# A Low-Cost Lane-Determination System Using GNSS/IMU Fusion and HMM-Based Multistage Map Matching

Mohamed Maher Atia, Allaa R. Hilal, Clive Stellings, Eric Hartwell, Jason Toonstra, William B. Miners, and Otman A. Basir

***Abstract*— This paper presents a low-cost real-time lane-determination system that fuses micro-electromechanical systems inertial sensors (accelerometers and gyroscopes), global navigation satellite system (GNSS), and commercially available road network maps. The system can be used for intelligent transportation systems, telematics applications, and autonomous driving. The system does not depend on visual markings or highly precise GNSS technology, such as DGPS or RTK, and it does not need explicit lane-level resolution maps. High-resolution estimation of the vehicle's position, velocity, and orientation is implemented by fusing inertial sensors with GNSS in a loosely coupled mode using extended Kalman filter. A curve-to-curve road-level map-matching is implemented using a hidden Markov model followed by a least-square regression step that estimates the vehicle's lane. The system includes a lane-change detector based on inertial sensors and the filtered vehicle's state. The system has been realized in real time and tested extensively on real-road data. Experiments showed robust map-matching in challenging road intersections and a 97.14% lane-determination success rate.**

***Index Terms*— Lane-determination, sensor fusion, map matching, hidden Markov models, extended Kalman filter, MEMS IMU, GPS.**

## I. Introduction

LANE determination is needed in several vehicular navigation applications. For example, a toll-charge system determines the vehicle's lane to enforce lane-dedication rules such as high-occupancy vehicle (HOV). Another example is self-driving cars where accurate lane determination is important for better guidance and control. Modern in-car navigation systems need lane information to perform lane-level, turn-by-turn guidance. Telematics applications is also a growing market that needs lane information for precise registration of telematics data.

Lane-determination systems have been actively considered in the last decade. Few articles that address this research can be found in early 90's [1], [2], [3]. The dominant technologies since the 90's have been vision and range-sensor- (e.g., light detection and ranging "LiDAR"). Vision/LiDAR lane determination systems depend on objects recognition and road features detection. Therefore, in addition to high computation requirements, they need clear visual features (e.g. lane markers) which may not be available in all scenarios. With the availability of low-cost global positioning and navigation technologies such as Global Positioning System "GPS" [4], micro-electromechanical systems (MEMS) inertial measurement units "IMU" [5] and geographical information systems (GIS), alternate lane-determination can be developed.

According to the FAA GPS Performance Analysis Report [6], horizontal accuracy of GPS (Standard Positioning Service "SPS") is within 3.351m with a 95% confidence level. Some recent technical reports such as [7] reported a 2.5m average horizontal accuracy over a 24-hours static test. A typical lane width is 3.65m which needs an unbiased precise positioning solution of much less than 1.825m. If a safety margin of 50% is considered, unbiased precise positioning of less than 0.9125m is needed. Therefore, a standard SPS GPS technology is not precise enough to accurately determine the vehicle's lane under conventional map matching techniques. Some recent research that address these issues has been reported in [8] and [9]. However, they require a high precision positioning technology (e.g. differential GPS "DGPS") and preprocessed lane-based road maps. Although GPS technology is significantly improved using several techniques such as DGPS, Augmented GPS (i.e. GPS enhanced by networks of ground reference stations or other satellites), Precise Positioning Services (PPS), or real-time kinematics (RTK) [4], [10], these techniques require additional infrastructures (e.g. reference stations coverage and communication links) and extra processing.

This paper contributes in the present efforts by proposing a low-cost lane-determination system that fuses measurements from a standard low-cost SPS GPS, MEMS IMU, and commercially available road-level network. The specifications of the systems components/sensors are given in the experimental work section in Table 2 and Table 3. The system proposes a

Manuscript received April 18, 2016; revised September 21, 2016; accepted February 5, 2017. This work was supported in part by Mitacs, Queen's University, and in part by Intelligent Mechatronic Systems Inc., under Mitacs Project 343584. The Associate Editor for this paper was K. Wang.

M. M. Atia is with the Department of Electronics, Carleton University, Ottawa, ON K1S 5B6, Canada (e-mail: mohamed.maher.atia@gmail.com).

A. R. Hilal, C. Stellings, E. Hartwell, J. Toonstra, and B. Miners are with Intelligent Mechatronic Systems Inc., Waterloo, ON N2J 2Z5, Canada (e-mail: ahilal@intellimec.com; clivestellings@intellimec.com; erichartwell@intellimec.com; jasontoonstra@intellimec.com; bminers@intellimec.com).

O. A. Basir is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: obasir@uwaterloo.ca).

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/TITS.2017.2672541

multi-phase design, where a sensor fusion engine is developed to fuse measurements from GPS and IMU using a loosely-coupled extended Kalman Filter (EKF) module. The second phase is a computationally efficient two-stage map-matching method using a Hidden-Markov-Model (HMM) [11] and a least-squares (LS) regression algorithm. The proposed system can handle temporary GNSS degradations and can detect lane changes. In addition, the proposed multi-stage map-matching method saves massive storage required to save explicit lane-level spatial reference information and it reduces the computational complexity of HMM decoding algorithm by reducing the number of the road segments HMM needs to decode.

This paper is organized as follows: section (II) provides a background and state of art about lane determination systems. Section (III) provides problem formulation. Section (IV) explains the methodology. Experimental work and results are given in section (V), followed by a conclusion section in VI.

## II. Related Work

Current car navigation systems are dominated by Global Navigation Satellite Systems (GNSS) which is commonly integrated with digital road network maps [12]–[14]. In order to enhance positioning precision and availability, dead-reckoning (DR) using extra sensors such as IMUs and speed measurements has been used [5]. The integration is commonly performed using sensors fusion algorithms such as Kalman Filtering (KF) and Particle Filtering (PF) [5], [10]. Precise figures on the accuracy of the state-of-the-art integrated car navigation systems may vary depending on the testing environment and fusion algorithms [15]–[16]. To provide lane information, a highly precise positioning with errors that are sufficiently below a half of a lane-width (0.9125m) and accurate lane-level maps are needed. Current commercially available maps are not commonly detailed and accurate enough for direct lane level matching. However, some map providers (e.g. NAVTEQ maps [17]) provide lane information such as number of lanes and legal directions. Although research in using vision and LiDAR for lane-determination is very active and promising [18], recent researches target the fusion of maps with GNSS and high-rate motion sensors (e.g. IMU) as low-cost and computationally efficient alternatives. Interesting work that realizes this approach can be found in [19] and [9] where DGPS is used to create refined maps with lane-level resolution. In this system, DGPS was used for positioning as well. This work utilized DR technique and fused IMU, DGPS, and enhanced lane-level maps using a PF. In support of this research direction, significant efforts are being performed to develop accurate lane-level maps [8]. Another proposed system can be found in [21] where the authors fuse DGPS with highly accurate lane-level maps. In [20], the lane-level maps are generated from geo-referenced, geo-rectified aerial images. The map-matching method applied in [20] is curve-to-curve matching [15]. The main drawbacks of such systems are the dependency on highly accurate geo-referenced, lane-level map resolution. They also assume DGPS availability. Working directly on lane-level resolution increases map-matching algorithms complexity due to the high number of map segments (lanes). An interesting method that utilizes driving

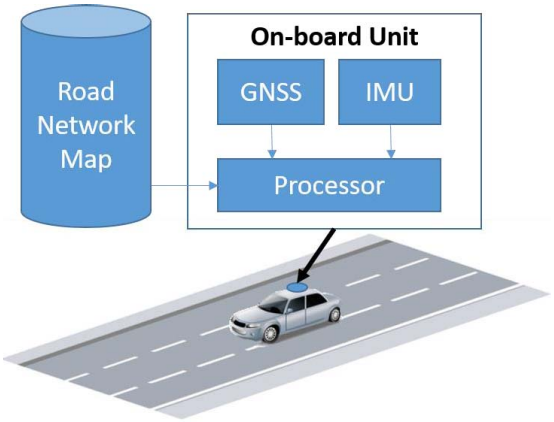


Fig. 1. General Block Diagram with an Illustrative Figure of the System.

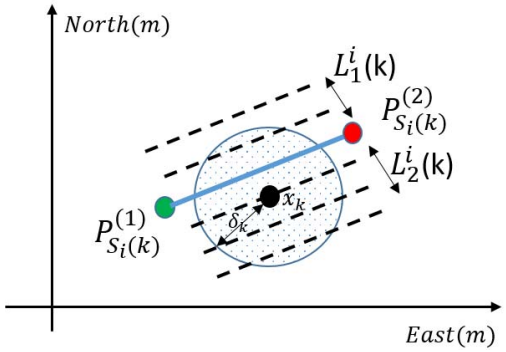


Fig. 2. Geometric Illustration of the Lane Determination Problem.

pattern heuristics and a probabilistic approach is presented in [21] with a claimed accuracy of 80%. This is one of the innovative systems that do not need geo-referenced, or geo-rectified, lane-level resolution maps. However, it needs lane information (mainly the number of lanes) at the current vehicle position. To further enhance accuracy, combining DGNSS, maps, and DR sensors with vision/LiDAR-based methods has been considered in [22].

## III. Problem Definition

The lane-determination problem considered in this work can be described as follows: given a standard SPS GPS receiver, a 3D MEMS IMU, and a network of geo-referenced road segments, it is required to determine the lane on which the vehicle is currently traveling. The problem and the desired system are depicted in Figure 1.

To further illustrate the problem, a geometric illustration is given in Figure 2. $\hat{x}_k$ is the vehicle position at time instant $k$ with error covariance $\delta_k^2$. The road's network be represented as a set of segments $\{S_1, S_2, \ldots, S_n\}$. Each road segment is defined by a straight line segment with start position $P_S^{(1)}$ and end position $P_S^{(2)}$. Curved roads are approximated by sufficient decomposition of straight line segments. $\{L_1^i, L_2^i\}$ is the number of lanes on both sides of the road segment $S_i$. Based on this notation and geometric illustration, the estimation problem that this paper is addressing is the determination of $\{S_i(k), l_i(k)\}$ where $S_i(k)$ is the estimated road segment on

which the vehicle is traveling at time $k$ and $l_i(k)$ is the lane the vehicle is traveling on.

## IV. The Proposed System

### A. Integrated Navigation Using Kalman Filter

Consider a dynamic system of states $x(t)$ given by:

$$\dot{x}(t) = f(x(t), u(t)) \tag{1}$$

$$y(t) = h(x(t)) \tag{2}$$

where $f(.)$ is a nonlinear system model, $u(t)$ is a noisy input, $y(t)$ is observables, $h(.)$ is a nonlinear measurement model. Noisy sensor inputs can be modelled as:

$$u(t) = s(t) + b_s(t) + \delta_s w(t) \tag{3}$$

where $s(t)$ is the true measurement, $b_s(t)$ is a deterministic error (i.e. bias), and $\delta_s$ is a stochastic random error component commonly modeled as Gaussian of covariance $\delta_s^2$, $w(t)$ is zero-mean Gaussian noise of covariance matrix $Q(t)$. If the system in (1) and (2) is linearized around an operating state, EKF can provide optimal estimate of the error states under unbiased Gaussian noisy measurements. By linearizing (1-2) we obtain:

$$\delta\dot{x}(t) = F(t)\delta x(t) + G(t)w(t) \tag{4}$$

$$\delta y(t) = H(t)\delta x + \mathrm{v}(t) \tag{5}$$

Where (5)

$$F(t) = \frac{\partial f(x(t), u(t))}{\partial x} \tag{6}$$

$$H(t) = \frac{\partial h(x(t))}{\partial x} \tag{7}$$

$$G(t) = \frac{\partial f(x(t), u(t))}{\partial w} \tag{8}$$

and $\mathrm{v}(t)$ is zero-mean Gaussian noise of covariance matrix $R(t)$. In discrete form, states are predicted by:

$$x_{k+1} = f(x_k, u_k) \tag{9}$$

$$P_{k+1} = \phi_k P_k \phi_k^T + Q_{d_{k+1}} \tag{10}$$

where $P_k$ is the error state covariance, $\phi_k$ is system transition matrix and $Q_{d_{k+1}}$ is system noise shaping matrix. $\phi_k$ and $Q_{d_{k+1}}$ can be approximated by [5]:

$$\phi_k \simeq (I + F_k T) \tag{11}$$

$$Q_{d_{k+1}} \simeq T^2 G Q G^T \tag{12}$$

where $T$ is the sampling period. When measurement $y_{k+1}$ is available, the states are corrected by:

$$K_{k+1} = P_{k+1} H_{k+1}^T (H_{k+1} P_{k+1} H_{k+1}^T + R_{k+1})^{-1} \tag{13}$$

$$\delta x_{k+1} = K_{k+1}\left[y_{k+1} - h(x_{k+1})\right] \tag{14}$$

$$P_{k+1} = (I - K_{k+1} H_{k+1}) P_{k+1} \tag{15}$$

More details about EKF derivation and approximations can be found in [5], [10], and [23]. In this work, the vehicle's state vector includes position, velocity, orientation, and IMU sensors biases as follows:

$$x_k \triangleq \left[p_k^{(l)}; v_k^{(l)}; \varphi_k^{lb}; b_a(t); b_\omega(t)\right] \tag{16}$$

where $p_k^{(l)}$ is the north-east-down (NED) position vector in local navigation frame $l$ (see Table 1 for all reference frames description), $v_k^{(l)}$ is the velocity vector, $\varphi_k^{lb}\,[\psi_k \theta_k \varphi_k]$ is the vehicle's orientation represented in Euler angles (heading, pitch, and roll), $b_a(t)$ and $b_\omega(t)$ are biases of IMU accelerometers and gyroscopes respectively. The vehicle motion model is given by:

TABLE I
Coordinate Reference Frames

| V. Frame | Definition |
|---|---|
| b: body frame | Origin: Vehicle center of mass<br>X: Longitudinal (forward) direction<br>Y: Transversal (lateral) direction<br>Z: Down (vertical) direction |
| l: local-level frame | Origin: Vehicle centre of mass<br>X: True north direction<br>Y: East direction<br>Z: Down vertical direction |
| Earth-Centered Earth-Fixed (ECEF) frame | Origin: Center of the Earth<br>Z: extends through the North Pole.<br>X: passes through the intersection of the equatorial plane and the prime meridian.<br>Y: completes the right-hand coordinate system in the equatorial plane. |

$$\dot{p}_{INS}^{(l)}(t) = v_{INS}^{(l)}(t) \tag{17}$$

$$\dot{v}_{INS}^{(l)}(t) = R_b^l(t)\left(a^{(b)}(t) - b_a(t)\right) + g^{(l)}(t) - \left(\omega_{EL}^{(l)} + 2\omega_{IE}^{(l)}\right) \mathrm{x}\, v_{INS}^{(l)}(t) \tag{18}$$

$$\dot{R}_b^l(t) = R_b^l(t)\,\Omega\left(\omega^{(b)}(t) - b_\omega(t)\right) - \Omega\left(\omega_{IE}^{(l)} + \omega_{EL}^{(l)}\right) R_b^l(t) \tag{19}$$

where $a^{(b)}(t)$ is the accelerometer signal in body-frame, $\omega^{(b)}$ is the gyroscope signal in body-frame, $R_b^l(t)$ is the vehicle's orientation direction cosine matrix, $\omega_{EL}^{(l)}$ is the rotation rate of $l$ frame with respect to ECEF frame due to motion on the ellipsoid surface of the Earth (i.e. transport rate [5]) and $\omega_{IE}^{(l)}$ is the earth rotation rate. $\Omega$ is the skew matrix operator [5], [10], and [23]. Biases can be modeled as Gauss-Markov(GM) [5] random process as follows:

$$\dot{b}_a(t) = -\beta_a b_a(t) + \sqrt{2\sigma_a^2 \beta_a} w(t) \tag{20}$$

$$\dot{b}_\omega(t) = -\beta_\omega b_\omega(t) + \sqrt{2\sigma_\omega^2 \beta_\omega} w(t) \tag{21}$$

where $\beta_a^{-1}$, $\beta_\omega^{-1}$ are time constants, $\sigma_a^2$, $\sigma_\omega^2$ are covariances that describes GM process parameters of accelerometer and gyroscopes biases respectively. Equations (17-21) constitute the vehicle's nonlinear system model $f(.)$ which is linearized to obtain a 15 state linear error model. The linearization process and details of numerical implementation of (17-21) can be found in [5], [10], and [23]. The measurement error model is defined as follows:

$$\begin{bmatrix} p_{k+1}^{(l)}\big|_{GPS} - p_{k+1}^{(l)}\big|_{INS} \\ v_{k+1}^{(l)}\big|_{GPS} - v_{k+1}^{(l)}\big|_{INS} \end{bmatrix} = H\delta x_{k+1} \tag{22}$$

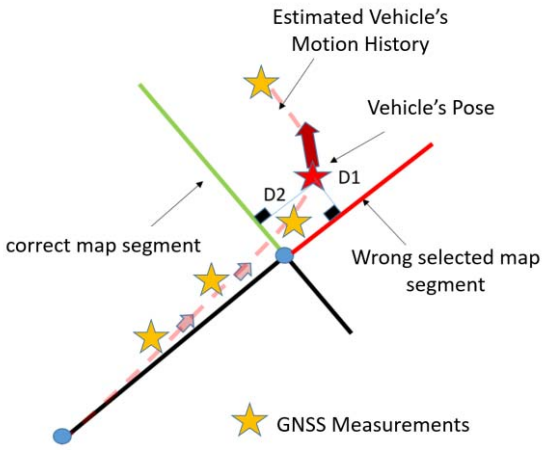


Fig. 3. Wrong map-segment selection in intersection.

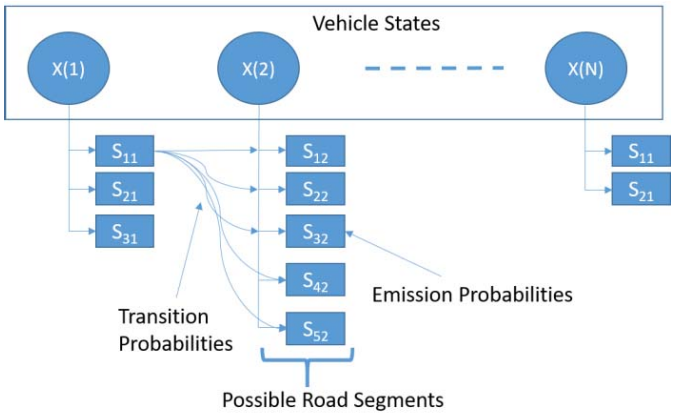


Fig. 4. Hidden Markov Model for Vehicle's State Map-matching.

where H is defined as follows:

$$H = \begin{bmatrix} I_{3x3} & 0_{3x3} & 0_{3x9} \\ 0_{3x3} & I_{3x3} & 0_{3x9} \end{bmatrix} \tag{23}$$

EKF is then performed by iteratively applying prediction/update steps described in (9-15).

### *B. HMM Map-Matching*

The simplest map-matching method is point-to-curve performed by searching for closest segments within a distance threshold [15]. However, this approach is sensitive to state estimation errors and it commonly fails in branching or dense parallel roads. Figure 3 shows a situation where pose error leads to wrong map segment due to dependency on distance measure only.

A better technique is to keep a record (buffer) of vehicle's poses and use it in the matching. This strategy is known as curve to curve matching [13], [15] and it generally considers the following constraints: 1) when projecting vehicle's position to multiple segments, segments that have legal direction closest to the vehicle's current heading should be preferred. 2) a transition from one segment to another should not violate the legal and logical road map connectivity. 3) a transition from one segment to another should not violate the vehicle's reasonable range of dynamics. 4) the geometry of vehicle poses must be consistent with the geometry of the selected road segments. To incorporate these constraints in a well-defined mathematical framework, HMM-based algorithm is proposed.

Markov model is a stochastic model that describes sequence of states $S = [s(1), s(2), s(3) \ldots, s(N)]$, where $s(k)$ depends only on $s(k-1)$. $S$ is called a Markov Process [11]. The transition from $s(k-1)$ to $s(k)$ is modeled by a conditional transition probability given by:

$$a_{ij} = p\left(s(k) = S_i \,|\, s(k-1) = S_j\right) \tag{24}$$

This transition probability forms a Markov model. The probability of any observed sequence under a certain Markov Process and Markov model $M$ is given by:

$$\begin{aligned} &p(s(1), s(2), s(3) \ldots, s(N) | M) \\ &\quad = p(s(1)) \prod_{i=1}^{N-1} p(s(i+1) | s(i)) \end{aligned} \tag{25}$$

If the states are not directly observable (hidden) but can be indirectly observed through a sequence of outputs $\{x(1), x(2), x(3) \ldots, x(N)\}$, the process is called a Hidden Markov Process. The HMM in this case is characterized by the transition probability and an emission probability; the probability a given state $s(k)$ generates an output $x(k)$:

$$p(s(k) \rightarrow x(k)) \tag{26}$$

A fundamental problem of HMM is: given a sequence of outputs $\{x(1), x(2), x(3) \ldots, x(N)\}$, what is the best sequence of states that explains the observed outputs. This problem is solved by selecting the sequence of states that maximizes the HMM probability as follows:

$$\begin{aligned} \bar{S} = &\arg\max_{S \in \{s(1), s(2), \ldots s(N)\}} \\ &\times \left\{ \begin{array}{l} p(s(1) \rightarrow x(1)) \prod_{i=1}^{N-1} p(s(i+1) | s(i)) \\ p(s(i+1) \rightarrow x(i+1)) \end{array} \right\} \end{aligned} \tag{27}$$

This estimation process is called "decoding" and it is solved using the Viterbi Algorithm [24]. In the proposed system, the hidden states represent map links and the observable outputs are the vehicle poses as shown in Figure 4. The emission probabilities are calculated as follows:

$$s_{il} = \frac{w_{il}}{\sum_{k \in M_i} w_{ik}} \tag{28}$$

where $M_i$ is a set of all map segments within 50m distance from current vehicle position in the $i^{th}$ epoch. The weighting parameter $w_{il}$ is calculated as follows:

$$w_{il} = Ae^{-aD_{il}} + Be^{-b\|\Delta A\|^2} \tag{29}$$

where $D_{il}$ is the distance from the vehicle's position in the $i^{th}$ epoch to its projection on the $l^{th}$ map segment. $\Delta A_{il}$ is the difference between the vehicle's heading and $l^{th}$ map segment "legal" heading. The parameters $A$,$a$,$B$, and $b$ are constants chosen to achieve balanced weighting of distance and direction of motion constraints. The transition probabilities are calculated as follows:

$$a_{lm}^{ij} = \frac{w_{lm}^{ij}}{\sum_{k \in L_l} w_{lk}^{ij}} \tag{30}$$

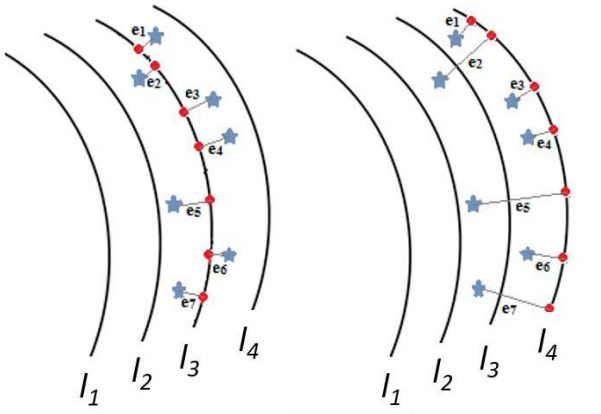


Fig. 5. Left, projection on lane 3. Right, projection on l4. Correct lane is l3.

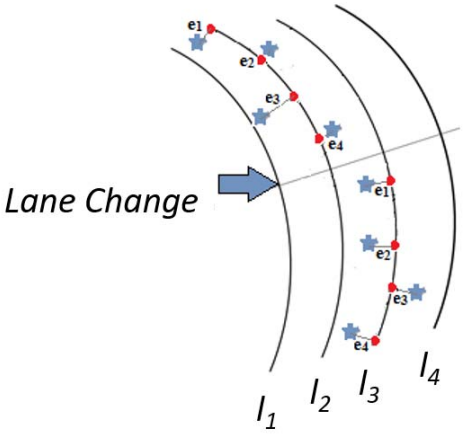


Fig. 6. Lane-Change Effect on Least-Square Error Calculation

where $a_{lm}^{ij}$ is the probability of transition from $l^{th}$ map link to $m^{th}$ map link given the vehicle' position at epochs $i$ and $j$ respectively. $L_l$ is the set of all map links connected to $l^{th}$ map link. The weighting parameter $w_{lm}^{ij}$ is calculated as follows:

$$w_{lm}^{ij} = C_{lm} e^{-c\|\Delta A_{ij} - \Delta A_{lm}\|} \tag{31}$$

where $\Delta A_{ij}$ is vehicle's heading change from $i^{th}$ to $j^{th}$ epochs, $\Delta A_{lm}$ is road segments' heading change from $l^{th}$ to $m^{th}$ segments and $c_{lm}$ is a connectivity parameter that determines if map segments $l^{th}$ and $m^{th}$ are topologically connected and $c$ is a constant parameter.

### C. Lane-Estimation Using Minimum Least-Squares

Once road segments have been selected, the projection of the vehicle's positions on segment lanes can be easily calculated knowing the lanes widths and number of lanes, as shown in Figure 5. The sum of squared errors for each lane $L_j$ is then calculated by:

$$E_{L_j} = \frac{1}{N}\sum_{i=1}^{N}\left[p_v(i) - p_{L_j}(i)\right]^2 = \frac{1}{N}\sum_{i=1}^{N} e_{ij}^2 \tag{32}$$

where $N$ is number of epochs, $p_{L_j}(i)$ is the projection of vehicle's position $p_v(i)$ on lane $L_j$. The lane associated with the minimum error is selected as the designated lane (see Figure 5 for a graphical illustration).

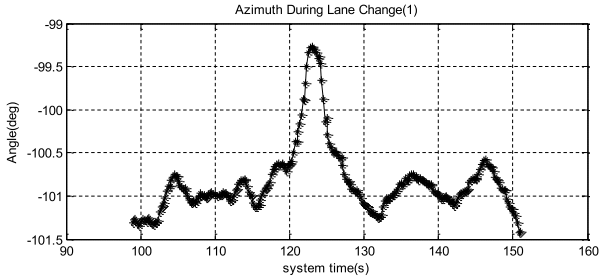


Fig. 7. Vehicle's Heading during Lane Change to Left

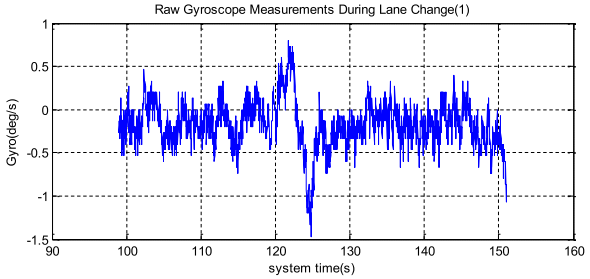


Fig. 8. Vehicle's Gyroscope Measurements during Lane Change to Left.

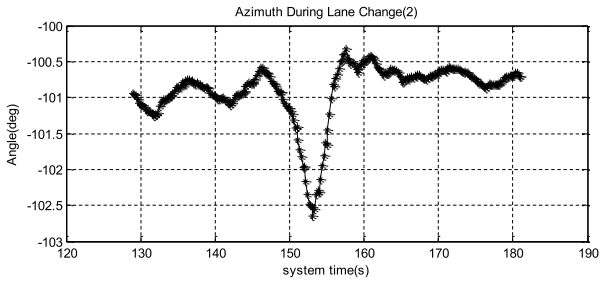


Fig. 9. Vehicle's Heading during Lane Change to Right.

### D. Lane-Change Detection

If a lane change occurred within the processed buffer of data, the least-square regression will not converge to the correct lane as depicted in Figure 6. Therefore, the buffer needs to be partitioned at the lane-switch locations. Thus, a lane-change detection module is developed. In this work, a lane-change detection method is designed based on capturing the patterns of the vehicle's orientation and raw gyroscope measurements. The heading and raw gyroscope measurements during two lane changes are depicted in Figure 7, Figure 8, Figure 9, and Figure 10. The general pattern that the lane change module detects is a peak or a valley in azimuth accompanied by a peak/valley or valley/peak sequence in the gyroscope measurements. To detect peaks and valleys, the standard deviation of a moving window of data is calculated and compared to a peak/valley threshold. If both gyro and azimuth peak/valleys sequence are consistent and matched with the pattern described above, a lane change is declared.

### E. GNSS Temporary Degradation

Although the experiments were mostly open-sky, it was noticed that the utilized GNSS receiver output does not reflect a lane change immediately. Instead, it might continue produce biased positioning estimate for few seconds. This causes a

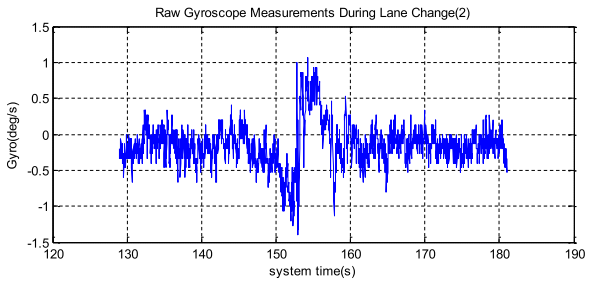


Fig. 10. Vehicle's Gyroscope Measurements during Lane Change to Right.

temporary error in lane determination. It was noticed also that sometimes GNSS provides temporary biased estimate that is not centered on the correct lane. This may be due to either internal receiver setting (e.g. motion models) or atmospheric errors. During these scenarios, there is an ambiguity in lane estimation. Therefore, the following two phases of processing are applied:

Acquisition Phase: In this mode, when the system starts, GNSS and IMU measurements are fused in a main EKF and HMM map-matching is performed and a lane is estimated. The innovation sequence of the main EKF which is the difference between the predicted state and GNSS updates, is calculated over a 10-seconds buffer of data. If innovation sequence is within a small threshold (30cm in this implementation) and no lane change has been detected, the acquisition phase is concluded and the tracking phase starts:

Tracking Phase: tracking phase will initiate two EKF filters. An instance of the EKF accepts position updates from the projection of the vehicle's position on the selected lane center, and the other EKF accepts GNSS position updates only. A discrepancy measure is evaluated between the two EKF instances for a short window of time (5 seconds in our implementation). This discrepancy measure is based on position estimation differences between the two EKF instances. If this discrepancy measure is higher than a threshold (1.5m in this particular implementation), a temporary GNSS deviation is declared and the system keep reporting current lane as the designated lane. If GNSS measurements started to be centered again on the new lane, a lane change is confirmed and the output of the first EKF instance will be considered to be the correct state. Otherwise, this lane change is declared as false and the second EKF output is considered to be the correct output. This strategy is more efficient than applying PF or unscented EKF since it instantiates only two EKF instances. A basic block diagram of the proposed system is shown in Figure 11.

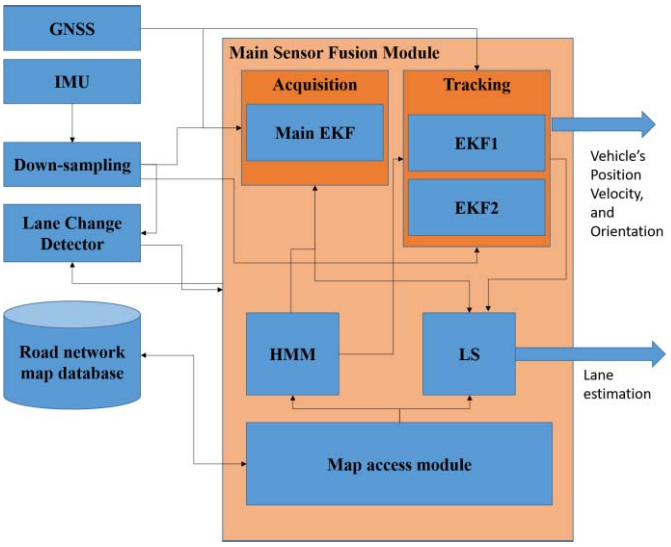


Fig. 11. Basic block diagram of the proposed system.

Fig. 12. Experimental Car (IMS Inc. Waterloo based).

TABLE II
u-BLOX 7 GNSS RECEIVER ACCURACY

| Parameter | Value |
|---|---|
| Horizontal position accuracy | 2.5 m |
| Velocity accuracy | 0.1 m/s |

TABLE III
3-SPACE SENSOR IMU SPECIFICATIONS

| Parameter | Value |
|---|---|
| Accelerometer noise density | 99μg/√Hz |
| Accelerometer range | ±4g |
| Gyro noise density | 0.009º/sec/√Hz |
| Gyro bias stability | 2.5º/hr |
| Gyro range | ±500/ |

## V. EXPERIMENTAL WORK AND RESULTS

The proposed system has been realized on an experimental setup consisting of a Gigabyte Intel Celeron N2807 1.58GHz Mini PC connected to Ublox EVK-7P kit GNSS receiver from Ublox and an automotive grade IMU 3D space sensor IMU from YOST Labs, and road network map data from NAVTEQ (now part of the HERE business unit [18]). A GPS-enabled camera (HP f310 car camcorder) was installed to capture video of the whole experiment, to be used as a ground truth to verify the results of our algorithms. The experimental car is shown in Figure 12 and the sensors specifications are given in Table 2 and Table 3. The effect of level arm (distance between IMU and GNSS antenna) was not considered in this implementation.

Three testing trajectories were collected during July 2015 through Highway 400 from Wilson Avenue

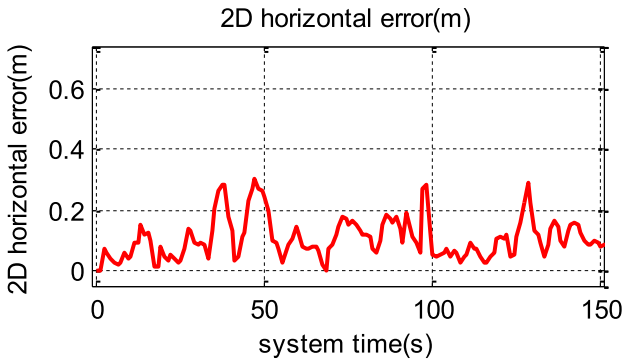


Fig. 13. 2D position horizontal error during acquisition phase.

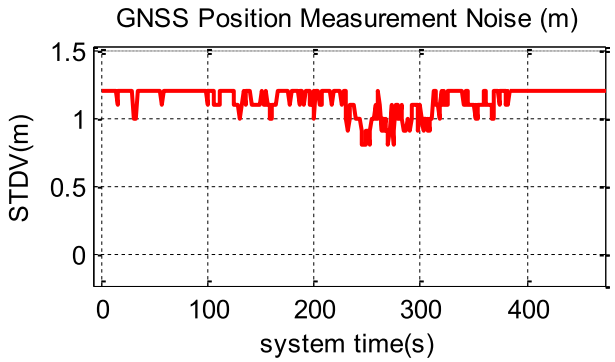


Fig. 14. Scaled 2D horizontal GNSS position measurement noise covariance during acquisition phase as reported by the receiver.

in the south to Davis Drive in the north. Approximately 65 km trip data was collected. The data included some urban areas but mostly open-sky. It also included challenging road intersections and roads joining/branching points.

The experimental setup was designed such that the Gigabyte computer box automatically starts when the vehicle's engine is turned on. A Linux OS was installed on the Gigabyte computer box and a data acquisition script written in Python was configured to automatically begin when the computer starts up. Measurements from the Ublox GNSS receiver at 1Hz and the 3D-Space IMU at 50 Hz were synchronized on the Gigabyte computer box. The main algorithm including GNSS/INS fusion and map matching was developed in native ANSI C language for efficient processing. Original raw IMU data was set to 50Hz down-sampled to 5Hz. Within this interval, the real-time system was able to fetch map information from a cached database file, perform basic prediction steps and implement the forward calculation of Viterbi algorithm (including calculation of emission and transition probabilities) that is needed for HMM map-matching step.

### *A. EKF Performance*

Figure 15 shows the convergence of the main EKF for longitude, latitude, and heading error covariance during acquisition phase. Figure 13 shows the error of the GNSS/INS integrated system which is within approximately 30cm. Figure 14 shows the GNSS position accuracy was scaled and used in the $R$ matrix to perform EKF measurement update. Figure 16 shows the convergence of accelerometer and gyroscope biases.

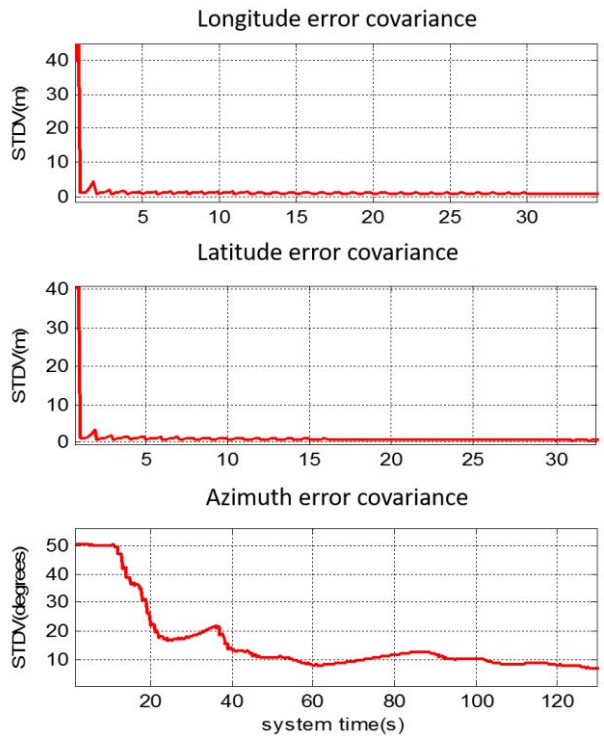


Fig. 15. EKF error state covariance of horizontal position and heading.

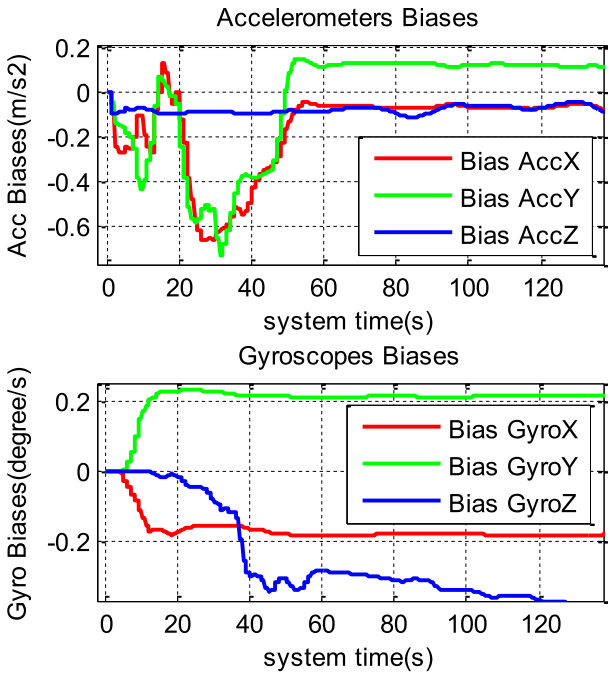


Fig. 16. Accelerometer and Gyroscope Biases during acquisition phase.

### *B. Map-Matching Accuracy*

The GNSS accuracy throughout the testing was typically within the SPS range (3.6-5m). The processing window was set to 50 epochs (10 seconds of measurements). The HMM-based map-matching algorithm performed well in all portions of the testing trajectories; including portions with challenging intersections, joints and exit/entry lanes. Graphical


Fig. 17. HMM-based map-matching. Green highlights the selected road links and. Dark circles are GPS output. Red track is the sensor fusion output.


Fig. 18. HMM-based map-matching in dense intersection. Green highlights the selected road links and. Dark circles are GPS output. Red track is the sensor fusion output.

representations of the HMM-based map-matching algorithm performance in challenging areas are shown in Figure 17, and Figure 18. On the other side, Figure 19 shows a portion in a parking lot where no detailed map coverage is available. With the exception of areas where no road network map coverage is available, such as most parking lots, the HMM-based map-matching algorithm achieved 100% accuracy in all testing trajectories. To detect these map-gap portions where no road network segment data is available, the normal distance between the vehicle's position and the nearest road segment is calculated. If it is larger than a threshold (fixed to 5m in this implementation), the area is declared as "no maps" and no lane estimation is applied to those areas.

### C. Lane-Determination Results

The estimated lanes were visually inspected record by record and results were saved in an Excel sheet. The results were written into a time-tagged file where each row can be easily visually inspected by looking at the portion of images corresponding to the same time-tag. The time-tag used was the UTC-time contained in the NMEA GNSS raw

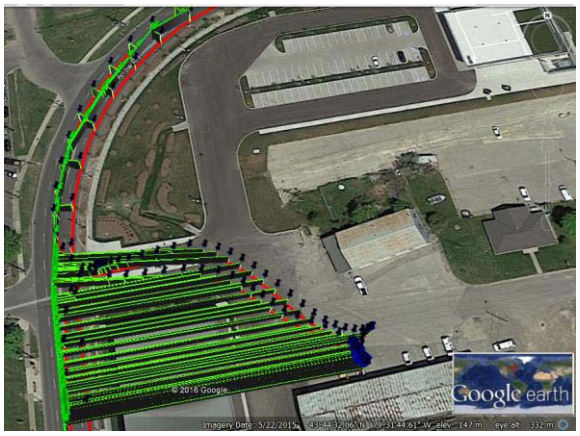


Fig. 19. HMM-based map-matching in a parking area. Green highlights the selected road links and. Dark circles are GNSS. Red track is the sensor fusion output.

TABLE IV
LANE-ESTIMATION ACCURACY

| Trajectory | Approximate Traveled Distance | Lane-Determination Accuracy |
|---|---|---|
| 1 | 23,234m | 95.2% |
| 2 | 20,630m | 98.32% |
| 3 | 21,231m | 97.9% |
| Overall | 65,095m | 97.14% |

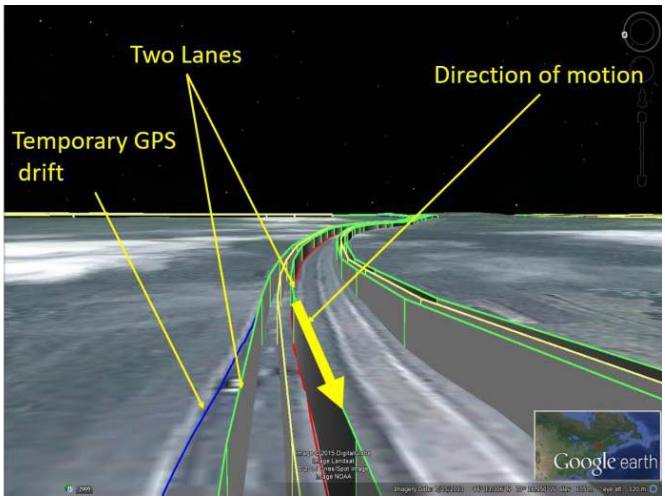


Fig. 20. Two lanes road. GPS (blue) is off road while sensor fusion output (red) follows the correct lane.

measurements. The overall accuracy of the proposed system in lane determination is shown in Table 4.

### D. Lane-Following and Sensor Fusion Performance

In the event of GNSS temporary drifts, the effect of the position update applied from the calculated lane position can be seen in Figure 20. In Figure 20, the GNSS (in blue) deviates while the sensor fusion output (in red) keeps the solution on the most recent selected lane. In Figure 21, a lane change

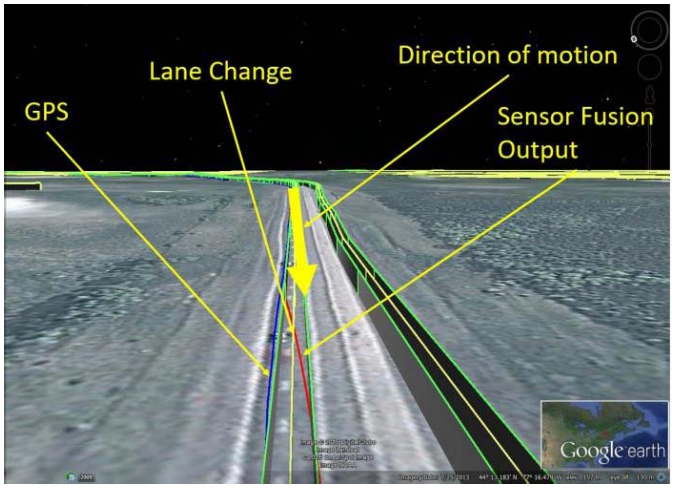


Fig. 21. Two-lane road. GPS (blue) is delayed during a lane change while sensor fusion output (red) follows the correct lane.

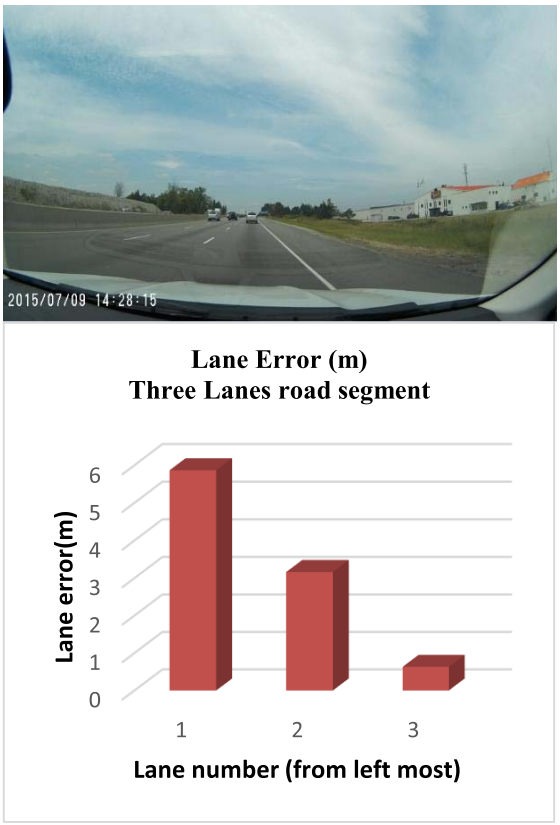


Fig. 22. Lane errors in a 3-lane road.

occurs but GNSS does not correctly follow it while the sensor fusion output keeps the solution on-track on the correct new lane.

### *E. Estimated Lane Errors*

Figure 22, Figure 23, and Figure 24 show example snapshots of the visual inspection software tool developed to evaluate the accuracy of the system. As can be seen in the figures, an image of the road that indicates the correct lane is displayed in the upper graph while the estimated lane information are displayed

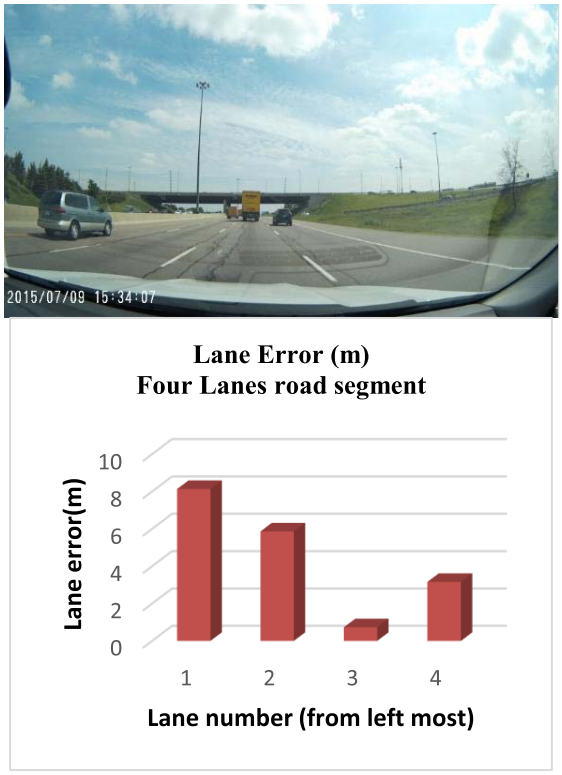


Fig. 23. Lane errors in a 4-lanes road.

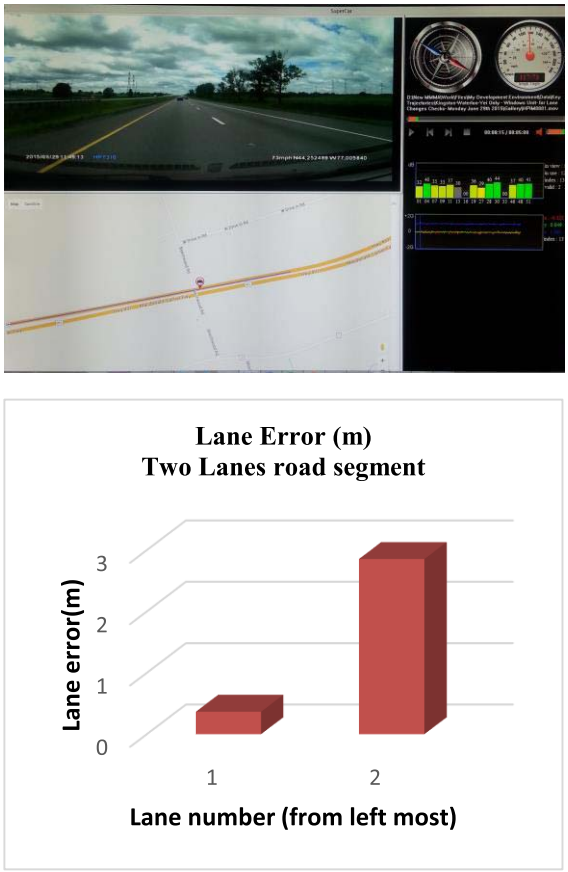


Fig. 24. Lane errors in a 2-lanes road.

along with road information including lane errors in the lower graph. Figure 23 shows that the system is able to identify the correct lane when the number of lanes is increased.

## VI. Conclusions and Future Work

This paper presented a low-cost lane-determination system that fuses MEMS IMU with standard SPS GPS technology and commercially available road network maps that provide unbiased road-center positioning. The system estimates the vehicle's 3D state using an EKF and then it applies an efficient two-phase multi-stage map-matching procedure using HMM and LS. The advantage of the proposed two-stage multi-resolution map matching method is the reduced complexity of HMM decoding step as compared to the case if HMM is applied directly over large number of lanes. This also eliminates the need for explicit storage of lane geo-spatial information. To guarantee a proper convergence, a lane-switch detector based on the vehicle's state and IMU measurements was developed. In contrast to existing and state-of-the-art systems, the proposed system does not need precise GNSS technology such as DGPS or RTK. The system has been tested widely in real-time using an EVK-7P kit GNSS receiver from Ublox and a 3D-space sensor IMU from YOST Labs. Real-road experiments showed a 97.14% success rate. The experiments have been performed mostly in open-sky. Future work will consider more challenging environments such as dense urban areas and the incorporation of more sensors such as vision and range sensors. Future work plans also include comparison analysis with other methods and analysis related to average speed, acceleration, and turns for better assessment of the GPS/IMU algorithm performance, and its impact on the lane estimation.

## References


[1] D. Ramaswamy, J. V. Medanic, W. R. Perkins, and R. F. Benekohal, "Lane assignment on automated highway systems," *IEEE Trans. Veh. Technol.*, vol. 46, no. 3, pp. 755–769, Aug. 1997.

[2] C. Kreucher and S. Lakshmanan, "LANA: A lane extraction algorithm that uses frequency domain features," *IEEE Trans. Robot. Autom.*, vol. 15, no. 2, pp. 343–350, Apr. 1999.

[3] M. Bertozzi and A. Broggi, "GOLD: A parallel real-time stereo vision system for generic obstacle and lane detection," *IEEE Trans. Image Process.*, vol. 7, no. 1, pp. 62–81, Jan. 1998.

[4] P. Misra and P. Enge, *Global Positioning System: Signals, Measurements, and Performance*. Lincoln, MA, USA: Ganga-Jamuna Press, 2011.

[5] J. Farrell, *Aided Navigation: GPS With High Rate Sensors*, New York, NY, USA: McGraw-Hill, 2008.

[6] W. J. Hughes Tech. Ctr., "Global positioning system (GPS), standard positioning service (SPS)," WAAS T&E Team, Atlantic City Int. Airport, NJ, USA, NSTB/WAAS T&E Team, Perform. Anal. Rep. #86, Jul. 2014.

[7] *U-Blox 7 GNSS Modules Datasheet*, document UBX-13003830, R07, U-blox, Thalwil, Switzerland, Nov. 2014.

[8] A. Joshi and M. R. James, "Generation of accurate lane-level maps from coarse prior maps and lidar," *IEEE Intell. Transp. Syst. Mag.*, vol. 7, no. 1, pp. 19–29, Spring 2015.

[9] R. Toledo-Moreo, D. Betaille, and F. Peyret, "Lane-level integrity provision for navigation and map matching with GNSS, dead reckoning, and enhanced maps," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 1, pp. 100–111, Mar. 2010.

[10] P. D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Norwood, MA, USA: Artech House, 2013.

[11] R. J. Elliott, L. Aggoun, and J. B. Moore, *Hidden Markov Models: Estimation and Control*. Springer Science+Business Media, LLC, 1995.

[12] L. Li, M. Quddus, and L. Zhao, "High accuracy tightly-coupled integrity monitoring algorithm for map-matching," *Transp. Res. C, Emerg. Technol.*, no. 36, pp. 13–16, Nov. 2013.

[13] S. Haibin, T. Jiansheng, and H. Chaozhen, "A integrated map matching algorithm based on fuzzy theory for vehicle navigation system," in *Proc. Int. Conf. Comput. Intell. Secur.*, Guangzhou, China, 2006, pp. 916–919.

[14] Q. Wu, X. Gu, J. Luo, P. Zhang, and X. Fang, "A vehicle map-matching algorithm based on measure fuzzy sorting," *J. Comput.*, vol. 9, no. 5, pp. 1058–1065, 2014.

[15] M. Hashemi and H. A. Karimi, "A critical review of real-time map-matching algorithms: Current issues and future directions," *Comput., Environ. Urban Syst.*, vol. 48, pp. 153–165, Nov. 2014.

[16] H.-J. Chu, G.-J. Tsai, K.-W. Chiang, and T.-T. Duong, "GPS/MEMS INS data fusion and map matching in urban areas," *Sensors*, vol. 13, no. 9, pp. 11280–11288, 2013.

[17] *HERE*, Accessed Jan. 19, 2016. [Online]. Available: https://company.here.com/here/

[18] S. Jung, J. Youn, and S. Sull, "Efficient lane detection based on spatiotemporal images," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 1, pp. 289–295, Jan. 2016.

[19] R. Toledo-Moreo, D. Betaille, F. Peyret, and J. Laneurit, "Fusing GNSS, dead-reckoning, and enhanced maps for road vehicle lane-level navigation," *IEEE J. Sel. Topics Signal Process.*, vol. 3, no. 5, pp. 798–809, Oct. 2009.

[20] J. Du and M. J. Barth, "Next-generation automated vehicle location systems: Positioning at the lane level," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 1, pp. 8–57, Mar. 2008.

[21] H. Aly, A. Basalamah, and M. Youssef, "LaneQuest: An accurate and energy-efficient lane detection system," in *Proc. IEEE PerCom*, Mar. 2015, pp. 163–171.

[22] A. Vu, A. Ramanandan, A. Chen, J. A. Farrell, and M. Barth, "Real-time computer vision/DGPS-aided inertial navigation system for lane-level vehicle navigation," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 2, pp. 899–913, Jun. 2012.

[23] P. G. Savage, *Strapdown Analytics*, 2nd ed. Maple Plain, MN, USA: Strapdown Associates, 2007.

[24] L. K. Balivada and K. P. Raju, *Optimization Techniques of Viterbi Algorithm: Performance Analysis of Different Algorithms*. LAP LAMBERT Academic Publishing, May 2012, pp. 60.


**Mohamed Maher Atia** received the B.S. and M.Sc. degrees in computer systems from Ain Shams University in 2000 and 2006, respectively, and the Ph.D. degree in electrical and computer engineering from Queen's University, Kingston, in 2013. He is an Assistant Professor with the Department of Electronics, Carleton University.

**Allaa R. Hilal** received the B.S. and M.Sc. degrees from German University in Cairo and the Ph.D. degree in electrical and computer engineering from University of Waterloo. She is the Director of Innovation and Emerging Technology, Intelligent Mechatronic Systems Inc., a world-renowned expert in the connected car technology.

**Clive Stellings** is a Senior SW Developer with Intelligent Mechatronic Systems Inc. He is an Expert in computer systems development and programming. He develops code and systems to realize new concepts and build testing environments to assist in improving software development frameworks.

**Eric Hartwell** received the B.S. and M.Sc. degrees in aerospace engineering from University of Toronto in 1976 and 1979, respectively. He is experienced in systems development using the wide spectrum of technologies. He is currently a Senior Software Developer with Intelligent Mechatronic Systems Inc.

**Jason Toonstra** received the B.S. and M.Sc. degrees from University of Manitoba in 1995 and 1997, respectively. He was with NORTEL as an Embedded Software Engineer and with Ciena/Catena Networks as a Lead Engineer. He is currently a Senior Director of Product Development with Intelligent Mechatronic Systems Inc.

**William B. Miners** received the M.Sc. degree from University of Guelph and the Ph.D. degree from University of Waterloo in 2002 and 2006, respectively. He is currently the VP Innovation with Intelligent Mechatronic Systems Inc. He transforms emerging technologies and research into practical solutions that help make vehicles safer, smarter, and greener.

**Otman A. Basir** received the M.Sc. degree in electrical engineering from Queen's University, Kingston, and the Ph.D. degree in systems design engineering from University of Waterloo, Waterloo, Canada, in 1989 and 1993, respectively. He is a Professor with the Department of Electrical and Computer Engineering, University of Waterloo.