Terrain Based GPS Independent Lane-Level Vehicle Localization using Particle Filter and Dead Reckoning

Hamad Ahmed* and Muhammad Tahir†
Department of Electrical Engineering, Lahore University of Management Sciences, Lahore, Pakistan 54890
Email: {hamad.ahmed*, tahir†}@lums.edu.pk

***Abstract*—The need of accurate and reliable positioning in various location-aware safety critical transportation applications is increasing day by day. The Global Positioning System (GPS) is not able to provide lane-level vehicle localization without the aid of differential corrections. It also suffers from signal outages in urban areas resulting in a complete loss of location information. Therefore, GPS independent localization methods are now being developed. In this domain, inertial sensors along with a terrain map have been successfully deployed to achieve sub-meter level accuracy in the longitudinal direction of the vehicle in an urban environment. However, lateral localization of the vehicle with good accuracy and computational efficiency remains a challenging topic. Existing algorithms are computationally intensive, and do not provide location information during the process of lane change by the vehicle. This information is very crucial as the risk of potential conflict with nearby vehicles is higher during lane changes. In this paper, we present a computationally efficient method for achieving lane-level localization in a multi-lane scenario by combining the particle filter with dead-reckoning. The particle filter provides the location information about a single lane while location information during the lane change maneuvers is provided by dead-reckoning. Lane-change maneuvers are detected by constantly observing the yaw rate of the vehicle. Developing a computationally efficient algorithm enables the GPS independent localization algorithm to be run on low cost micro-controllers making its deployment feasible for packaged devices. Experiments performed on an instrumented vehicle show the superiority of the proposed algorithm on the existing ones.**

I. Introduction

Numerous driving related applications developed in the past couple of years have been focused on making driving as smart, safe and automated as possible. These developed solutions have reduced the risk of accidents through lane keeping and active steering systems, increased road throughput by fleet management & lane allocation, and caused lots of convenience to the drivers by automated toll collection and ticketing systems. Accurate and reliable vehicle location information is a key element of all these applications. The accuracy requirement on the vehicle location information has also increased with the increasing sophistication level of these applications. Now, lane-level localization is a must requirement for all the applications mentioned above.

The Global Positioning System (GPS) has been the standard for providing vehicle localization for decades. The visibility of GPS satellites in urban environments is very poor due to urban canyons and tunnels which degrade the quality of location information from GPS. If unobstructed GPS signal is somehow available, differential corrections provided by Differential-GPS (DGPS) are required to increase the accuracy to lane-level. Unfortunately, DGPS subscription is only available in a very few countries, and is costly. Therefore the efforts have been focused towards developing solutions that are either independent of GPS or can overcome the visibility and low accuracy issues of GPS for lane-level positioning.

Camera and LIDAR based solutions have been extensively proposed and studied in the literature [1] [2]. They rely on detecting the lane markings by image processing to estimate the lane of travel. However the camera suffers from visibility problem in poor light conditions and the LIDAR is inaccurate due to road reflections. Further, both of these sensors are costly and require hefty image/signal processing to extract lane information which increases the on-board processing requirements. A simple and cost-effective solution is to integrate an Inertial Measurement Unit (IMU) with GPS receiver for aiding [3]. GPS/IMU integration not only increases the accuracy but also provides location during signal outages. However, the IMU based location begins to drift if the GPS outage lasts for a long time, as the case with tunnels.

To resolve these predicaments, recent research efforts have developed solutions which provide location information completely independent of GPS. In this category, the use of terrain map of a road segment (road elevation and bank angles) along with an IMU sensor (gyroscope & accelerometer) inside a Particle Filter have been successfully demonstrated for achieving the accuracy at decimeter levels [4]. However, in this work, there has been an inherent assumption that the vehicle never changes its lane of motion. For lateral localization and lane change estimation two approaches, namely Bayesian Belief (BB) algorithm and Lateral Particle Filter (LPF), were proposed in a later work [5]. Both of these approaches increase the overall computations of the system thus prohibiting it to function on low end microprocessors. Another problem with these approaches is that they only provide location while the vehicle is traveling in a single lane. They do not provide location during the lane change. As one can observe that the event of lane changing is the most crucial during which the risk of potential collisions is the highest. Hence, a location estimate during this event is utmost desired for safety critical applications. Also, the work described in [4], [5] assumes the use of highly accurate IMU sensors which are quite expensive.

We have been working on enabling the IMU based terrain aided localization with low cost MEMS-IMU sensors suited for low cost processors which will enable a low cost finished device for commercial use on large scale. Recently, we proposed a novel Kalman filter to extract the accurate roll angle of the vehicle which is subsequently used inside a Particle Filter for longitudinal vehicle localization. The proposed solution achieves good accuracy using low cost MEMS-IMU sensors [6]. In this work, we extend our previous study to develop a new algorithm which estimates the lane of travel without using any sensor other than the IMU and without increasing the computational complexity of the algorithm so that it is still able to function on a low cost microprocessor. We monitor the yaw rate of the vehicle to detect the lane change maneuver. The algorithm is only triggered in the events of lane change, thus keeping the computational cost low. We also introduce dead reckoning into our algorithm so that it also provides solution during the maneuvers of lane changing. Hence we achieve lateral localization along with longitudinal localization to complete the solution.

The rest of this paper is organized as follows. Section II briefly describes the single lane longitudinal localization algorithm using

Particle filter algorithm. Section III presents the proposed approach for lane-level localization in the presence of lane change maneuvers. Section IV describes the experiments that were performed to test, validate and compare the algorithm. Finally, section V concludes the work based on the discussion carried out in previous section.

II. Single Lane Longitudinal Localization

Before describing the proposed algorithm, an explanation of the longitudinal localization algorithm is in order. The road terrain consists of both road elevation and bank angles. The elevation angle affects the vehicle pitch and the bank angle affects the vehicle roll. In order to localize a vehicle using road terrain, information regarding any one of these angles is required. As we have shown in our previous work, obtaining the roll angle from low cost sensors is computationally efficient as compared to pitch angle. Hence we used only the roll angle in our localization algorithm [6]. The given terrain map of a single road segment consists of longitude and latitude values of the road with the bank (or roll) angle of the road at that location. This readings are spatially spread at every $0.1m$ distance of the road segment. This map is generated using extremely accurate IMU sensors and DGPS so that they can be considered as a ground truth and serve as an accurate localization reference.

The in-vehicle roll sensor which is a low cost MEMS IMU, provides the roll angle after fusing the measurements from a gyroscope and accelerometer through a Kalman filter as described in [6]. For matching this roll angle with the already available terrain map, we used the particle filter. Note that at the start of the drive, the vehicle can be present anywhere along the map hence the probability distribution describing the initial location of vehicle is uniform (non Gaussian). Any probabilistic technique that assumes a Gaussian distribution cannot be used here. Particle filters are sequential Monte-Carlo methods with the ability to work on non-Gaussian densities. The particle filter used here is the third algorithm described in [7]. Following are main steps of this algorithm.

A. Drawing the particles

At the start of the algorithm, N particles are distributed uniformly across the entire map due to uniform distribution at the beginning of the drive. Each particle $i = 1...N$ has a location X, a roll angle value $\phi_{p,i}$ and a weight q_i. The weight of a particle is an indicator of the match between its roll angle value and the value measured by the in-vehicle sensor. The range of this weight is from 0 to 1. At the beginning, all particles are assigned an equal weight $1/N$.

B. State Update

The odometery of the vehicle is also available through the CAN bus and is used to update the location of each particle at every time instant k. The location of each particle is projected forward according to the speed of the vehicle v from the odometer reading. A variance equal to the variance of the odometery is added to each particle's position.

$$X_k = X_{k-1} + dX + dO \tag{1}$$

where $dX = \Delta t \times v$ is the forward projected distance, Δt is the sampling time and dO is the added variance.

C. Measurement Update

For each particle i, its roll angle $\phi_{p,i}$ is compared with the roll measured by the IMU sensor ϕ_a and the particle's weight q_i is calculated using

$$q_i = \frac{exp(-\frac{1}{2R}.(\phi_a - \phi_{p,i})^2)}{\Sigma_{j=1}^{N}(exp(-\frac{1}{2R}.(\phi_a - \phi_{p,j})^2))} \tag{2}$$

where R is the variance of the roll measured by the IMU sensor. The particles with roll similar to that measured by the sensor will get a higher weight from this function.

D. Resampling

At each step, the number of effective particles is calculated by

$$N_{eff} = \frac{1}{\Sigma_{i=1}^{N}(q_i)^2} \tag{3}$$

When number of effective particles are below a defined threshold N_T, the particles are re-sampled according to the re-sampling algorithm given in [7]. The resampling algorithm is presented below in Algorithm 1. The particles with roll very different from the sensor measurements get a lower weight and the particle population eventually becomes less effective. The resampling step draws the particles again from the cumulative density resulting in more particles being drawn from the neighborhood of the higher weighted particles. This causes the filter to converge when the variance of all the particle's location becomes less than a threshold. The mean location of the particles is the location of the vehicle. So, given that the vehicle travels in the same lane on the road segment, the above algorithm not only finds its location in the mapped segment but also continues to track it. The complexity of the problem however, increases if lane changing is taken into account because each lane will have its own terrain map and the particle filter will need to operate on the correct map for localizing and tracking the vehicle. Therefore whenever the vehicle changes its lane, the correct terrain map should be switched automatically.

Algorithm 1. Algorithm for Particle Resampling

$c = cumsum(q_i)$
$u_1 = rand(1).N^{-1}$
$i = 1$
for $j = 1...N$ **do**
  $u_j = u_1 + (j-1).N^{-1}$
  while $u_j > c_i$ **do**
    $i = i + 1$
  end while
  $X_j = X_i$
  $q_j = N^{-1}$
end for

III. Proposed Methodology

The proposed approach extends the Particle Filter based single lane localization as described in the previous section to multi-lane localization by incorporating a lane change detection and dead-reckoning modules. Suppose the vehicle is traveling on a road with two lanes L_1 and L_2. At the start of the drive, the Particle Filter algorithm essentially has two maps to localize the vehicle as the vehicle could be present anywhere on either lanes. Hence the total number of particles are spread over both the lanes. Eventually the filter converges to the right lane providing longitudinal position of the vehicle.

Now, given that vehicle has been initially localized by the Particle Filter in the correct lane, Fig. 1 shows the next steps of the proposed approach which are described below.

Lane change detection : When the vehicle is changing lanes, the gyroscope aligned with the yaw axis of the vehicle measures the yaw rate and hence it can be used as an indication that the vehicle is in the process of changing lanes. The yaw rate observed during a typical

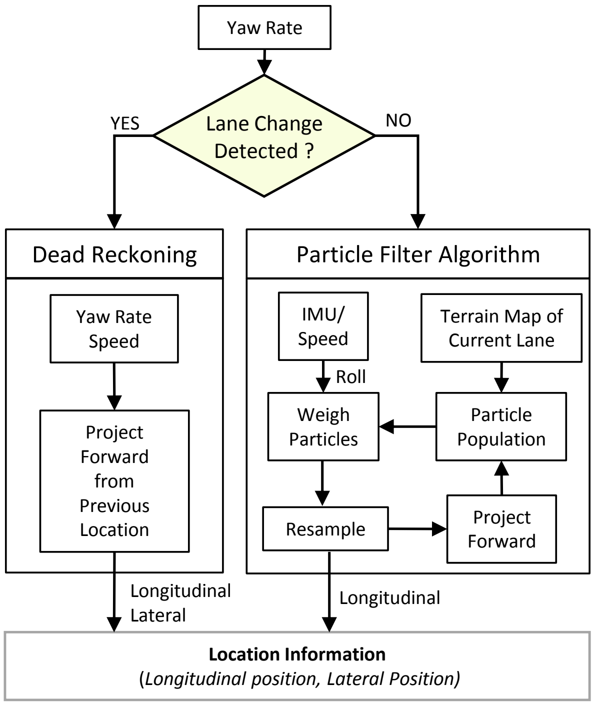


Fig. 1. Block diagram describing the proposed approach

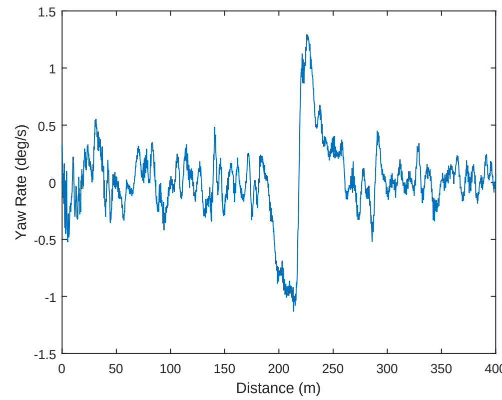
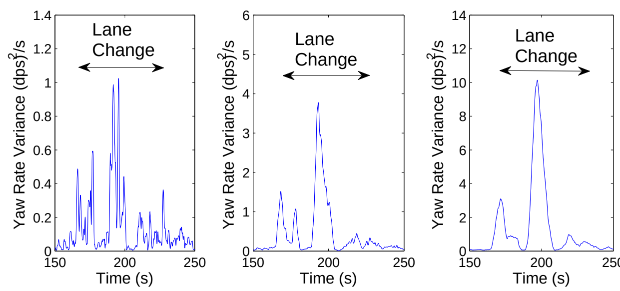


Fig. 2. Yaw rate during a typical lane change maneuver



(a) (b) (c)

Fig. 3. Variances of vehicle yaw rate over different window sizes (a) 0.2s (b) 0.5s (c) 1s

lane change is shown in Fig. 2. It is evident that the yaw rate exhibits a specific behavior due to lane changing. By taking the variance of the gyroscope measurements over a pre-defined window length M, we can get an indication whenever the car changes lanes. The choice of window length M is interesting. There is a trade-off between accuracy and the delay to lane change detection which depends on the window length M . If a large window is chosen, there will be a sufficient delay in the detection process. If a small window is chosen, then the signal may remain buried in the noise floor. We collected data for a lane change and recorded the variance of yaw rate over different window sizes. Fig. 3a indicates that the window size of 0.1s is too small to detect the lane change. On increasing it to 0.5s, there is a visible peak in the variance (Fig. 3b) which can be used as an indication that the vehicle is changing its lane. Further increase in the window sizes will only increase the delay in the detection process. The optimal value of threshold to trigger the lane change detection can be set via a machine learning classifier which can be trained to separate the two cases of driving i.e. straight lane drive and lane changing maneuver. However for simplicity, here we set this value manually by observing the variance over several lane change drives.

Lane change aid to localization module : Suppose the vehicle was initially occupying L_1. The particle filter will be tracking the vehicle in L_1 using the terrain map of L_1. When the yaw rate monitor senses a lane change, it notifies the localization module which switches to dead reckoning instead of Particle Filter tracking. This is due to the reason that the vehicle is no longer in L_1 and the particle filter will continue to track it in L_1 causing lateral location error.

Dead-Reckoning during lane change : In dead reckoning, the x and y coordinates of the vehicle at time t are updated from the coordinates at time $t-1$ by using the speed v and yaw rate $\dot{\psi}_t$ by the equation

$$\begin{bmatrix} x \\ y \end{bmatrix}_t = \begin{bmatrix} x \\ y \end{bmatrix}_{t-1} + \Delta t \left(\begin{bmatrix} cos(\psi_{t-1} + \Delta t.\dot{\psi}_t \\ sin(\psi_{t-1} + \Delta t.\dot{\psi}_t \end{bmatrix} .v \right) \tag{4}$$

It is evident that the accuracy of this location estimate will only hold for a short duration and the error will grow because the integration process will accumulate any noise present in the sensors. However, lane changing is a short maneuver lasting for a few seconds, in which the accuracy of the dead reckoning does not deteriorate. Hence we get a location estimate even during the process of lane changing.

Particle Filter after lane change : When the variance of yaw rate goes below the threshold, the vehicle is assumed to have shifted to the desired lane and is not maneuvering anymore. Now the terrain map for the appropriate lane should be used for further tracking the vehicle. We choose the terrain map of that lane, in which the dead reckoning placed the vehicle. The particle filter starts tracking the vehicle on this new map.

IV. Results and Discussion

A. Experiments

STMicroelectronics 3-axis MEMS accelerometer 'LSM303DHLC' and gyroscope 'L3GD20' were mounted on the test vehicle. Experiment was performed in the vicinity of the Lahore University of Management Sciences. The terrain map of both the lanes of the road was available, and is shown in Fig. 5. The vehicle was driven for some time in the first lane and then shifted to the second lane. The roll measured by the vehicle is also shown in Fig. 5 for the test drive. It can be seen that the vehicle travels in the first lane for about $160m$ and then starts to shift lanes. At about $230m$, the vehicle has entered lane 2 and the measured roll angle now resembles the terrain map of the second lane.

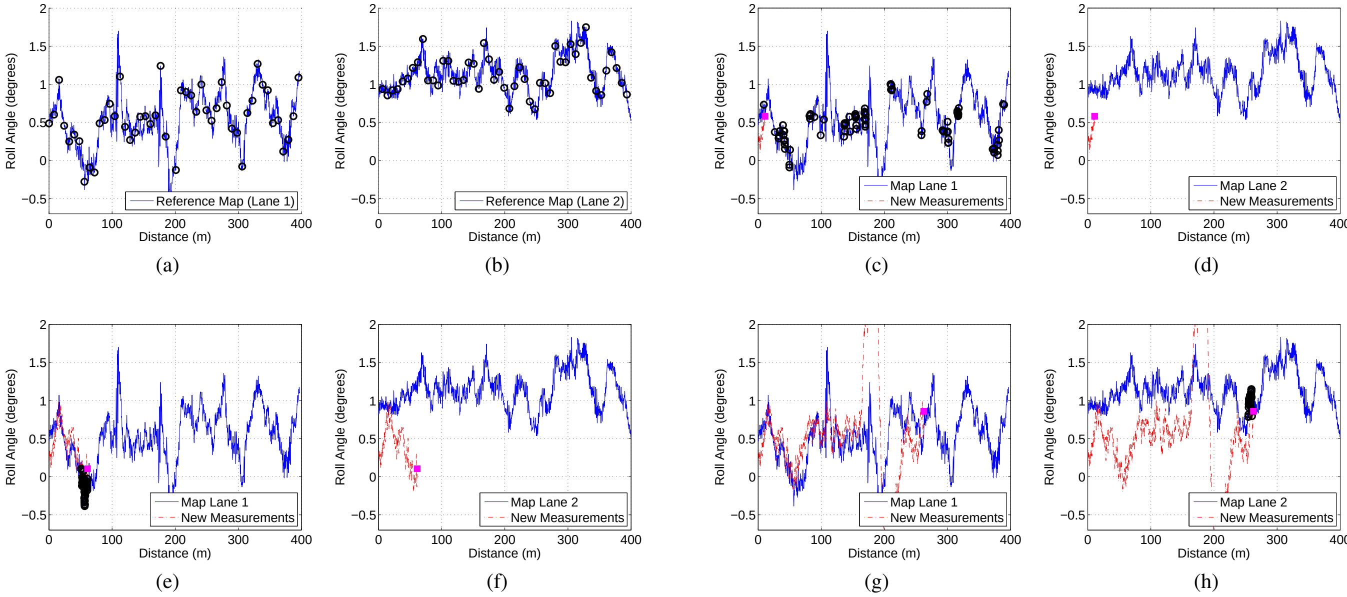


Fig. 4. Particle distribution over the two lanes for different distances traveled (a)-(b) Start (c)-(d) 10m (e)-(f) 50m (g)-(h) 250m

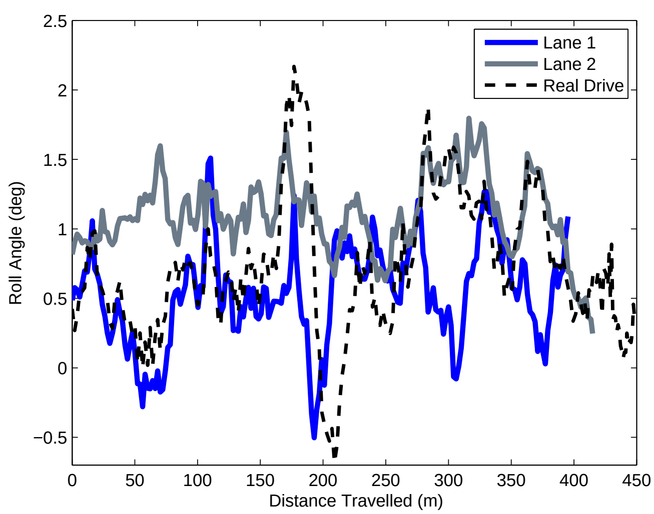


Fig. 5. Terrain map of both lanes and the roll measured in a test drive

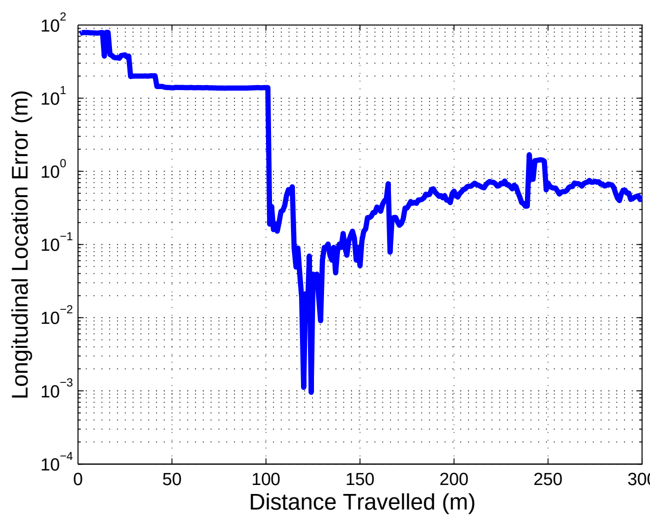


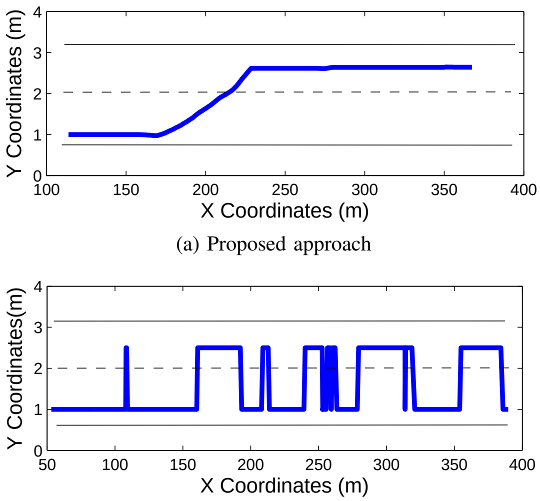
Fig. 6. Error in position estimate with distance traveled

B. *Vehicle localization using the proposed approach*

For the proposed approach, the particle filter was implemented with $N = 100$ particles. The window size for monitoring yaw rate was set to $0.5s$. Half of the particles were spread in lane 1 (Fig. 4a) and half in lane 2 (Fig. 4b) as initially the vehicle can be present in any lane. After only $10m$ of travel, all the particles are converged to lane 1 (Fig. 4c-4d) which indicates that the vehicle has been localized in lane 1. The longitudinal localization was achieved after $50m$ of travel (Fig. 4e-4f) where the error was within $1m$ which can be observed from Fig. 6. At $165m$, the vehicle initiates a lane change. The yaw rate monitors detects it at 168 meters. At this point, the dead reckoning takes over and keeps tracking the vehicle until the variance goes below the threshold. The new lane map is decided based on the lane in which the vehicle was localized by the dead reckoning solution. As shown in Fig. 4g-4h, the particles now have shifted to lane 2 because the dead reckoning solution localized the vehicle in lane 2 as shown in Fig. 7a. Fig. 6 indicates that the proposed approach was able to achieve sub-meter level accuracy during the complete drive of the vehicle.

C. *Comparison with the BB algorithm*

To compare the proposed techniques, we also implemented BB algorithm as well from [5]. Other algorithm described in [5] i.e. LPF algorithm is infeasible for a low cost packaged device as it requires an extra sensor to measure the vehicle yaw. Hence, it was not considered for comparison. The BB algorithm terms the vehicle occupying any lane as a 'state'. An assumption on the likelihood of changing lanes is assumed, and the beliefs of each state are propagated at periodic time intervals. These beliefs are then updated in next step using the



(a) Proposed approach

(b) BB algorithm

Fig. 7. Vehicle localization during the test drive

measured roll value. The state with higher belief is termed as the lane of travel of the vehicle.

Fig. 7a & 7b show the localization results of the proposed approach and BB algorithm during the same test drive. In the case of BB algorithm, it can be seen that the location information during the lane changing is meaningless which indicates the inability of the algorithm to localize during lane change maneuvers. another problem with the BB algorithm is that it estimates incorrect lane indices during certain intervals e.g. between $260 - 290m$ and $320 - 350m$. The reason is that the terrain map of both lanes is quite similar in these regions (clear from Fig. 5) which confuses the BB algorithm. Also, the BB algorithm has to make assumptions on lane change likelihood. In reality, this assumption is very hard to make and calculate. The proposed approach does not make any assumption of this kind.

D. Computational complexity comparison

The BB algorithm propagates and updates the state beliefs at periodic time intervals in addition to running the expensive Particle Filter. Also, the number of beliefs propagated and updated at each time instance is equal to the number of lanes. Hence the computational complexity increases with the number of lanes. In the proposed method, only the particle filter is running, and an accompanying yaw rate monitor which is simply a variance estimator. The complexity of the proposed approach does not scale with the increase in number of lanes. Hence, the proposed approach is computationally efficient as compared to the BB algorithm.

V. Conclusion

In this work, we have extended the terrain based localization algorithm using low cost IMU sensors to detect lane changes and localize the vehicle in the lateral direction as well, thus completing the localization solution. The proposed method also provides location information during lane change maneuvers which were missing in previous algorithms. It is also computationally efficient making it feasible for implementation on low cost microprocessors. The method was experimentally verified and compared with an existing algorithm in the literature. Future works aims to extend this work for incorporation of curved and geometrically complex road networks.

References

[1] Amol Borkar, Monson Hayes, and Mark T Smith. A novel lane detection system with efficient ground truth generation. *Intelligent Transportation Systems, IEEE Transactions on*, 13(1):365–374, 2012.

[2] Christopher Rose, Jordan Britt, John Allen, and David Bevly. An integrated vehicle navigation system utilizing lane-detection and lateral position estimation systems in difficult environments for gps. *Intelligent Transportation Systems, IEEE Transactions on*, 15(6):2615–2629, 2014.

[3] Rafael Toledo Moreo, B Úbeda, A Skarmeta, Miguel A Zamora Izquierdo, et al. High integrity imm-ekf based road vehicle navigation with low cost gps/ins. 2007.

[4] A Dean, R Martini, and S Brennan. Terrain-based road vehicle localisation using particle filters. *Vehicle System Dynamics*, 49(8):1209–1223, 2011.

[5] Adam J Dean and Sean N Brennan. Terrain-based road vehicle localization on multi-lane highways. In *American Control Conference, 2009. ACC'09.*, pages 707–712. IEEE, 2009.

[6] Hamad Ahmed and M. Tahir. Terrain-based vehicle localization using low cost mems imu sensors. In *83rd Vehicular Technology Conference*. IEEE, 2016.

[7] M Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *Signal Processing, IEEE Transactions on*, 50(2):174–188, 2002.