2008 IEEE Intelligent Vehicles Symposium
Eindhoven University of Technology
Eindhoven, The Netherlands, June 4-6, 2008

Realtime Experiments in Markov-Based Lane Position Estimation Using Wireless Ad-Hoc Network

Thanh-Son Dao, Luke Ng, Christopher Michael Clark and Jan Paul Huissoon

Abstract— **This paper presents real-time experimental results for a new lane positioning system using Markov localization based on inter-vehicle communication. The proposed system uses low-cost GPS receivers to provide vehicle locations. The system also combines a low-pass Butterworth filter and a particle filter for GPS receiver noise rejection. To study the new lane positioning system, a multi-threaded program in C++, that enables the communication between vehicles and determines their lane positions in real-time, was developed. Experiments using this software validate the effectiveness of the lane positioning system.**

I. INTRODUCTION

INTELLIGENT transportation systems are being developed in many different countries with the objective of improving road traffic efficiency and safety. Many systems have been proposed and a large number of transportation research applications would benefit from lane-level positioning capability. For example, such a capability could be used for a lane-level navigation system which advises the driver as to which lane should be chosen to reach the specified destination without requiring excessive last-minute lane changing. Another approach is to use probe vehicles that measure lane-specific traffic conditions on a freeway. This is particularly important when understanding the efficiency of weaving road sections that may unnecessarily cause recurrent congestion [1].

Due to the importance of such applications, much effort has been put in lane finding/positioning and commercially available lane finding/detection systems are already appearing. Ieng et al. [2] dealt with multi-lane detection using multiple cameras. Pierre-Yves and Jeff [3] used lane-level navigation systems with a high level DGPS/DR sensor integration system and a map database. Their system was able to detect which lane a car was driving in and when a car was changing lanes. Other research directions can be found in [4][5].

These systems either required image processing algorithms and/or costly equipment such as highly accurate sensors, high performance computers, etc. With that in mind, we approach the problem by using Inter-Vehicle Communication (IVC) and low-cost GPS to determine lane positions. Besides the cost effectiveness, this approach can also increase the applicability of the system in areas where the infrastructure for DGPS is not available for systems as described in [3]. IVC and co-operative driving systems have been under development for some time [6][7]. With the availability of GPS systems, it is practical to locate a vehicle with certain accuracy. However, GPS data do not provide the exact positions of vehicles (sometimes even placing the vehicle beside the road) due to degradation or multi-path problems. Therefore, it is challenging to determine the exact lane that a vehicle is traveling in, even when a digital map of the road network is available.

This paper builds from our recent research in the area of low-cost lane level position determination [8]. Introduced here is a software architecture that enables real-time implementation of the method proposed in [8]. Also, unlike our previous work, a filtering scheme to reject GPS receiver noise and improve system performance is presented, highlighting details of our Butterworth filter and particle filter implementations. Finally, new real-time experimental results are presented to validate the system. Our fundamental assumption is the existence of a number of vehicles equipped with GPS receivers. Each vehicle has the ability to communicate with other vehicles within a certain radius and is able to send its information regarding position to other vehicles via a wireless ad-hoc network.

This paper is organized as follows: Section II gives a short introduction to the architecture of the lane positioning system. Section III deals with the Markov localization algorithm in detail and its specific application to the lane positioning problem. Section IV proposes a method to reject GPS receiver noise. The experiments are discussed in Section V, and some concluding remarks are provided in Section VI.

II. PROGRAM ARCHITECTURE

To demonstrate the lane positioning algorithm, an in-car localization software was written in C++ which is responsible for reading GPS data from sensors as well as ensures the smooth communication between the vehicles via wireless ad-hoc network and estimation of vehicles' lane positions from sharing information. Each vehicle is equipped with a GPS receiver and a processor to implement the lane positioning algorithm and to communicate with other vehicles. The GPS data for the vehicles is fed into a position filter to reject the measurement noise from receivers. The in-car

---

This work was funded by AUTO21 Canada.

T. S. Dao is a PhD candidate at the Lab for Autonomous and Intelligent Robotics, Dept. of Mechanical Engineering, University of Waterloo, ON, Canada N2L 3G1 `tsdao@engmail.uwaterloo.ca`

L. Ng is a PhD candidate at the Lab for Autonomous and Intelligent Robotics, Dept. of Mechanical Engineering, University of Waterloo, ON, Canada N2L 3G1 `l4ng@engmail.uwaterloo.ca`

C. M. Clark is an Assistant Professor at the Computer Science Department, California Polytechnic State University, San Luis Obispo, CA, USA 93407 `cmclark@calpoly.edu`

J. P. Huissoon is a Professor at the Dept. of Mechanical Engineering, University of Waterloo, ON, Canada N2L 3G1 `jph@mecheng1.uwaterloo.ca`

978-1-4244-2569-3/08/$20.00 ©2008 IEEE.

localization software is organized in a multi-thead program. The functionality for the threads are as follows:

1) *TCP/IP socket server thread:* to send GPS data to other vehicles,
2) *Client threads:* to receive GPS data from other vehicles,
3) *GPS thread:* to read GPS measurements from sensor, i.e., GPS receiver,
4) *Filters thread:* to reject receiver measurement noise from raw GPS data,
5) *Localization thread:* to conduct localization algorithm for lane positions estimation.

The architecture for the system is illustrated in Fig. 1 and a screenshot of the localization software GUI is shown in Fig. 2.

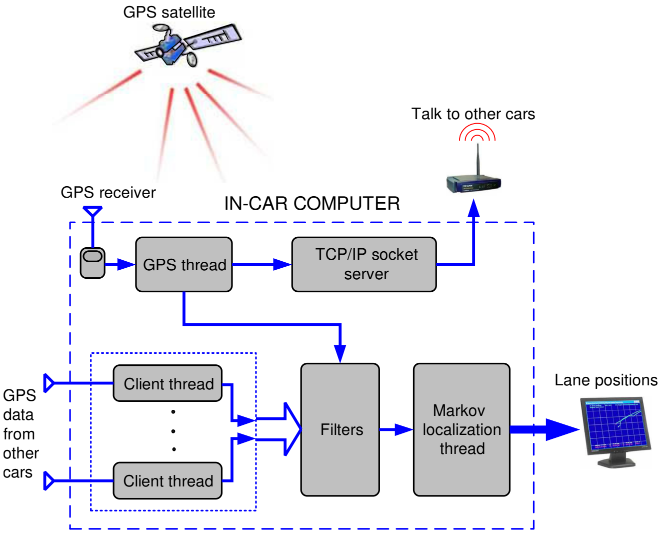
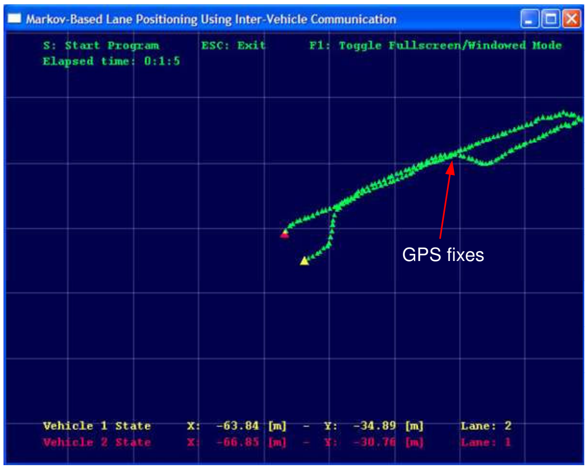


Fig. 1. Program architecture.


Markov-Based Lane Positioning Using Inter-Vehicle Communication
GPS fixes

```
S: Start Program      ESC: Exit      F1: Toggle Fullscreen/Windowed Mode
Elapsed time: 0:1:5

Vehicle 1 State    X:  -63.84 [m]  -  Y:  -34.89 [m]    Lane: 2
Vehicle 2 State    X:  -66.85 [m]  -  Y:  -30.76 [m]    Lane: 1
```

Fig. 2. Screenshot of localization program.

III. MARKOV LOCALIZATION

The target of Markov localization within this work is to estimate lane positions of vehicles traveling on highways from GPS data using IVC. Within the framework of this paper, we only briefly describe how Markov localization is used in the proposed lane positioning system. Further details about Markov localization and associated issues can be found in [8].

Let us start with the simplest case: two vehicles traveling on a two-lane road. The two vehicles are assumed close enough to be able to communicate with each other. Within this Markov localization approach, call $P(v_{1,t} = l_a, v_{2,t} = l_b)$ the probability that vehicle 1 is traveling in lane a and vehicle 2 is in lane b at time t, where a and b are either 1 or 2. There are four possibilities in total, which are $P(v_1 = l_1, v_2 = l_1)$, $P(v_1 = l_1, v_2 = l_2)$, $P(v_1 = l_2, v_2 = l_1)$, and $P(v_1 = l_2, v_2 = l_2)$. Initially, these probabilities are equally set to 0.25. In the course of the vehicles' mission (i.e., for $t > 0$), P is updated through two basic steps: (1) *prediction*, and (2) *correction*.

A. Prediction

In the prediction step, the state of a vehicle is modeled through the conditional probability $P(v_{i,t} = l_a | v_{i,t-1} = l_j)$, which denotes the probability for a motion action that carries vehicle i from lane j to lane a (j and a can be equal). When the vehicle moves, $P(v_{i,t} = l_a | v_{i,t-1} = l_j)$, which models the uncertainty in the vehicle's dynamics, is used to compute the probability distribution at time t as

$$P(v_{i,t} = l_a) \leftarrow \sum_{j=1}^{2} P(v_{i,t} = l_a | v_{i,t-1} = l_j) P(v_{i,t-1} = l_j), \quad (1)$$

where $P(v_i = l_a)$ is the probability that vehicle i is traveling in lane a. This step is repeated for both vehicles and then the product of them is used to form the combined probabilities:

$$P(v_{1,t} = l_a, v_{2,t} = l_b) = P(v_{1,t} = l_a) P(v_{2,t} = l_b). \quad (2)$$

The conditional probabilities $P(v_{i,t} = l_a | v_{i,t-1} = l_j)$ for a vehicle are computed by comparing its current position estimate to the position at the last time step $t-1$. For example, by calculating the shortest distance ΔD from the vehicle position estimate from the GPS reading at time t to the line passing through the two most recent position estimates, the probability at which vehicle 1 switches to lane a (see Fig. 3) can be computed.

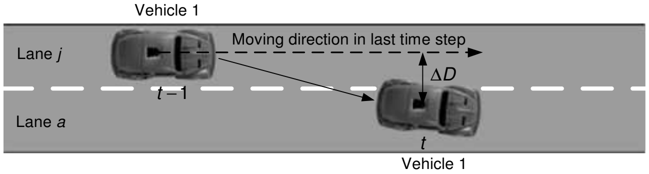


Fig. 3. Prediction step: this figure illustrates a vehicle switching lane and its position is unknown. The shortest distance ΔD from the position estimate at time t to the line passing through the two most recent position estimates is used to calculate the conditional probability.

B. Correction

Denote z as the vehicle position estimates from GPS measurements that come in at time step t for both vehicles, and $P(z|v_1 = l_a, v_2 = l_b)$ as the probability of perceiving z when the two vehicles are in lanes a and b respectively. When the GPS measurements are taken into account, $P(z|v_1 = l_a, v_2 = l_b)$ is used to update the probability distribution at time t according to Bayes' rule

$$P(v_{1,t} = l_a, v_{2,t} = l_b|z_t) \leftarrow \\ \leftarrow \frac{P(z_t|v_{1,t} = l_a, v_{2,t} = l_b)P(v_{1,t} = l_a, v_{2,t} = l_b)}{P(z_t)}, \tag{3}$$

where $P(z_t)$ has the purpose of normalizing the sum of all $P(v_{1,t} = l_a, v_{2,t} = l_b|z_t)$.

At time t, the new measurement z is the perpendicular distance from one vehicle to the other vehicle's direction of motion (see Fig. 4). The conditional probability $P(z|v_1 = l_a, v_2 = l_b)$ can be calculated based on z using the probability density function $\frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(z-\mu)^2}{2\sigma^2}}$. The mean $\mu = w \times (l_a - l_b)$ where w is the lane width. For example, in the case $l_a = 2$ and $l_b = 1$, the conditional probability $P(z|v_1 = l_a, v_2 = l_b)$ is closer to 100% when the measurement z is closer to the lane width w.

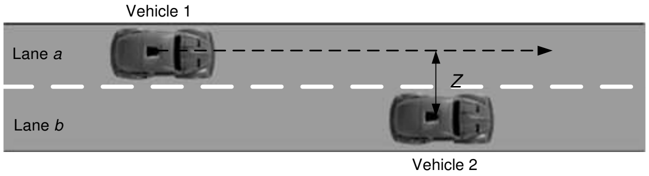


Fig. 4. Correction step: this figure illustrates two vehicles traveling on a highway. The distance z is used to update the belief in prediction step.

The result for two-vehicle-and-two-lane-road can be extended to the general case. Denote the number of vehicles that are communicating with each other as n_v, and and the number of lanes as n_l. Let $P(v_{1,t} = l_a, v_{2,t} = l_b, ..., v_{n_v,t} = l_x)$ be the probability that vehicle 1 is traveling in lane a, vehicle 2 is in lane b, etc., at time t where a, b, ..., x take on values between 1 and n_l. The probability distribution at time t for the prediction step is given by

$$P(v_{i,t} = l_a) \leftarrow \sum_{j=1}^{n_l} P(v_{i,t} = l_a|v_{i,t-1} = l_j)P(v_{i,t-1} = l_j), \tag{4}$$

and

$$P(v_{1,t} = l_a, v_{2,t} = l_b, ..., v_{n_v,t} = l_x) = \\ = P(v_{1,t} = l_a)P(v_{2,t} = l_b)...P(v_{n_v,t} = l_x). \tag{5}$$

Bayes' rule for the correction step is

$$P(v_{1,t} = l_a, v_{2,t} = l_b, ..., v_{n_v,t} = l_x|z_t) \leftarrow \frac{QS}{P(z_t)}. \tag{6}$$

where $Q = P(z_t|v_{1,t} = l_a, v_{2,t} = l_b, ..., v_{n_v,t} = l_x)$ and $S = P(v_{1,t} = l_a, v_{2,t} = l_b, ..., v_{n_v,t} = l_x)$.

The next section discusses the problem of GPS receiver noise elimination to enhance lane estimation results.

IV. GPS RECEIVER NOISE REJECTION

Error from GPS generally comes from several sources including satellite clock, ephemeris error, ionospheric effects, tropospheric effects, and the geometry of visible satellites. This set of errors from GPS will be common for vehicles. Other errors are local to the different receivers and include RF noise from the environment, receiver noise and resolution, multi-path, and receiver clock error. Aside from multi-path, errors from this second set are generally smaller than those common to the receivers. Hence if multi-path is not a problem (as is the case for many open highways), then common errors from receivers that are relatively close to one another can be eliminated by utilizing relative distance between vehicles.

Other errors that need to be eliminated are the errors caused by receiver noise. Receiver noise, which is dependent on the design of antenna, the method used for the analogue to digital conversion, the correlation process, etc., sometimes can falsely indicate a vehicle is switching lanes. The high pitched receiver noise can be rejected using a low-pass filter. In the proposed lane positioning system, a combination of a low-pass Butterworth filter [10] fused with a particle filter [9] can satisfy this task.

A low-pass Butterworth filter is used to smooth the raw data distance measurements between two vehicles. More specifically, let the raw data measurement be z_t^r =distance$(GPS_t^{car_1}, GPS_t^{car_2})$ where $GPS_t^{car_1}$ and $GPS_t^{car_2}$ are raw GPS measurements from two vehicles. The distance z_t^r is then smoothed using a low-pass Butterworth filter producing the output z_t^b. However, Butterworth filter causes time delay in the output due to phase-shift. The main objective of the particle filter is to compensate this delay while preserving the smoothed feature of the output curve. Particle filter, also known as Sequential Monte Carlo methods (SMC), is a sophisticated model estimation technique based on simulation. Particle filter is used to reduce the number of states based on estimating the posterior probability distribution on the state. It does so by "tracking" a variable of interest as the variable evolves over time. In our problem, the variable of interest at time t, which is the relative distance z_t, is represented as a set of M particles $\mathbf{S}_t = (z_t[i], w_t[i]) : i = 1...M$, where the index i denotes the particle. Each particle consists of an estimate of the variable of interest and a weight $w_t[i]$ defining the contribution of this particle to the overall estimate of the variable. Let $P(z_t^r|z_t[i])$ be the probability of perceiving z_t^r given $z_t[i]$. The weight $w_t[i]$ can be calculated by

$$w_t[i] = P(z_t^r|z_t[i]). \tag{7}$$

The calculated weights are then employed to generate a new set of particles for the next time step (re-sampling) using roulette wheel selection.

The particle filtering algorithm can be illustrated by pseudocode in Algorithm 1.

In experiments, the number of particles was chosen to be 500 particles so that the filtering algorithm can work in real-time on Intel® Pentium®-M 1.4GHz processors (computers

Algorithm 1 Particle Filter

```
Require: A set of particles at time 0: $\mathbf{S}_0 = (z_0[i], w_0[i]) : i = 1...M$
while $t > 0$ do
    $z_t^r$ = Distance($GPS_t^{car_1}, GPS_t^{car_2}$)
    $z_t^b$ = Butterworth($z_t^r$) {Smooth raw distance}
    for $i = 1...M$ do
        $z_t[i]$ = Propagate($z_{t-1}[i], z_t^b$) {Prediction with $z_t^b$}
        $w_t[i] = P(z_t^r | z_t[i])$ {Update the weights}
    end for
    for $i = 1...M$ do
        $w_t[i] = \frac{w_t[i]}{\sum_{j=1}^{M} w_t[j]}$ {Normalize the weights}
    end for
    $\mathbf{S}_t = (z_t[i], w_t[i]) : i = 1...M$
    $\mathbf{S}_t$ = Resampling($\mathbf{S}_t$) {Generate new set of particles}
end while
```

used in the experiments). Fig. 5 shows the GPS plot for the two laptops used in an experiment. The raw relative vs. filtered distances between the two computers (see Fig. 6) shows the effectiveness of the filtering scheme. It can be observed that the random walk caused by the GPS receiver noise was rejected effectively.

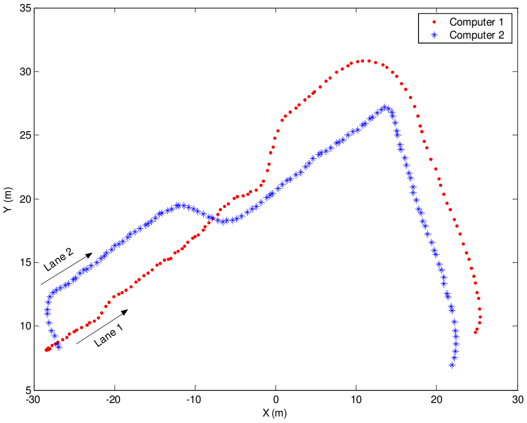


Fig. 5. GPS measurements for two computers.

V. EXPERIMENTS

To further enhance the results from the real-road tests described in [8] where the lane positions of the cars used in the tests were estimated off-line, this section discusses the experiments in real-time at low speeds with the new software (see Section II) and the new filtering scheme.

To evaluate the localization system at low-speed conditions, a number of experiments were conducted on Wilhelm St in Kitchener, ON, Canada with two persons holding two laptops, both are Compaq Evo N620c, equipped with two Garmin 18 GPS receivers. The communication between the laptops was made through the IEEE 802.11b standard D-Link DWL-AG660 Wireless Adapters. The experimental apparatus are shown in Fig. 7. The walking speeds for both persons were estimated from GPS data to be approximately 5km/h.

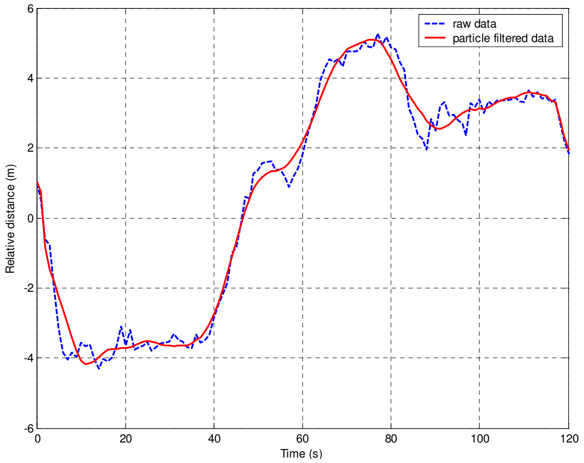


Fig. 6. Relative distance from raw GPS data vs. filtered data.

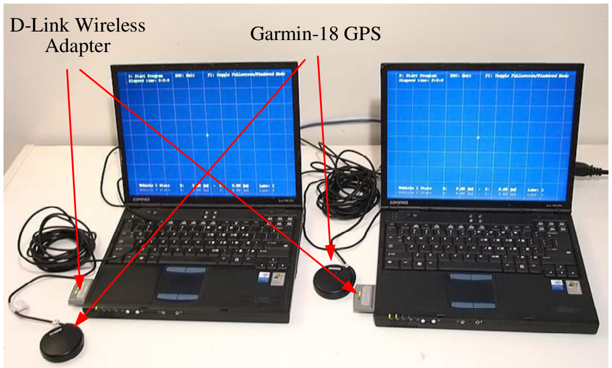


Fig. 7. Experiment apparatus.

The GPS measurements for both computers in the first experiment are plotted in Fig. 5. The strategy was as follows: Initially, both person 1 (computer 1) and person 2 (computer 2) were in lane 1. After 6 seconds, person 2 switched to lane 2. Person 2 moved back to lane 1 after 44 seconds. Person 1 switched to lane 2 at the 63th second and stayed in lane 2 until the end of the test. The resulting probability distributions and estimated lane positions are shown in Fig. 8 and Fig. 9, respectively. The estimation appears to be quite good with the estimated lane positions following the actually values very closely. Fig. 8 shows how confident the system was about its estimation.

In the second experiment (see Fig. 10 for GPS plots) person 1 (computer 1) was initially in lane 1 and person 2 (computer 2) was in lane 2. After 12 seconds, person 1 switched to lane 2. Person 1 moved back to lane 1 after 28 seconds and finally moved to lane 2 after 63 seconds. Person 2 switched to lane 1 at the 45th second and stayed in lane 1 until the end of the test. Fig. 11 and Fig. 12 show the resulting probability distributions and estimated lane positions, respectively. It is interesting to note that the system sometimes could anticipate the lane changing action before the vehicles completely moved to the other lanes as indicated in Fig. 12 at around the 28nd second. This can

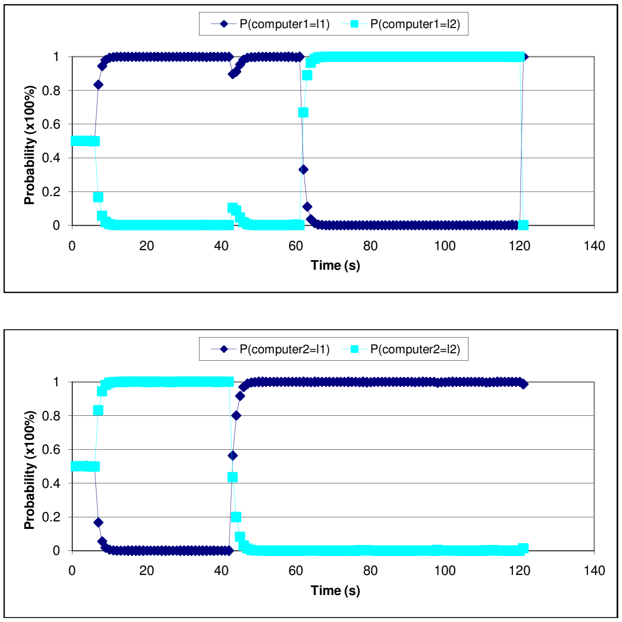


Fig. 8. Probability distributions for experiment 1.

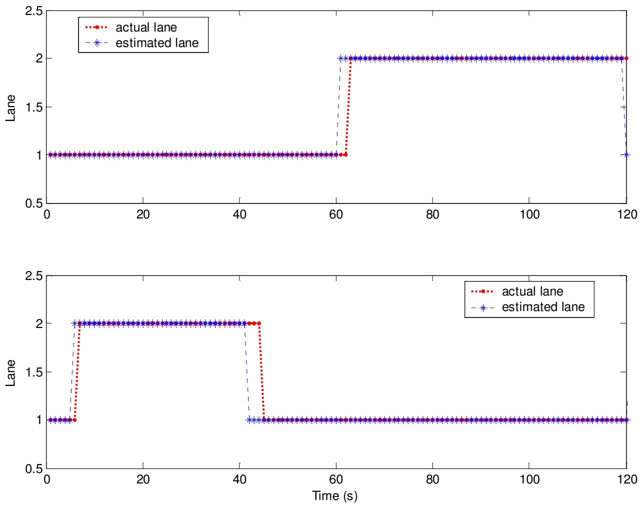


Fig. 9. Estimated vs. actual lane positions for computer 1 (top) and computer 2 (bottom).

be explained by the fact that when a vehicle/person starts making a lane change maneuver, its lateral displacement notifies the prediction step of the localization algorithm that the vehicle is about to make a lane change before the lane changing is completed. In this case, this happened when person 1's made a sharp turn at the 28th second as can be seen in Fig. 10.

The third experiment dealt with a three-lane situation on the same road section. The lane estimation results are given in Fig. 13 and Fig. 14. It can be seen from Fig. 14 that both computers started off with the wrong estimation as the computers initially had no knowledge about the lanes they occupied and the initial lane positions were assumed to be the lowest lane, i.e., lane 1. Fig. 13 shows that the system was not very confident about the estimation in the first 7 seconds of the experiment. However, as soon as person 1 started moving to lane 1 at around the 10th second, the estimated lane positions for both computers quickly converge to the actual values. This experiment indicates that Markov localization algorithm can work well without prior knowledge about the vehicle's initial lane position. It is, therefore, possible to localize the vehicle from scratch and to recover from localization failures or GPS outage in which the positions of the vehicle are lost.

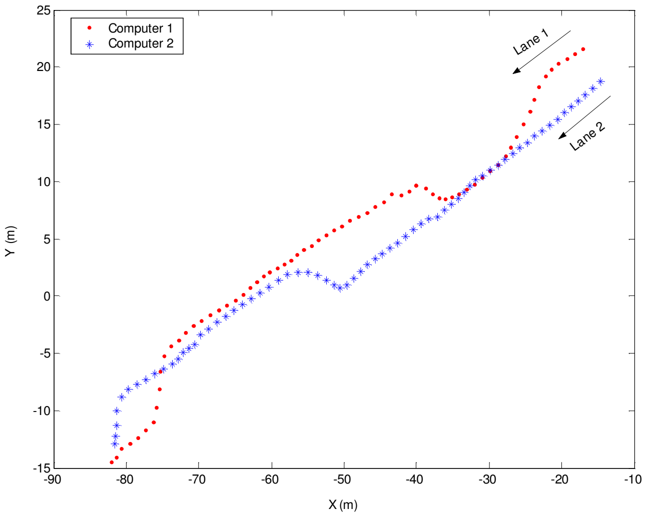


Fig. 10. GPS measurements for experiment 2.

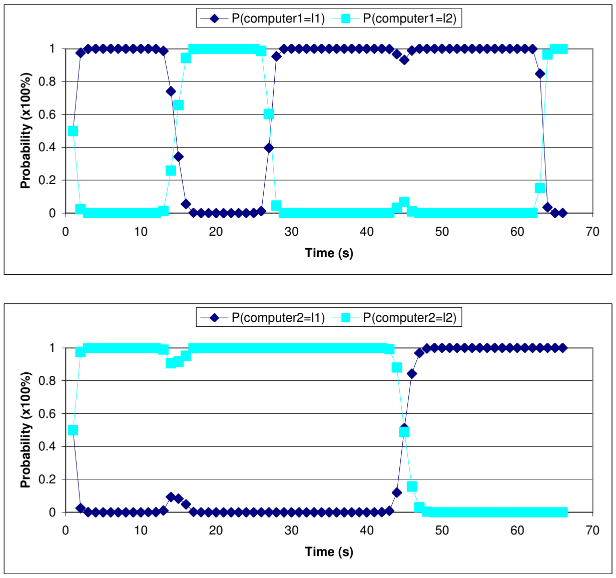


Fig. 11. Probability distributions for experiment 2.

ACKNOWLEDGMENT

Financial support for this research has been provided by AUTO21 Canada.

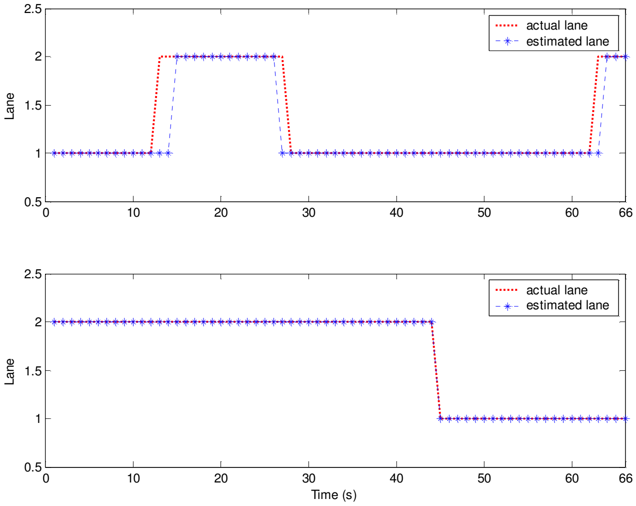


Fig. 12. Estimated vs. actual lane positions for computer 1 (top) and computer 2 (bottom).

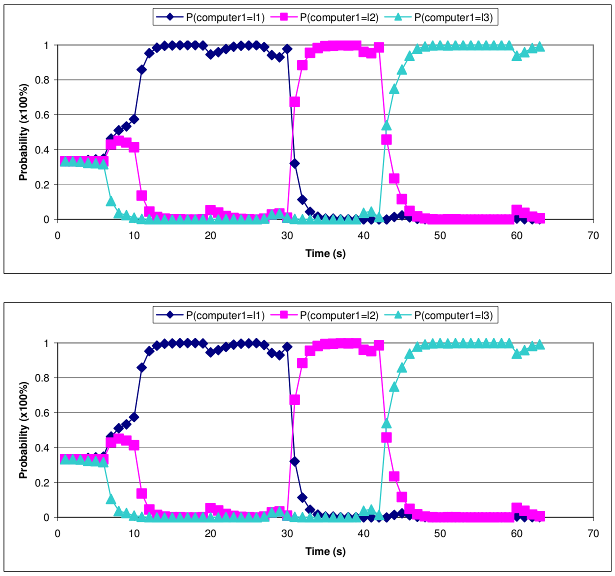


Fig. 13. Probability distributions for experiment 3.

VI. CONCLUSION

The implementation of a new lane position estimation system that uses a Markovian approach based on cooperative driving models has been discussed in this paper.

In comparison to conventional lane positioning methods which usually deal with complicated image processing techniques and/or expensive equipment, the proposed method only requires low cost GPS receivers, IVC, and a simple localization algorithm. Experimental results have shown the efficiency of the algorithm. It is also important to note from the experimental results that in practise low cost GPS receivers can be effectively fused with a filter to obtain low-noise GPS measurements, rather than resorting to the use of more expensive sensors.

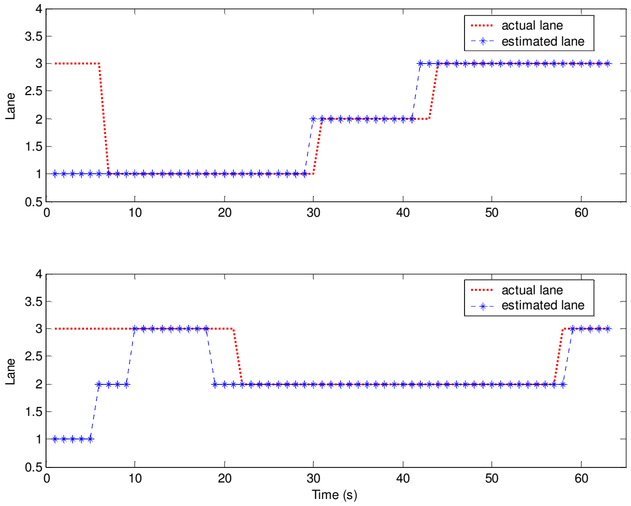


Fig. 14. Estimated vs. actual lane positions for computer 1 (top) and computer 2 (bottom).

The limitation of the proposed strategy lies in the fact that it only uses GPS data to estimate lane positions. This might be challenging where GPS data is not available or GPS signal is blocked completely by large obstacles like in a long tunnel. One possible solution to this problem is to fuse the GPS data with another type of sensor such as an Inertial Measurement Unit (IMU) until GPS data is again available.

REFERENCES

[1] J. Du, J. Masters and M. Barth, ”Lane-level positioning for in-vehicle navigation and automated vehicle location (AVL) systems”, in *Proc. Of IEEE Intelligent Transportation Systems Conference*, Washington, USA, 2004, pp. 35-40.

[2] S. S. Ieng, J. Vrignon, D. Gruyer and D. Aubert, ”A new multi-lane detection using multi-camera for robust vehicle localtion”, in *Proc. Of IEEE Intelligent Vehicles Symposium*, Nevada, USA, 2005, pp. 700-705.

[3] G. Pierre-Yves and K. Jeff, ”Enhanced navigation system for road telematics”, in *Proc. Of 3rd Swiss Transportation Research Conference*, 2003, pp. 1-16.

[4] K. Y. Chiu and S. F. Lin, ”Lane detection using color-based segmentation”, in *Proc. Of the IEEE Intelligent Vehicle Symposium*, 2005, pp. 706-711.

[5] B. D. Stewart, I. Reading, M. S. Thomson, T. D. Binnie, K. W. Dickinson and C. L. Wan, ”Adaptive lane finding in road traffic image analysis”, in *Proc. Of the 7th IEEE Int. Conference on Road Traffic Monitoring and Control*, London, UK, 1994, pp. 133-136.

[6] Z. D. Chen, H. T. Kung and D. Vlah, ”Ad hoc relay wireless networks over moving vehicles on highways”, in *Proc. Of the 2nd ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2001.

[7] W. Enkelmann, ”FleetNet applications for inter-vehicle communication”, in *Proc. Of the IEEE Intelligent Vehicles Symposium*, Ohio, USA, 2003, pp. 162-167.

[8] T. S. Dao, K. Y. K. Leung, C. M. Clark and J. P. Huissoon, ”Markov-based lane positioning using intervehicle communication”, to appear in *IEEE Trans. on Intelligent Transportation Systems*, vol. 8, no. 4, Dec 2007.

[9] I. M. Rekleitis, ”A particle filter tutorial for mobile robot localization TR-CIM-04-02”, Centre for Intelligent Machines, McGill University, Canada, 2003.

[10] B. A. Shenoi, *Introduction to digital signal processing and filter design*, John Wiley & Sons, 2005.