Trajectory Estimations Using Smartphones

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

***Abstract*—This paper investigates whether the smartphones' built-in sensors can accurately predict future trajectories for a possible implementation in a vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) system. If smartphones could be used, vehicles without the V2V/V2I technology could use them to tap into the V2V/V2I infrastructure and help to populate the gap of vehicles off the V2V/V2I grid. To evaluate this, we set up a dead-reckoning system that uses Kalman filters to predict the future trajectory of a vehicle, information that could be used in a V2V/V2I system to warn drivers if the trajectories of vehicles will intersect at the same time. Then, we use a vehicle with accelerometer, GPS, and speedometer sensors mounted on it and evaluate its accuracy in predicting the future trajectory. Afterward, we place a smartphone securely on the vehicle's dashboard, and we use its internal accelerometer and GPS to feed the same dead reckoning and Kalman filter setup to predict the future trajectory of the vehicle. We end by comparing both results and evaluating if a smartphone can achieve similar accuracy in predicting the future trajectory of a vehicle. Our results show that some smartphones could be used to predict a future position, but the use of their accelerometer sensors introduces some measurements that can be incorrectly interpreted as spatial changes.**

***Index Terms*—Dead reckoning, Kalman filter, position estimation, smartphones, trajectory paths, vehicle-to-infrastructure (V2I), vehicle-to-vehicle (V2V).**

I. Introduction

THE overall function of intelligent transport systems is to improve decision making, often in real time, improving the operation of the entire transport system. This can go from systems with intelligent route planning implemented to avoid some specific type of traffic in certain areas [1], to keeping track of the position of the vehicle for infrastructure assessment [2], to systems designed to aid with the prevention of collisions between the vehicles [3], [4]. For this study, the research focuses on evaluating the use of smartphones as an intermediate step to accelerate the implementation of vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I), which could be used to prevent collisions.

Fig. 1. Illustration for V2V and V2I from [17].

There are two main types of collision avoidance systems: self-sufficient and interactive systems. Self-sufficient systems are those that can obtain enough information from their own sensors, such as those in [5]–[7], where they placed sensors around the vehicle to maintain a safe following distance or to detect vehicles in the surroundings. Interactive systems are those that, as the name implies, interact with the infrastructure and/or other vehicles, such as researched in [8]–[10], where their systems send spatial information to nearby vehicles to estimate the probability of a future collision. While self-sufficient systems are limited to line-of-sight detection, the interactive systems account for scenarios farther ahead or even around corners or intersections by predicting and communicating the future estimated trajectories.

The V2V and V2I areas are being well researched these days [11]–[17], as government is carefully evaluating the implementation of new technologies to make our roads safer (Fig. 1). In an article published on February 3, 2014, by the United States Department of Transportation [18], the National Highway Safety Administration announced its decision to begin taking the next steps toward implementing V2V technology in all new cars and trucks, after years of research and unprecedented coordination between industry and across government.

When the steps toward implementing V2V technology are defined, car and truck manufacturers will be mandated to enable this in their new vehicles. The challenge faced is that, because V2V relies on other vehicles nearby also supporting V2V technology, there will be a gap of many years when the V2V/V2I will not be able to show its true potential in improving road safety. In an article published by Forbes on March 14, 2013 [19], they calculated that the age of the average vehicle on the road is at a record high of 10.8 years, which means that there are vehicles on the roads that are 20 years old. Keeping this in mind, it is a long time to wait to ensure full V2V/V2I reliability.

Manuscript received December 2, 2014; revised June 9, 2015 and August 10, 2015; accepted August 19, 2015. Date of publication September 14, 2015; date of current version November 6, 2015. This work was supported in part by the United States Department of Transportation through the University of Vermont Transportation Research Center and in part by the National Science Foundation through the Faculty Early Career Development Program under NSF ECCS Grant 1054333.

C. Barrios was with the Department of Electrical Engineering, University of Vermont, Burlington, VT 05405 USA. He is now with GlobalFoundries, Burlington, VT 05406 USA (e-mail: cbarrios@uvm.edu).

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

D. Huston is with the Department of Mechanical Engineering, 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/TIE.2015.2478415

The scientific contribution of this research includes the evaluation of using the smartphone to predict future trajectories for a possible implementation as a temporary hook into the V2V/V2I infrastructure in older vehicles. Allowing drivers of older vehicles the possibility of taking advantage of this new technology would not only benefit them, but it would also benefit the rest of the V2V/V2I-enabled vehicles, as the number of vehicles participating in the system would be much greater. Smartphones are already being used in the transportation field, and one example is the mobile application DriveWell, created by Cambridge Mobile Telematics [20], where the smartphone's built-in sensors are used to provide a driver safety scoring and tips on how to improve it.

Since this research wants to evaluate the use of a smartphone's built-in sensors for a setup that could be used in a V2V/V2I system, it will focus on the prediction of a vehicle's future trajectory and compare the results with the use of more robust sensors mounted on a vehicle to predict the same future trajectory. Given that multiple sensors will be used, some type of sensor fusion will be needed to use the different measurements in the prediction.

II. Sensor Fusion Techniques

Multisensor data fusion (MSDF) techniques are used in many diverse fields, although most of the literature addresses the fields of military target tracking or autonomous robotics [21]. MSDF is required to combine and process data, which has been traditionally performed by some form of Kalman [22] or Bayesian filters. Furthermore, there can be two ways of setting up an MSDF system: centralized or decentralized. While a centralized approach suffices for most common scenarios where the sensors are synchronous, a decentralized approach is more convenient when the sensors should be treated independently [23]–[28], as with asynchronous sensors.

In [29], the authors discuss one solution that they have developed: the optimal asynchronous track fusion algorithm (OATFA), which evolved from their earlier research on an asynchronous/synchronous track fusion [30]. They base their technique in the interacting multiple model (IMM) algorithm, but they replaced the conventional Kalman filters with their OATFA (which contains several Kalman filters of its own). The OATFA treats each sensor separately, passing the output from each to a dedicated Kalman filter, departing from the idea that the best way to fuse data is to deliver them all to a central fusion engine. The results from the IMM-OATFA show position estimation errors half of those of what the conventional IMM produces. However, as pointed out by the same authors in [31], all measurement data must be processed before the fusion algorithm is executed. With a similar approach as the technique described previously, the authors of [32] 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 arise where, for some reasons, a sensor is delayed in providing its data or is offline for a few cycles. The whole system needs to wait, as it is designed to work with certain data at specific rates.

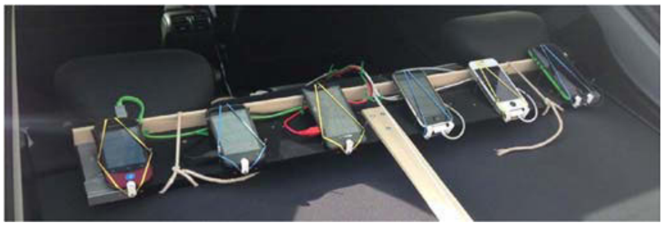

Fig. 2. Smartphones securely mounted in the trunk of a hatchback vehicle.

To evaluate whether smartphones can properly estimate future trajectories to be considered as an option to fill in the V2V/V2I implementation gap, a 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 [33]. Looking at 3 s ahead of time was chosen as a reference point that is double the reaction time of an average human being. In reality, this number will probably vary in relation to the speed and the type of the vehicle since a faster or heavier vehicle will need more time to slow down, but it is taken as a reference point (Fig. 2).

III. Position Estimation With Kalman Filters

For this research, the core method chosen to estimate a future position of a vehicle is the use of KF. The KF [34] was first proposed in the 1960s, and it has been the most commonly used technique in target tracking and robot navigation since. The basic KF has been presented as a form of Bayesian filter [35], which is an optimal estimator for linear Gaussian systems. From a series of noisy measurements, the KF is capable of estimating the state of the system in a two-step process: correct and then predict [36]–[38].

The elements of this state vector (x) are the following: position, velocity, and acceleration of the vehicle. The position (x_v) and velocity (v_v) components of the state estimate have x and y components to them (east-west and north-south directions), and the acceleration (a_v) has n and t components to it (normal and tangential acceleration). Therefore, the full state vector matrix will be $X = (x_x, x_y, v_x, v_y, a_n, a_t)$.

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 data set that specifies the estimated accuracy in the observation errors between all possible pairs of vertical levels.

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

The KF has a long history of accurately predicting the future states of a moving target and has been applied to many different fields [39]–[46], including transportation, which is why it was selected for this research. Because one KF estimates the future

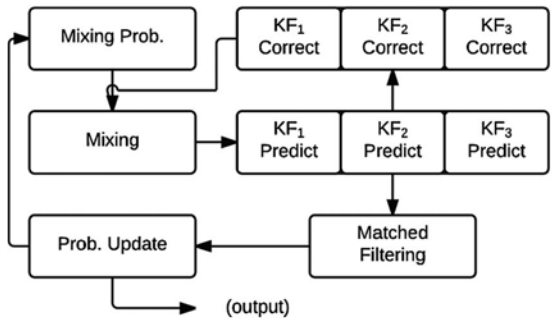

Fig. 3. Flowchart for three KFs in an IMM framework.

position of a vehicle using one spatial movement model, there is a need to set up several KFs to account for the different spatial states in which the vehicle can be found. Having multiple KFs running in parallel at the same time requires a framework that can obtain one weighted answer.

This research opted for the use of IMM, which can calculate the probability of success of each KF model at every filter execution, providing a combined solution for the vehicle behavior [47]–[50]. These probabilities are calculated according to a Markov model for the transition between maneuver states, as detailed in [51]. To implement the Markov model, it is assumed that, at each execution time, there is a probability p^{ij} that the vehicle will make the transition from model state i to state j.

In Johnson and Krishnamurthy's paper [52], they describe the IMM as a recursive suboptimal algorithm that consists of four core steps, interacting with the KF steps as illustrated in Fig. 3.

The four-step IMM process starts with the calculation of the mixing probabilities, which uses the transition matrix and the previous iteration mode probabilities $\mu_{k-1}(i)$ to compute the normalized mixing probabilities $\mu_k(i|j)$. The mixing probabilities are recomputed each time the filter iterates before the mixing step.

The second step uses the mixing probabilities, which are used to compute new initial conditions for each of the n filters. The initial state vectors are formed as the weighted average of all of the filter state vectors from the previous iteration x_{k-1}^{oj}. The error covariance corresponding to each of the new state vectors is computed as the weighted average of the previous iteration error covariance's conditioned with the spread of the means (P_{k-1}^{oj}).

The third step calculates mode matched filtering, using the $\hat{x}_{k-1}^{0j}$ and P_{k-1}^{0j}, and the bank of n Kalman filters produces outputs $\hat{x}_k^j$, the covariance matrix P_k^j, and the probability density function $f_n(z_k)$ for each filter (n).

The fourth set of calculations begins once the new initial conditions are computed; the filtering step generates a new state vector, an error covariance, and a likelihood function for each of the filter models. The probability update step then computes the individual filter probability $(\mu_k(j))$ as the normalized product of the likelihood function and the corresponding mixing probability normalization factor.

The estimate and covariance combination is used for output purposes only $\hat{x}_k = \sum_{j=1}^{n} \mu_k^j \cdot \hat{x}_k^j$; it is not part of the algorithm recursions.

IV. Position Estimation Framework Using GPS and Accelerometer Sensors

For the evaluation of the use of smartphones to predict a future position of a vehicle, only those sensors common across all devices were selected, which, in this case, are the GPS and accelerometer sensors. With the measurements obtained from these sensors, the future position estimation is obtained.

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 accelerometer provides normal acceleration (a_n) and tangential acceleration (a_t). The different models to be used in this setup are defined as follows, which represents the different spatial states the vehicle can be found in:

$$\begin{aligned}
&\text{Constant Location Model (CL)}\\
&\quad s(k) = s(k-1) + \sigma_{p_s}\\
&\quad v(k) = 0\\
&\quad a(k) = 0\\
&\text{Constant Velocity Model (CV)}\\
&\quad s(k) = s(k-1) + v(k-1)\cdot\Delta k + \sigma_{p_s}\\
&\quad v(k) = v(k-1) + \sigma_{p_v}\\
&\quad a(k) = 0\\
&\text{Constant Acceleration Model (CA)}\\
&\quad s(k) = s(k-1) + v(k-1)\cdot\Delta k\\
&\qquad\qquad + \frac{1}{2}a(k-1)\cdot\Delta k^2 + \sigma_{p_s}\\
&\quad v(k) = v(k-1) + a(k-1)\cdot\Delta k + \sigma_{p_v}\\
&\quad a(k) = a(k-1) + \sigma_{p_a}.
\end{aligned}$$

The aforementioned three KF models are used as part of an IMM setup to merge each of the KF predictions and obtain one single predicted position, as described in Section IV. When not all sensor measurements are available to properly populate the three KF models, the DR approach described in Section VI is used to handle the asynchronous data.

V. MSDF Setup

This research initially looks at predicting future positions by running the IMM system at 1 Hz with all vehicle-mounted (VM) sensors online in every iteration and then running it at 10 Hz, where some sensors are offline and measurements are missing in many of the iterations. A measurement could be missing, either due to the sensor not being able to take the measurement (system running at a faster frequency than the sensor, malfunction, or no satellites in view for the GPS) or due to the processing CPU not being able to read/write fast enough. When a measurement is absent and the value is needed for the models, the missing values are calculated from the measurements obtained by the remaining sensors based on previous real measurements, not estimations, when available.

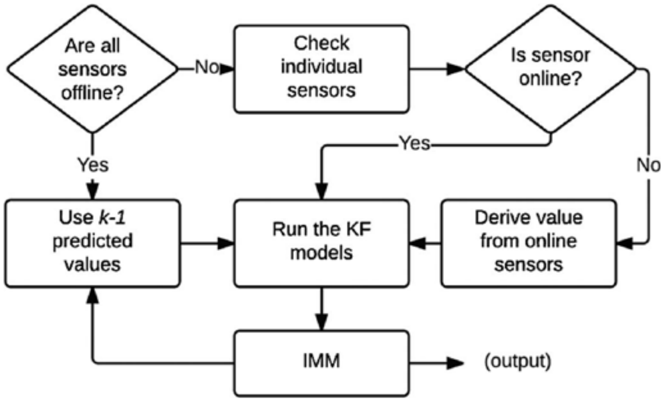

Fig. 4. Flowchart of the position estimation framework used by the VM and smartphone sensors.

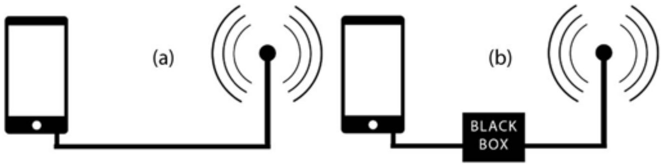

Fig. 5. (a) SP supporting V2V/V2I communication technology. (b) SP needing an external device (black box) for communication.

In this research, because the system is running at the rate of the accelerometer (10 Hz), the missing measurements come from the GPS and ScanTool VM sensors which run at only 1 Hz. Instead of using previously estimated values for the missing measurements, this setup uses the real measurements obtained from the accelerometer and uses the following equations to derive the missing measurements:

$$v(k) = v(k-1) + a(k) \cdot \Delta k$$

$$s(k) = s(k-1) + v(k-1) \cdot \Delta k + \frac{1}{2} a(k) \cdot \Delta k^2.$$

Using this DR approach, as outlined in Fig. 4, predictions are more accurate than waiting until all measurements are available again or predicting these measurements the second time, using only previously estimated values. Only when all measurements are missing, which will be very unlikely, the system will use all of the previously estimated values to feed the KF models and obtain the new position estimation.

VI. Car and Smartphone Sensor Setup for a V2V/V2I System

As described toward the end of Section II, this research will estimate the future position of the vehicle 3 s away using the framework described in Section IV, which is something that could be shared with other vehicles (V2V) or with the infrastructure (V2I) as part of some collision avoidance systems. As the communication standards are defined for the V2V and V2I systems, smartphones will fall into one of the two different categories as briefly illustrated in Fig. 5: 1) already supporting the technology required for the communication or 2) needing an external device, illustrated as a black box and connected via USB/Bluetooth/Wi-Fi to manage the communication aspect. This is briefly illustrated in Fig. 5. Since smartphones and V2V/V2I are designed independently of each other, the use of an external device (black box) is more likely to be the case to enable smartphones to participate in a V2V/V2I system.

The VM sensors specifically set up for this specific task will be used, like manufacturers will implement in their vehicles. Smartphone (SP) sensors will also need to be evaluated when used for position estimation to determine if they yield similar results.

To properly evaluate if smartphones can be used in a V2V/V2I system, this research plans to set a baseline by running the VM measurements through the position estimation framework defined in Section IV and to calculate the position errors in the estimations by comparing them to the actual GPS data. Once the baseline is established and a determination of what are the amounts of prediction errors obtained, the individual SP measurements will be fed into the same position estimation framework, and the error will be calculated in the position estimations. This research can then proceed to compare the results between the different sensors used and evaluate whether the smartphones' built-in sensors yield similar prediction errors or not.

The setup on the VM sensors consists of a Garmin 16HVS GPS receiver running at 1 Hz and a Crossbow three-axis accelerometer running at 10 Hz. An AutoEnginuity ODBII ScanTool (which obtains the velocity from the vehicle's internal system at 1 Hz) is also available, but it will not be used in this evaluation because the smartphones that this research is using do not have a way of connecting into the ODBII system. The data from the sensors used are postprocessed from the different log files recorded on an earlier date and matched based on time stamps. Since these VM sensors were mounted on a van from the University of Connecticut, they will be labeled as UConn data throughout this research.

For the smartphones used in this evaluation, some were selected from different manufacturers and at different price ranges to identify if there is some limitation on which ones can accurately predict the future trajectory of a vehicle. Also, smartphones are used with different operating systems as well to improve the evaluation experiment and take that into account as well. They were mounted securely on the vehicle to ensure that the accelerometer readings truly reflect the dynamics of the vehicle. Because several smartphones were running at the same time, they were mounted in the trunk, where they would still have a clear view of the sky, as shown in Fig. 2, but a more common implementation would be to mount only one of them on the dashboard. Fig. 2 shows six smartphones, but one of them did not record any GPS data so it had to be removed from this experiment. The smartphones used in the evaluation of accurately predicting future trajectories are listed in Table I.

All smartphones listed previously have a built-in accelerometer sensor that can take measurements at 10 Hz, but no details were found on their model or sensitivity. These smartphones also have a built-in GPS sensor, and only very basic information

TABLE I
SMARTPHONE SPECS

Manufacturer	*Model*	*OS*	*Rel. Date*	*Base Price*
Alcatel	OneTouch 908F	Android 2.2	6/2011	$130
HTC	Desire C	Android 4.0	6/2012	$150
LG	Lucid VS 840	Android 2.3	4/2012	$300
Apple	iPhone 3GS	iOS 5.1	6/2009	$199*
Apple	iPhone 5	iOS 7.01	9/2012	$650

Details about these smartphones obtained from gsmarena.com.
* Subsidized price; this model could not be purchased without a contract, so real price could be two or three times more.

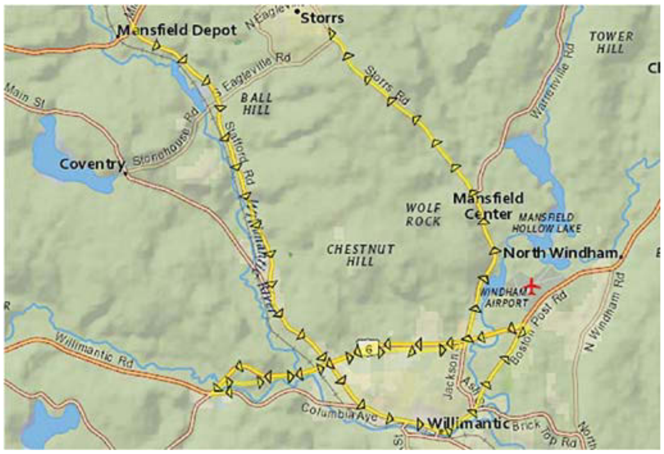

Fig. 6. Map of the recorded route near the University of Connecticut.

was found about them. iPhone 5 has an A-GPS/GLONASS sensor, while the other four smartphones do not have support for GLONASS (Global Navigation Satellite System by the Russians). Also, both Apple smartphones can take measurements from the GPS sensor at 10 Hz, while the other three smartphones can only take measurements at 1 Hz.

Some smartphones also have a three-axis gyro sensor and a compass, which could be used as well to better estimate a position of a vehicle, but to match more closely the sensors mounted on the vehicle and have a more equal comparison, they were not used in this experiment.

The measurements from the internal sensors of the iPhone smartphones are recorded by running the SensorLog v1.4 application written by Thomas. The sensors' measurements on the Android smartphones are recorded using the Data Recording v1.0.2.0 application written by Wolf. The data used are labeled by smartphone manufacturer, except where more than one device per manufacturer, in which case the data were labeled by model name.

To properly exercise the position estimation framework described in Section IV, the route shown in Fig. 6 for this evaluation was selected, which contains several curves (smooth and sharp) and straight paths, driven at different speeds in the larger and smaller roads. There were also some traffic lights on the way and even a U-turn, providing also some stop and go scenarios. The route driven, shown in Fig. 6, is approximately 44 km long and takes about 45 min to drive all of it.

VII. Evaluation Criteria

To evaluate whether smartphones can properly predict a future trajectory and be considered as a possible solution to fill in the V2V/V2I implementation gap, the position estimation error between the VM sensors (UConn) and the SP sensors was selected using the same KF models and IMM framework. To start, the position estimation error between both setups will be evaluated for the whole trajectory recorded. Also, since position estimation errors tend to increase during nonstraight paths, this research will also divide the trajectory recorded into smooth and sharp curves. To determine whether a set of consecutive points in the trajectory is a curve or a straight line, the change in the heading (angle) between the current heading and the heading 2 s before was looked at; if more than 5°, then it was defined as a curve. Moreover, to determine if the curve is a sharp one, the change has to be greater than 20°; otherwise, it was defined as a smooth curve.

To calculate the position estimation error in each step, this research will compare the estimated position to the actual position measured by the GPS 3 s later. This will allow a data set of calculated errors to be built for the whole trajectory, and then, it can be divided into the route sections described in the previous paragraph. This research will look at average prediction errors and root-mean-square (rms) prediction errors to try to evaluate whether the sensors built into smartphones can properly fill in the V2V/V2I implementation gap

$$\mathrm{RMS} = \square\frac{\square_{k}^{k\square} (\mathrm{err}_k - \mathrm{err}_{k-1})^2}{(k^{\square} - k)}.$$

For this experiment, a tolerance of 10% from the position estimation errors obtained from the VM sensors will be used; therefore, if a smartphone yields more than 10% higher estimation error than the VM sensors, then it will be concluded that such a smartphone cannot properly predict a future trajectory and hence could not be considered as a possible temporary solution to fill in the V2V/V2I implementation gap.

VIII. Experimental Evaluation

A. Data Set Characteristics

The characteristics of the complete recorded data set are shown in Table II, where the mean and standard deviation for the position difference between each second, velocity, normal acceleration, and tangential acceleration are displayed.

Looking at Table II, it is quickly noticed that the values between the distance and velocity columns are very similar, as expected, because this research is measuring the position change every 1 s. Also, as mentioned in Section VI, the UConn data were obtained on an earlier date, so it can be seen that, overall, the University of Connecticut van was driven a little bit faster than the vehicle with the smartphones. Also, because all five smartphones were in the same vehicle, their sensor measurements should have been very similar, which is not the case in several places. For example, for sharp curves, the two iPhones seemed to be moving at a much faster speed than the other three devices, while during straight paths, they

TABLE II
REPRESENTATIVE DATA SETS

	Device	*Distance* (m)	*Velocity* (m/s)	*Acc.norm* (m/s^2)	*Acc.tang* (m/s^2)
Whole Trajectory	UConn	18.14±8.99	17.95±8.45	-0.17±0.58	0.61±0.60
	Alcatel	16.31±8.40	16.30±7.95	-0.12±0.73	0.66±0.95
	HTC	16.21±8.24	16.29±8.01	-0.09±0.56	0.64±0.68
	LG	16.79±9.13	16.45±7.83	-0.18±0.58	0.61±0.64
	iPhone3GS	16.59±9.37	16.21±7.96	-0.22±0.74	0.51±0.88
	iPhone5	16.25±9.09	16.24±8.18	-0.11±0.79	0.62±0.82
Straight Paths	UConn	19.43±8.29	19.02±8.14	-0.16±0.48	0.55±0.57
	Alcatel	18.76±6.92	18.72±6.70	-0.12±0.72	0.64±0.95
	HTC	18.40±6.56	18.38±6.51	-0.10±0.54	0.62±0.57
	LG	18.98±7.74	18.42±6.65	-0.18±0.56	0.59±0.50
	iPhone3GS	16.96±9.04	16.63±7.77	-0.20±0.74	0.50±0.88
	iPhone5	16.23±9.24	16.23±8.29	-0.10±0.78	0.60±0.73
Smooth Curves	UConn	16.19±9.26	15.81±8.03	-0.29±0.89	0.74±0.63
	Alcatel	14.66±7.90	14.58±7.36	-0.11±0.75	0.73±0.98
	HTC	12.76±8.32	12.92±7.99	-0.07±0.64	0.78±0.91
	LG	14.01±9.28	13.53±7.65	-0.21±0.64	0.68±0.82
	iPhone3GS	12.47±11.31	11.99±8.49	-0.34±0.70	0.58±0.87
	iPhone5	17.99±6.64	17.86±5.83	-0.16±0.89	0.82±1.29
Sharp Curves	UConn	8.52±9.17	9.91±7.65	-0.10±0.89	1.01±0.68
	Alcatel	7.13±9.60	7.49±8.35	-0.16±0.79	0.63±0.88
	HTC	5.29±9.05	5.97±8.68	-0.07±0.64	0.59±0.94
	LG	5.48±8.86	7.09±8.01	-0.16±0.66	0.59±1.03
	iPhone3GS	16.30±12.13	13.50±9.49	-0.15±1.05	0.70±0.89
	iPhone5	12.00±8.54	12.18±8.54	-0.50±1.04	0.62±1.58

Values represent median ± standard deviation of all sensor measurements collected by each device (~25000 data points).

seemed to be moving a little slower than the rest. The tangential acceleration for all devices seems to be fairly consistent across all devices, while the normal acceleration is not as consistent, especially when smooth and sharp curves were observed, which could imply that some sensors are more sensitive than others.

B. Position Estimation Setup

To set up the IMM, it is necessary to calculate the transition probability matrix, so the GPS position measurements for the whole trajectory shown in Fig. 6 were used. From this full GPS log that contained multiple scenarios, it was determined which transition occurs frame by frame by comparing the actual measurements from the GPS to the smoothed measurements. The smoothing of the data was done with a rolling window using a combination of median smoothing, split the sequence, and Hann's sequence, which removed any abrupt changes from the data. The type of spatial change, such as no change, a constant change, and so on, determines each transition. Similarly, by calculating the covariance of the differences in the measurements to each other, the measurement noise covariance matrix (R) was obtained. Finally, by calculating the covariance of the differences in the measurements compared to their respective x and y components, the process covariance noise (Q) for each KF was obtained. From this type of information, calculating the frequency the vehicle changes from one state to another, the transition probability matrix is derived.

Next, each of the devices was run through the same IMM system using the KF models described in Section IV, and for each new measurement obtained from any of the sensors, the system predicts the position of where the vehicle is going to be 3 s later in time.

TABLE III
RMS PREDICTION ERROR (3 s AHEAD)

	UConn at 1Hz	*UConn at 10Hz*
Whole Trajectory	1.68 ±3.21	1.29 ±2.64
Straight Paths	1.27 ±2.66	0.92 ±1.65
Smooth Curves	2.60 ±3.73	2.28 ±4.87
Sharp Curves	3.43 ±4.98	2.51 ±4.36

Contains values representing median prediction error in meters ± standard deviation.

TABLE IV
RMS PREDICTION ERROR (3 s AHEAD)

	UConn	*Alcatel*	*HTC*	*LG*	*iPhone3GS*	*iPhone5*
Whole	1.29	1.88	1.00	1.13	2.34	1.41
Trajectory	±2.64	±2.62	±1.03	±1.25	±3.00	±2.08
Straight	0.92	1.77	0.89	0.98	2.21	1.35
Paths	±1.65	±2.56	±1.10	±1.13	±2.81	±2.05
Smooth	2.28	1.89	0.97	1.07	3.49	1.86
Curves	±4.87	±2.14	±0.97	±1.23	±4.02	±2.40
Sharp	2.51	2.11	1.21	1.43	4.78	2.13
Curves	±4.36	±3.83	±1.23	±1.59	±4.70	±2.05

Values representing median prediction error in meters ± standard deviation by all devices for trajectory shown in Fig. 6 (~25000 data points).

C. Evaluation of Position Estimation Error by System Rate

This part of the experiment is to make sure that the system defined in Section V, which runs at the rate of its fastest sensor, yields better results than running the system at the rate of its slowest sensor. First, the VM sensor's data set (Uconn) is run through the IMM system at the rate of 1 Hz when all sensors are online at each iteration. Then, the same data set is run through the same IMM system but running at 10 Hz, so now missing values from offline sensors are calculated based on measurements from online sensors using a DR approach. For this part of the experiment, all of the VM sensors available were used, including the AutoEnginuity ODBII ScanTool.

Accuracy in predicting where the vehicle will be 3 s later in time is the factor to observe to be able to evaluate if running the IMM system at the rate of its fastest sensor yields smaller errors. Calculating the rms prediction error by comparing each predicted position with each actual GPS measurement 3 s later in time, Table III is compiled.

Taking a quick look at Table III, it can be observed that the rms prediction errors for the system running at 10 Hz are smaller than when the system is run at 1 Hz, especially during curves. Even during straight paths, some improvements can be observed because the system is detecting a change in the vehicle's speed faster and can accordingly recalculate its prediction.

D. Evaluation of Position Estimation Error by Device

Now that it is shown that running this research's system at the rate at the fastest sensor yields better predictions of future positions, all smartphones are also run through this setup, and their corresponding prediction errors are recorded in Table IV. This table displays the rms distance between the predicted and actual positions. This prediction error can only be calculated when the time stamps between the predicted position and GPS reading match. It is assumed that the GPS reading is correct,

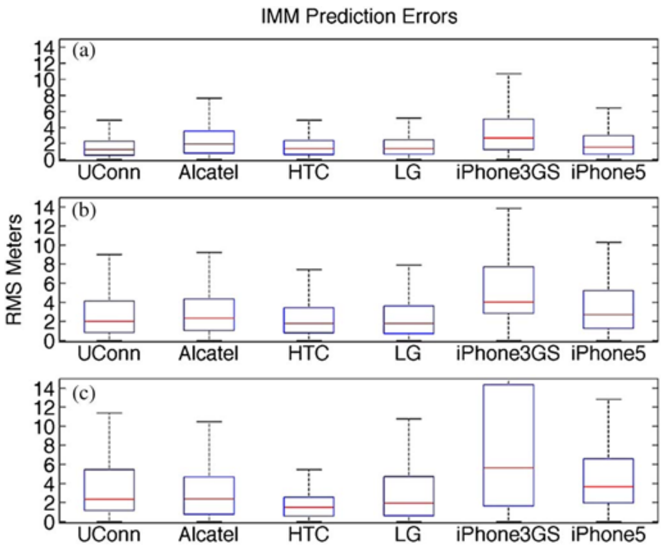

Fig. 7. Prediction errors during (a) straight paths, (b) smooth curves, and (c) sharp curves.

and it calculates the distance vector to the predicted position. Then, the mean and standard deviation were calculated for all of the calculated rms error vectors for the whole trajectory and also for those classified by segment types.

As expected, the prediction error was less during straight paths, and it increased during curves. Based on the values recorded in Table IV, the prediction errors can double during curves. Also, the prediction errors for smooth curves were better than during the sharp curves, which makes sense because, in a smooth curve, the vehicle is changing its heading less abruptly than in a sharp curve, allowing the system more time to recalculate and correct its next prediction.

This research also observed that the prediction error was not the same between all devices, and sometimes, a device that has a small prediction error in one segment type may not be as good on a different segment type, making it hard to draw conclusions from Table IV. In spite of these results, if one looks at the percent deviation of prediction errors compared to the UConn results, it can be narrowed down to the HTC and LG cellphones having the smaller prediction errors overall and meeting the tolerance of no more than 10% more error than obtained with the UConn sensors.

Fig. 7 is another way of representing the prediction errors for each of the devices in the different segment types. The boxplots display the median value as the solid line dividing the box in two, and then, the upper and lower halves of the boxes represent the interquartiles, which together represent 50% of the calculated prediction errors. The upper whisker indicates that 75% of the errors fall below it, and the lower whisker indicates the 25% marker. With this in mind, it can be seen that, for the straight paths, except for the iPhone3GS, the boxes are very short, which means that the prediction errors have a high level of agreement. One can also see small boxes in the smooth and sharp curves for the HTC and LG, so it can be observed that their predictions are fairly consistent most of the time, unlike the boxplot for the iPhone3GS where it is a very large box, indicating a very low level of agreement between the predictions. Also, the lower the boxes to the x-axis, the smaller the prediction errors, so a small box close to the x-axis, like the HTC in sharp curves or the UConn in straight lines, represents a very accurate prediction system. Again, looking at the boxplots for the five smartphones, one can visually pick the HTC and LG to be fairly good, then maybe Alcatel and iPhone5, although it looks like the iPhone5 is not as reliable as Alcatel during sharp curves.

TABLE V
RMS PREDICTION ERROR (3 S AHEAD)

	UConn	*Alcatel*	*HTC*	*LG*	*iPhone3GS*	*iPhone5*
Whole	1.46	3.22	1.81	1.56	6.34	1.79
Trajectory	±0.97	±3.64	±1.87	±1.65	±6.95	±2.18
Straight	0.69	2.96	1.42	0.77	4.50	0.95
Paths	±0.53	±4.60	±2.11	±0.58	±5.76	±0.74
Smooth	1.69	3.13	1.75	1.84	11.57	1.57
Curves	±0.54	±2.18	±1.43	±2.13	±6792	±0.85
Sharp	1.98	3.52	2.67	2.19	14.73	2.69
Curves	±0.97	±3.81	±2.03	±1.26	±6.24	±3.12

Values represent median prediction error in RMS meters ± standard deviation by all devices for the selected curve (~550 data points).

When looking at the iPhone3GS results, both in Table IV and Fig. 7, it can be observed that this device has prediction errors much larger than the other devices. It seems that this device has a problem, losing its signal quite often, introducing more errors to what was assumed to be the "true" position. Looking more into this topic, we have found several Apple discussion forums (discussions.apple.com) where users have reported very inaccurate GPS locations when using the iPhone3GS running iOS5.

To look at a subset of the whole route shown in Fig. 6, a couple of curves were selected, and the results represented in a similar way but only for this small subset of the data set. This selected segment of the route represents only 0.8 km (36% straight path, 44% smooth curve, and 20% sharp curve), which takes around 10 s to go through.

When looking at Table V, the first difference that might be observed when comparing it to the results for the whole route shown in Table IV is that the average prediction error for the whole trajectory of the selected subset is different. In this case, straight paths are a small percentage of the whole selected subset, while smooth curves are the most abundant. For this very specific set of curves, the UConn data are better than any of the smartphones in all trajectory types. The LG device yields the smallest prediction errors of all of the smartphones, which are still within the selected 10% tolerance when comparing to the UConn results. The next best devices seem to be the HTC and iPhone5 smartphones, where, despite having prediction errors over the 10% tolerance, their prediction errors are around 20% worse than the UConn results.

Another difference one can observe in Table V is that, unlike the results in Table III, the HTC device did not seem to perform as well in this selected set of curves than when evaluated over the whole route. Even when looking at the results for smooth and sharp curves, the HTC results were always worse than the UConn prediction errors, which is not the case when looking at the data in Table IV. In Table V,

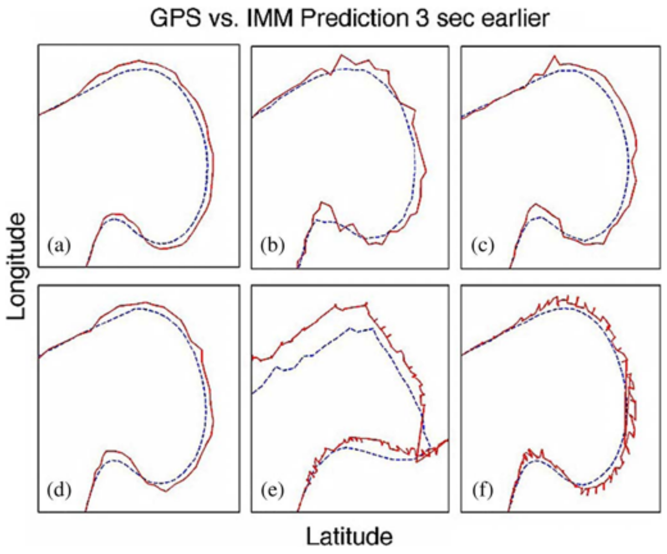

Fig. 8. Dotted blue lines represent GPS measurements, while solid red lines represent IMM position predicted 3 s earlier for (a) UConn, (b) Alcatel, (c) HTC, (d) LG, (e) iPhone3GS, and (f) iPhone5.

it seems that most smartphones performed worse than the UConn setup in this set of curves. Since this is consistent across all smartphones, it can be concluded that there was something on the curves that affected the prediction, like bumps or maybe unleveled pavement, especially over the sharp curve section.

A visual representation of the prediction errors during the small subset of curves previously mentioned is shown in Fig. 8, where each predicted trajectory is compared to the actual GPS position measured 3 s later. One can observe that, for some devices, there is a smooth trajectory of predicted positions, like those for the UConn and LG, closely followed by the HTC, but it can also be observed that some other devices are constantly correcting its predicted position drastically, causing all those spikes during the curves. One positive thing of looking at the predicted position errors as shown in Fig. 8 is that, despite conclusions obtained from Table V that the iPhone5 was predicting a future position better than several of the other devices, this research would not think that this is a reliable device after looking at Fig. 8. Therefore, even though Fig. 8 cannot be used by itself to draw some final conclusions, it is a very useful addition to Table V.

An important lesson learned in this research was that the accelerometer sensor could provide misleading information based on road slopes. Table VI shows a data subset where velocity was increasing but tangential acceleration was decreasing, implying that the vehicle was slowing down. Looking at the data, it can be observed that the vehicle was actually speeding up, and the decrease on the tangential acceleration was actually because the road was going up (altitude increase) and not because the vehicle was slowing down.

TABLE VI
MISLEADING ACCELEROMETER READINGS

Timestam p	*Altitude* (m)	*Velocity* (m/s)	*Acc.tang* (m/s^2)
33:17	117.93	18.06	-0.508
33:18	118.03	18.30	-0.593
33:19	118.12	18.53	-1.061
33:20	118.27	18.77	-1.168
33:21	118.80	19.07	-1.858

Misleading accelerometer measurements due to altitude changes.

IX. CONCLUSION

The built-in sensors of some smartphones were evaluated to predict the future trajectory of a vehicle and their prediction errors compared to those obtained by using VM sensors. The results shown in Tables IV and V indicate that smartphones yield similar or better prediction errors and could therefore be used in older vehicles to participate in a V2V/V2I system. Although smartphone price seemed to play a small role, the HTC smartphone is one of the cheaper ones used in this experiment, and it performed quite well in some scenarios. The more expensive LG device yielded more reliable results in more, so price could be a factor, but then, iPhone 5, being the most expensive one, did not contribute well to the price factor. If the results of the trajectory prediction errors for the whole trajectory driven for this experiment (shown in Table IV) are observed, it can be concluded that all smartphones, except for the iPhone3GS, yield prediction errors up to the 10% more than the UConn errors. It can be observed that the results from the smartphones perform, in most scenarios described in Table IV, even better than the baseline set by the VM sensors (UConn), some even almost 50% better. With these results, it can be safely concluded that the smartphones' sensor trajectory prediction accuracy is sufficiently reliable to be considered for a temporary hook to enable an older vehicle to participate in a V2V/V2I system.

Future research will include the use of other sensors found in some of the smartphones used (gyroscope and magnetic compass) to evaluate if they improve prediction of a vehicle's trajectory and how these predicted trajectories can fit into collision avoidance systems.

REFERENCES

[1] 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.

[2] 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.

[3] T. Taleb, A. Benslimane, and K. B. 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.

[4] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," ROBOMECH J., vol. 1, no. 1, pp. 1–14, Dec. 2014.

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

[6] 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, Emerging Technol., vol. 18, no. 4, pp. 611–625, Aug. 2010.

[7] C. Hakyoung, L. Ojeda, and J. Borenstein, "Accurate mobile robot dead-reckoning with a precision-calibrated fiber-optic gyroscope," IEEE Trans. Robot. Autom., vol. 17, no. 1, pp. 80–84, Feb. 2001.

[8] R. Toledo-Moreo, M. Pinzolas-Prado, and J. M. 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.

[9] J. Ueki, J. M. Y. Nakamura, Y. Horii, and H. Okada, “Development of vehicular-collision avoidance support system by inter-vehicle communications,” in Proc. 59th IEEE Veh. Technol. Conf., May 2004, vol. 5, pp. 2940–2945.

[10] Y. Wang, S. Wang, and M. Tan, “Path generation of autonomous approach to a moving ship for unmanned vehicles,” IEEE Trans. Ind. Electron., vol. 62, no. 9, pp. 5619–5629, Sep. 2015.

[11] J. J. Blum and A. Eskandarian, “A reliable link-layer protocol for robust and scalable intervehicle communications,” IEEE Trans. Intell. Transp. Syst., vol. 8, no. 1, pp. 4–13, Mar. 2007.

[12] J. Wen-Long and W. Recker, “An analytical model of multihop connectivity of inter-vehicle communication systems,” IEEE Trans. Wireless Commun., vol. 9, no. 1, pp. 106–112, Jan. 2010.

[13] B. D. Chiara, F. Deflorio, and S. Diwan, “Assessing the effects of inter-vehicle communication systems on road safety,” IET Intell. Transp. Syst., vol. 3, no. 2, pp. 225–235, Jun. 2009.

[14] S. Sohaib and D. K. C. So, “Asynchronous cooperative relaying for vehicle-to-vehicle communications,” IEEE Trans. Commun., vol. 61, no. 5, pp. 1732–1738, May 2013.

[15] M. Fogue et al., “Automatic accident detection: Assistance through communication technologies and vehicles,” IEEE Veh. Technol. Mag., vol. 7, no. 3, pp. 90–100, Sep. 2012.

[16] L. Chunhua, K. T. Chau, W. Diyun, and G. Shuang, “Opportunities and challenges of vehicle-to-home, vehicle-to-vehicle, and vehicle-to-grid technologies,” Proc. IEEE, vol. 101, no. 11, pp. 2409–2427, Nov. 2013.

[17] J. C. Ferreira, V. Monteiro, and J. L. Afonso, “Vehicle-to-anything application (V2Anything App) for electric vehicles,” IEEE Trans. Ind. Informat., vol. 10, no. 3, pp. 1927–1937, Aug. 2014.

[18] D. Freidman, “V2V: Cars Communicating to Prevent Crashes, Deaths, Injuries,” US Dept. Transp. (DOT), Washington, DC, USA, Feb. 3, 2014. [Online]. Available: http://www.dot.gov/fastlane/v2v-cars-communicating-prevent-crashes-deaths-injuries

[19] J. Gorselanv, “Cars That Can Last for 250,000 Miles (or More),” Forbes Corp., New York, NY, USA, Mar. 14, 2013. [Online]. Available: http://www.forbes.com/sites/jimgorzelany/2013/03/14/cars-that-can-last-for-250000-miles/

[20] “Data Analytics and Processing,” Cambridge Mobile Telematics, Cambridge, MA, USA, Aug. 2013. [Online]. Available: http://www.cmtelematics.com/product/data-analytics

[21] R. C. Luo, C. C. Y. , and K. L. Su, “Multisensor fusion and integration—Approaches, applications, and future research directions,” IEEE Trans. Sens., vol. 2, no. 2, pp. 107–119, Apr. 2002.

[22] J. B. Gao and C. J. Harris, “Some remarks on Kalman filters for the multisensor fusion,” Inf. Fusion, vol. 3, no. 3, pp. 191–201, 2002.

[23] S. C. Felter, “An overview of decentralized Kalman filter techniques,” in Proc. IEEE Southern Tier Tech. Conf., Apr. 1990, pp. 79–87.

[24] M. Hua, T. Bailey, P. Thompson, and H. Durrant-Whyte, “Decentralised solutions to the cooperative multi-platform navigation problem,” IEEE Trans. Aerosp. Electron. Syst., vol. 47, no. 2, pp. 1433–1449, Apr. 2011.

[25] E. M. Nebot, M. Bozorg, and H. F. Durrant-Whyte, “Decentralized architecture for asynchronous sensors,” Auton. Robots, vol. 6, no. 2, pp. 147–164, Apr. 1999.

[26] 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.

[27] 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.

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

[29] A. T. Alouani and T. R. Rice, “On optimal asynchronous track fusion,” in Proc. 1st IEEE Australian Symp. Data Fusion, Nov. 1996, pp. 147–152.

[30] G. A. Watson, T. R. Rice, and A. T. Alouani, “Optimal track fusion with feedback for multiple asynchronous measurements,” in Proc. SPIE Acquisition, Tracking Pointing, Apr. 2000, vol. XIV, pp. 20–33.

[31] M. W. Owen and S. C. Stubberud, “Interacting multiple model tracking using a neural extended Kalman filter,” in Proc. IEEE Int. Joint Conf. Neural Netw., 1999, vol. 4, pp. 2788–2791.

[32] J. Wang, 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.

[33] 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.

[34] R. Kalman, “A new approach to linear filtering and prediction problems,” Trans. ASME, vol. 82, no. 1, pp. 34–45, Mar. 1960.

[35] 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.

[36] G. Welch and G. B., “An Introduction to the Kalman Filter,” Dept. Comput. Sci., Chapel Hill, NC, USA, SIGGRAPH Course Notes, 2001.

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

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

[39] S. Y. Chen, “Kalman filter for robot vision: A survey,” IEEE Trans. Ind. Electron., vol. 9, no. 11, pp. 4409–4420, Nov. 2012.

[40] C.-L. Lin, Y.-M. Chang, C.-C. Hung, C.-D. Tu, and C.-Y. Chuang, “Position estimation and smooth tracking with a fuzzy-logic-based adaptive strong tracking Kalman filter for capacitive touch panels,” IEEE Trans. Ind. Electron., vol. 62, no. 8, pp. 5097–5108, Aug. 2015.

[41] 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.

[42] 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.

[43] B. Aubert, J. Régnier, S. Caux, and D. Alejo, “Kalman-filter-based indicator for online interturn short circuits detection in permanent-magnet synchronous generators,” IEEE Trans. Ind. Electron., vol. 62, no. 3, pp. 1921,1930, Mar. 2015.

[44] R. K. Singleton, E. G. Strangas, and S. Aviyente, “Extended Kalman filtering for remaining-useful-life estimation of bearings,” IEEE Trans. Ind. Electron., vol. 62, no. 3, pp. 1781–1791, Mar. 2015.

[45] C. F. Graetzel, B. J. Nelson, and S. N. Fry, “A dynamic region-of-interest vision tracking system applied to the real-time wing kinematic analysis of tethered drosophila,” IEEE Trans. Autom. Sci. Eng., vol. 7, no. 3, pp. 463–473, Jul. 2010.

[46] F. Auger et al., “Industrial applications of the Kalman filter: A review,” IEEE Trans. Ind. Electron., vol. 60, no. 12, pp. 5458–5471, Dec. 2013.

[47] 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.

[48] 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.

[49] L. Wenling and J. Yingmin, “Location of mobile station with maneuvers using an IMM-based cubature Kalman filter,” IEEE Trans. Ind. Electron., vol. 59, no. 11, pp. 4338–4348, Nov. 2012.

[50] 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.

[51] H. A. P. Blom and Y. Bar-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.

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

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

He has been with IBM since receiving the B.S. degree in 1999. He first started in the information technology field and then moved to the IBM Semiconductor Research and Development Center. He was with IBM Global Services, Morristown, NJ, in 2001–2003 and with IBM Microelectronics, Burlington, in 2003–2015. Since 2015, he has been with GlobalFoundries, Burlington.

Yuichi Motai (M'01) 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.