussian shape membership functions were used here; however, other shapes proved possible). The original mean and spread of the membership functions are computed using fuzzy clustering techniques [50], [51]. The membership values are evaluated at the first layer and the fuzzy t-norm operator $(\prod)$ is implemented at the second layer. A normalized firing strength $(\bar{W}_i)$ is computed at the third layer. The INS position error is computed as

$$ \text{if } P_{\mathrm{INS}} \in P_1 \text{ and } T \in T_1, \text{ then } f_1 = p_1 P_{\mathrm{INS}} + q_1 T + r_1 \tag{41} $$

$$ \text{if } P_{\mathrm{INS}} \in P_{\sim 2} \text{ and } T \in T_{\sim 2}, \text{ then } f_2 = p_2 P_{\mathrm{INS}} + q_2 T + r_2 \tag{42} $$

$$ \vdots $$

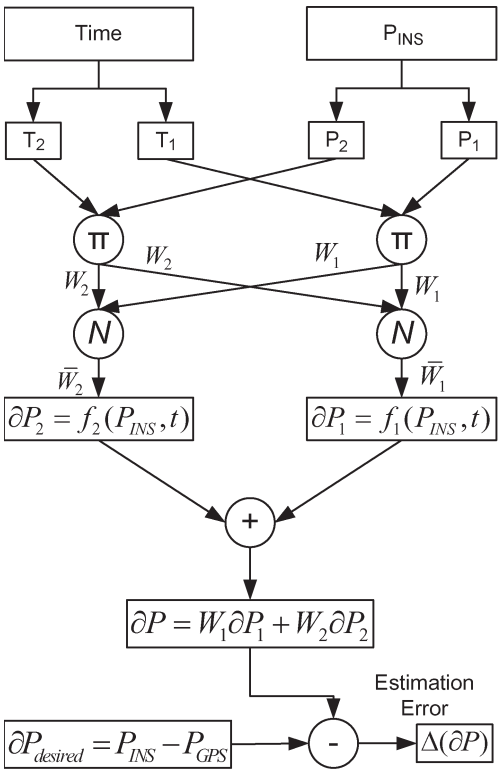

Fig. 7. ANFIS architecture for INS/GPS integration.

$$\text{if } P_{\mathrm{INS}} \in P_{\sim n} \text{ and } T \in T_{\sim n}, \text{ then } f_n = p_n P_{\mathrm{INS}} + q_n T + r_n \tag{43}$$

$$\delta P_{\mathrm{INS}} = \sum_{i=1}^{n} \bar{W}_i f_i \tag{44}$$

where p_i, q_i, and r_i are the consequent linear parameters originally determined using a least squares approach. $P_{\sim n}$ represents a fuzzy set defined over the P_{INS} domain, $T_{\sim n}$ represents a fuzzy set defined over the T domain, and $\bar{W}_i$ represents the mean normalized weight of the ith rule as shown in Fig. 7.

The difference between δP_{INS} and $\delta P_{\mathrm{INS|GPS}}$ is the training error $(\Delta(\delta p))$ of the ANFIS module. To minimize the training error, the ANFIS module updates the learning rule-base (membership function parameters and consequent parameters) using a hybrid learning approach that combines the least squares and back propagation techniques until a certain minimal RMSE is reached [50], [51].

When the satellite signal is blocked, the system is switched to the prediction mode, where the P–δP module is used to predict the INS position error using the latest learning rule-base obtained before losing the satellite signals. The error is then removed from the corresponding INS position component to obtain the corrected INS position (P_{INSC}). In this study, three P–δP modules are developed to provide complete navigation solution in the three axes for a moving vehicle represented by the three position components.

It should also be highlighted that there are several sources of errors that contribute to the overall position error of the vehicle in different ways. These sources include the inertial sensors bias drift and scale factor instability, the initial misalignment error during INS initialization, and the error in the heading angle of the vehicle (i.e., the azimuth error). The azimuth error, for example, can have a significant contribution to the position error along both the east and north position components, particularly at high speeds since it is modulated by the vehicle velocity. Although all these error sources are not explicitly represented in the architecture shown in Fig. 6, it is evident that the P–δP module incorporates the effects of such errors in two ways. First, it establishes the learning rule-base to pattern vehicular navigation performance using exemplar navigation scenarios that have been affected by the same set of sources of errors. Second, it considers the input parameters as interval data with membership values rather than crisp values that provide room for uncertainty in the input parameters due to these sources of errors.

To use the P–δP module in a temporal INS/GPS integration, a sliding window with a certain window size ω is considered. For each of the ANFIS modules, the number of samples (equal to ω) of INS position component P_{INS} and the corresponding GPS position P_{GPS} are acquired from both systems. The INS position is considered as the input to the P–δP module, and the error between P_{INS} and P_{GPS} is considered to be the corresponding desired response $(\delta P_{\mathrm{INS|GPS}})$. The update procedure of the P–δP learning rule-base starts after collecting the ωth sample of both INS and GPS position components. Before considering the next INS and GPS samples, the ANFIS module is trained until a certain minimum RMSE is reached or after a certain number of training epochs is completed. To guarantee timely operation of the system, the update procedure is terminated at the end of these training epochs, even if the desired RMSE is not achieved.

While the GPS signal is available, the data window continues to slide collecting new samples from INS and GPS position components. Since the GPS signal is available during the update mode, both P_{INSC} and the position error between P_{INSC} and P_{GPS} are used to train ANFIS to mimic the dynamics present within the last data window. This results in a new ANFIS rule-base, which is used to provide an estimate for the INS position error at time $i+1(\delta P_{\mathrm{INS}}(i+1))$. Therefore, the INS position is continuously updated, and the INS position errors (estimated by the ANFIS module) are removed, thus keeping accurate INS position components available in case of any GPS outages.

2) Cross Validation: To optimize the operation of the ANFIS-based module, a cross validation algorithm has been adopted. Cross validation is a data-partitioning technique that allows iterative partitioning of samples into two sets of data: the first set can be used in building the simulation model (training), while the second set is used for testing the model [28]. Among different cross validation techniques, the *hold-out* method is utilized in this study due to its suitability for real-time implementation. As shown in Fig. 8, the hold-out method splits the data into two sets, namely 1) the training set and 2) the validation set. The function "*approximator*" is trained to fit a function using the training set only [28]. Then, the function approximator is used to predict the output values for the data in the validation set. While this method has the advantage of being less computationally expensive, it might yield a high variance. However, it is the most suitable for real-time implementation.

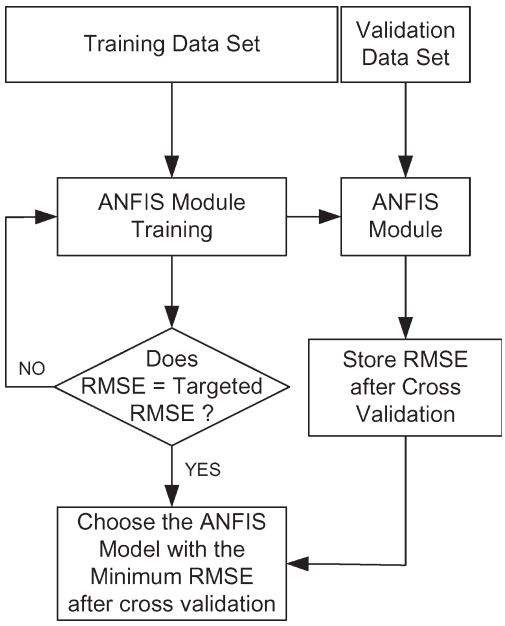

Fig. 8. Cross validation procedure during the training of an ANFIS module.

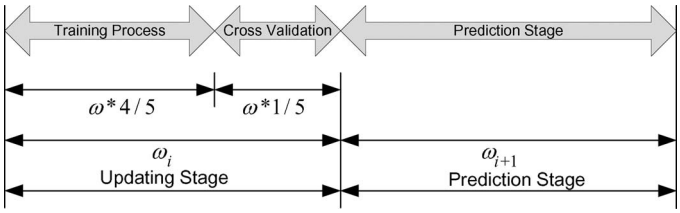

Fig. 9. Timeline index for the model stages sequences.

For real-time INS/GPS integration, a nonoverlapping and moving window with a certain window size ω is considered. The P–δP ANFIS module consists of three main stages. The first stage considers initializing the ANFIS module by acquiring both the INS and GPS original position components (P_{INS} and P_{GPS}) up to the window size ω (from time $t = 0$ to $\omega - 1$). At time t equal to ω, the second stage is started, namely, the update stage. While the GPS signal is still available, the data window collects new samples from INS and GPS position components (from $t = \omega$ to $2\omega - 1$). The updating stage is divided into two consequent parts, namely 1) the training session and 2) the cross validation session. In case of GPS outages, the latest updated ANFIS module (after training and cross validation) is utilized in the prediction mode to predict the INS position error. Fig. 9 shows the time line index for the sequences of both the update and prediction stages.

3) Performance and Limitations: The limitation of ANFIS modules is that they cannot be used with MEMS-based sensors as standalone without the aid of KF. This is because the output of the low-cost sensors degrades quickly over time and must be corrected at regular intervals. The growing positional error affects the performance of the ANFIS module since, after only a short period, the mechanized IMU output is no longer reliable [29]. KF has traditionally been used to predict the error of the IMU and to update the output using GPS. However, since the optimal KF output is nearly exact to the GPS values when GPS signals are present, there is no error available with which

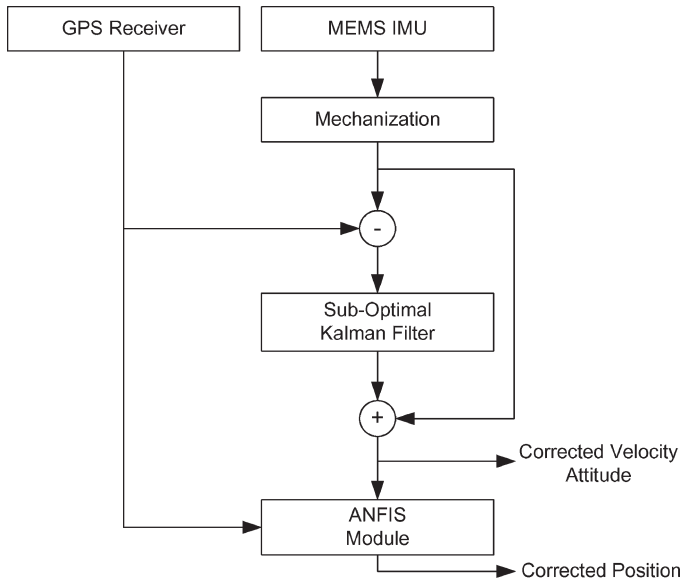

Fig. 10. Flowchart of an augmented ANFIS and a suboptimal KF.

the NF module can train. To overcome this limitation, the KF output can be detuned by increasing the GPS position measurement covariance values to provide suboptimal performance that artificially creates a small error between the KF output and the GPS. This suboptimal KF solution can now be used to train the NF module so that it can outperform the optimal KF in periods where GPS outages occur. Fig. 10 shows the flow diagram of the augmented KF–ANFIS INS/GPS integration. The suboptimal or detuned KF provides corrected velocity and attitude information and suboptimal position. The suboptimal position information is sent to the ANFIS module, where it is used with the GPS position for training. While training the ANFIS module, the cross validation procedure operates on 20% of the INS/GPS window of position data to improve the training performance and to avoid overfitting problems. When a GPS outage is detected, the ANFIS module uses its latest parameters to predict the INS position error and to correct the KF position.

One other important limitation for ANFIS is that it is not a recursive algorithm. The windowing criterion described earlier will enable near-real-time operation. Thus, there will be a delay at the beginning before providing its output and correcting for the INS errors. Moreover, the ANFIS training and cross validation procedures are relatively more computationally expensive compared to KF. The ANFIS module, however, is similar to KF in requiring a relatively longer design time and tuning of its parameters to achieve the desired performance during both the update and prediction stages and also in delivering the desired accuracy.

It should be noted that the proper choice of the window size is vital to guarantee the desired accuracy while ensuring system robustness in real time. The complexity of choosing the window size is related to its dependence on the level of vehicle dynamics, the length of the GPS outages, and its significance on the update procedure. Therefore, there is a tradeoff in choosing a small or a large window size. Large window sizes enable mimicking significant details of the latest vehicle dynamics, and thus, the module becomes reliable during long

Fig. 11. Data-collection equipment mounted inside the road test vehicle.

GPS outages. However, large window sizes may complicate the update procedure and result in a long processing time. On the other hand, a fast and robust update procedure can be achieved by using a small window size due to the reduced level of the dynamic nature of INS and GPS data. Moreover, using relatively small window sizes prevents considering inaccurate position information provided by the ANFIS module during the first outage for the prediction of the INS position components. However, relatively small window sizes may cause the system to lose reliability in case of relatively long GPS outages. In this study, we determined that a 45-s window size provides adequate performance and positioning accuracy.

III. Experimental Work

A. Equipment

For this paper, two IMUs were used. The first was the Crossbow IMU300CC-100 MEMS-grade IMU, which was integrated with the low-cost Trimble Lassen SQ GPS receiver that utilizes the methods suggested in this paper. The second IMU was the Honeywell HG1700 tactical grade IMU, which was integrated with the Novatel OEM4 GPS receiver. The Honeywell IMU and the Novatel GPS were integrated using an off-the-shelf assembly, i.e., the G2 Pro-Pack SPAN unit, which was also developed by Novatel. The Novatel unit provides a tightly coupled INS/GPS navigation solution, which is used as a reference to compare the performance and the effectiveness of the proposed methods when applied to MEMS-based sensors. The specifications of the two IMUs can be found in Table I. The experimental test setup, which is shown in Fig. 11, was built as a platform around the rear bench seat mounts for any full-size General Motors van. The power requirements were delivered by the vehicle's 12-V dc source and a 1800-W ac power supply to run the computer and configurable dc power supply for voltages other than 12 V. Fig. 11 shows all of the equipment installed inside the van, ready for a road test.

B. Data Acquisition

Data acquisition was carried out using two software programs. The Novatel GPS system and Honeywell HG1700 (high-end tactical grade IMU) used proprietary off-the-shelf software from Novatel that communicates with the sensors and logs both INS and GPS data.

A communication and data logging system was developed for the Crossbow IMU so that it could be integrated and synchronized with the Trimble GPS receiver. Both the Trimble GPS receiver and the Crossbow IMU use the RS-232 communication protocol. A new Labview (National Instruments Corporation, TX)-based module was developed to communicate and log the Crossbow IMU data and synchronize them with GPS position and velocity data. In the module interface, the communication settings such as the "port identification," "read delay," and "string to write" can be changed. The Labview module sends a command to the Crossbow IMU, requesting a reading at a predefined data rate, and acquires the response. The module reads and writes to the RS-232 serial port of the Crossbow IMU and converts the 18-B message into rotation rates and acceleration in the x, y, and z directions. Moreover, it time stamps each record with GPS time from the Trimble GPS before saving the IMU data to a file.

C. Road Test Trajectories

Several road trajectory tests were carried out using the above-described setup. As shown in Fig. 12, a trajectory from Smith Falls, ON, to Kingston, ON (trajectory 1), was chosen for analysis in this paper since it contained many of the features typical to real-world trajectories. It has urban roadways in Smith Falls, in Perth, and in highway sections in-between. In addition, the terrain varies with many hills, trees, winding turns, and section of straight roads. The ultimate check for the proposed system's accuracy is during GPS signal degradation and blockage, which was simulated and intentionally introduced during postprocessing, in which 45-s simulated GPS outages (shown as blue/gray circles overlaid on the map in Fig. 12) were introduced into the trajectory to assess the performance of sensor error models and the ANFIS module. The intentionally introduced GPS outages are given such that they encompass all conditions of a typical trip, including straight portions, turns, slopes, high speeds, slow speeds, and stops. The second trajectory consisted of only urban roadways inside the city of Kingston (trajectory 2), as shown in Fig. 13.

IV. Results and Discussion

A. Impact of the Second-Order AR Model Over the First-Order GM Model

1) Trajectory 1 Results—KF: The impact of the second-order AR model for each inertial sensor error on the overall positioning accuracy will be examined and compared to the conventional method using a first-order GM model during nine GPS outages intentionally introduced into the trajectory. Figs. 14 and 15 compare, respectively, the maximum and rms values of the position errors of the second-order AR model to the conventional first-order GM error model during the nine GPS outages. It can be determined from this figure that, overall, the second-order AR model provided a 20% higher positioning accuracy over the first-order GM model for 45-s GPS outages. We observed that during periods of longer GPS outages, a

TABLE I
IMU SPECIFICATIONS

Specifications	IMU 300CC-100	IMU HG1700
Update Rate	>100 Hz	100Hz
	Gyroscope	
Range	±100 deg/s	±1000 deg/s
Bias	< ±2.0 deg/s	1.0 deg/hr
Scale Factor	< 1%	150 ppm
Angle Random Walk	< 2.25 deg/$\sqrt{hr}$	0.125 deg/$\sqrt{hr}$
	Accelerometer	
Range	± 2g	± 50g
Bias	< ±30mg	1.0 mg
Scale Factor	<1%	300 ppm
Velocity Random Walk	<0.15 m/s/$\sqrt{hr}$	No spec
Linearity	< 1%	500 ppm

Fig. 12. Trajectory 1 from Smith Falls, ON, to Kingston, ON.

Fig. 13. Trajectory 2 in Kingston, ON.

significant improvement in the positioning accuracy can be obtained when using the second-order AR model for the inertial sensor errors. In relatively shorter GPS outages of 30 s or less, the performance of the first-order GM model becomes quite comparable with that of the second-order AR model. As apparent in Figs. 14 and 15, for relatively long GPS outages (like 45 s, as assumed in this study), the second-order AR model showed superior performance during all GPS outages, except for outages 5 and 9, where the performance of the first-order GM model was slightly better. Fig. 16 and Fig. 17(a) and (b) show the sections of the trajectory during GPS outages 4, 5, and 9, respectively. Fig. 16 shows the performance during GPS outage 4, which was on a relatively straight portion of a highway. The GM model influenced the KF so that it overestimated both the east and north velocities, while the second-order AR model contained the KF estimation to well within limits and provided a 56% higher accuracy. The two cases where GM model performed better are depicted in Fig. 17 for GPS outages 5 and 9. As can be observed in Fig. 17, there was only a slight difference in the performance between the two methods, particularly for GPS outage 9. We believe that higher order AR models for some of the inertial sensors may lead to a superior accuracy over the traditional first-order GM models; however, it may complicate the error model used by KF.

We determined that in most cases, both the second-order AR and first-order GM models provided comparable levels of accuracy for a straight-line section of a trajectory, particularly for short outages. However, if a GPS outage took place in turns or in winding sections, or when the outages were longer, second-order AR model helped KF in producing better positioning accuracies, as depicted in Figs. 14 and 15 for GPS outages 4, 6, 7, and 8.

In general, the second-order AR stochastic error models of MEMS-based inertial sensors benefit the positioning accuracy in cases where the overall system tends to provide large position errors during GPS longer outages. The high position errors (over the 100-m level) that occurred during some of the above 45-s GPS outages were noticed to transpire when a GPS outage began immediately prior to a change in vehicle dynamics, such as a sharp turn or a stop-and-go traffic. These high errors

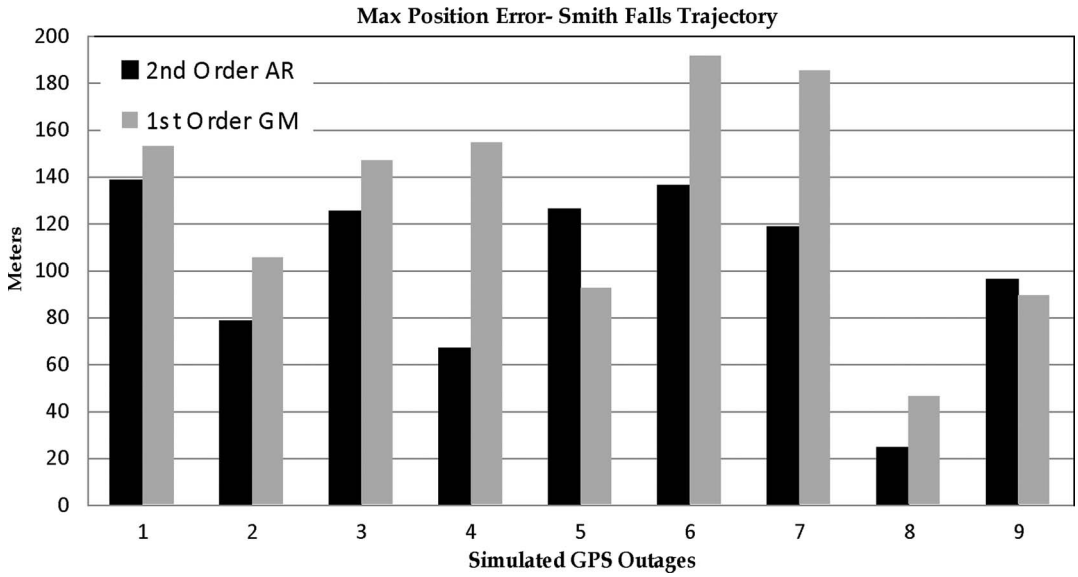

Fig. 14. Maximum position error during 45-s GPS outages while using first-order GM and second-order AR error models.

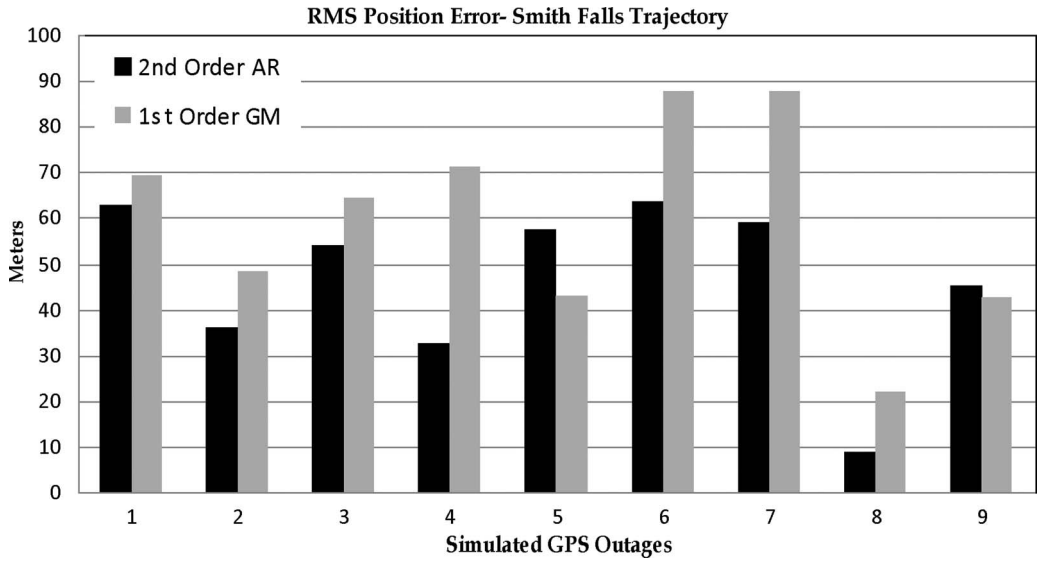

Fig. 15. RMS position error during 45-s GPS outages while using first-order GM and second-order AR error models.

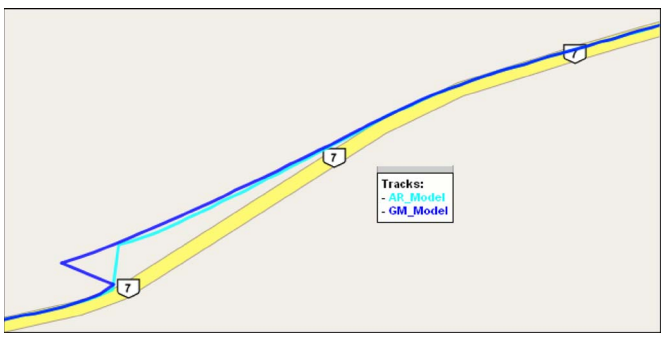

Fig. 16. Performance during GPS outage 4.

are likely caused by the nonlinear and stochastic parts of the INS errors, and they cannot be mitigated appreciably using the second-order AR model of inertial sensor errors. Another reason for the relatively large position errors during the 45-s GPS

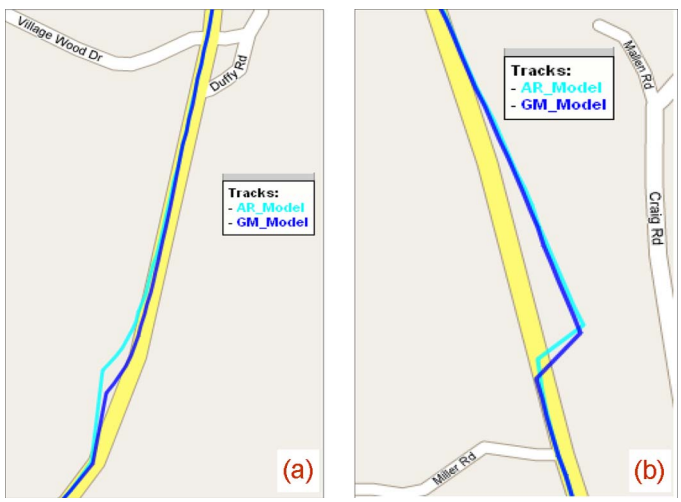

Fig. 17. (a) Performance during GPS outage 5. (b) Performance during outage 9.

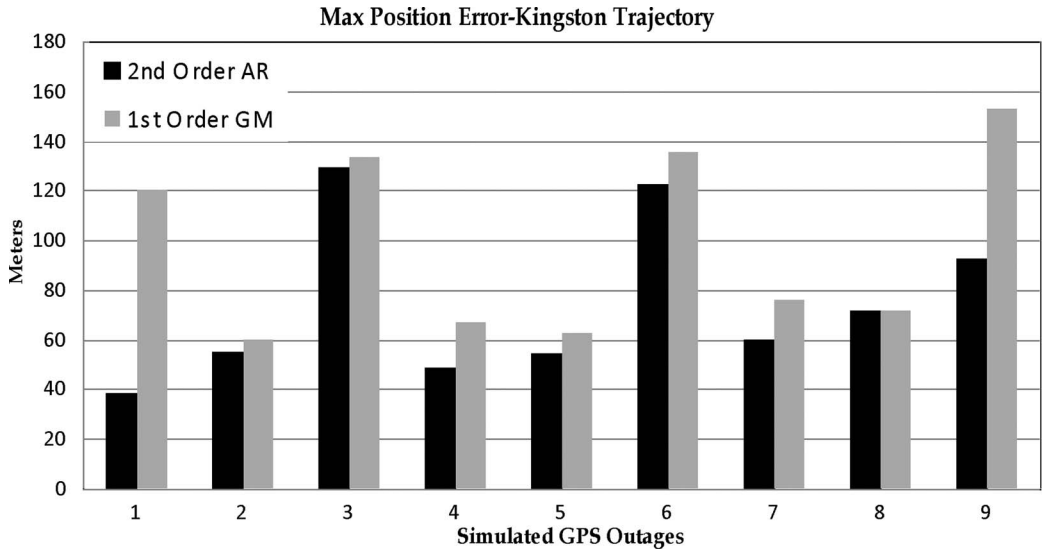

Fig. 18. Maximum position errors during 45-s GPS outages while using first-order GM and second-order AR error models.

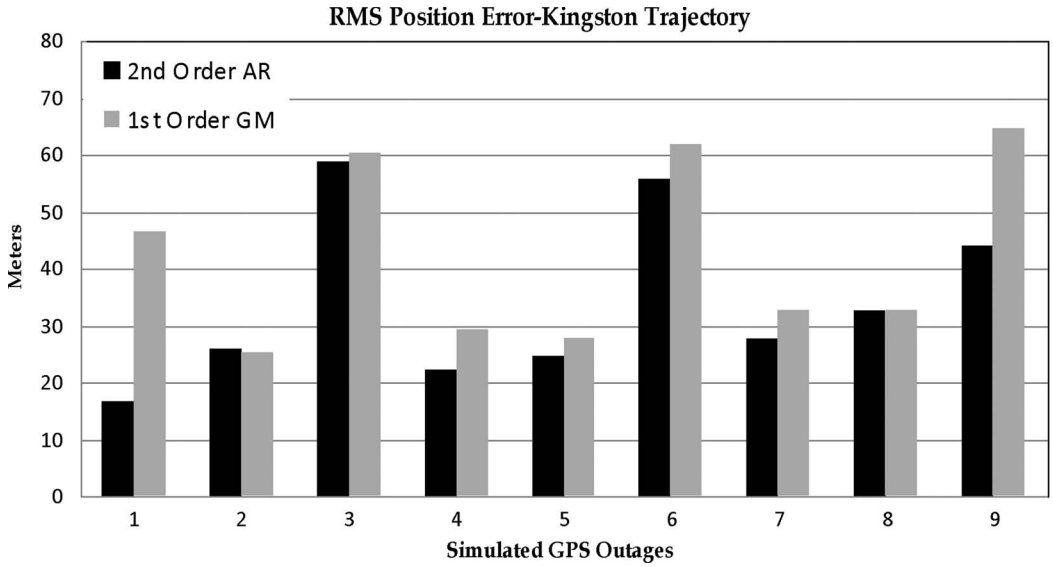

Fig. 19. RMS errors during nine 45-s simulated GPS outages.

outages is the nonstationary part of the INS position errors, which is due to the azimuth misalignment modulated by the vehicle velocity. In this trajectory, the vehicle was moving in highway regions at relatively higher speeds (close to 100 km/h). Such high speeds modulated azimuth errors and resulted in large position errors along the east and north directions in different sections of the trajectory. These kinds of nonstationary position errors cannot be extenuated by either the second-order AR or first-order GM model.

2) Trajectory 2 Results—KF: To further validate the usefulness of the second-order AR model for inertial sensors, trajectory 2 within the city of Kingston was used for this purpose. The maximum and rms position errors during nine 45-s GPS outages of the Kingston trajectory are plotted in Figs. 18 and 19. It is worth mentioning here that the relatively small position errors are encountered while driving at lower speeds typical of in-city driving than higher speeds used on highways. During the Kingston trajectory, except for GPS outages 3 and 6, all other outages show a relatively small error when compared to the Smith Falls trajectory, which mostly includes high-speed driving. During the Kingston trajectory, the overall accuracy when using the AR model was 22% higher than the GM model and provided less position errors for all nine GPS outages, except for outage 8, where errors were almost the same. Fig. 20 shows the superior performance of the AR model over the GM model while the vehicle was going into a turn during GPS outage 1. During this GPS outage, the second-order AR model assisted KF to perform well, showing a 68% accuracy improvement. During GPS outage 8 (Fig. 21), KF provided approximately the same positioning accuracy (errors of 71 m) when using both the models. GPS outage 9 (Fig. 22) turned out to be really challenging, as it consisted of a turn on a steep downslope. The KF module using the GM model produced an error of 152 m, while using second-order AR model, KF produced an error of only 92 m, which is about a 40% improvement in accuracy during this GPS outage.

Fig. 20. Performance during GPS outage 1.

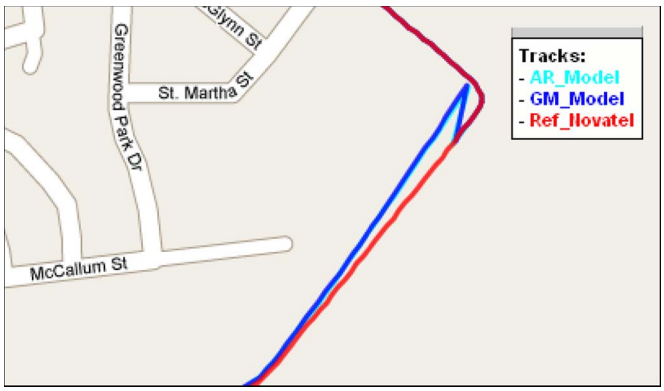

Fig. 21. Performance during GPS outage 8.

Fig. 22. Performance during GPS outage 9.

B. Impact of the Augmented KF–ANFIS Module

Using suboptimal KF with an ANFIS module for each position component in a cascaded scheme (discussed above) proved to be very advantageous not only in reducing the INS nonstationary and nonlinear position errors but also in showing a very consistent level of performance for relatively long GPS outages—a normal occurrence in urban canyons. The optimum ANFIS performance was achieved using a 45-s window size, six two-sided Gaussian membership functions for the position information, and one for the time input. The RMSE threshold of 10^{-4} was used during the training procedure.

1) Trajectory 1 Results—KF–ANFIS: Figs. 23 and 24 show a comparison of maximum and rms position errors between the KF (utilizing the second-order AR model for each inertial sensor) and the ANFIS module that used a suboptimal KF solution for the same nine 45-s GPS outages in the first trajectory. The augmented ANFIS module was able to reduce the position error for all GPS outages, except for outage 8, with an overall improvement of 60% for this trajectory. From Figs. 23 and 24, it can be noticed that both the KF and augmented KF–ANFIS methods provided comparable accuracies during GPS outage 8. The rms value of the position error for the augmented KF–ANFIS solution during the same GPS outage was slightly better than the KF solution. Over the nine GPS outages, the augmented KF–ANFIS module provided an average maximum error of only 32 m and an rms of 13 m compared with an average maximum error of 101 m and an rms of 46 m for the standalone KF. It can be determined from Figs. 23 and 24 that the nonlinear modeling of an INS position error was able to maintain the maximum position errors within the range of 10–50 m (with an rms of 5–25 m), as opposed to the range of 20–140 m (with an rms of 9–60 m) achieved by KF (with the second-order AR model).

To further demonstrate the improvements of the augmented KF–ANFIS module over the standalone KF, three of the GPS outages were chosen, and the trajectories were plotted in Figs. 25(a) and (b) and 26. Fig. 25(a) shows a typical applicability of KF–ANFIS, where GPS outage 1 was introduced in a turn and KF–ANFIS adapted to the turn fairy well, as opposed to KF alone, which digressed by a large amount. Fig. 25(b) shows GPS outage 6, where the KF–ANFIS position error was the highest among all the other GPS outages but was still much better than the second lowest error of standalone KF solution. During this GPS outage, the KF–ANFIS solution was able to provide an approximately 60% better positioning accuracy than the standalone KF. In Fig. 26, an interesting scenario can be noticed where both the KF and KF–ANFIS methods tried to obtain the vehicle position during GPS outage 3 while it was going through a curve. The KF–ANFIS solution followed the true trajectory more closely, providing an approximately 80 m better positioning accuracy than KF.

2) Trajectory 2 Results—KF–ANFIS: The augmented KF–ANFIS method was further validated using the Kingston trajectory shown in Fig. 13. As demonstrated in Fig. 27 (for the maximum position error) and Fig. 28 (for the rms position error), similar performance was obtained for this trajectory as well, with an overall accuracy improvement of about 70% over the standalone KF. This validates that the performance of the KF and the augmented KF–ANFIS modules were repeatable. Fig. 29(a) shows a flawless performance of the KF–ANFIS module during GPS outage 8, where the KF–ANFIS solution stayed within the road boundaries (which are depicted by a red/dark reference trajectory as a road map was not available for this part) for the whole 45-s outage. As shown earlier, the performance during GPS outage 9 for the Kingston trajectory was always challenging for KF while utilizing the second-order AR model. However, the augmented KF–ANFIS module outperformed the standalone KF by about 60%, with less than 40 m maximum and 16 m rms position errors during this outage as well.

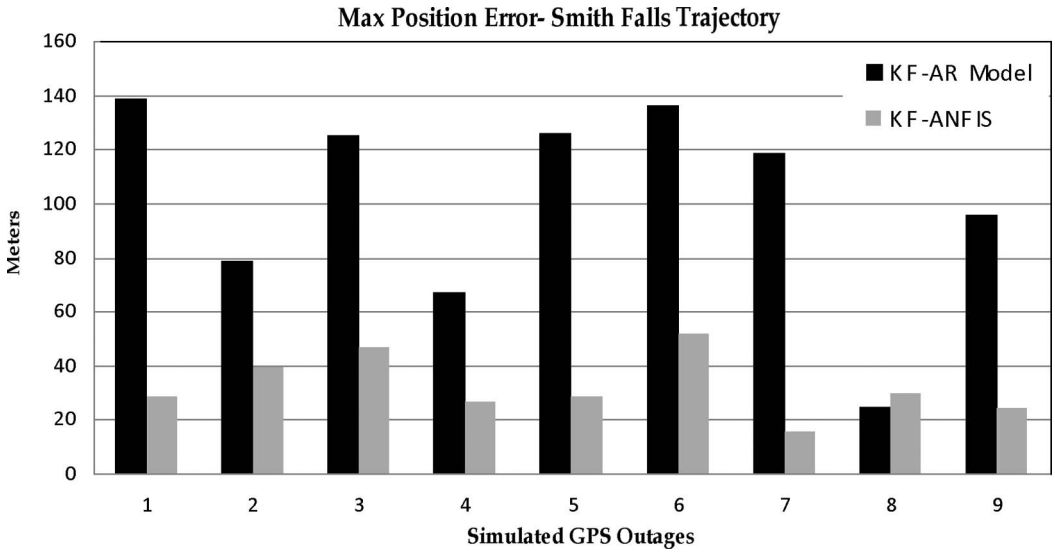

Fig. 23. Maximum position errors during 45-s GPS outages, comparing KF-ANFIS with KF using the second-order AR error model.

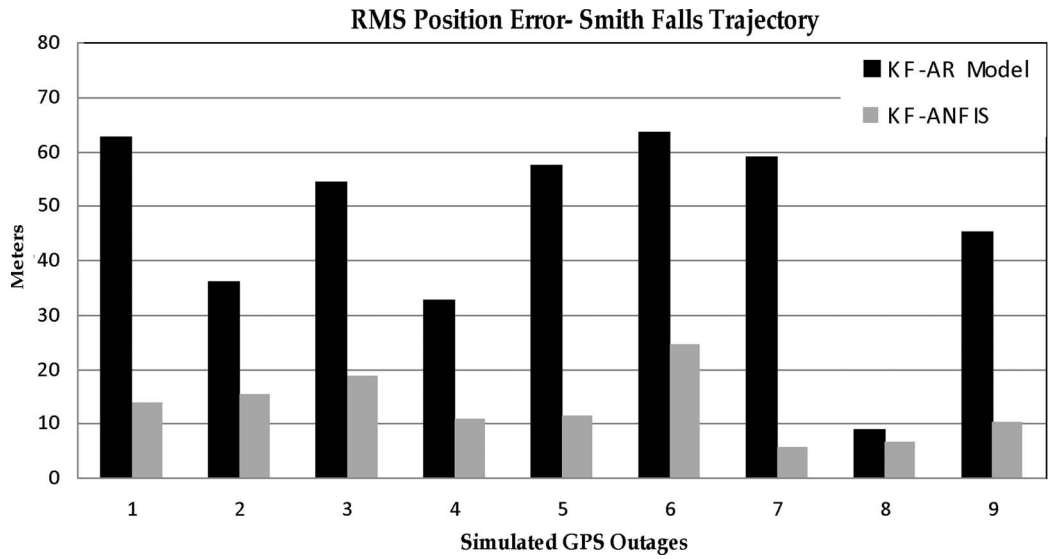

Fig. 24. RMS position errors during 45-s GPS outages, comparing ANFIS with KF using the second-order AR error model.

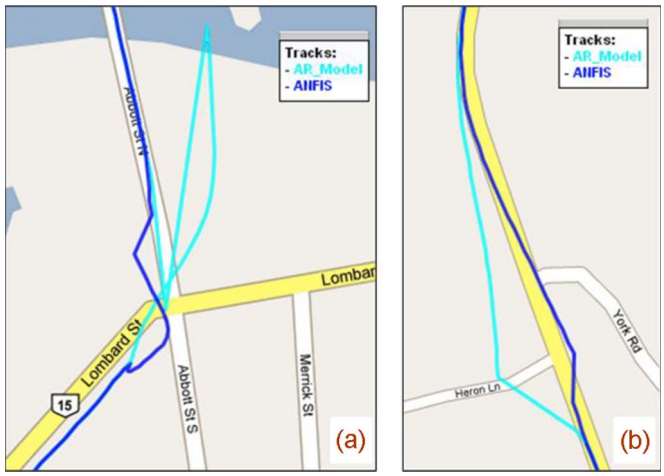

Fig. 25. (a) Performance during GPS outage 1. (b) Performance during GPS outage 6.

Fig. 26. Performance during GPS outage 3.

C. Comparison With Contemporary Research

Owing to different hardware, trajectory scenarios, duration of the GPS outages, and the variations in implementations, an

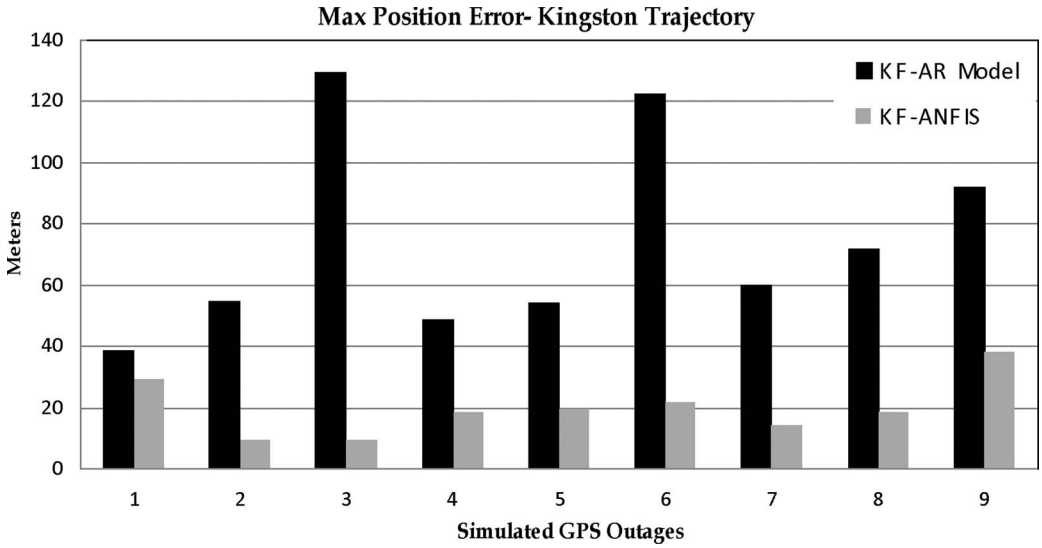

Fig. 27. Maximum position errors during 45-s GPS outages, comparing ANFIS with KF using the second-order AR error model.

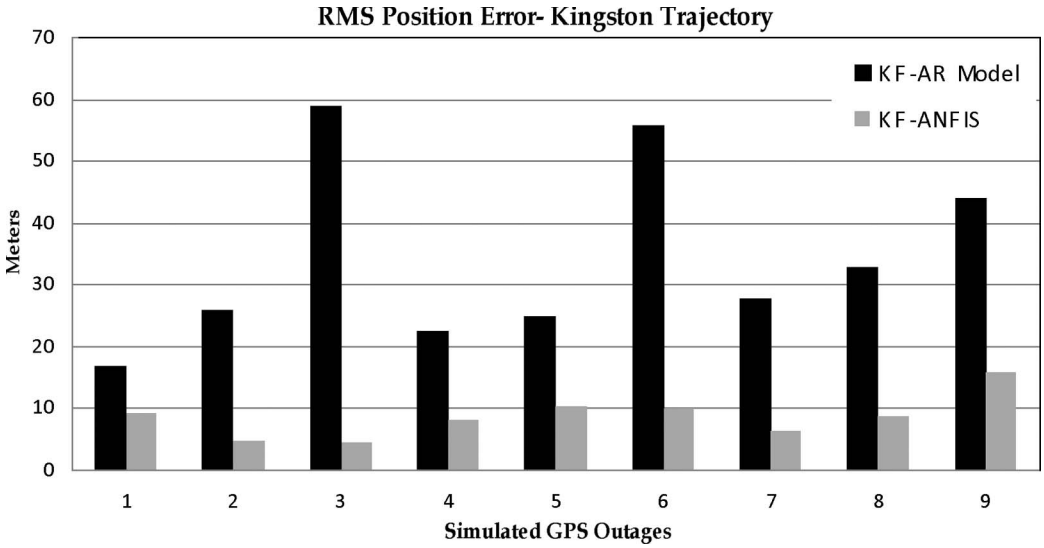

Fig. 28. RMS position errors during 45-s GPS outages, comparing ANFIS with KF using the second-order AR error model.

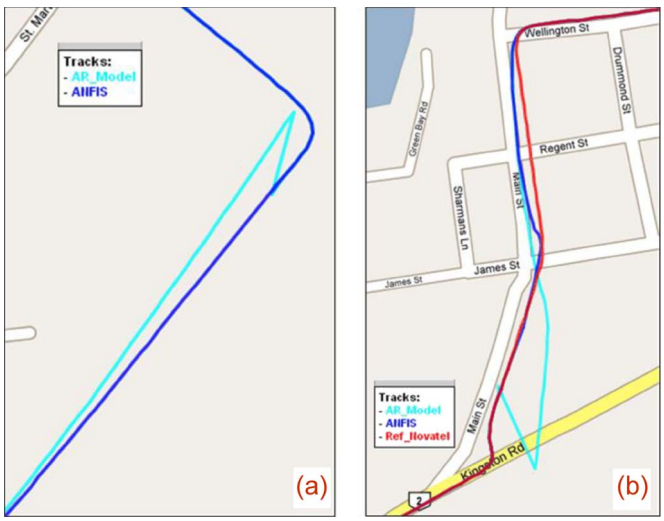

Fig. 29. (a) Performance during GPS outage 8. (b) Performance during GPS outage 9.

accurate and fair comparison is usually difficult. In this part of this paper, we compare the overall performance of our method to a couple of recently published methods in the field of MEMS-based INS/GPS.

Abdel-Hamid *et al.* [52] used ANFIS to improve the positioning accuracy during GPS outages while using MEMS IMU but employing a different architecture than the one utilized in this paper. The GPS update performed by Abdel-Hamid *et al.* [52] was based on the more accurate differential GPS instead of the single-point positioning adopted in this study. Moreover, their road test was based on only one trajectory, which was mostly straight road portions with 90° turns, as opposed to two trajectories used in this study, encompassing all the real-life scenarios. The reported accuracy improvement of their method was around 75%–80% over 30-s GPS outages, whereas our method showed a 70% accuracy improvement over 45-s GPS outages. In addition to giving a similar level of accuracy over longer GPS outages, and for more realistic trajectories, the architecture of our augmented KF–ANFIS module is simpler

and can be adopted in real time. The method published by Abdel-Hamid *et al.* [52] considers too many input parameters to the ANFIS module and, thus, requires a longer training time. Thus, it may not be suitable for real-time operation.

Goodall *et al.* [53] also employed neural networks for MEMS-based INS/GPS integration and reported an overall accuracy improvement of 25% over the standard KF approach for 60-s GPS outages. It was noted that their trajectories were relatively easy with straight-line movements and 90° turns. Moreover, they did not show consistent levels of accuracy and improvement over the conventional techniques for all GPS outages. On the other hand, the method proposed in this paper showed accuracy improvements on almost all the GPS outages introduced in both trajectories.

V. Conclusion

This paper has suggested two methods of performance enhancement of MEMS-based multisensor systems for low-cost navigation applications at two different levels. The first was at the inertial sensor level by stochastically modeling their bias errors. The second was at the information fusion level by augmenting the conventional KF module with AI for the nonlinear modeling of INS position errors.

The second-order stochastic error model adopted for each inertial sensor, coupled with gain limiting and invalid data rejection schemes, provided, in general, a 20% accuracy improvement over the conventional first-order GM modeling during several 45-s GPS outages for two different trajectories. It was also observed that the second-order AR model showed superior performance over the first-order GM model for relatively long GPS outages, which may happen in urban canyons. For short GPS outages, both models may provide a comparable level of accuracy.

Although, in the enhancement obtained after utilizing the second-order AR model, there are still significant effects of nonlinear and nonstationary INS errors that influence the overall system accuracy in the long term, particularly when driving at relatively high speeds. Additional accuracy improvements were achieved by augmenting the KF with an ANFIS module for the nonlinear modeling of INS position errors. This resulted in an approximately 64% accuracy improvement over KF that utilizes the second-order AR model for the same GPS outages. The level of accuracy obtained using the augmented KF–ANFIS module was consistent over the entire trajectory. Despite being promising, the flip side of ANFIS is its long design time and high computational burden for real-time implementation.

Future research work related to this study may focus on new architectures that can enable updating both the INS position and velocity instead of the position only. Optimizing the operation of the ANFIS module by reliable tuning of its parameters using some automated techniques, like genetic algorithms, may significantly improve the overall performance. Finally, the real-time implementation is a major challenge and requires dedicated research toward the development of an embedded module for an augmented KF–ANFIS INS/GPS information fusion of low-cost systems.

Appendix A

$$\Phi_{13} = -\frac{\dot{\varphi}}{M+h}$$
$$\Phi_{15} = \frac{1}{M+h}$$
$$\Phi_{21} = \dot{\lambda}\tan\varphi$$
$$\Phi_{23} = -\frac{\dot{\lambda}}{N+h}$$
$$\Phi_{24} = \frac{1}{(N+h)\cos\varphi}$$
$$\Phi_{36} = 1$$
$$\Phi_{41} = 2\omega_e(V_u\sin\varphi + V_n\cos\varphi) + V_n\dot{\lambda}/\cos\varphi$$
$$\Phi_{44} = \frac{-\dot{h}}{N+h} + \dot{\varphi}\tan\varphi\frac{(M+h)}{N+h}$$
$$\Phi_{45} = (2\omega_e + \dot{\lambda})\sin\varphi$$
$$\Phi_{46} = -(2\omega_e + \dot{\lambda})\cos\varphi$$
$$\Phi_{48} = f_u$$
$$\Phi_{49} = -f_n$$
$$\Phi_{51} = -V_e 2\omega_e\cos\varphi - \dot{\lambda}/\cos\varphi$$
$$\Phi_{54} = -2(\omega_e + \dot{\lambda})\sin\varphi$$
$$\Phi_{55} = \frac{-\dot{h}}{M+h}$$
$$\Phi_{56} = -\dot{\varphi}$$
$$\Phi_{57} = -f_u$$
$$\Phi_{59} = f_e$$
$$\Phi_{61} = -2\omega_e V_e\sin\varphi$$
$$\Phi_{63} = 2g/R$$
$$\Phi_{64} = 2(\omega_e + \dot{\lambda})\cos\varphi$$
$$\Phi_{65} = 2\dot{\varphi}$$
$$\Phi_{67} = f_n$$
$$\Phi_{68} = -f_e$$
$$\Phi_{73} = \frac{-\dot{\varphi}}{M+h}$$
$$\Phi_{75} = \frac{1}{M+h}$$
$$\Phi_{78} = (\omega_e + \dot{\lambda})\sin\varphi$$
$$\Phi_{79} = -(\omega_e + \dot{\lambda})\cos\varphi$$
$$\Phi_{83} = \frac{\dot{\lambda}\cos\varphi}{N+h}$$
$$\Phi_{84} = -\frac{1}{N+h}$$
$$\Phi_{87} = -(\omega_e + \dot{\lambda})\sin\varphi$$
$$\Phi_{89} = -\dot{\varphi}$$
$$\Phi_{91} = -\omega_e\cos\varphi - \frac{\dot{\lambda}}{\cos\varphi}$$
$$\Phi_{93} = \frac{\dot{\lambda}\sin\varphi}{N+h}$$
$$\Phi_{94} = -\frac{\tan\varphi}{N+h}$$
$$\Phi_{97} = (\omega_e + \dot{\lambda})\cos\varphi$$
$$\Phi_{98} = \dot{\varphi}$$

$$R_b^l = \begin{pmatrix} \cos(y)\cos(r) - \sin(y)\sin(p)\sin(r) & -\sin(y)\cos(p) & \cos(y)\sin(r) + \sin(y)\sin(p)\cos(r) \\ \sin(y)\cos(r) + \cos(y)\sin(p)\sin(r) & \cos(y)\cos(p) & \sin(y)\sin(r) - \cos(y)\sin(p)\cos(r) \\ -\cos(p)\sin(r) & \sin(p) & \cos(p)\cos(r) \end{pmatrix}$$

where

- φ latitude;
- $V_{(\cdot)}$ east, north, and up velocities;
- λ longitude;
- $f_{(\cdot)}$ acceleration from x, y, and z accelerometers;
- h altitude;
- M meridian radius.

APPENDIX B

$$^1\Phi_{3\times 9} = \begin{pmatrix} 0 & 0 & \Phi_{13} & 0 & \Phi_{15} & 0 & 0 & 0 & 0 \\ \Phi_{23} & 0 & \Phi_{23} & \Phi_{24} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \Phi_{36} & 0 & 0 & 0 \end{pmatrix}$$

$$^2\Phi_{3\times 9} = \begin{pmatrix} \Phi_{41} & 0 & 0 & \Phi_{44} & \Phi_{45} & \Phi_{46} & 0 & \Phi_{48} & \Phi_{49} \\ \Phi_{51} & 0 & 0 & \Phi_{54} & \Phi_{55} & \Phi_{56} & \Phi_{57} & 0 & \Phi_{59} \\ \Phi_{61} & 0 & \Phi_{63} & \Phi_{64} & \Phi_{65} & 0 & \Phi_{67} & \Phi_{68} & 0 \end{pmatrix}$$

$$^3\Phi_{3\times 9} = \begin{pmatrix} 0 & 0 & \Phi_{73} & 0 & \Phi_{75} & 0 & 0 & \Phi_{78} & \Phi_{79} \\ \Phi_{81} & 0 & \Phi_{83} & \Phi_{84} & 0 & 0 & \Phi_{87} & 0 & \Phi_{89} \\ \Phi_{91} & 0 & \Phi_{93} & \Phi_{94} & 0 & 0 & \Phi_{97} & \Phi_{98} & 0 \end{pmatrix}$$

R_b^l is shown at the top of the page, where

- p pitch;
- r roll;
- y yaw.

REFERENCES

[1] A. El-Rabbany, *Introduction to GPS—The Global Positioning System*. Norwood, MA: Artech House, 2002.

[2] E. D. Kaplan and C. J. Hegarty, *Understanding GPS Principles and Applications*, 2nd ed. Norwood, MA: Artech House, 2006.

[3] M. S. Grewal, L. R. Weill, and A. P. Andrews, *Global Positioning Systems, Inertial Navigation, and Integration*, 2nd ed. Hoboken, NJ: Wiley-Interscience, 2007.

[4] J. A. Farrell, *The Global Positioning System and Inertial Navigation*. New York: McGraw-Hill, 1998.

[5] R. V. C. Wong, K. P. Schwarz, and M. E. Cannon, "High-accuracy kinematic positioning by GPS–INS," *J. Inst. Navigat.*, vol. 35, no. 2, pp. 275–287, Summer 1988.

[6] J. Cao, S. Chen, Y. Li, and A. R. C. Ren Chongyu, "Design of integrated navigation system based on information fusion technology for the intelligent transportation system," in *Proc. 6th Int. Conf. ITS Telecommun.*, 2006, pp. 1248–1251.

[7] R. G. Brown and P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering: With MATLAB Exercises and Solutions*, 3rd ed. New York: Wiley, 1997.

[8] M. S. Grewal and A. P. Andrews, *Kalman Filtering: Theory and Practice Using MATLAB*, 2nd ed. New York: Wiley, 2001.

[9] G. Minkler and J. Minkler, *Theory and Application of Kalman Filtering*. Palm Bay, FL: Magellan, 1993.

[10] D. Simon, "Kalman Filtering," *Embedded Syst. Program.*, vol. 14, no. 6, pp. 72–79, Jun. 2001.

[11] P. Zarchan, *Fundamentals of Kalman Filtering: A Practical Approach*, 2nd ed. Reston, VA: AIAA, 2005.

[12] R. Toledo-Moreo, M. A. Zamora-Izquierdo, B. Ubeda-Miarro, and A. F. Gomez-Skarmeta, "High-integrity IMM–EKF-based road vehicle navigation with low-cost GPS/SBAS/INS," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 3, pp. 491–511, Sep. 2007.

[13] J. Bernstein, "An overview of MEMS inertial sensing technology," *Sensors*, vol. 20, no. 2, pp. 14–21, Feb. 2003.

[14] A. Lawrence, *Modern Inertial Technology*. New York: Springer-Verlag, 1998.

[15] D. Titterton and J. Weston, *Strapdown Inertial Navigation Technology*, 2nd ed. Reston, VA: AIAA, 2005.

[16] A. Gelb, *Applied Optimal Estimation*. Cambridge, MA: MIT Press, 1974.

[17] P. Vanicek and M. Omerbasic, "Does a navigation algorithm have to use Kalman filter?" *Can. Aeronaut. Space J.,* vol. 45, no. 3, pp. 292–296, 1999.

[18] D. Obradovic, H. Lenz, and M. Schupfner, "Fusion of sensor data in Siemens car navigation system," *IEEE Trans. Veh. Technol.*, vol. 56, no. 1, pp. 43–50, Jan. 2007.

[19] G. Dissanayake and S. Sukkarieh, "The aiding of a low-cost strapdown inertial measurement unit using vehicle model constraints for land vehicle applications," *IEEE Trans. Robot. Autom.*, vol. 17, no. 5, pp. 731–747, Oct. 2001.

[20] R. Hirokawa, K. Nakakuki, K. Sato, and R. Ishihara, "Threading the maze: GPS/INS, landmark sensing, and obstacle avoidance," *GPS World*, vol. 15, no. 11, pp. 20–26, Nov. 2004.

[21] S. Nassar, A. Noureldin, and N. El-Sheimy, "Improving positioning accuracy during kinematic DGPS outage periods using SINS/DGPS integration and SINS data de-noising," *Survey Rev.*, vol. 37, no. 292, pp. 426–438, Apr. 2004.

[22] A. Noureldin, D. Irvine-Halliday, and M. P. Mintchev, "Accuracy limitations of FOG-based continuous measurement-while-drilling surveying instruments for horizontal wells," *IEEE Trans. Instrum. Meas.*, vol. 51, no. 6, pp. 1177–1191, Dec. 2002.

[23] A. Noureldin, A. Osman, and N. El-Sheimy, "A neuro-wavelet method for multi-sensor system integration for vehicular navigation," *J. Meas. Sci. Technol.*, vol. 15, no. 2, pp. 404–412, Feb. 2004.

[24] K. W. Chiang, A. Noureldin, and N. El-Sheimy, "The utilization of artificial neural networks for multi-sensor system integration in navigation and positioning instruments," *IEEE Trans. Instrum. Meas.*, vol. 55, no. 5, pp. 1606–1615, Oct. 2006.

[25] R. Sharaf and A. Noureldin, "Sensor integration for satellite-based vehicular navigation using neural networks," *IEEE Trans. Neural Netw.*, vol. 18, no. 2, pp. 589–594, Mar. 2007.

[26] C. M. Bishop, *Neural Networks for Pattern Recognition*. New York: Oxford Univ. Press, 1995.

[27] A. Hiliuta, R. Landry, and F. Gagnon, "Fuzzy corrections in a GPS/INS hybrid navigation system," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 40, no. 2, pp. 591–600, Apr. 2004.

[28] M. M. R. Taha, A. Noureldin, and N. El-Sheimy, "Improving INS/GPS positioning accuracy during GPS outages using fuzzy logic," in *Proc. Inst. Navigat. ION GPS/GNSS*, Sep. 2003, pp. 449–508.

[29] R. Sharaf, M. Reda-Taha, M. Tarboushi, and A. Noureldin, "Merits and limitations of using adaptive neuro-fuzzy inference system for real-time INS/GPS integration in vehicular navigation," *Soft Comput.*, vol. 11, no. 6, pp. 588–598, Jun. 2007.

[30] L. Semeniuk and A. Noureldin, "Bridging GPS outages using neural network estimates of INS position and velocity errors," *Meas. Sci. Technol.*, vol. 17, no. 10, pp. 2783–2798, Oct. 2006.

[31] W. Abd-Elhamid, A. Osman, and A. Noureldin, "Wavelet multi-resolution analysis for enhancing the performance of integrated GPS and MEMS-based navigation systems," *Geomatica*, vol. 59, no. 1, pp. 61–72, Mar. 2005.

[32] J. Skaloud, A. Bruton, and K. Schwarz, "Detection and filtering of short-term ($1/f^{\gamma}$) noise in inertial sensors," *J. Inst. Navigat.*, vol. 46, no. 2, pp. 97–107, Summer 1999.

[33] K.-W. Chiang, "INS/GPS integration using neural networks for land vehicular navigation applications," Ph.D. dissertation, Dept. Geomat., Univ. Calgary, Calgary, AB, Canada, 2004.
[34] W. Abdel-Hamid, "Accuracy enhancement of integrated MEMS-IMU/GPS system for land vehicular navigation applications," Ph.D. dissertation, Dept. Geomat. Eng., Univ. Calgary, Calgary, AB, Canada, Jan. 2005.
[35] M. D. Eberts, "Performance enhancement of MEMS based INS/GPS integration for low cost navigation applications," M.S. thesis, Dept. Elect. Comput. Eng., Roy. Mil. College Canada, Kingston, ON, Canada, 2007.
[36] C. Burrus, R. Gopinath, and H. Guo, *Introduction to Wavelets and Wavelet Transforms—A Primer*. Englewood Cliffs, NJ: Prentice–Hall, 1998.
[37] Y. T. Chan, *Wavelet Basics*. Norwell, MA: Kluwer, 1996.
[38] A. Graps, "An introduction to wavelets," *IEEE Comput. Sci. Eng.*, vol. 2, no. 2, pp. 50–61, Summer 1995.
[39] R. Polikar, *The Wavelet Tutorial*. [Online]. Available: http://users.rowan.edu/~polikar/WAVELETS/
[40] C. G. Johnston, "High resolution wavelet de-noising for MEMS-based navigation systems," M.S. thesis, Dept. Elect. Comput. Eng., Roy. Mil. College Canada, Kingston, ON, Canada, 2007.
[41] A. Papoulis and S. U. Pillai, *Probability, Random Variables, and Stochastic Processes*. New York: McGraw-Hill, 2002.
[42] S. Nassar, "Improving the inertial navigation system (INS) error model for INS and INS/DGPS applications," Ph.D. dissertation, Univ. Calgary, Calgary, AB, Canada, Nov. 2003.
[43] M. Hayes, *Statistical Digital Signal Processing and Modeling*. Hoboken, NJ: Wiley, 1996.
[44] L. Jackson, *Digital Filters and Signal Processing*. Norwell, MA: Kluwer, 1975.
[45] J. Burg, "Maximum entropy spectral analysis," Ph.D. dissertation, Dept. Geophys., Stanford Univ., Stanford, CA, 1975.
[46] G. Welch and G. Bishop, *An Introduction to the Kalman Filter*. New York: ACM, 2001.
[47] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. Hoboken, NJ: Wiley, 2001.
[48] D. J. Biezad, *Integrated Navigation and Guidance Systems*. Reston, VA: AIAA, 1999.
[49] P. S. Maybeck, *Stochastic Models, Estimation, and Control*, vol. 2. New York: Academic, 1982.
[50] J. S. Rogers, C. T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. Englewood Cliffs, NJ: Prentice–Hall, 1997.
[51] K. Passino and S. Yurkovich, *Fuzzy Control*. Menlo Park, CA: Addison-Wesley, 1998.
[52] W. Abdel-Hamid, A. Noureldin, and N. El-Sheimy, "Adaptive fuzzy prediction of low-cost inertial-based positioning errors," *IEEE Trans. Fuzzy Syst.*, vol. 15, no. 3, pp. 519–529, Jun. 2007.
[53] C. Goodall, Z. Syed, and N. El-Sheimy, "Improving INS/GPS navigation accuracy through compensation of Kalman filter errors," in *Proc. IEEE 64th VTC—Fall*, 2006, pp. 1–5.

Aboelmagd Noureldin (M'98–SM'08) received the B.Sc. degree in electrical engineering and the M.Sc. degree in engineering physics from Cairo University, Giza, Egypt, in 1993 and 1997, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Calgary, Calgary, AB, Canada, in 2002.

He is a Cross-Appointment Associate Professor with the Department of Electrical and Computer Engineering, Queen's University, Kingston, ON, Canada, and the Department of Electrical and Computer Engineering, Royal Military College (RMC) of Canada, Kingston. He is also the founder and the leader of the Navigation and Instrumentation Research Group, RMC. His research interests include artificial intelligence, digital signal processing, spectral estimation and denoising, wavelet multiresolution analysis, and adaptive filtering, with emphasis on their applications in mobile multisensor system integration for navigation and positioning technologies.

Dr. Noureldin is currently the Chair of the *Alternative Integration Methods* Research Group of the International Association of Geodesy (IAG–SC 4.1). He is the recipient of Natural Sciences and Engineering Research Council, Canada Foundation for Innovation, and Ontario Innovation Trust research awards from the Government of Canada.

Tashfeen B. Karamat received the B.S. degree in avionics from the Pakistan Air Force Academy, Risalpur, Pakistan, in 1989 and the M.Eng. degree in computer engineering from Queen's University, Kingston, ON, Canada, in 2006. He is currently working toward the M.A.Sc. degree with the Royal Military College of Canada, Kingston.

He has served over 14 years in aviation in various capacities. He was involved with the repair, maintenance, and overhaul of aircraft structures, engines, and avionics equipment, including HF/UHF/VHF communication sets, direction-finding systems, auto pilots, GPS, flight instruments, motors, generators, flight-management systems, and navigation systems.

Mr. Tashfeen is a member of the Canadian Institute of Geomatics.

Mark D. Eberts received the B.Eng. degree in mechanical engineering and the M.A.Sc. degree in electrical engineering from the Royal Military College (RMC) of Canada, Kingston, ON, Canada, in 1999 and 2007. His master's research focused on sensor fusion and integration algorithms for INS/GNSS.

He is an Aerospace Engineering Officer with the Canadian Air Force. In 1999, he was with the 435 Transport and Rescue Squadron, Winnipeg, MB, Canada, as an Engineering Technical Officer for the CC130 Hercules Squadron. In 2003, he was appointed to the Executive Staff of the First Canadian Air Division Headquarters, Winnipeg, as an Executive Assistant to the Brigadier General in charge of air force logistics, training, engineering, and maintenance. In 2005, he was selected to attend postgraduate training at the RMC. He is currently an Avionics Design Engineer with the Aerospace and Telecommunications Engineering Support Squadron, Trenton, ON.

Ahmed El-Shafie received the B.Sc. and M.Sc. degrees from Cairo University, Giza, Egypt, in 1993 and 1998, respectively, and the Ph.D. degree in water resources management and planning from the Cairo University in 2003, under a collaborative academic channel program with the Department of Civil Engineering, University of Calgary, Calgary, AB, Canada.

Between 2004 and 2007, he was a Postdoctoral Fellow with the Department of Electrical and Computer Engineering, Royal Military College of Canada and Queen's University, Kingston, ON, Canada. He is currently a Senior Lecturer with the Smart Engineering System, Department of Civil and Structural Engineering, University Kebangsaan Malaysia, Bangi, Selangor, Malaysia. His research interests include artificial intelligence techniques and their applications to several engineering applications, with emphasis on hydrological processes, environmental and water resources, dam and reservoir operation, and multisensor system integration.