

# On-Bicycle Vehicle Tracking at Traffic Intersections Using Inexpensive Low-Density Lidar

Zhenming Xie and Rajesh Rajamani

**Abstract**— This paper explores the challenges in developing an inexpensive on-bicycle sensing system to track vehicles at a traffic intersection. In particular, opposing traffic with vehicles that can travel straight or turn left are considered. The estimated vehicle trajectories can be used for collision prevention between bicycles and left-turning vehicles. A compact solid-state 2-D low-density Lidar is mounted at the front of a bicycle to obtain distance measurements from vehicles. Vehicle tracking can be achieved by clustering based approaches for assigning measurement points to individual vehicles, introducing a correction term for position measurement refinement, and by exploiting data association and interacting multiple model Kalman filtering approaches for multi-target tracking. The tracking performance of the developed system is evaluated by both simulation and experimental results. Two types of scenarios that involve straight driving and left turning vehicles are considered. Experimental results show that the developed system can successfully track cars in these scenarios accurately in spite of the low measurement density of the sensor.

## I. INTRODUCTION

There were 818 bicyclists deaths within the US in 2015. An additional estimated 45,000 bicyclists were injured in crashes [1]. According to a study of crash data over 10 years in Minneapolis, 41% of bicyclist-motorist crashes happen at intersections, and another 40% occur within 50 feet of intersections [2]. The most common bicyclist pre-crash maneuver is bicyclist riding across an intersection (46%) [2]. The “left hook” is one of the most prevalent bike-car crash types (19%) [3]. This paper focuses on tracking opposing direction vehicles at a traffic intersection and identifying collision danger from any of these vehicles turning left.

According to a report by IEEE Spectrum (January 2017), bicycles are probably the most difficult detection problem that autonomous vehicle systems face [4]. Cyclists are vulnerable on roads because they are difficult to detect and predict by both human motorists and autonomous vehicles. The situation is especially complicated when it comes to intersections where the traffic is complex involving multiple types of maneuvers. Research and development on vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) networks are ongoing, in which researchers have also tried to bring the bicycles into the system (V2B) [5]. However, to fully deploy these “V2X” technologies on all cars and achieve high real-world penetration among all

cars operating on the road, the deployment could take many years or even decades. On the other hand, it is possible to immediately address the problem by using inexpensive collision prevention technology on the bicycle’s side. Previous work on bicycle-based collision systems has focused only on tracking rear vehicles behind the bicycle [6].

In this paper, it is proposed to develop an on-bike Lidar based compact sensing system, which estimates kinematic states of vehicles in opposing direction traffic and provides visual and/or audio warnings to both the motorists and the cyclists under dangerous situations at traffic intersections (Fig. 1). The estimation should be good enough so that when it is integrated with a proper warning triggering metric, both false positive and false negative rates are low.

A lot of previous research has been conducted on cars using Lidar sensors for collision avoidance, autonomous driving and simultaneously localization and mapping (SLAM). However, most of the related work is done using relatively high-resolution Lidar sensors, where the horizontal angular resolution is normally better than  $1^\circ$ . It is impractical to implement such high-density Lidar systems on a bicycle where size, weight and cost (here we consider a cost of a few hundred dollars, instead of tens of thousands of dollars in autonomous driving systems) are crucial and computational power is limited. Also, as a collision avoidance system instead of an autonomous driving system, the need for differentiating different types of targets is not necessary, which justifies the possibility of using inexpensive low-resolution sensors suitable for a bicycle.

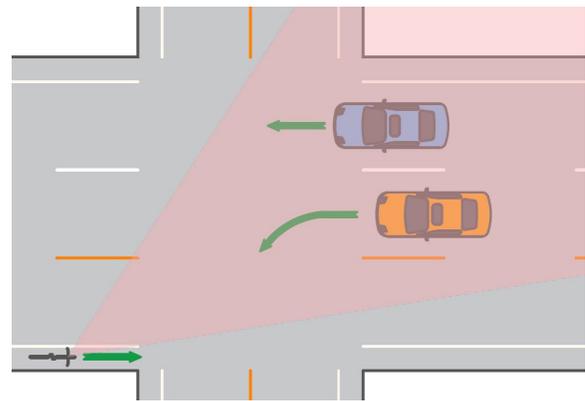


Fig. 1. A bicycle with a Lidar and two opposing traffic vehicles at a traffic intersection

This research was supported in part by a research grant from the National Science Foundation (NSF Grant PFI-1631133).

Zhenming Xie and Rajesh Rajamani are with the Department of Mechanical Engineering, University of Minnesota, Twin Cities, Minneapolis, MN 55455 USA (E-mail: xiexx592@umn.edu; rajamani@umn.edu, TEL: 612-626-7961).

## II. SENSOR SETUP AND OBJECTIVE

The sensor used in this project is the LeddarTech<sup>®</sup> Vu8 solid-state 2-D low resolution Lidar. It has 8 segments/beams

and a horizontal field of view of  $48^\circ$ . Therefore, the angular uncertainty of each segment/beam is  $48^\circ/8 = 6^\circ$ , which is a high uncertainty and makes the estimation problem challenging, as demonstrated later. The sensor is mounted stationary at the front of a bicycle, oriented  $33^\circ$  to the left as shown in Fig. 2. Data from the sensor is processed by a Teensy 3.5 microprocessor located on the bicycle.



Fig. 2. The LeddarTech<sup>®</sup> Vu8 mounted on a bicycle

While the sensor field of view is limited, under the setup here the sensor can obtain measurement points from opposing vehicles no matter whether the traffic intersection has one, two, or three lanes in each direction, as shown in Fig. 3.

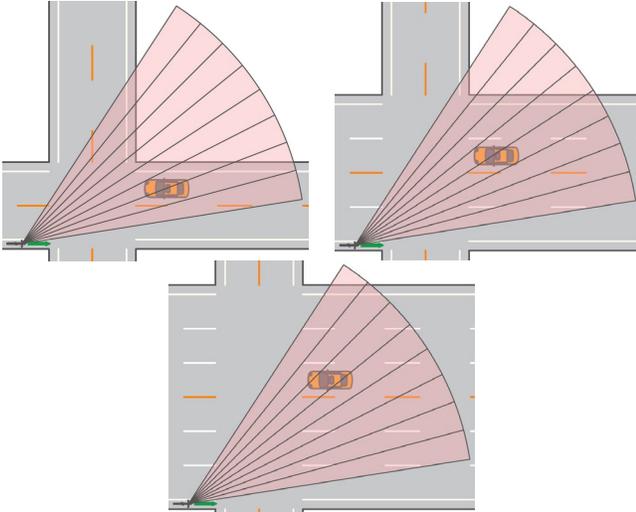


Fig. 3. Sensor coverage in different lane configurations at a traffic intersection (1 lane, 2 lanes and 3 lanes of opposing traffic with sensor range set to be 30 m)

With the stated sensor setup, the preliminary objective of this paper is to estimate trajectories of opposing vehicles that may go straight or turn left. If a left-turning vehicle causes danger to the bicyclist, while the bicyclist is crossing a traffic intersection, the event will be predicted and an auditory/visual warning will be provided to the involved motorist (from an automatic electronic horn on the on-bike system).

### III. OVERVIEW OF TRACKING SYSTEM DESIGN

A flow chart of the overall tracking system design is shown in Fig. 4. First, clustering is applied to raw measurements to detect vehicles and obtain approximate initial positions of vehicles. A correction term for each measured position is introduced to account for the laser reflection

location on the vehicle to improve the position measurement quality. The corrected position measurements are then input to the data association module, in which either a nonlinear Kalman filter update of a tracked target, or initialization of a new target, will happen. The components in Fig. 4. will be discussed in the rest of the paper. Specifically, approaches to computing measured positions of vehicles, and approaches to estimating states of each vehicle, will be discussed in section IV and section V, respectively.

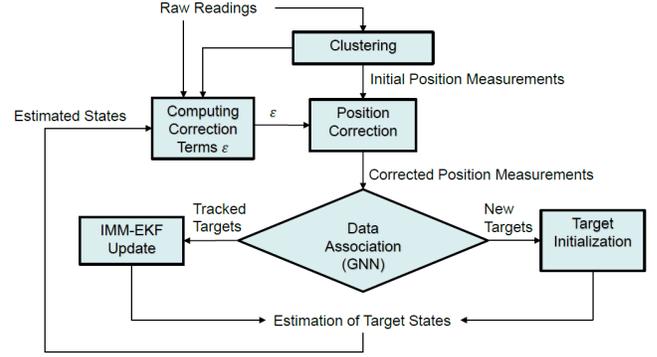


Fig. 4. Flowchart overview of the tracking system

## IV. DETECTION OF VEHICLES

### A. Clustering for Assigning Measurement Points to Target Vehicles

Due to the extremely low angular resolution ( $6^\circ$ ) of the sensor, there is a high ambiguity in the problem of which measurement point is obtained from which target (“target” and “vehicle” are used interchangeably in this article). It is expected that only one or two measurement points will be obtained from a vehicle when the vehicle is more than 20 meters away. Also, the intrinsic characteristics of raw measurement points in polar coordinates make the measurement uncertainty vary as the measured distance varies. A clustering algorithm is used to assign raw measurement points to targets. To evaluate the distance or dissimilarity between any pair of measurement points, the smaller value of two distances: the Mahalanobis distance and the Euclidean distance, is used (1):

$$a_{ij}^2 = \min\{(p_i - p_j)^T (R_{mi} + R_{mj})^{-1} (p_i - p_j), K_{Euc}^2 \|p_i - p_j\|_2^2\}, \quad (1)$$

$$i, j = 1, 2, \dots, N_m,$$

where  $p_i$  and  $p_j$  are positions of measurement points  $i$  and  $j$ ,  $R_{mi}$  and  $R_{mj}$  are the corresponding covariance matrices,  $N_m$  is the number of measurement points from a sensor sampling,  $K_{Euc}$  is a weighting parameter.

Under this distance measure, two measurement points are “similar” if:

a) they are likely to originate from the same part of a target (described by small Mahalanobis distance, mainly for targets far from the sensor), or

b) they are likely to originate from different parts of a target (described by small Euclidean distance, mainly for targets close to the sensor).

For example, if a front corner of a vehicle (far from the sensor) occupies two adjacent beams of the sensor, these two

measurement points have small Mahalanobis distance (they have relatively high uncertainty) but relatively large Euclidean distance (the centers of them are far from each other, note that a target does not need to fully occupy the two beams for the two measurement points to be obtained). In the case where a vehicle is close to the sensor, there can be several measurement points from different parts of the vehicle but the Euclidean distances between them are relatively small.

A proximity matrix  $\{a_{ij}^2\}$  is formed based on (1) and input to an agglomerative hierarchical clustering algorithm with complete linkage. The number of clusters is determined by cutting the dendrogram at a certain inter-cluster distance level  $M$ , which ensures that the distance between any pair of measurement points within each cluster is less than  $M$ :

$$\|p_i - p_j\| < M, \forall p_i, p_j \in C, \quad (2)$$

where  $C$  is any cluster. Fig. 5 demonstrates the situation where 4 measurement points are assigned to 2 clusters/targets. In this case,  $p_1$  and  $p_2$  are considered “similar” and they form a cluster, same for  $p_3$  and  $p_4$ . But the two clusters are separated because the distance between them is greater than  $M$ .

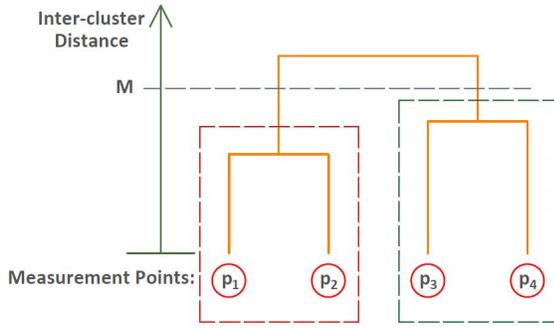


Fig. 5. Dendrogram of hierarchical clustering for assigning measurement points to targets/vehicles

### B. Obtaining Corrected Position Measurements of Vehicles

After the clustering phase, measurement points have been assigned to vehicles, with the number of clusters being the number of detected targets/vehicles. For each vehicle, we are interested in finding a representative point whose kinematic states can approximate the vehicle’s kinematic states. The idea of tracking a vehicle by finding its center based on its geometry is not practical here, since detailed geometric information of any target is hardly available due to the low sensor angular resolution. If the center of measurement points (in the same cluster) is simply taken to represent the center of a vehicle, this “center” will have undesired motion created by vehicle moving from one sensor segment to another (again because of the low angular resolution), which does not represent the maneuver of the vehicle. It is therefore proposed to track a virtual point (call it the “closest point”) with the minimum longitudinal and lateral distances of any vehicle (note that this virtual point may not be on the vehicle, see Fig. 6), and use its estimated kinematic states to represent the vehicle’s states.

To track the “closest point”, we first compute its initial position using raw measurement points, and then introduce a correction term  $\varepsilon$  for further possible improvement based on its approximate dynamic model.

For any target (cluster), its initial position measurement is obtained by finding the point with the minimum longitudinal distance ( $p_{mx}$ ) and the point with the minimum lateral distance ( $p_{my}$ ) from all points in the cluster.

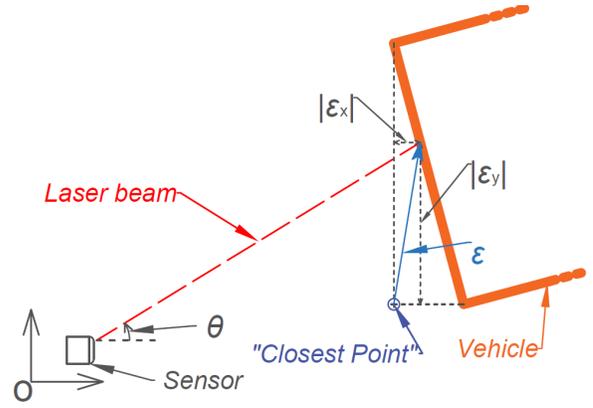


Fig. 6. The “closest point” and the  $\varepsilon$  vector

The initial position measurement and its covariance are computed as

$$\begin{aligned} z_0 &= C_x p_{mx} + C_y p_{my}, \\ \text{cov}[z_0] &= C_x \text{cov}[p_{mx}] C_x^T + C_y \text{cov}[p_{my}] C_y^T \\ C_x &= \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, C_y = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}. \end{aligned} \quad (3)$$

It is challenging to accurately and continuously obtain the closest point position measurements. For example, for an opposing vehicle, the  $y$  component of  $p_{my}$  may not reflect the minimum lateral distance because at some point  $p_{my}$  could be actually obtained from the front of the vehicle and a beam closer to the left corner of the vehicle (which could provide a “better” lateral distance) does not return a reading due to its large incident angle. However, at some point in the past, some beam should hit the left corner and that measurement could reflect the minimum lateral distance. Therefore, we try to find a way to retain this type of minimum distance information for a period of time. A 2-D correction term  $\varepsilon$ , which is a vector that connects a measurement point and the closest point of a vehicle (Fig. 6), is introduced to refine the initial measurements using the information from historical measurements.

Specifically, each measurement point  $p_i = (p_{ix}, p_{iy})^T$ ,  $i=1,2,\dots,N$  ( $N$  is the number of measurement points from a vehicle) is attached with a vector  $\varepsilon_i$ . Each component of  $\varepsilon_i$ ,  $\varepsilon_{ix}$  and  $\varepsilon_{iy}$ , has one of the two flags attached, “valid” or “invalid”. We “initialize” the two components of  $\varepsilon$  in the following way:

$$\varepsilon_{ix} = \begin{cases} 0, & \text{if } p_i = p_{mx}, \varepsilon_{ix} \text{ is invalid} \\ p_{ix} - x_{min}, & \text{if } p_i \neq p_{mx} \end{cases} \quad (4)$$

$$\varepsilon_{iy} = \begin{cases} 0, & \text{if } p_i = p_{my}, \varepsilon_{iy} \text{ is invalid} \\ p_{iy} - y_{min}, & \text{if } p_i \neq p_{my} \end{cases} \quad (5)$$

Here  $x_{min}$  and  $y_{min}$  are the minimum longitudinal distance ( $x$  component of  $p_{mx}$  in (3)) and the minimum lateral distance ( $y$  component of  $p_{my}$  in (3)) obtained by inspecting all the measurement points from the same target. After a component is initialized, its flag will be set to “valid”. If  $p_i = p_{mx}$ ,  $\varepsilon_{ix}$  is valid or  $p_i = p_{my}$ ,  $\varepsilon_{iy}$  is valid, the corresponding component  $\varepsilon_{ix}/\varepsilon_{iy}$  will not be initialized. Instead they will be “propagated” based on their dynamic equations discussed below.

To derive the dynamic equation of  $\varepsilon$ , we analyze the sliding motion of a laser reflection point. Here we assume that a) the reflection surface of any measurement point is a line in 2-D and that b) the relative velocity between the sensor and a target

is perpendicular to a reflection surface. In other words, vehicles are assumed to have rectangular shapes and zero slip angles (with respect to the sensor frame). Also, when analyzing the dynamics of  $\varepsilon$ , the laser beam is approximated as a “narrow beam” with “small width”.

In Fig. 7, a vehicle’s position change in time interval  $\Delta t$  ( $\Delta t$  should be “small” and in practice, equal to the sensor sampling time) is shown (with the vehicle being represented by part of a rectangle), where  $(v, \psi)$  is the velocity vector of the vehicle,  $\psi_l$  is the orientation of the vehicle, and  $\theta$  is the azimuth angle of a laser beam. From the geometry, we can compute  $\Delta \varepsilon_x$  and  $\Delta \varepsilon_y$  (the change of the  $\varepsilon$  vector in  $\Delta t$ ) as:

$$\Delta \varepsilon_x = -v\Delta t[\sin(\psi_l - \psi) + \cos(\psi_l - \psi) \tan(\theta - \psi_l)] \sin \psi_l, \quad (6)$$

$$\Delta \varepsilon_y = v\Delta t[\sin(\psi_l - \psi) + \cos(\psi_l - \psi) \tan(\theta - \psi_l)] \cos \psi_l. \quad (7)$$

In (6) and (7),  $v$  and  $\psi$  are in the vehicle state vector being estimated,  $\psi_l$  can be approximated by  $\psi$  using the assumption that vehicles have zero slip angles with respect to the sensor frame, yielding the dynamic equation:

$$\begin{bmatrix} \varepsilon_x \\ \varepsilon_y \end{bmatrix}_{k+1} = \begin{bmatrix} \varepsilon_x - v\Delta t \tan(\theta - \psi) \sin(\psi) \\ \varepsilon_y + v\Delta t \tan(\theta - \psi) \cos(\psi) \end{bmatrix}_k. \quad (8)$$

It can be verified that (8) is applicable for  $\psi \in [0, 2\pi)$ . Since (8) is approximated, after initialized, the  $\varepsilon$  vector attached to a beam will only be propagated by its dynamic equation in a short period of time ( $< 0.5$  second), after which it will be discarded (set to be “invalid”) if it does not get initialized again.

Using the computed  $\varepsilon$  vectors, the corrected closest point position measurement can be computed as

$$z = C_x[p_{mx} - \varepsilon_x(p_{mx})] + C_y[p_{my} - \varepsilon_y(p_{my})], \quad (9)$$

$$C_x = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, C_y = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix},$$

where  $\varepsilon_x(p_{mx})$  and  $\varepsilon_y(p_{my})$  are the corresponding  $\varepsilon$  components of  $p_{mx}$  and  $p_{my}$ . Note that  $\varepsilon_x(p_x)$  and  $\varepsilon_y(p_y)$  are used only when they are valid and they bring the measurement closer to the prediction. Invalid  $\varepsilon_x$  or  $\varepsilon_y$  will be substituted by zero in (9). This “correction” of position measurements based on the computed  $\varepsilon$  vectors is used for targets without significant turning motion. For targets with significant turning motion, (8) is less effective and it may degrade the measurement quality, therefore the “correction” is disabled. This can be done by setting  $\varepsilon_x(p_x)$  and  $\varepsilon_y(p_y)$  to be zeros in (9).

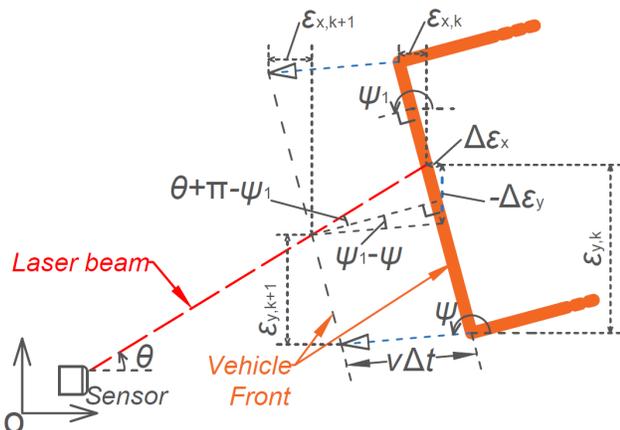


Fig. 7. Sliding motion of a laser reflection point on a surface of a vehicle

The  $\varepsilon$  information is useful in the sense that it retains information about the closest point position from the historical values of the lidar sensor measurements.

## V. STATE ESTIMATION

### A. Data Association

Data association is required when multiple targets are tracked simultaneously. The data association problem is the problem of associating new measurements to existing tracks. There are many data association approaches in the literature [7], with different optimality and computational complexity. Here the global nearest neighbor filter (GNN) is applied because it is considered sufficient for this application and its computational complexity is relatively low.

Specifically, for each target from a new sampling, its statistical distance to each existing track is computed:

$$d_{ij}^2 = r_{ij}^T S_{ij}^{-1} r_{ij} + \ln(|S_{ij}|), \quad i = 1, 2, \dots, N_o, j = 1, 2, \dots, N_t, \quad (10)$$

where  $N_t$  is the number of targets being tracked until the last sampling,  $N_o$  is the number of new measurements (targets detected in a new sampling),  $r_{ij}$  is the measurement residual and  $S_{ij}$  is the corresponding residual covariance:

$$r_{ij} = z_i - \hat{z}_j, \quad S_{ij} = H P_j H^T + R_i. \quad (11)$$

$z_i$  is a measurement from a new target  $i$ , and  $\hat{z}_j$  is a predicted position of an existed track  $j$ .  $H$  is from the output equation which will be discussed later,  $P_j$  is the state covariance of the existed track  $j$ , and  $R_i$  is the measurement uncertainty. Before applying GNN, a gating test, which is based on the Mahalanobis distances, is done to eliminate unlikely observation-to-track pairing. The term  $\ln(|S_{ij}|)$  is used for penalizing tracks with high prediction uncertainty. Therefore, it ensures that tracks with missing measurement updates, and resulting larger covariance matrices, do not “steal” measurements from higher quality tracks [7]. Once all the distances are computed, a measurement-to-track assignment matrix  $\{d_{ij}^2\}$  is formed. The objective of GNN is to find a globally optimized assignment  $\{x_{ij}\}$ , which minimizes the total distance of all the matches:

$$\begin{aligned} & \arg \min_{\{x_{ij}\}} \sum d_{ij}^2 \cdot x_{ij}, \\ & \text{subject to } x_{ij} \in \{0, 1\}, \sum_i x_{ij} = 1, \sum_j x_{ij} = 1. \end{aligned} \quad (12)$$

The assignment problem can be solved by applying the Kuhn-Munkres or the Auction algorithm [7].

### B. Extended Kalman Filter Design

After solving the data association problem, the measurement update for each track is carried out by an extended Kalman filter under the interacting multiple model framework [8] (IMM-EKF). Two dynamic models are employed, a constant turn model with polar velocity [9] (CTP) and a constant velocity model with zero turn rate (non-turning model). The state vector for estimating each vehicle’s maneuver contains five states, including the position of the vehicle (the closest point) in Cartesian coordinates, the velocity vector in polar coordinates and the turn rate:

$$X = [x, y, v, \psi, \dot{\psi}]^T. \quad (13)$$

The constant turn model with polar velocity can be described by its state space equation in discrete time:

$$\begin{aligned}
x_{k+1} &= x_k + \frac{2}{\dot{\psi}_k} v_k \sin\left(\frac{\dot{\psi}_k \Delta t}{2}\right) \cos\left(\psi_k + \frac{\dot{\psi}_k \Delta t}{2}\right) + w_{11,k}, \\
y_{k+1} &= y_k + \frac{2}{\dot{\psi}_k} v_k \sin\left(\frac{\dot{\psi}_k \Delta t}{2}\right) \sin\left(\psi_k + \frac{\dot{\psi}_k \Delta t}{2}\right) + w_{12,k}, \\
v_{k+1} &= v_k + w_{13,k}, \\
\psi_{k+1} &= \psi_k + \dot{\psi}_k \Delta t + w_{14,k}, \\
\dot{\psi}_{k+1} &= \dot{\psi}_k + w_{15,k}, \\
w_{1,k} &= (w_{11,k}, w_{12,k}, w_{13,k}, w_{14,k}, w_{15,k})^T \sim N(0, Q_{1,k}).
\end{aligned} \tag{14}$$

While  $w_{13,k}$  and  $w_{15,k}$  are used for modeling processing noise, nonzero values of  $w_{11,k}$ ,  $w_{12,k}$ , and  $w_{14,k}$  can be used for taking into account the model uncertainty.

The constant velocity model is described as:

$$\begin{aligned}
x_{k+1} &= x_k + v_k \Delta t \cos(\psi_k) + w_{21,k}, \\
y_{k+1} &= y_k + v_k \Delta t \sin(\psi_k) + w_{22,k}, \\
v_{k+1} &= v_k + w_{23,k}, \\
\psi_{k+1} &= \psi_k + w_{24,k}, \\
\dot{\psi}_{k+1} &= 0, \\
w_{2,k} &= (w_{21,k}, w_{22,k}, w_{23,k}, w_{24,k}, w_{25,k})^T \sim N(0, Q_{2,k}).
\end{aligned} \tag{15}$$

Similar to the CTP model, here  $w_{23,k}$  and  $w_{24,k}$  are used for modeling processing noise, nonzero values of  $w_{21,k}$  and  $w_{22,k}$  can be used for taking into account the model uncertainty.

At this point, position measurements (in Cartesian coordinates) of the closest point of vehicles have already been obtained from the previous stage, therefore, the measurement equation is linear:

$$z_k = \begin{bmatrix} x_k \\ y_k \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \\ v_k \\ \psi_k \\ \dot{\psi}_k \end{bmatrix} + R_k. \tag{16}$$

## VI. SIMULATION AND EXPERIMENTAL RESULTS

### A. Simulation Results

The simulation study is based on multiple MATLAB scripts and functions developed for this project. Several specific scenarios at a traffic intersection with different opposing vehicle trajectories are simulated. Here the scenarios with an opposing vehicle going straight / turning left are shown (Fig. 8). In the shown simulation scenarios, the initial velocity of the bicycle is set to be 3 m/s. The initial velocity of straight-going vehicles is set to be 10 m/s, with some acceleration and deceleration. The initial velocity of left-turning vehicles is set to be 8 m/s. Scenarios where multiple vehicles are present are also simulated (e.g. combining (a)(b) in Fig. 8).

Simulation results are shown in Fig. 9. All the magenta parts of the estimation indicate that the estimation is based on a single measurement point without the “ $\epsilon$ ” information and therefore with low fidelity. All the cyan parts indicate that significant turning motion is detected from the target vehicle. The blue ellipses indicate the uncertainty of the position estimation with 99% confidence. The two red dashed lines in

each trajectory estimation figure indicate the sensor horizontal field of view.

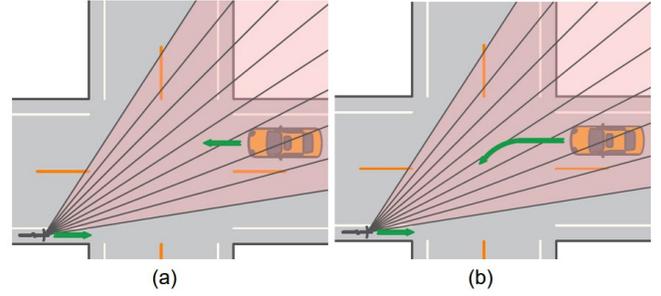


Fig. 8. Vehicle maneuver simulation scenarios: a) opposing vehicle going straight, b) opposing vehicle turning left

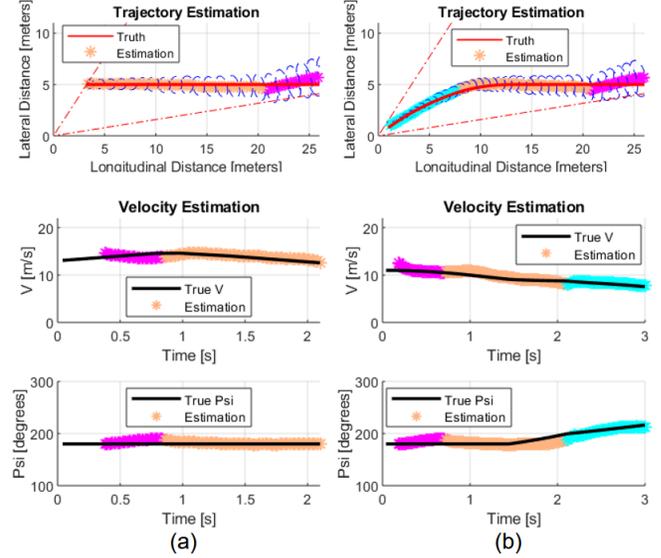


Fig. 9. Simulation results: trajectory estimation and velocity estimation, a) opposing vehicle going straight, b) opposing turning left

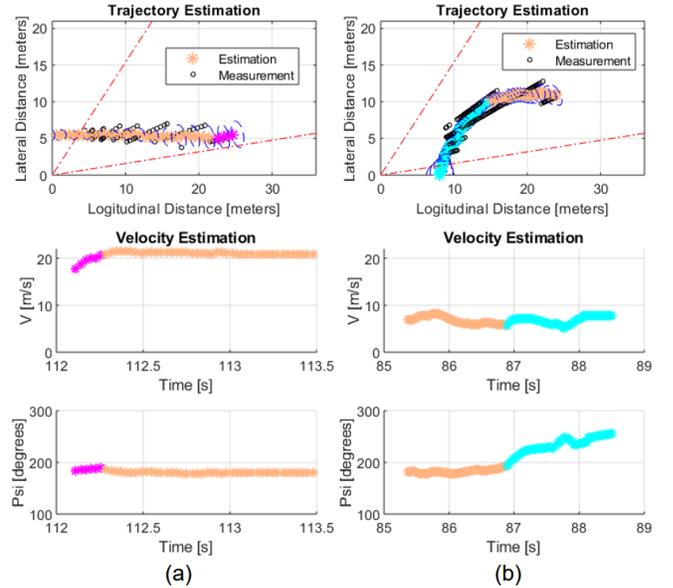


Fig. 10. On-road experimental test results: trajectory estimation and velocity estimation, a) opposing vehicle going straight b) opposing vehicle turning left

In all the scenarios, at the beginning there are magenta parts, meaning the estimation error could be relatively large. Also, the uncertainty ellipses (of position estimation) are relatively large, matching the estimation error. As the vehicle comes closer, the magenta parts are gone, meaning more information is now being obtained from the measurements (the “ $\epsilon$ ” values are valid and more than one measurement point is obtained from the vehicle). The uncertainty ellipses also become smaller. Despite the large angular measurement uncertainty of the sensor, the estimated trajectories and velocities are smooth and they match the true values pretty well. For scenarios with multiple vehicles, the estimation performance is similar to the performance of the single vehicle scenarios in Fig. 8, when there is no severe occlusion between vehicles.

### B. Experimental Results

Experiments are conducted for similar scenarios as in the simulation study. Results from controlled experiments are similar to results from the simulation study. In Fig. 10, results from the same scenarios are shown. These results are generated by on-road experimental tests with real-world uncontrolled vehicles. In the scenario with left-turning vehicles, vehicles have larger lateral and longitudinal distances (than in simulation) because the bicycle can only stay at the boundary of a traffic intersection during testing (instead of going into an intersection and creating a dangerous situation).

The on-road test results show similar patterns as the simulation results. The raw measurements associated to a vehicle are plotted for comparison with the estimation. Since the “mean” of the raw measurements are from centers of beams, they show “discontinuous” patterns due to the low angular resolution. Despite this, continuous estimation is achieved by the designed tracking system.

## VII. CONCLUSIONS

This paper described the development of an inexpensive on-bicycle sensing system which estimates trajectories of opposing direction vehicles at traffic intersections. Despite the low measurement density of the proposed Lidar sensor, vehicle tracking was achieved by clustering based approaches for assigning measurement points to individual target vehicles, novel estimation approaches for determining the longitudinal and lateral positions of a vehicle using new system models, and by exploiting data association and interacting multiple model Kalman filtering approaches for the multi-target tracking problem. Simulation results and on-road test results were shown to demonstrate the tracking performance.

## VIII. REFERENCES

- [1] National Center for Statistics and Analysis, "Bicyclists and other cyclists: 2015 data," National Highway Traffic Safety Administration, Washington, DC, 2017.
- [2] "Understanding Bicyclist-Motorist Crashes in Minneapolis, Minnesota," City of Minneapolis Public Works Department, Minneapolis, 2013.
- [3] "City Transportation Safety," Traffic, Parking, and Transportation Department, Community Development Department, City of Cambridge, MA, 2014.
- [4] P. Fairley, "The Self-Driving Car's Bicycle Problem," *IEEE Spectrum*, 31 January 2017. [Online]. Available: <https://spectrum.ieee.org/cars-that-think/transportation/self-driving/the-selfdriving-cars-bicycle-problem>.
- [5] M. Jenkins, D. Duggan and A. Negri, "Towards a connected bicycle to communicate with vehicles and infrastructure : Multimodel alerting interface with Networked Short-Range Transmissions (MAIN-ST)," in *2017 IEEE Conference on Cognitive and Computational Aspects of Situation Management (CogSIMA)*, Savannah, GA, 2017.
- [6] W. Jeon and R. Rajamani, "Rear Vehicle Tracking on a Bicycle Using Active Sensor Orientation Control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 8, pp. 2638-2649, 2018.
- [7] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*, Norwood, MA: Artech House, 1999.
- [8] Y. Bar-Shalom, X. R. Li and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*, New York: John Wiley & Sons, INC, 2001.
- [9] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking. Part I. Dynamic models," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1333-1364, 2003.