Intelligent Forecasting Using Dead Reckoning With Dynamic Errors

Cesar Barrios, Yuichi Motai, *Senior Member, IEEE*, and Dryver Huston

***Abstract*—A method for integrating, processing, and analyzing sensing data from vehicle-mounted sensors for intelligent forecasting and decision-making is introduced. This dead reckoning with dynamic errors (DRWDEs) is for a large-scale integration of distributed resources and sensing data intervehicle collision avoidance system. This sensor fusion algorithm is introduced to predict the future trajectory of a vehicle. Current systems that predict a vehicle's future trajectory, necessary in a network of collision avoidance systems, tend to have a lot of errors when the vehicles are moving in a nonstraight path. Our system has been designed with the objective of improving the estimations during curves. To evaluate this system, our research uses a Garmin 16HVS GPS sensor, an AutoEnginuity OBDII ScanTool, and a Crossbow three-axis accelerometer. Using Kalman filters (KFs), a dynamic noise covariance matrix merged together with an interacting multiple models (IMMs) system, our DRWDE produces the future position estimation of where the vehicle will be 3 s later in time. The ability to handle the change in noise, depending on unavailable sensor measurements, permits a flexibility to use any type of sensor and still have the system run at the fastest frequency available. Compared with a more common KF implementation that runs at the rate of its slowest sensor (1 Hz in our setup), our experimental results showed that our DRWDE (running at 10 Hz) yielded more accurate predictions (25%–50% improvement) during abrupt changes in the heading of the vehicle.**

***Index Terms*—Collision avoidance, course correction, dead reckoning, global positioning system, Kalman filters, road vehicles, sensor fusion.**

I. Introduction

SENSOR fusion and tracking techniques have potential applications for the vehicle and the infrastructure as introduced in [1], something we can appreciate from the industrial sensing intelligence found in intelligent transport systems (ITSs) area [2]. The overall function of ITS is to improve decision-making, often in real time, improving the operation of the entire transport system. This can range from systems with intelligent route planning implemented to avoid some specific type of traffic from certain areas [3], to registering the position of vehicle-borne sensors for infrastructure assessment [4], to systems designed to prevent collisions between the users [5]. This research could fall under the latter category.

A collision avoidance system is only as good as its accuracy in warning the driver—either human or automated. An accurate system will minimize the number of false warnings, so the driver takes them seriously. Designing the architecture of this type of system involves using many sensors, for intelligent control and decision-making, and finding the right balance among the number of sensors implemented, type, and their overall contributions to the intelligent forecasting system.

There are mainly two types of designs for a collision avoidance system. Self-sufficient systems are those that can obtain enough information by themselves, such as those in [6]–[8]. Interactive systems are those that interact with the infrastructure or other vehicles to detect a dangerous scenario, such as researched in [9]–[11], where their systems send spatial information to nearby vehicles to judge the possibility of a collision in the future. While self-sufficient systems are limited to line-of-sight detection, the interactive systems are not limited by this but are more complex. Estimating the future trajectory of a vehicle requires multiple sensors that need to be merged together and put through a set of prediction models.

Multisensor data fusion (MSDF) techniques are required to combine and process data [12], [13]. This has been traditionally performed by some form of Kalman [14] or Bayesian filters [15]; however, in recent years, there has been a trend toward the use of soft techniques such as fuzzy logic and artificial neural networks [16], [17]. Furthermore, there can be two ways of setting up an MSDF system: 1) centralized; or 2) decentralized. A centralized approach suffices for most common scenarios where the sensors are synchronous, but a decentralized approach is convenient when the sensors should be treated independently [18]–[22], as with asynchronous sensors.

In [23], the authors discuss the optimal asynchronous track fusion algorithm (OATFA), which evolved from their earlier research on an asynchronous/synchronous track fusion (ASTF) [24]. They use the interacting multiple model (IMM) algorithm, but replace the conventional Kalman filters (KFs) with their OATFA. The OATFA treats each sensor separately, passing the output from each to a dedicated KF, departing from the idea that the best way to fuse data is to deliver it all to a central fusion engine. The paper's IMM-OATFA results show position estimation errors that are about half of conventional IMM setups. However, as pointed out by the same authors in

Manuscript received August 31, 2015; revised November 20, 2015; accepted December 23, 2015. Date of publication January 01, 2016; date of current version December 06, 2016. This work was supported in part by the United States Department of Transportation through the University of Connecticut Transportation Research Center, the CAREER award 1054333 from the National Science Foundation. Paper no. TII-15-1351.

C. Barrios is with the Electrical Engineering Department, University of Vermont, Burlington, VT 05405 USA (e-mail: cesar.barrios@uvm.edu).

Y. Motai is with the Electrical Engineering Department, Virginia Commonwealth University, Richmond, VI 23284 USA (e-mail: ymotai@vcu.edu).

D. Huston is with the Mechanical Engineering Department, University of Vermont, Burlington, VT 05405 USA (e-mail: dryver.huston@uvm.edu).

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/TII.2015.2514086

[25], all measurement data must be processed before the fusion algorithm is executed. Similarly, the authors of [26] create asynchronous holds, where, from a sequence of inputs sampled at a slow sampling rate, it generates a continuous signal that may be discretized at a high sampling rate. Despite the benefits of making the asynchronous system into a synchronous one by using these methods, restrictions are observed where, if for some reason, a sensor is delayed in providing its data or is offline for a few cycles, the whole system needs to wait.

In [27]–[29], the authors also look into problems of getting measurements from multiple sensors, but they focus on measurements being out-of-sequence and not on missing measurements. Therefore, while this is a very important topic on some scenarios, for the system that was used in this study, having all the sensors being processed locally, it will be assumed that all measurements are in the correct sequence, and there should not be a reason for some of them getting out-of-sequence.

Another method to fuse asynchronous sensors is discussed in [30]. In this paper, the authors synchronize the output of the sensors by estimating the data of the slower sensors for the time stamps where no data were available from them. Even though the method used to estimate the unavailable readings is very rudimentary (based only on the previous reading), this idea allows the system to run at the fastest frequency of its sensors. This difference, compared to the previously referenced papers, allows the system to make any corrections as soon as data are available, making its estimations more accurate in some scenarios.

The contribution of this paper is a dead reckoning (DR) system that runs at the frequency of its fastest sensor to update its prediction as soon as a change is detected. The difference from other DR implementations, subject to cumulative errors, is that our dead reckoning with dynamic error (DRWDE) continually updates the noise covariance matrices when any sensor remains offline by innovating a dynamic Q matrix in the KFs. This constant modification of the true weight of each measurement helps to counteract the cumulative error of the DR when the measurements are estimated and not real.

II. Position Estimation Techniques

The KF [31] was first proposed in the 1960s and has been shown to be a form of Bayesian filter [32]. From a series of noisy measurements, the KF is capable of estimating the state of the system in a two-step process: 1) correct; and 2) predict [33]–[35].

The KF has a long history of accurately predicting future states of a moving object and has been applied to many different fields [36]–[39], including transportation, which is why it was chosen for this research. The elements of the state vector used ($\boldsymbol{x}$) are: the position, velocity, and acceleration of the vehicle; all available from the different sensors. Keep in mind that the position (xv) and velocity (vv) components of the state estimate have an x and y component to them (east–west and north–south directions), and the acceleration (av) has an n and t component to it (normal and tangential acceleration). So, the state vector matrix will be $X = (xx, xy, vx, vy, an, at)$.

The estimated error covariance (P) for the state estimate is based on the relationships between each of the elements to the others. The error covariance matrix is a dataset that specifies the estimated accuracy in the observation errors between all possible pairs of vertical levels.

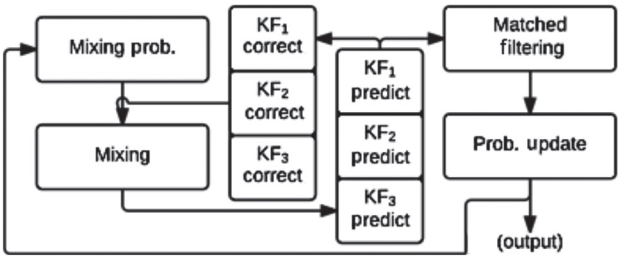

Fig. 1. Flowchart for the three KF in an IMM framework.

Together with P, the Jacobian matrix of the measurement model (H), the measurement noise covariance (R), and with the measurement noise (σ_m), are used to calculate the Kalman gain (K). Once the K is calculated, the system looks at the measured data (Z) to identify the error of the predicted position obtained from the dynamic models defined, and uses it to adjust P.

The IMM framework was used in this system [44], as shown in Fig. 1. It can calculate the probability of success of each model at every filter execution, providing combined solution for the vehicle behavior [40]–[42]. These probabilities are calculated according to a Markov model for the transition between maneuver states, as detailed in [43]. To implement the Markov model, it is assumed that at each execution time, there is a probability p^{ij} that the vehicle will make a transition from model state i to state j.

III. DRWDE Using KFs

This system uses three different sensors: 1) a Garmin 16HVS GPS receiver and Fugawi 3 GPS navigation software; 2) an AutoEnginuity OBDII ScanTool (which obtains the velocity from the vehicle's internal system); and 3) a Crossbow three-axis accelerometer. This set of sensors offers data at different rates (asynchronous) and also at the same rates (synchronous); one measurement from two of the sensors overlap (homogeneous) but most of them do not (heterogeneous). The accelerometer measures normal and tangential acceleration every tenth of a second, the ScanTool measures velocity every 1 s, and the GPS measures position, velocity, and heading every 1 s (timing precise).

A problem with some of the existing research, as mentioned in Section I, is that sensors can unexpectedly go offline and not provide data when expected. The system in this study will need to handle this without slowing down the running frequency of the overall system and then wait for the sensor to come back online. This in turn means that the system can run at the frequency of its fastest sensor. If the system can continue to run and handle the missing data, it will allow for a quicker correction of the estimation if a change occurs in the spatial movement of the vehicle.

A. System Architecture

In this setup, the GPS sensor provides the location (s_x, s_y), the velocity (v), and the angle of direction (β) using north as the zero. Then the ScanTool sensor provides the velocity (v),

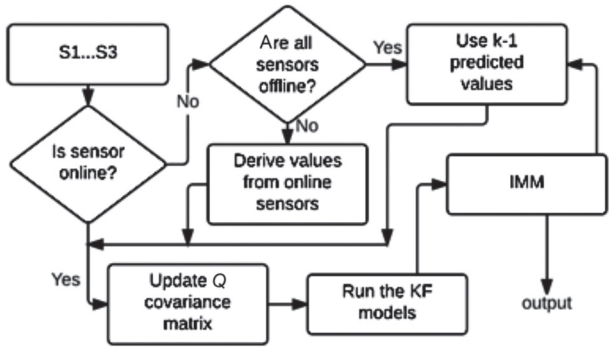

Fig. 2. Flowchart of our DRWDE system.

and the accelerometer provides normal acceleration (a_n) and tangential acceleration (a_t). The jerk j (acceleration change) in this study's equations is included as the factor responsible for the noise in the measurements; therefore, the jerk term is represented as the prediction noise (σ_p). The different linear dynamic models to be used in the KF used in this research are defined as follows.

1) Constant location (CL)

$$\begin{aligned} s(k) &= s(k-1) + \sigma_{p_s} \\ v(k) &= 0 \\ a(k) &= 0. \end{aligned}$$

2) Constant velocity (CV)

$$\begin{aligned} s(k) &= s(k-1) + v(k-1)\cdot\Delta k + \sigma_{p_s} \\ v(k) &= v(k-1) + \sigma_{p_v} \\ a(k) &= 0. \end{aligned}$$

3) Constant acceleration (CA)

$$\begin{aligned} s(k) &= s(k-1) + v(k-1)\cdot\Delta k \\ &\quad + \frac{1}{2}a(k-1)\cdot(\Delta k)^2 + \sigma_{p_s} \\ v(k) &= v(k-1) + a(k-1)\cdot\Delta k + \sigma_{p_v} \\ a(k) &= a(k-1) + \sigma_{p_a}. \end{aligned}$$

In the flow of this setup (Fig. 2), when a sensor goes offline and the data needed for the models are not present, e.g., velocity, the missing data are derived from the data obtained by the remaining online sensors, making this estimation more accurate than only using the offline previous reading of the sensor to estimate what would be its current value. This is insufficient, however, as the longer a sensor remains offline, the more noise is accumulated in the estimation of its value, which in turn affects the overall prediction of the future spatial location of the vehicle. To handle this properly, we have to modify dynamically the noise covariance matrices.

B. Q Matrix in the KF

The process noise covariance (Q) of the KF is defined based on the estimated prediction noise (σ_p). A simple approach is made to estimate Q using an extensive dataset of common scenarios. For this system, because this research wanted to be able to handle sensors going offline at any given time and for any given period of time, an innovative method was devised that makes the Q matrix dynamic, allowing the noise to vary depending on the number of iterations the different variables go through without getting an actual measurement from the corresponding sensor.

1) Mathematical Framework for Improvement: A discrete and dynamic linear system and measurement equation can be generally expressed as shown below, where k is the current instance and $k+1$ is the future instant for which the data are being estimated. For the linear discrete system, x is the state, F is the state transition function, and B is the control input function. For the measurement equation, Z is the sensor reading, H is the sensor function, and n is the sensor noise covariance

$$\begin{aligned} x_{k+1} &= F_k \cdot x_k + B_k \cdot u_k + w_k \\ Z_{k+1} &= H_{k+1} \cdot x_{k+1} + n_{k+1}. \end{aligned}$$

Given the intermediate data for the instant t_k between the instances when all sensors are online (t_i and t_{i+1}), it is possible to make a prediction for the instant t_{k+1} based on the data at instant t_k, which will most probably result in a better prediction than if using the instant t_i. There are two possible approaches to handle the missing data when sensors are offline.

The first option is to fill in the missing measurements with those of $\hat{x}_{k+1}$, which is the prediction to the intermediate instant. The risk for the minimal quadratic error [45] is $(\hat{x} - E(x))^t \cdot M\,(\hat{x} - E(x)) + \mathrm{trace}\,[\mathrm{MP}]$, where M is the defined positive matrix of the quadratic error, and $P = E\ (\hat{x} - E(x)) \cdot (\hat{x} - E(x))^t$, with the corresponding reduction in the actual measurements when sensors are online.

The second option is to calculate, with the current data obtained from the online sensors, the noise errors for the given small time interval, and obtain a better approximation of the missing measurements, with the goal of obtaining a better Q covariance matrix.

In the first option, the error will generally be greater, hence the interval is greater $\hat{x}$ and $E(x)$. In the second option, there may not always be a relationship that will yield a good estimation, but experimental runs can help evaluate this approach to determine if the estimation is indeed better.

As proven in Appendix I, a smaller trace of the Q matrix would suppose a general improvement in the covariance matrix of the process, and therefore the resulting estimation. However, if a sufficiently general condition is required, then there is a need to study the matrix $E\ m_{k+1}m_{k+1}^t$ for each specific case, where m_k represents the status of a specific sensor at a given instant.

To approximate the unknown magnitudes, if $x = (x_0 x_1 \ldots x_n)^t$ verifies that $x_{l+1} = \dot{x}_l\ \forall l = 0, 1, \ldots, n-1$, and x_n is the function we have for known measurements in the intermediate instances, it is possible to approximate any x_p for $p = 0, 1, \ldots, n-1$ through a Taylor polynomial as shown in Appendix II.

With the new data obtained from the online sensors, the process can be repeated for the next intermediate instances $t_{k+1}, t_{k+2}, \ldots$, which, in general, the error will continue to increase as the time gap increases. The exact value of the errors will be unknown in general, so this research will have to be bounded through statistical estimations; even though, in reality, the actual implementation, and not the theoretical validation of the formulas, will determine if there is an improvement in the estimations. For this, it must be taken into account that, due to the cumulative error accumulated with each iteration, an excessive number of iterations will be counterproductive, and will make the estimations worse.

In the case that the function of the more frequent known measurements x_q is not x_0 or x_n, it will suffice to consider on one hand $(x_0 \ldots x_q)$, and on the other hand $(y_0 \ldots y_m) = (x_0 \ldots x_m)$, and proceed with each group accordingly. If there were more functions with known measurement data, in general, the remaining would be estimated using the closest one, or one of the closest ones.

2) Dynamic Process Noise Covariance (Q): In the case when all the sensors are available, the formulas for the *CA* models will depend on the location, velocity, and acceleration measurements in a given instant, and also will depend on the prediction noise σ_p. In this case, σ_p is based on the jerk (j), which will have a variable and unknown value. Based on the Lagrange form of the remainder of Taylor's formula, there is a value for j which will yield the exact measurements. Therefore, to set an upper bound of the expected value (E), it suffices to identify an upper bound for j, and calculate the corresponding integrals to obtain each E. However, because in a real-time execution of the system all the values of j are not known ahead of time, this research made it a moving range, so the system can dynamically tune itself.

In summary, to determine $Q = E[\sigma_p \sigma_p^T]$, this research starts by defining j_k (acceleration change) as the least upper bound (supremum) of the dataset collected so far, i.e., $\max\{|j_{t_k}|, |j_{n_k}|, |j_{t_{k-1}}|, |j_{n_{k-1}}| \ldots |j_{t_{k_0}}|, |j_{n_{k_0}}|\}$.

If the state vector defined in Section II-A and the Kalman models defined in Section III-A are used, and if j_n is to the right of j_t, and for the *CA* model (3), $x(k) = F(k) \cdot x(k-1) + \sigma_p$ has

$$x(k) = \begin{bmatrix} 1 & \Delta k & \frac{1}{2}(\Delta k)^2 \\ 0 & 1 & \Delta k \\ 0 & 0 & 1 \end{bmatrix} \cdot x(k-1) + \begin{bmatrix} \frac{1}{6} j (\Delta k)^3 \\ \frac{1}{2} j (\Delta k)^2 \\ j(\Delta k) \end{bmatrix}. \quad (4)$$

Furthermore, in this system, it will also take into account the error in the estimations for location, velocity, and acceleration when the sensor providing the corresponding value is offline, and consider for how long it has been offline.

Now, given $M_k(x)$ as the total measurement error of a variable x such that in the step when all sensors are online $m = 0$, and in the following m step(s), only the accelerometer sensor is online. Because sensors can go offline independently of each other, a different m is needed to identify each sensor: m_1 for the GPS sensor, m_2 for the ScanTool sensor, and m_3 for the accelerometer.

Therefore, this research can now define Q for the CA model as follows:

$$Q_{CA} = \begin{bmatrix} M(s)^2 & M(sv) & M(sa) \\ M(vs) & M(v)^2 & M(va) \\ M(as) & M(av) & M(a)^2 \end{bmatrix}. \quad (5)$$

Therefore, each of the process error elements in the Q matrix can be derived. For the position elements (x/y), it is obtained that $E[M^2(s)] \leq \frac{(\Delta k)^6}{36} \sum_{i=0}^{m_1} j_{k-i}^2$, with the details shown in Appendix III. Then, using a similar approach, it was found that for the velocity elements (x/y) $E[M^2(v)] \leq \frac{(\Delta k)^4}{4} \sum_{i=0}^{m_1} j_{k-i}$, and, finally, for the acceleration elements (n/t), it was derived $E[M^2(a)] \leq (\Delta k)^2 \sum_{i=0}^{m_1} j_{k-i}^2$. Moreover, for the nonzero elements outside of the diagonal, it was calculated that $E[M(s \cdot v)] = E[M(v \cdot s)] \leq \frac{(\Delta k)^5}{12} \sum_{i=0}^{m_1} j_{k-i}^2$.

For a given tangential or normal acceleration, the locations and velocities in the axis directions can be any; therefore, the location and velocity variables are independent from the value of the tangential or normal accelerations. Similarly, the tangential and normal accelerations are independent of each other. Therefore, the expected value of those errors is zero, and the final Q matrix that will dynamically increase the corresponding measurement error in relation to how long some sensors (m_i) have been offline (Δk) is defined as follows:

$$Q_{CA} = \begin{bmatrix} \frac{(\Delta k)^6}{36} \sum_{i=0}^{m_1} j_{k-i}^2 & \frac{(\Delta k)^5}{12} \sum_{i=0}^{m_1} j_{k-i}^2 & 0 \\ \frac{(\Delta k)^5}{12} \sum_{i=0}^{m_1} j_{k-i}^2 & \frac{(\Delta k)^4}{4} \sum_{i=0}^{m_1} j_{k-i} & 0 \\ 0 & 0 & (\Delta k)^2 \sum_{i=0}^{m_1} j_{k-i}^2 \end{bmatrix}. \quad (6)$$

Using a similar approach as shown above, this research can derive the dynamic Q matrix for the *CV* model as follows:

$$Q_{CV} = \begin{bmatrix} \frac{(\Delta k)^4}{4} \sum_{i=0}^{m_1} a_{k-i}^2 & \frac{(\Delta k)^3}{2} \sum_{i=0}^{m_1} a_{k-i}^2 \\ \frac{(\Delta k)^3}{2} \sum_{i=0}^{m_1} a_{k-i}^2 & (\Delta k)^2 \sum_{i=0}^{m_1} a_{k-i}^2 \end{bmatrix}. \quad (7)$$

Also, for the *CL* model

$$Q_{CL} = (\Delta k)^2 \sum_{i=0}^{m_1} v_{k-i}^2. \quad (8)$$

These Q matrices will be used in the KF prediction step to estimate the error covariance for each of the models. Also, as shown in the Q matrices above, the moment a sensor comes back online $(m_i = 0)$, the corresponding element in the dynamic Q matrix can be reset to its minimum value.

IV. Evaluation Criteria

To verify the improvements of using the DRWDE, we will implement and compare the results of the following setups:

1) synchronous sensors using a common KF + IMM implementation (GPS at 1 Hz, ScanTool at 1 Hz, and accelerometer at 1 Hz);
2) asynchronous sensors using our dynamic DRWDE implementation (GPS at 1 Hz, ScanTool at 1 Hz, and accelerometer at 10 Hz).

The first setup is to get the IMM working at 1 Hz, which will only run when all sensors are online, therefore not really using the dynamic part of the Q matrix.

The DRWDE setup is to increase the frequency of the system to 10 Hz to try to take advantage of all the readings from the accelerometer and try to correct the predictions sooner, instead of having to wait for all sensors to come back online, as in this first setup. This second setup uses the dynamic Q matrix technique described in Section III-B to account for the error in the estimation of the data when some sensors are offline.

Using the above two setups helps to track improvements to the overall trajectory prediction when the frequency of the system increases along with the proper handling of the accumulated error in the predictions. If this DRWDE is flexible enough to handle all the different synchronous and asynchronous, homogeneous and heterogeneous data from the sensors in use, improvements should be seen on the predicted future locations, and the system should be able to detect and correct a spatial change in the vehicle much sooner than when the system is forced to run at the speed of its slowest sensor.

The evaluation criteria will be based on comparing the actual prediction errors for both the DRWDE and the IMM 1 Hz systems against the true location data obtained from the GPS receiver. Both systems will be run through the same trajectory, and the results looked at in several different ways. First, this research will look at the average prediction error for whole trajectory, but then also separate the trajectory into straight lines, smooth curves, and sharp curves, to better evaluate both systems in the different scenarios. This research will also select one specific smooth curve and one specific sharp curve, and it will look at those results in greater detail, using error histograms and calculating root-mean-square (RMS) and mean absolute percentage error (MAPE) values using the actual and predicted position S of both systems

$$RMS = \sqrt{\frac{\sum_{k} (S_k - S_{k-3})^2}{(k - k)}}, \quad \text{MAPE} = \frac{1}{n} \sum_{t} \left| \frac{A_t - F_t}{A_t} \right| 100.$$

V. Experimental Performance of the DRWDE System

A. Dataset Characteristics

The dataset consists of measurements from the three sensors while driving a vehicle for over 1 h. The trajectory followed is shown in Fig. 3, where the vehicle followed the route marked in red.

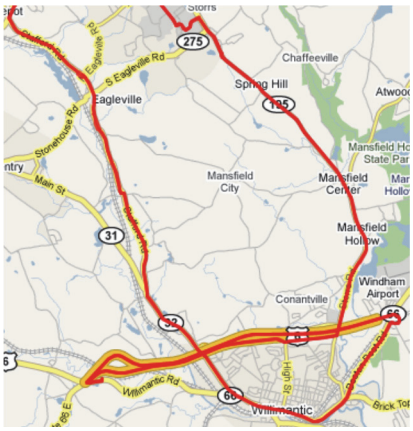

Fig. 3. Map of overall trajectory in Mansfield City, CT (Google Maps).

For this experiment, the GPS sensor takes measurements of the current geographical coordinates in degrees, heading in degrees, and velocity in miles per hour every 1 s. These measurements were converted to meters, radians, and meters per second, respectively.

The ScanTool reads the measurements of the velocity determined by sensors coupled to the wheels of the vehicle in miles per hour every 1 s. This measurement is more accurate than the one obtained from the GPS, so it is used instead of the one from the GPS (except when it is not available).

The last sensor used in this experiment is an accelerometer, which takes measurements of the normal and tangential accelerations in volts every 0.1 s. Using a calibration formula provided by the manufacturer of the sensor, the conversion is units to meters per second squared.

The trajectory selected for this research is shown in Fig. 3. The selected route was selected to include straight and nonstraight paths, and also road types driven at different speeds, such as highways, local routes, small streets, and even going through town with several traffic lights and stop signs. This dataset includes most possible scenarios a vehicle could be traveling through.

To be able to create a useful dataset of the data recorded from the trajectory shown in Fig. 3, this research had to create scripts to map the values from the log file of each sensor to each other, using the timestamp as the common reference between them. In the end, a dataset was created with all the desired measurements in columns, with all available readings in a row for each timestamp. Because only the accelerometer works at 10 Hz, many of the rows only contain acceleration measurements, and this is where the system comes into action and takes advantage of these extra measurements. Table I shows the average and standard deviations of the data used, to take a general look at the characteristics of the dataset worked with.

For this experiment, the focus was on predicting a trajectory when the vehicle is going through curves, which are the more problematic areas. To be able to evaluate this better, the dataset of the whole trajectory was classified into straight lines, smooth curves, and sharp curves. To determine whether a set of consecutive points in the trajectory was a curve or a straight line, the change in the heading after a period of 2 s was observed; if it

TABLE I
REPRESENTATIVE DATASETS

	Distance (m)	Velocity (m/s)	Acc.norm (m/s^2)	Acc.tang (m/s^2)
Whole trajectory	16.34	15.20	−0.44	0.69
	±6.97	±5.86	±1.10	±0.58
Smooth curves	19.04	17.66	−1.69	0.04
	±5.58	±4.97	±1.72	±0.53
Sharp curves	10.59	9.36	0.37	1.19
	±6.84	±5.78	±1.34	±0.58

Values represent median ± standard deviation of all data points used.

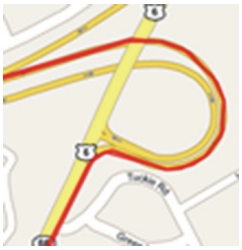

Fig. 4. Map of selected turn for testing (Google Maps).

was more than 2°, then it was defined as a curve. Moreover, to determine if the curve was a sharp one, the change had to be greater than 10°; otherwise, it was defined as a smooth curve.

Moreover, the system to estimate a future position of a vehicle will be set up to determine where the vehicle will be 3 s later, which is based on the average human reaction time of 1.5 s to stop a vehicle [46]. In reality, this value would also need to take into account the vehicle's weight and speed to properly determine minimum stopping time necessary.

Looking at Table I, it can be seen that the dataset used for this experiment agrees with how a vehicle would be driven under normal conditions. For example, the standard deviations are not very different from each other for the distance and velocity measured by the sensors, which is expected, as the values do not change much from one point to the next for an average vehicle driving on normal roads. The average for distance and velocity is smaller for the smooth curves than for the sharp curves, which means that the vehicle's speed is more constant through the smooth curves than the sharp curves. The change in movement for sharp curves agrees with how a vehicle would behave in such a scenario, as it will usually have to slow down considerably while turning and then accelerate again as the driver gets a handle on the curve.

Since the main problem with trajectory estimation is during curves based on research reviewed in Section I, this research selected a specific curved scenario from Fig. 3 and use that dataset to evaluate the DRWDE system and its performance.

The section of the trajectory shown in Fig. 4 was selected because it has a sharp curve and then a smooth constant curve, which should be a good scenario to test if the system can correct its prediction when the vehicle enters the curve, and maintain it through the whole curve. Sharper curves allow our dynamic system to be tested properly as the curve ends up being very short and does not allow a slower system to estimate a trajectory during the actual turn if it lasts only a few seconds. The "selected smooth curve" refers to the longer curve in Fig. 4 (~30 s of data), and the "selected sharp curve" represents the small curve (bottom left) shown in Fig. 4 as well (~ 10 s of data).

TABLE II
AVERAGE PREDICTION ERROR

	DRWDE (3 s)	DRWDE (5 s)	IMM 1Hz (3 s)	IMM 1Hz (5 s)
Whole trajectory	2.719 ±2.030	5.844 ±3.237	3.044 ±1.800	5.854 ±3.193
All smooth curves	2.811 ±1.925	5.633 ±3.825	2.972 ±1.737	5.357 ±3.282
All sharp curves	3.236 ±2.844	5.063 ±3.175	4.456 ±3.307	5.270 ±3.003
Selected smooth curve	3.051 ±1.173	5.892 ±3.200	3.212 ±1.205	5.483 ±3.124
Selected sharp curve	2.277 ±2.388	4.210 ±1.442	4.090 ±2.241	5.093 ±2.981

Values represent median prediction error in meters ± standard deviation of all data points used for both 3- and 5-s-ahead predictions.

The DRWDE setup for this experiment, as explained in Section III, runs at the frequency of its fastest sensor (10 Hz), and uses the dynamic matrices accounting for the accumulated noise of the missing measurements. Moreover, as mentioned in Section IV, data will be running through a common IMM implementation (synchronous sensors) to be able to compare results to the DRWDE setup.

Since the common IMM can only run at the frequency of its slowest sensor, this research defined Δk to be 1 s (1 Hz), and, because all sensors are available during each iteration of the system, this setup does not utilize the dynamic portion of the Q matrix defined in section B.

Moreover, to properly compare this run to the 10 Hz run, it cannot be assumed that the vehicle would move in a straight line between each second, so ten intermediate points between each second based on the dynamics of the vehicle were defined. This allows us to more accurately compare both runs visually.

B. Evaluation of the Prediction Error

Following the evaluation criteria defined in Section IV, the data recorded from the trajectory shown in Fig. 3 were executed through both systems. The results for the overall trajectory, all smooth and sharp curves, and the selected smooth and sharp curves were recorded in Table II. Keep in mind that the DRWDE is running at 10 Hz, where only the accelerometer can provide a measurement at every system iteration, while the GPS and ScanTool provide only one reading every ten iterations, leaving it to our dynamic Q implementation to account for the accumulated error in predicting these missing measurements.

Table II shows the average prediction errors for both the DRWDE run and the IMM 1 Hz run for broader scenarios as well as for our selected curves. If the results of 3-s-ahead prediction for the whole trajectory were observed, only a negligible improvement was seen, as expected, since the number of sharp curves in the whole trajectory is very small. Similarly, there is almost no improvement if all smooth curves in the trajectory were observed when compared to the IMM 1 Hz. However, since the DRWDE was created to react quickly to changes, it was observed that when taken into account all sharp curves,

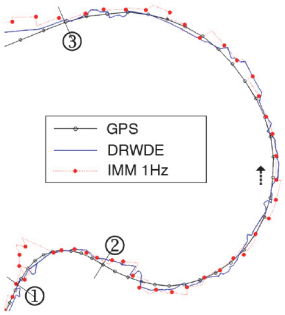

Fig. 5. Comparison between actual path (GPS) and predicted paths by both systems (DRWDE 10 Hz and IMM 1 Hz) for the selected curves. Sharp curve between (1) and (2) and smooth curve between (2) and (3). Direction of movement shown by arrow.

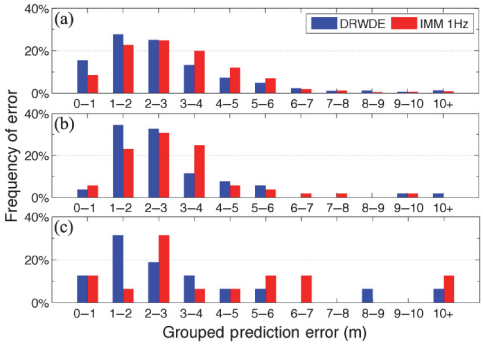

Fig. 6. Frequency of each system's 3-s-ahead prediction error. Top (a) uses data from the whole trajectory; middle (b) only represents data from all smooth curves; and bottom (c) represents data from all sharp curves.

improvements to the 3-s-ahead estimation were seen that are considerably greater for the DRWDE system (3.2 versus 4.5 m).

If now the focus is on the selected smooth and sharp curves for the 3-s-ahead prediction, the result of the IMM run at 1 Hz is shown in Fig. 5. The red-dotted line shows the predicted location every second (red dots) and the intermediate points derived in between each second (dotted line) to simply show visually what may be happening in between each second.

Moreover, Table II shows results for 5-s-ahead predictions. As expected, the earlier in time a position is predicted, the more errors there will be as well as the less reliable to prediction is, as shown by the larger standard deviation values for the estimation errors.

Now, for the DRWDE run, Δk was defined to be 0.1 s, which is the period of its fastest sensor (10 Hz). Since only the accelerometer runs at 10 Hz, there will be many system iterations where the other sensors will be offline, and this is where the dynamic Q variance introduced in Section III-B comes into play. The result of the DRWDE run at 10 Hz is also shown in Fig. 5, as the blue solid line.

Fig. 5 displays the actual trajectory of the vehicle represented by the GPS line, and then the predicted locations 3 s earlier in time by both the IMM 1 Hz run and the DRWDE 10 Hz run (prediction performance is shown later in Fig. 7). It can be observed that both the 1 Hz and the 10 Hz runs behave somewhat similarly during the smooth curve; this is also represented quantitatively in Table II.

The average error in the predicted locations during the selected smooth curve is only slightly better for the DRWDE (3.0 versus 3.2 m). The benefits are clearly seen in the selected sharp curve, where the average error is much lower for the DRWDE (2.3 versus 4.1 m). Looking at Fig. 5, it can be seen that, as the vehicle enters the sharp curve (bottom left), the slower system (red dotted line) is estimating its location to be in more of a straight line, as the vehicle is traveling in a straight line before taking the exit ramp (see Fig. 4). It can even be seen that there are three red dots (each dot represents 1 s) before the system realizes that the vehicle is turning and can adjust its 3-s-ahead prediction accordingly. Looking at the blue line representing the DRWDE run, it can be seen that its line is a lot closer to where the vehicle actually moves through 3 s later in time. The DRWDE 10 Hz system is able to react and correct its future prediction much quicker, using its dynamic covariance matrices to take into account how long a measurement has not been corrected by an actual sensor. As shown in Table II, in the selected sharp curve, a difference of over 1.5 m in accuracy between the two systems can be seen, which is a significant improvement.

C. RMS and MAPE Error Distribution

A simple visualization of the error distribution for the "whole trajectory," "all smooth curves," and "all sharp curves" prediction errors is shown in Fig. 6. The charts have the individual prediction errors categorized into groups, where group "0–1" in the x-axis contains all the prediction errors that fall between 0 and 1 m, and the y-axis shows how frequently the errors fall in each of the groups.

Looking at the histograms in Fig. 6, it can be observed how the DRWDE system tends to be more often in the first groups, which represent less prediction error. The taller the bars on a given group means that more often the error falls in that error group; therefore, the taller the blue bars on the smaller groups, the more accurate the system is.

In Fig. 6(a), it can only be seen that the DRWDE outperform the IMM 1 Hz by a small amount when looking at the overall trajectory, and a larger difference when looking at the results for all smooth curves in Fig. 6(b). However, when all sharp curves in Fig. 6(c) is observed, a more distinct difference in the prediction accuracy between the DRWDE and the IMM 1 Hz can be seen. To analyze the results for selected smooth and sharp curves specifically, as shown in Fig. 5, Fig. 7 was created.

Fig. 7(a) represents the error between the estimated future distance the vehicle will travel in the following 3 s and the actual distance traveled as recorded by the GPS sensor for the selected smooth and sharp curves. Time zero in the figure is set a few seconds before the vehicle enters the sharp curve shown in Fig. 4. Right at the beginning of the sharp curve, the error in the estimation is quite large for both systems, and that is due because the vehicle is moving in a somewhat straight path, so the estimated future position assumes that the vehicle will continue to move in the same direction. As soon as the vehicle enters the sharp curve, the first system to detect this change in direction is the DRWDE 10 Hz, as expected, as it can detect this

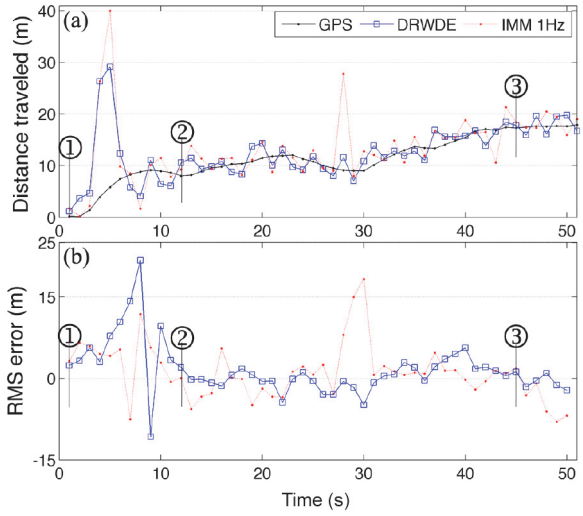

Fig. 7. Position error for a 3-s-ahead prediction during out selected curves as shown in Fig. 5. Top (a) displays actual versus predicted distance traveled per second and bottom (b) displays rms error in each prediction.

TABLE III
MAPE

	DRWDE	IMM 1Hz
Whole trajectory	0.0589	0.0610
All smooth curves	0.0424	0.0417
All sharp curves	0.0987	0.0642
Selected smooth curve	0.0320	0.0333
Selected sharp curve	0.0325	0.0641

Values represent 3-s-ahead prediction errors.

change using the accelerometer, while the GPS is still offline. Once the GPS sensor is back online, the 1 Hz system can also detect this change and can correct its prediction. The upward trend of the lines in Fig. 7 simply indicates that the vehicle is slowly increasing its velocity and is covering more distance in the same period of time (3 s). Only dots at each full second are shown to be able to compare between the two systems.

For a different view of the kind of errors the DRWDE 10 Hz system has, Fig. 7(b) was created, which shows the RMS error between the estimated future location (3 s later) and the actual location measured by the GPS. Table III shows the MAPE prediction accuracy of this system for the different segment types.

Looking at Fig. 6 and Table II, it can be concluded that the DRWDE setup really stands out when abrupt changes occur in the movement of the vehicle, and, only then, the fast reaction time shows substantial improvements in the prediction.

D. Computational Complexity

For completeness, it was also looked into how much of an extra load it is to run the DRWDE system with the dynamic noise matrices compared to the simpler approach of the 1 Hz IMM system. Because the dataset had already been recorded, only the processing time of the system itself was measured. If taking into account the processing time of the sensors, especially the accelerometer, the CPU times would be even larger.

TABLE IV
COMPUTATIONAL COMPLEXITY

	tic/toc (s)	cputime (s)	Data points	Avg. load
DRWDE	389.72	382.88	17,525	1.31
IMM 1Hz	48.53	47.29	2,187	1.30

Measurements taken on system running through the whole trajectory.

Table IV shows different MATLAB commands used to measure CPU times for each of the systems. All two commands (tic/toc and cputime) measure actual CPU time used by the MATLAB code, but this research is showing both to get a better idea on the accuracy of the measurements. The column tic/toc represents actual start/stop time of execution, while cputime displays the actual CPU time in that was spent executing the code. The system was run on a machine with a dual core 2.0 GHz CPU.

As expected, Table IV shows that the DRWDE 10 Hz system requires a lot more processing power than the simpler IMM 1 Hz system. This is as expected, since the DRWDE system has to handle close to 10 times more data points, and, therefore, yields much longer CPU times. On the same token, if looking at the last column, it can be observed that the average load times for every record processed is almost the same for both systems, which shows that the extra computational requirements of the DRWDE's dynamic error processing and measurement noise matrices are not significant at all.

VI. Conclusion

The key contribution of this research's DRWDE system is the introduction of dynamic noise covariance matrix merged together by an IMM. The longer a sensor remains offline, the less accurate the overall prediction is, so the dynamic Q presented in Section III-B tells the system how true is the value being used.

This DRWDE setup only had three sensors, of which only one of them was running at 10 Hz. The accelerometer is very sensitive to changes in the road, including road bumps; so, relying on this sensor to estimate the values of the other sensors when they were offline had its challenges. However, looking at Section V, it can be concluded that by properly handling the accumulating error for missing measurements, running the system at a higher frequency can yield better predictions, especially when abrupt changes occur. The key here was to be able to accurately account for the accumulating error when sensors go offline and remain offline for an unknown amount of time.

An improvement to this system could be to add more sensors running at high frequencies, for redundancy and to minimize the times sensors are offline. Moreover, this system could be combined with our previous researches [47], [48], where the predicted location is compared against geographical information system (GIS) to reduce false predictions.

APPENDIX

I. MATHEMATICAL LIMITATION FOR IMPROVED ESTIMATIONS

Because $x_{i+1} = [I + F_{k+1}(t_{i+1} - t_{k+1})] + w_{k+1}^{[0]}(t_{i+1} - t_{k+1}) = [I + F_{k+1}(t_{i+1} - t_{k+1})]x_{k+1} + w_{k+1}^{[0]}(t_{i+1} - t_{k+1}) - [I + F_{k+1}(t_{i+1} - t_{k+1})]u_{k+1}$, the corresponding process covariance matrix will be $m = w_{k+1}^{[0]}(t_{i+1} - t_{k+1}) - [I + F_{k+1}(t_{i+1} - t_{k+1})]\,u_{k+1}$. If φ_l is the vector formed by the elements of row l from F_{k+1}, $|\varphi_l \cdot u_{k+1}| \le |\varphi_l| \cdot |w_{k+1}|$, and if we operate at $E\left[m_{k+1}^t m_{k+1}\right] \le E\left[w_i^2\right](t_{is+1} - t_{k+1})^2 + E\left[w_{k+1}^2\right][1 + (\varphi_l w_{k+1}^{[0]})(t_{i+1} - t_{k+1})]^2$, we can define the trace of the covariance matrix of the process as $E\left[w_i w_i\right](t_{i+1} - t_{k+1})^2 = E\left[w_i^2\right](t_{i+1} - t_{k+1})^2$, which will show an improvement when

$$E\left[w_i^2\right](t_{i+1} - t_{k+1})^2 + E\left[w_{k+1}^2\right] \times \left[1 + (\varphi_l w_{k+1})(t_{i+1} - t_{k+1})\right]^2 < E\left[w_i^2\right](t_{i+1} - t_i)^2$$

which can be rewritten as

$$E\left[w_{k+1}^2\right] < \text{trace}(Q_i) \cdot \frac{1 - \left(\frac{t_{i+1} - t_{k+1}}{t_{i+1} - t_i}\right)^2}{[1 + (\varphi_l)(t_{i+1} - t_{k+1})]^2}.$$

II. TAYLOR POLYNOMIAL REPRESENTATION WITH ITS RESPECTIVE ERROR

$$x_p(t) = x_p(t_i) + \frac{x_{p+1}(t_i)}{1!}\Delta t_k + \cdots + \frac{x_n(t_i)}{(n-p)!}(\Delta t_k)^{n-p} + \int_{t_i} \frac{1}{(n-p)!}\dot{x}_n(t_i - y)^{n-p}dy.$$

The measurements of the variables will have an error. Given $\bar{x}_l$ the obtained measurement of x_l, the corresponding error $\varepsilon_l = \bar{x}_l - x_l$, in this first step, is due to $w(t_i)\,\Delta t_k$. Then, we can accordingly modify the Taylor polynomial as follows:

$$x_p(t_j) = x_p(t_i) + \frac{x_{p+1}(t_i)}{1!}\Delta t_k + \cdots + \frac{x_n(t_i)}{(n-p)!}(\Delta t_k)^{n-p} - \sum_{m=p} \frac{\varepsilon_m(t_i)}{(m-p)!} + \int_{t_i} \frac{1}{(n-p)!}\dot{x}_n(t_i - y)^{n-p}dy.$$

With this procedure, the measurement $\bar{x}_p(t_i)$ of $x_p(t_i)$ will have an error of

$$\varepsilon_j = \sum_{m=p} \frac{\varepsilon_m(t_i)}{(m-p)!} + \int_{t_i} \frac{1}{(n-p)!}\dot{x}_n(t_i - y)^{n-p}dy.$$

If the function of which we have known measurements in $t_j \in (t_i, t_{i+1})$ is x_0, then

$$x_{p+1}(c) = \dot{x}_p(c) = \frac{\bar{x}_p(t_k) - \bar{x}_p(t_i)}{\Delta t_k} = \bar{x}_{p+1}(t_i) + \frac{\bar{x}_{p+2}(t_i)}{2!}(\Delta t_k)^1 + \ldots + \frac{\bar{x}_n(t_i)}{(n-p)!}(\Delta t_k)^{n-p-1} - \frac{1}{\Delta t_k}\left[\varepsilon_p(t_k) - \sum_{m=p} \frac{\varepsilon_m(t_i)}{(m-p)!} + \int_{t_i} \frac{1}{(n-p)!}\dot{x}_n(t_i - y)^{n-p}dy\right].$$

For a given $c \in (t_i, t_k)$, we can approximate $x_{p+1}(t_k)$ as

$$x_{p+1}(t_j) = \bar{x}_{p+1}(t_i) + \frac{\bar{x}_{p+2}(t_i)}{2!}(\Delta t_k)^1 + \cdots + \frac{\bar{x}_n(t_i)}{(n-p)!}(\Delta t_k)^{n-p-1}$$

with an error of

$$\varepsilon_{p+1}(t_j) = x_{p+1}(c) - x_{p+1}(t_k) + \frac{1}{\Delta t_k}\left[\varepsilon_p(t_k) - \sum_{m=p} \frac{\varepsilon_m(t_i)}{(m-p)!} + \int_{t_i} \frac{1}{(n-p)!}\dot{x}_n(t_i - y)^{n-p}dy\right]$$

where the difference $x_{p+1}(c) - x_{p+1}(t_j)$, which depends on the stability of $x_{p+1}(t)$, is expected to be lower as Δt_k is small.

III. PROOF OF THE EXPECTED VALUE CALCULATIONS FOR EACH PREDICTION NOISE (σ_p) ELEMENT IN THE PROCESS NOISE COVARIANCE (Q) MATRIX TO SHOW HOW TO ARRIVE AT (6) STARTING FROM (5)

Derivation for $E\left[M^2(s)\right]$

$$E\left[M_k^2(s)\right] = E\left[M_{k-1}(s) + \frac{1}{6}j_k(\Delta k)^3\right]^2 = E\left[M_{k-1}^2(s) + 2M_{k-1}(s)\frac{1}{6}j_k(\Delta k)^3 + \frac{1}{6}j_k(\Delta k)^3\right]^2 = E\left[M_{k-1}^2(s)\right] + 2 \cdot E\left[M_{k-1}(s)\right] \cdot E\left[\frac{1}{6}j_k(\Delta k)^3\right] + E\left[\frac{1}{36}j_k^2(\Delta k)^6\right] = E\left[M_{k-1}^2(s)\right] + E\left[\frac{1}{36}j_k^2(\Delta k)^6\right]$$

and

$$E\left[\sigma_p \sigma_{p_1}\right] = E\left[\sigma_{p_1}^2\right] = E\left[\frac{1}{6}(j_t \sin\beta + j_n \cos\beta)(\Delta k)^3\right]^2 = \frac{(\Delta k)^6}{36} \cdot E\left[(j_t^2 \sin^2\beta + j_n^2 \cos^2\beta) + 2E\left[j_t\right]E\left[j_n\right]E\left[\sin\beta + \cos\beta\right]\right] \le \frac{(\Delta k)^6}{36} \cdot E\left[(j^2 \sin^2\beta + j^2 \cos^2\beta) + 2 \cdot 0 \cdot 0 \cdot E\left[\sin\beta + \cos\beta\right]\right] = \frac{j^2(\Delta k)^6}{36} \cdot E\left[(j_t^2 \sin^2\beta + j_n^2 \cos^2\beta)\right] = \frac{j^2(\Delta k)^6}{36}.$$

Therefore,

$$E\left[M_k^2(s)\right] \le \frac{1}{36}j_k(\Delta k)^2 + \frac{1}{36}j_{k+1}(\Delta k)^2 + E\left[M_k^2(s)\right]$$
$$= \frac{1}{36}j_k^2(\Delta k)^6 + \cdots + \frac{1}{36}j_{k+1}^2(\Delta k)^6 + E[\sigma_{p_1}^2]$$
$$\le \frac{(\Delta k)^6}{36}\sum_{i=0}^{k+1} j_{k-i}^2.$$

REFERENCES

[1] J. C. Miles and A. J. Walker, "The potential application of artificial intelligence in transport," *IEEE Proc. Trans. Intell. Transp. Syst.*, vol. 153, no. 3, pp. 183–198, Sep. 2006.

[2] F. Wang, A. Broggi, and C. C. White, "The road to transactions on intelligent transportation systems: A decade's success [transactions on ITS]," *IEEE Intell. Transp. Syst. Mag.*, vol. 1, no. 4, pp. 29–32, Winter 2009.

[3] V. Di Lecce and A. Amato, "Route planning and user interface for an advanced intelligent transport system," *IET Intell. Transp. Syst.*, vol. 5, no. 3, pp. 149–158, Sep. 2011.

[4] X. Xu, T. Xia, A. Venkatachalam, and D. Huston "The development of a high speed ultrawideband ground penetrating radar for rebar detection," *J. Eng. Mech.*, vol. 139, no. 3, pp. 272–285, Mar. 2013.

[5] T. Taleb, A. Benslimane, and K. Ben Letaief, "Toward an effective risk-conscious and collaborative vehicular collision avoidance system," *IEEE Trans. Veh. Technol.*, vol. 59, no. 3, pp. 1474–1486, Mar. 2010.

[6] F. Jimenez and J. Eugenio Naranjo, "Improving the obstacle detection and identification algorithms of a laserscanner-based collision avoidance system," *Transp. Res. C. Emerg. Technol.*, vol. 19, no. 4, pp. 658–672, Aug. 2011.

[7] R. Toledo-Moreo and M. A. Zamora-Izquierdo, "Collision avoidance support in roads with lateral and longitudinal maneuver prediction by fusing GPS/IMU and digital maps," *Transp. Res. C. Emerg. Technol.*, vol. 18, no. 4, pp. 611–625, Aug. 2010.

[8] L. Jiun-Ren, T. Talty, and O. K. Tonguz, "A blind zone alert system based on intra-vehicular wireless sensor networks," *IEEE Trans. Ind. Informat.*, vol. 11, no. 2, pp. 476–484, Apr. 2015.

[9] R. Toledo-Moreo, M. Pinzolas-Prado, and J. Manuel Cano-Izquierdo, "Maneuver prediction for road vehicles based on a neuro-fuzzy architecture with a low-cost navigation unit," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 2, pp. 498–504, Jun. 2010.

[10] J. Ueki, J. Mori, Y. Nakamura, Y. Horii, and H. Okada, "Development of vehicular-collision avoidance support system by inter-vehicle communications," in *Proc. 59th Veh. Technol. Conf. (VTC'04)*, May 2004, vol. 5, pp. 2940–2945.

[11] H. D. Weerasinghe, R. Tackett, and H. Fu, "Verifying position and velocity for vehicular ad-hoc networks," *Sec. Commun. Netw.*, vol. 4, no. 7, pp. 785–791, Jul. 2011.

[12] R. C. Luo and C. C. Chang, "Multisensor fusion and integration: A review on aproaches and its applications in mechatronics," *IEEE Trans. Ind. Informat.*, vol. 8, no. 1, pp. 49–60, Feb. 2012.

[13] D. Bruckner, H. Zeilinger, and D. Dietrich, "Cognitive automation survey of novel artificial general intelligence methods for the automation of human technical environments," *IEEE Trans. Ind. Informat.*, vol. 8, no. 2, pp. 206–215, May 2012.

[14] J. B. Gao and C. J. Harris, "Some remarks on Kalman filters for the multisensor fusion," *Elsevier Inf. Fusion*, vol. 3, pp. 191–201, 2002.

[15] M. Kafai and B. Bhanu, "Dynamic bayesian networks for vehicle classification in video," *IEEE Trans. Ind. Informat.*, vol. 8, no. 1, pp. 100–109, Feb. 2012.

[16] R. R. Murphy, "Sensor Fusion," in *Handbook of Brain Theory and Neural Networks*. Cambridge, MA, USA: A Brandford Book, MIT Press, 2003.

[17] K. Y. Chan, S. Khadem, T. S. Dillon, V. Palade, J. Singh, and E. Chang, "Selection of significant on-road sensor data for short-term traffic flow forecasting using the Taguchi method," *IEEE Trans. Ind. Informat.*, vol. 8, no. 2, pp. 255–266, May 2012.

[18] S. C. Felter, "An overview of decentralized Kalman filter techniques," in *Proc. IEEE South. Tier Tech. Conf.*, Apr. 1990, pp. 79–87.

[19] M. Hua *et al.*, "Decentralised solutions to the cooperative multi-platform navigation problem," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 47, no. 2, pp. 1433–1449, Apr. 2011.

[20] D. Herrero-Perez and H. Martinez-Barbera, "Modeling distributed transportation systems composed of flexible automated guided vehicles in flexible manufacturing systems," *IEEE Trans. Ind. Informat.*, vol. 6, no. 2, pp. 166–180, May 2010.

[21] M. Vallee, M. Merdan, W. Lepuschitz, and G. Koppensteiner, "Decentralized reconfiguration of a flexible transportation system," *IEEE Trans. Ind. Informat.*, vol. 7, no. 3, pp. 505–516, Aug. 2011.

[22] H. M. Wang, Q. Yin, and X. Xia, "Fast Kalman equalization for time-frequency asynchronous cooperative relay networks with distributed space-time codes," *IEEE Trans. Veh. Technol.*, vol. 59, no. 9, pp. 4651–4658, Nov. 2010.

[23] G. A. Watson, T. R. Rice, and A. T. Alouani, "An IMM architecture for track fusion," in *Proc. SPIE Signal Process. Sens. Fusion Target Recognit.*, 2000, vol. IX, pp. 2–13.

[24] A. T. Alouani and T. R. R. Rice, "On optimal asynchronous track fusion," in *Proc. IEEE 1st Aust. Symp. Data Fusion*, Nov. 1996, pp. 147–152.

[25] G. A. Watson, T. R. Rice, and A. T. Alouani, "Optimal track fusion with feedback for multiple asynchronous measurements," in *Proc. SPIE Acquisit. Track. Point. XIV*, vol. 4025, Apr. 2000, pp. 20–33.

[26] L. Armesto *et al.*, "Probabilistic self-localization and mapping–An asynchronous multirate approach," *IEEE Rob. Autom. Mag.*, vol. 15, no. 2, pp. 77–88, Jun. 2008.

[27] Y. Bar-Shalom and H. M. Chen, "IMM estimator with out-of-sequence measurements,"*IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 1, pp. 90–98, Jan. 2005.

[28] X. Shen *et al.*, "Optimal centralized update with multiple local out-of-sequence measurements," *IEEE Trans. Signal. Process.*, vol. 57, no. 4, pp. 1551–1562, Apr. 2009.

[29] X. Shen *et al.*, "Globally optimal distributed kalman fusion with local out-of-sequence-measurement updates," *IEEE Trans. Autom. Control*, vol. 54, no. 8, pp. 1928–1934, Aug. 2009.

[30] W. Jiangxin, S. Y. Chao, and A. M. Agogiono, "Validation and fusion of longitudinal positioning sensors in AVCS," in *Proc. Amer. Control Conf.*, 1999, vol. 3, pp. 2178–2182.

[31] R. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME*, vol. 82, pp. 34–45, 1960.

[32] Y. Ho and R. Lee, "A Bayesian approach to problems in stochastic estimation and control," *IEEE Trans. Autom. Control*, vol. 9, no. 4, pp. 333–339, Oct. 1964.

[33] G. Welch and G. Bishop, "An introduction to the Kalman filter," Dept. Comput. Sci., Chapel Hill, NC, USA, SIGGRAPH Course Notes, 2001.

[34] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation With Applications to Tracking and Navigation*. Hoboken, NJ, USA: Wiley, 2001.

[35] Y. Bar-Shalom and X. R. Li, *Estimation and Tracking: Principles, Techniques and Software*. Norwood, MA, USA: Artech House, 1993.

[36] B. Mokaberi and A. A. G. Requicha, "Drift compensation for automatic nanomanipulation with scanning probe microscopes," *IEEE Trans. Autom. Sci. Eng.*, vol. 3, no. 3, pp. 199–207, Jul. 2006.

[37] P. N. Pathirana, A. V. Savkin, and S. Jha, "Location estimation and trajectory prediction for cellular networks with mobile base stations," *IEEE Trans. Veh. Technol.*, vol. 53, no. 6, pp. 1903–1913, Nov. 2004.

[38] X. Xu, Z. Xiong, X. Sheng, J. Wu, and X. Zhu, "A new time synchronization method for reducing quantization error accumulation over real-time networks: theory and experiments," *IEEE Trans. Ind. Informat.*, vol. 9, no. 3, pp. 1659–1669, Aug. 2013.

[39] M. H. Kim, S. Lee, and K. C. Lee, "Kalman predictive redundancy system for fault tolerance of safety-critical systems," *IEEE Trans. Ind. Informat.*, vol. 6, no. 1, pp. 46–53, Feb. 2010.

[40] L. Hong, "Multirate interacting multiple model filtering for target tracking using multirate models," *IEEE Trans. Autom. Control*, vol. 44, no. 7, pp. 1326–1340, Jul. 1999.

[41] E. Mazor, "Interacting multiple model methods in target tracking: A survey," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 34, no. 1, pp. 103–123, Jan. 1998.

[42] S. J. Lee, Y. Motai, and H. Choi, "Tracking human motion with multichannel interacting multiple model," *IEEE Trans. Ind. Informat.*, vol. 9, no. 3, pp. 1751–1763, Aug. 2013.

[43] H. A. P. Blom and Y. Shalom, "The interacting multiple model algorithm for systems with markovian switching coefficients," *IEEE Trans. Autom. Control*, vol. 33, no. 8, pp. 780–783, Aug. 1988.

[44] L. A. Johnson and V. Krishnamurthy, "An improvement to the interactive multiple model (IMM) algorithm," *IEEE Trans. Signal Process.*, vol. 49, no. 12, pp. 2909–2923, Dec. 2001.

[45] M. S. Grewal and A. P. Andrews, *Kalman Filtering Theory and Practice Using MATLAB*, 2nd ed. Hoboken, NJ, USA: Wiley, 2001, p. 132.

[46] M. Green, "How long does it take to stop? Methodological analysis of driver perception-brake times," *Transp. Hum. Factors*, vol. 2, no. 3, pp. 195–216, Sep. 2000.
[47] C. Barrios and Y. Motai, "Improving estimation of vehicle's trajectory using the latest global positioning system with Kalman filtering," *IEEE Trans. Instrum. Meas.*, vol. 60, no. 12, pp. 3747–3755, Dec. 2011.
[48] C. Barrios, Y. Motai, and D. Huston, "Trajectory estimations using smartphones," *IEEE Trans. Ind. Electron.*, vol. 62, no. 12, pp. 7901–7910, Dec. 2015.

Cesar Barrios received the B.S. and M.S. degrees from the New Jersey Institute of Technology, Newark, NJ, USA, and the Ph.D. degree in 1999, 2001, and 2014, respectively, all in electric engineering.

He has worked with IBM, Essex Junction, VT, USA, since his graduation in 1999, and with GLOBALFOUNDRIES, Essex Junction, since 2015. His research interests include information technology and semiconductor research and development center.

Yuichi Motai (M'01–SM'13) received the B.Eng. degree in instrumentation engineering from Keio University, Tokyo, Japan, in 1991; the M.Eng. degree in applied systems science from Kyoto University, Kyoto, Japan, in 1993; and the Ph.D. degree in electrical and computer engineering from Purdue University, West Lafayette, IN, USA, in 2002.

He is currently an Associate Professor of Electrical and Computer Engineering with Virginia Commonwealth University, Richmond, VA, USA. His research interests include the broad area of sensory intelligence, particularly in medical imaging, pattern recognition, computer vision, and sensory-based robotics.

Dryver Huston received the B.S. degree in mechanical engineering from the University of Pennsylvania, Philadelphia, PA, USA, in 1980, and the M.A. and Ph.D. degrees in civil engineering from Princeton University, Princeton, NJ, USA, in 1982 and 1986, respectively.

He is a Professor with the School of Engineering, University of Vermont, Burlington, VT, USA. His research interests include sensors, structural health monitoring, and self-healing systems.