2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

Cooperative Perception and Localization for Cooperative Driving

Aaron Miller[1], Kyungzun Rim[1], Parth Chopra[2], Paritosh Kelkar[2], and Maxim Likhachev[1]

Abstract— **Fully autonomous vehicles are expected to share the road with less advanced vehicles for a significant period of time. Furthermore, an increasing number of vehicles on the road are equipped with a variety of low-fidelity sensors which provide some perception and localization data, but not at a high enough quality for full autonomy. In this paper, we develop a perception and localization system that allows a vehicle with low-fidelity sensors to incorporate high-fidelity observations from a vehicle in front of it, allowing both vehicles to operate with full autonomy. The resulting system generates perception and localization information that is both low-noise in regions covered by high-fidelity sensors and avoids false negatives in areas only observed by low-fidelity sensors, while dealing with latency and dropout of the communication link between the two vehicles. At its core, the system uses a set of Extended Kalman filters which incorporate observations from both vehicles' sensors and extrapolate them using information about the road geometry. The perception and localization algorithms are evaluated both in simulation and on real vehicles as part of a full cooperative driving system.**

I. INTRODUCTION

Vehicles are currently being developed with varying levels of driver assistance and autonomy capabilities. There are already cars on the road today that have some ability to sense their surroundings and provide driver assistance but are unable to drive autonomously without constant human supervision. These cars are designated as Level 2, or L2, vehicles. Meanwhile, Level 4 (L4) vehicles are fully autonomous and do not require human supervision in the areas in which they are approved to drive. Such vehicles are currently in development, but are not yet available, and will take a long time to replace existing cars on the road even once they are available. For a long period of time, the road will be shared by L4, L2, and lower capability vehicles.

Standards are already in place for vehicle-to-vehicle (V2V) communication, allowing vehicles to communicate wirelessly, albeit at low bandwidth and limited range. Because of this, it is desirable for a limited number of vehicles with high-fidelity sensors (L4 vehicles) to be able to share information from their own perception systems with less capable (L2) vehicles. This allows an L2 vehicle to achieve "Affordable Autonomy through Cooperative Sensing & Planning" (Figure 1), where an L2 vehicle is able to operate autonomously without expensive sensors by receiving perception information and a suggested trajectory from an L4 vehicle. The L2 vehicle then fuses the perception information from both vehicles, the L4 vehicle and itself, and uses it to generate a safe plan which follows the L4 vehicle but also avoids obstacles that might not be visible to the L4. This paper is concerned specifically with the process of fusing the perception and localization measurements and not the other parts of the system.

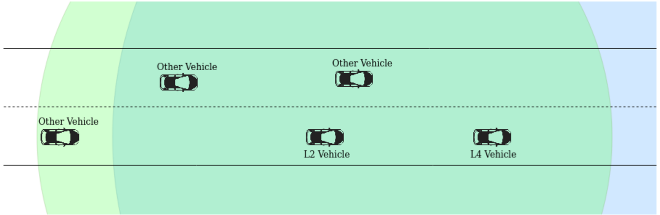


Fig. 1: An L2 vehicle engaged in cooperative sensing and planning with an L4 vehicle. The unoccluded L4 sensor range is shown in blue, and the L2 sensor range in green.

There are several challenges involved in such a system. First, the bandwidth of a V2V link is not high, meaning that raw sensor information cannot be shared and fused in the same way that sensors would be fused on a single vehicle. Second, there is some latency associated with the network communication. At highway speeds (approximately 55mph and above), this means that measurements of each observed vehicle must be extrapolated separately and accurately for robust results. And finally, the system must be able to handle a drop in communication or in quality of perception data from the L4. If another vehicle cuts between the L2 and the L4, or if the L4 goes out of range of the L2, there must be some amount of time where the L2 is able to operate autonomously before the driver is able to take over.

We present a cooperative perception and localization system which deals with these issues. Our system shares observed vehicles ("tracks") and their associated uncertainties along with localization estimates and uncertainties. These estimates are matched to vehicles observed by the L2, allowing elimination of false negatives due to occlusion or limited sensor range. The associated measurements are then fused together in an Extended Kalman filter to give high-fidelity estimates of tracked vehicle states at the current time. Because of this EKF architecture, the system degrades gracefully over time if communication drops out, producing reliable perception until the driver is able to take over. The system is validated both in a simulator and on physical test vehicles equipped with typical L2- and L4-capable sensor suites.

II. RELATED WORK

There has been a good deal of work in the cooperative driving space. One set of approaches has come primarily from a connected vehicles perspective with less emphasis on sensing. Many of these approaches were demonstrated

[1]The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA
[2]Honda R&D Americas, Inc, Ann Arbor, MI, USA

978-1-7281-7395-5/20/$31.00 ©2020 IEEE

in the Grand Cooperative Driving Challenge [1]. This required competitors to demonstrate various connected vehicle behaviors, such as platooning, using shared localization information from the connected vehicles, but did not require interaction with vehicles only observed by sensors.

Some work has also been done to fuse perception data from multiple vehicles. Several works have done this using occupancy grids [2] [3] [4] or raw point clouds [5]. These approaches successfully combine static obstacles from multiple vehicles, eliminating false negatives and mitigating occlusions. They do have some ability to deal with dynamic obstacles either at low speed or low latency. However, due to the loss of representation of individual vehicles and the lack of velocity information, these approaches cannot fuse observations of other vehicles at highway speeds while tolerating latency in communication.

Other approaches include using Probability Hypothesis Density (PHD) filters [6], which are able to represent uncertainty both in locations of observed objects and in the number of observed objects. However, these approaches require high communication bandwidth and latency is not considered.

Rauch et al. [7] do fuse individual vehicles by representing them as point clouds, using a single point with uncertainty for each corner of each observed vehicle. This successfully fuses tracks with lower bandwidth requirements, but suffers from higher estimation errors than expected and does not consider latency compensation necessary for operation at highway speeds.

On the localization side, [8] uses multiple sensors to decide the global position of a Micro Aerial Vehicle (MAV). In this paper, as they use sensors having different frequencies, they process delayed measurements and integrate them with a Multi Sensor Fusion Extended Kalman Filter. This method succeeds in compensating for delayed measurements and localizing the MAV in both indoor and outdoor environments. However, the error of this approach is too high to apply to a vehicle running at high speed.

Our approach focuses on the ability to operate at highway speeds while dealing with limited network bandwidth and latency.

III. FORMULATION

A. Sensor Suites

The approach we present is general and can be applied regardless of the types of sensors on each vehicle. With that being said, the domain for the experiments is highway driving. In our experiments, the L4 vehicle is equipped with high-resolution sensors typical of an L4 test vehicle today - Lidar and Radar covering 360 degrees around the vehicle as well as cameras for perception, and an RTK GPS for localization. The L2 vehicle, on the other hand, is equipped only with Radar and a forward-facing camera for perception and a lower resolution GPS-INS system for localization. Each vehicle is equipped with a DSRC radio for V2V communication. Diagrams showing similar vehicles to the ones that we tested on are shown in Figure 2.

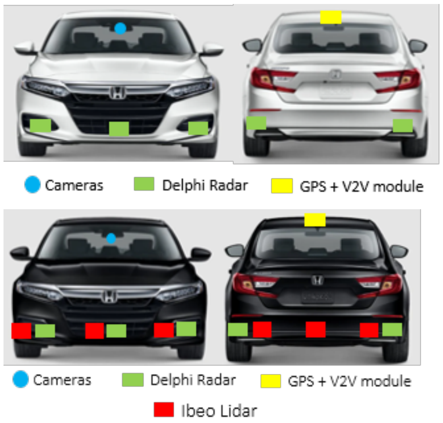


Fig. 2: Sensor layouts of the two test vehicles. The L2 is above and the L4 is below.

B. Mathematical Formulation

We start by denoting the L2 vehicle state by $\boldsymbol{x}_t^{\text{L2}}$ and the L4 vehicle state by $\boldsymbol{x}_t^{\text{L4}}$. Similarly, vehicles other than the L2 or the L4 have states $\boldsymbol{x}_t^i$. The state of each vehicle is a vector $\boldsymbol{x} = \begin{bmatrix} x & y & \theta & v & \omega \end{bmatrix}^T$, where x, y, and θ are the pose of the center of the rear axle, v is the signed speed, and ω is the angular velocity. For our cooperative localization algorithm, we will omit the angular velocity (and equivalently, wheel angle) because it is considered to be an internal part of the state, not observed by the localization system. Then, the state of the whole environment can be summarized as $E_t = (\boldsymbol{x}_t^{\text{L2}}, \boldsymbol{x}_t^{\text{L4}}, \boldsymbol{x}_t^1, \boldsymbol{x}_t^2, ..., \boldsymbol{x}_t^N)$.

Several types of measurements are assumed to be available, and all are assumed to be normally distributed. First, each vehicle has some localization system which provides measurements

$$\begin{aligned} \boldsymbol{z}_t^{\text{L2}} &= \boldsymbol{x}_t^{\text{L2}} + \boldsymbol{\varepsilon}_t^{\text{L2}}, \boldsymbol{\varepsilon}_t^{\text{L2}} \sim \mathcal{N}(0, \Sigma_t^{\text{L2}}) \\ \boldsymbol{z}_t^{\text{L4}} &= \boldsymbol{x}_t^{\text{L4}} + \boldsymbol{\varepsilon}_t^{\text{L4}}, \boldsymbol{\varepsilon}_t^{\text{L4}} \sim \mathcal{N}(0, \Sigma_t^{\text{L4}}) \end{aligned} \tag{1}$$

with independent Gaussian noise $\boldsymbol{\varepsilon}_t^{\text{L2}}$ and $\boldsymbol{\varepsilon}_t^{\text{L4}}$, respectively.

The perception system is assumed to both have some false negatives and have some error in estimated state of the tracked vehicles. These false negatives can have a variety of causes, the most common being occlusions and limited sensor range. Formally, we assume there is some probability $p_{\text{FN}}^{\text{L2}}(\boldsymbol{x}_t^i, E_t)$ that track i is undetected by the L2 vehicle's perception system at time t, and similarly probability $p_{\text{FN}}^{\text{L4}}(\boldsymbol{x}_t^i, E_t)$ that track i is undetected by the L4 vehicle at time t, where the subscript FN stands for false negative. For vehicles with states $\boldsymbol{x}_t^i$ which are detected at time t, we have measurements

$$\begin{aligned} \boldsymbol{z}_t^{i,\text{L2}} &= \boldsymbol{x}_t^i + \boldsymbol{\varepsilon}_t^{i,\text{L2}}, \boldsymbol{\varepsilon}_t^{i,\text{L2}} \sim \mathcal{N}(0, \Sigma_t^{i,\text{L2}}) \\ \boldsymbol{z}_t^{i,\text{L4}} &= \boldsymbol{x}_t^i + \boldsymbol{\varepsilon}_t^{i,\text{L4}}, \boldsymbol{\varepsilon}_t^{i,\text{L4}} \sim \mathcal{N}(0, \Sigma_t^{i,\text{L4}}), \end{aligned} \tag{2}$$

for vehicles detected by the L2 and L4, respectively, again with independent Gaussian noise $\boldsymbol{\varepsilon}_t^{i,\text{L2}}$ and $\boldsymbol{\varepsilon}_t^{i,\text{L4}}$.

IV. PERCEPTION METHODS

The overall algorithm uses a set of Extended Kalman Filters (EKFs), one for each tracked vehicle. The state of each EKF is reinitialized each time a new set of tracks is available from the L4 vehicle; during normal operation, this measurement is delayed by communication latency, and during a drop in communication this measurement is delayed even more. We reinitialize the state instead of fusing the whole stream of measurements from the L4 because the L4 perception system is assumed to have its own filtering scheme, and the measurement we receive is assumed to contain all information from older measurements made by the L4's sensors. More recent measurements from the L2 perception system are then fused in these EKFs to get a current estimate of the tracked vehicles' states. For this to be possible, we must know which track observed by the L2 corresponds to each track observed by the L4, as well as which tracks were only observed by one of the two vehicles. Finding this correspondence is nontrivial, especially when covariances of estimates aren't uniform. The overall perception algorithm (shown in Algorithm 1) then has two essential components: a MATCH function which takes two sets of tracks and returns a set of pairings between them, and a set of EKFs which make up the PREDICT and UPDATE functions used to extrapolate and fuse the individual track states. It uses these to extrapolate the current set of tracks $S_{t-1|t-1}$ to a timestamp t when a measurement is available, producing the set $S_{t|t-1}$, then associate the measurements with existing tracks and update them to produce the set $S_{t|t}$, before repeating the process until the current time is reached.

This algorithm handles drops in communication with no further modifications - when no new messages are being received, it is still able to take the last received measurement from the L4 vehicle and iteratively fuse more recent measurements from the L2; as the time since the communication failure increases, the output gradually becomes noisier and closer to the L2's raw perception output.

A. Matching Tracks between Vehicles

Each measurement must be associated correctly with a measurement from the other vehicle; cases where a vehicle is not observed by both the L2 and the L4 should be correctly identified. This is formulated as a minimum-weight bipartite matching problem, where a measurement can be matched either to a measurement from the other vehicle or left unmatched. A cost is associated with each of these, coming from the log-likelihood of the measurement under that match. For a measurement $\boldsymbol{z}_t^{i,\mathrm{v}}$ from vehicle $\mathrm{v} \in \{\mathrm{L2}, \mathrm{L4}\}$ that is left unmatched, the cost is $-\ln p_{\mathrm{FN}}^{\mathrm{v}}(\boldsymbol{z}_t^{i,\mathrm{v}}, E_t)$. For a pair of measurements $(\boldsymbol{z}_t^{i,\mathrm{L2}}, \Sigma_t^{i,\mathrm{L2}})$ and $(\boldsymbol{z}_t^{j,\mathrm{L4}}, \Sigma_t^{j,\mathrm{L4}})$ that are matched to each other, the cost is

$$\begin{aligned} C(\boldsymbol{z}_t^{i,\mathrm{L2}}, \boldsymbol{z}_t^{j,\mathrm{L4}}) = &-\ln(1 - p_{\mathrm{FN}}^{\mathrm{L2}}(\boldsymbol{z}_t^{i,\mathrm{L2}}, E_t)) \\ &-\ln(1 - p_{\mathrm{FN}}^{\mathrm{L4}}(\boldsymbol{z}_t^{j,\mathrm{L4}}, E_t)) \\ &-\ln f(\boldsymbol{z}_t^{i,\mathrm{L2}}; \hat{\boldsymbol{x}}_t^{ij}, \Sigma_t^{i,\mathrm{L2}}) \\ &-\ln f(\boldsymbol{z}_t^{j,\mathrm{L4}}; \hat{\boldsymbol{x}}_t^{ij}, \Sigma_t^{j,\mathrm{L4}}), \end{aligned} \tag{3}$$

where $f(\cdot; \mu, \Sigma)$ is the PDF (probability density function) of the multivariate normal distribution and $\hat{\boldsymbol{x}}_t^{ij}$ is the mean of the estimate formed by fusing measurements $\boldsymbol{z}_t^{i,\mathrm{L2}}$ and $\boldsymbol{z}_t^{j,\mathrm{L4}}$. This represents the negative log-likelihood of detecting both vehicles as well as measuring them at the given positions. It should be noted that E_t is unknown. In our experiments, a constant $p_{\mathrm{FN}}^{\mathrm{L2}}$ and $p_{\mathrm{FN}}^{\mathrm{L4}}$ were sufficient; however, it would also be possible to use an approximation of E_t and a more sophisticated function for p_{FN}.

Once a cost is assigned to each matching, the problem is solved by the Hungarian algorithm [9] [10] [11], which finds the best cost matching in cubic time in the total number of measurements from both vehicles. An example matching problem, and its formulation as a bipartite graph to be solved by the Hungarian algorithm, is shown in Figure 3.

B. EKF per Tracked Vehicle

Each tracked vehicle has its own EKF with state space $[x, y, \theta, v, \omega]^T$, where x, y, and θ are the pose in SE(2), v is the signed speed, and ω is the angular velocity. The angular velocity is not expected to be measured; instead, we use the road curvature and vehicle speed v to calculate a prior for ω. The PREDICT function assumes constant speed and angular velocity to extrapolate the given set of states to the desired time, while adding process noise. The UPDATE function is then the standard EKF update which takes the filter state and covariance $(\hat{\boldsymbol{x}}, \hat{\Sigma})$ and measurement $(\boldsymbol{z}, \Sigma)$ and returns the new state and covariance. Because the L2 perception system is assumed to have some internal filtering scheme that would cause sequential measurements to be highly correlated, we do not fuse every measurement we get from the L2 perception system; instead, we subsample down to a frequency at which sequential measurements are not highly correlated.

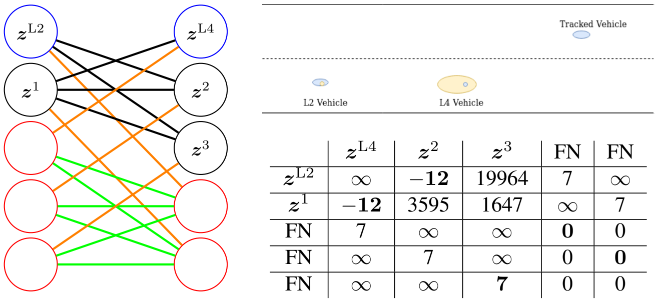
V. LOCALIZATION METHODS

A. Estimation of Global Position of L2

To get a more accurate global position estimate for the L2 vehicle, the measurements from the L4 vehicle, which have a higher resolution, are fused with measurements from the L2 vehicle. We have two measurements of the L2 vehicle

Algorithm 1 Fuse Perception

```
$t_{\mathrm{L4}}$ = Time of last available measurement from the L4
$T$ = Time of last available measurement from the L2
$S_{t_{\mathrm{L4}}|t_{\mathrm{L4}}} = \{\hat{\boldsymbol{x}}^i_{t_{\mathrm{L4}}}, \hat{\Sigma}^i_{t_{\mathrm{L4}}}\} = \{\boldsymbol{z}^{i,\mathrm{L4}}_{t_{\mathrm{L4}}}, \Sigma^{i,\mathrm{L4}}_{t_{\mathrm{L4}}}\}$
for $t = t_{\mathrm{L4}} + 1, ..., T$ do
    $S_{t|t-1}$ = PREDICT($S_{t-1|t-1}$, $t$)
    $M$ = MATCH($S_{t|t-1}$, $\{\boldsymbol{z}_t^{j,\mathrm{L2}}, \Sigma_t^{j,\mathrm{L2}}\}_{j=1}^{M_t}$)
    $S_{t|t} = \emptyset$
    for match in $M$ do
        if match is a false negative from one vehicle then
            Add one measurement ($\hat{\boldsymbol{x}}^i_{t|t-1}$ or $\boldsymbol{z}_t^{j,\mathrm{L2}}$) to $S_{t|t}$
        else
            Add UPDATE($\hat{\boldsymbol{x}}^i_{t|t-1}, \boldsymbol{z}_t^{j,\mathrm{L2}}$) to $S_{t|t}$
return PREDICT($S_{T|T}$, $t_{now}$)
```



	z^{L4}	z^2	z^3	FN	FN
z^{L2}	∞	$\mathbf{-12}$	19964	7	∞
z^1	$\mathbf{-12}$	3595	1647	∞	7
FN	7	∞	∞	**0**	0
FN	∞	7	∞	0	**0**
FN	∞	∞	**7**	0	0

Fig. 3: Example toy matching problem. The scenario is shown in the top right; tracks and localization from the L4 vehicle (with their covariances in x and y) are shown in blue, and tracks and localization from the L2 are shown in yellow. In the representation of the matching problem as a bipartite graph (on the left), each blue node represents a localization measurement, black nodes represent perception measurements, and red nodes represent false negatives. The left column shows the L2 localization and one track detected by the L2, while the right column shows the L4 localization and two tracks detected by the L4 perception system. Each edge represents an allowed connection - no edge between two nodes is the same as infinite cost. Black edges are associated with pairs of measurements and have costs given by the function C. Orange edges represent assigning a detection as undetected by the other vehicle and have costs $-\ln p^{\mathrm{v}}_{\mathrm{FN}}$. Green edges allow FN nodes to be matched with each other for 0 cost, which is necessary for this to be a well-defined bipartite matching problem. The table of edge costs is shown in the bottom right; edges forming the optimal matching are in bold.

position. The first measurement is received from GPS-INS on the L2 vehicle; the second measurement is received from the L4 perception system, from which we calculate the L2 vehicle location by adding the global position measurement of the L4 vehicle to the measurement of the L2 vehicle position relative to L4 vehicle. These measurements from the L4 vehicle are delayed by time τ. To use these two measurements, we define the dynamical system

$$\begin{aligned} \boldsymbol{x}_t^{\mathrm{L2}} &= A_t \boldsymbol{x}_{t-\tau}^{\mathrm{L2}} + \boldsymbol{\varepsilon}_t, \boldsymbol{\varepsilon}_t \sim \mathcal{N}(0, Q_t) \\ A_t &= \begin{bmatrix} 1 & 0 & 0 & cos(\theta)\tau \\ 0 & 1 & 0 & sin(\theta)\tau \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \end{aligned} \tag{4}$$

where $\boldsymbol{\varepsilon}_t$ is the process noise, and measurement models

$$\begin{aligned} \boldsymbol{z}_{t-\tau}^{\mathrm{L4}} &= \boldsymbol{x}_{t-\tau}^{\mathrm{L4}} + \boldsymbol{\varepsilon}_{t-\tau}^{\mathrm{L4}}, \boldsymbol{\varepsilon}_{t-\tau}^{\mathrm{L4}} \sim \mathcal{N}(0, \Sigma_{t-\tau}^{\mathrm{L4}}) \\ \boldsymbol{z}_{t-\tau}^{\mathrm{L2-L4}} &= \boldsymbol{x}_{t-\tau}^{\mathrm{L2-L4}} + \boldsymbol{\varepsilon}_{t-\tau}^{\mathrm{L2-L4}}, \boldsymbol{\varepsilon}_{t-\tau}^{\mathrm{L2-L4}} \sim \mathcal{N}(0, \Sigma_{t-\tau}^{\mathrm{L2-L4}}) \\ \boldsymbol{z}_t^{\mathrm{L2}} &= \boldsymbol{x}_t^{\mathrm{L2}} + \boldsymbol{\varepsilon}_t^{\mathrm{L2}}, \end{aligned} \tag{5}$$

where $\boldsymbol{\varepsilon}_{t-\tau}^{\mathrm{L4}}$ is the measurement noise of the sensors for the global position of the L4 vehicle. $\boldsymbol{\varepsilon}_{t-\tau}^{\mathrm{L2-L4}}$ is the measurement noise of the sensors which measure the position of the L2 vehicle relative to the L4 vehicle. $\boldsymbol{\varepsilon}_t^{\mathrm{L2}}$ is the measurement noise of the sensors which measure the global position of the L2 vehicle. We use an Extended Kalman Filter modified by using the measurement from the L4 vehicle $\boldsymbol{z}_{t-\tau}^{\mathrm{L2}}$ to calculate $\hat{\boldsymbol{x}}_{t-\tau}^{\mathrm{L2}}$ in the prediction step instead of using the previous step's L2 vehicle estimate $\hat{\boldsymbol{x}}_{t-1}^{\mathrm{L2}}$, because all measurements are assumed to come from filters which already account for all past information.

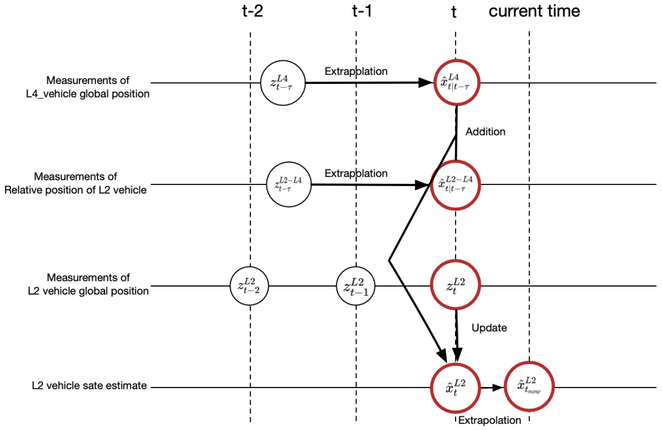


Fig. 4: Diagram of measurements and state estimates of the L2 vehicle. t means the time when we process measurements and "current time" is when our algorithm finishes.

To estimate the L2 vehicle state, we first calculate $\boldsymbol{z}_{t-\tau}^{\mathrm{L2}}$, where τ is a time latency between measured time and time when we process measurements, by adding measurements of the global position of the L4 vehicle and the L2 vehicle's position relative to the L4 vehicle.

$$\begin{aligned} \boldsymbol{z}_{t-\tau}^{\mathrm{L2}} &= \boldsymbol{z}_{t-\tau}^{\mathrm{L4}} + \boldsymbol{z}_{t-\tau}^{\mathrm{L2-L4}} \\ &= \boldsymbol{x}_{t-\tau}^{\mathrm{L2}} + \boldsymbol{\varepsilon}_{t-\tau}^{\mathrm{L4}} + \boldsymbol{\varepsilon}_{t-\tau}^{\mathrm{L2-L4}} \end{aligned} \tag{6}$$

As $\boldsymbol{z}_{t-\tau}^{\mathrm{L2}}$ is also a normal distribution, we use the mean of $\boldsymbol{z}_{t-\tau}^{\mathrm{L2}}$ as $\hat{\boldsymbol{x}}_{t-\tau}^{\mathrm{L2}}$. In the prediction step, we predict the L2 vehicle state $\hat{\boldsymbol{x}}_{t|t-\tau}^{\mathrm{L2}}$ by using the system model and $\hat{\boldsymbol{x}}_{t-\tau}^{\mathrm{L2}}$. In the update step, we calculate the state estimate $\hat{\boldsymbol{x}}_t^{\mathrm{L2}}$ by updating the predicted state with the measurement from the L2 vehicle $\boldsymbol{z}_t^{\mathrm{L2}}$.

B. Compensating for Latency with Prediction

We compensate the measurement from the L2 vehicle which has a lower resolution with measurements received from the L4 vehicle. As in Figure 4, when there is latency in the communication between the L2 vehicle and the L4 vehicle, we compensate for the L4 global position and the L2-L4 relative distance measurements by extrapolation. Extrapolation is implemented with the same algorithm used in the Kalman Filter prediction step using the model in Eq. 4. However, if the latency is longer than 1 second, we only use the measurement from the L2 vehicle, because predicting the state with measurements from the L4 vehicle does not reduce the resulting localization error.

VI. EXPERIMENTS AND RESULTS

We tested our perception and localization system both in simulation and on real vehicles. Testing in simulation allowed us to study how the system responds to various amounts of latency and to gather statistics on how the system performs. On the real vehicle, we are able to gather some statistics as well. However, we rarely have ground truth positions for the tracked vehicles, so we are primarily concerned with demonstrating that the perception system generates smooth output which eliminates false negatives. In the following subsections, we present results from testing perception and localization in simulation and on real vehicles.

A. Simulation

Experiments in simulation were done in a highway driving scenario using the VTD simulator. The road is a divided highway with a speed limit of 60mph (26.8m/s) and moderate to heavy traffic traveling at approximately that speed. The simulator generates perception from both the L2 and L4 vehicles by applying limited sensor range to each (200m for the L4 and 100m for the L2), along with Gaussian noise on the poses and speeds of all observed vehicles (.12m and .5m/s on position and velocity respectively on the L4, and .25m and .5m/s on the L2). Similarly, the simulator adds Gaussian noise to the localization available to the system for each vehicle (.01m on the L4 and .1m on the L2). Random latency is also injected into the communication between the vehicles; the typical delay is around 100ms, but there are random spikes of higher latency as well.

1) Perception: First, the RMS error and the 99th percentile error of the fused perception output is shown in Figure 5. It remains in a high-quality range (on the level of the L4 perception system) for all typical latencies; for longer delays from the last measurement, as caused by drops in communication, the error increases, but remains in a usable range for long enough that the system can continue operating until the system is able to ask the driver to take control.

The rate of mismatched tracks is also low; in experiments in simulation, only 0.017% of measurements were incorrectly associated between the L2 and L4 perception systems.

2) Localization: We evaluate the performance of our localization system by using the RMS error. The error is calculated by comparing the estimated location with the ground truth location. We have 2 error distributions in Figure 8: the measurement location error is for location estimated by measurements from only the L2 vehicle and updated location error is for cooperative localization.

The average of the updated location error is 0.074m, which is lower than the average of the measurement location error, 0.127m. The updated location error is small enough to localize the car at highway speeds.

B. Real vehicles

Experiments were run on data from real vehicles on a test track and on a public road. The test track is similar to a divided highway with three lanes. Vehicles were not driving at highway speed, but instead at approximately 12m/s on the test track and up to 18m/s on the public road.

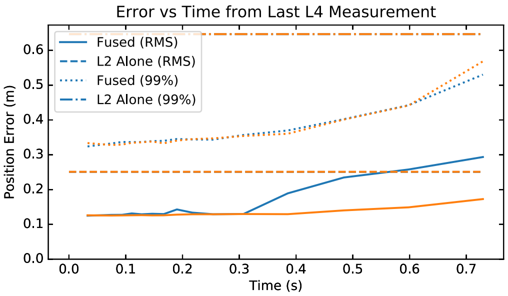


Fig. 5: RMS and 99th-percentile errors between the ground truth track locations and output of the fused perception system. Position error is split into on-track (blue) and cross-track (orange) error. The horizontal axis shows the time since the last measurement from the L4 perception system; the average error increases with time, but remains near or below the L2 perception system on its own up to 1s of latency.

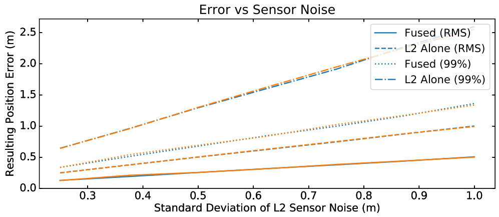


Fig. 6: Change in perception error with varying amounts of noise in simulation, with L4 and L2 sensor noise increased in proportion to each other. As in Figure 5, blue and orange are on-track and cross-track error respectively. Error increases linearly with sensor noise, as would be expected.

1) Perception: The perception system was tested on several scenarios on the test track and a public road, with other vehicles in each scenario. The perception system successfully matched the vehicles from the L2 and L4 perception systems correctly, eliminating false negatives (which were due to occlusions and sometimes to blind spots). An example detection is shown in Figure 9. Because we did not have access to ground truth poses or velocities for most of the tracked vehicles, we do not study the accuracy of the predicted tracks on the real data. However, we do have one example of the system tracking a vehicle equipped with an RTK to provide ground truth position; this is shown in Figure 7, where the system handles false negatives from both the L2 and L4 perception systems at different times and generates a smooth output trajectory close to the ground truth.

2) Localization: For the localization system, we get measurements of the L4 vehicle global location $z_{t-\tau}^{\mathrm{L4}}$ from the RTK GPS on the L4 vehicle and the global position of the L2 vehicle z_t^{L2} from GPS-INS on the L2 vehicle. In our tests, we use an RTK mounted on the L2 to simulate the position of the L2 vehicle relative to the L4 vehicle $z_{t-\tau}^{\mathrm{L2-L4}}$ instead of the L4 perception system because the L4 did not detect the L2 vehicle consistently. We add white noise to the

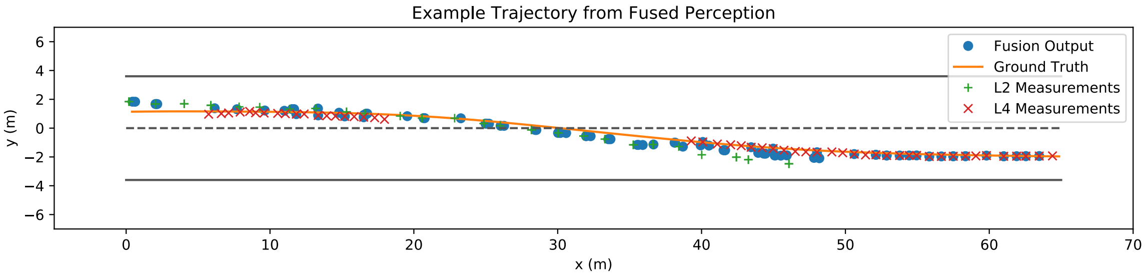


Fig. 7: The trajectory, both raw and fused, of a vehicle tracked by the perception system. The vehicle is traveling from left to right while changing lanes. The fusion system successfully deals with false negatives from both perception systems, while also having less error than the L2 system alone. Because the tracked vehicle also has an RTK, we can use this as ground truth.

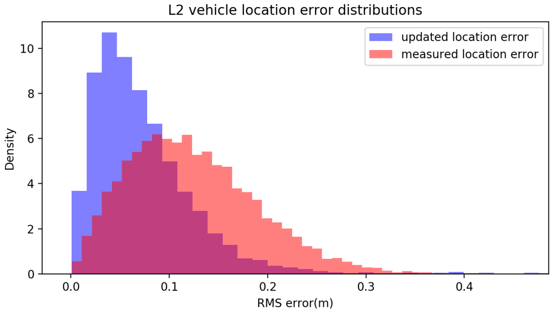


Fig. 8: Distributions of localization error in simulation. The blue histogram shows the RMS error of the fused L2 vehicle location. The red histogram shows the RMS error of the L2 localization measurement before fusion. RMS error is defined as the total displacement $\sqrt{x^2+y^2}$.

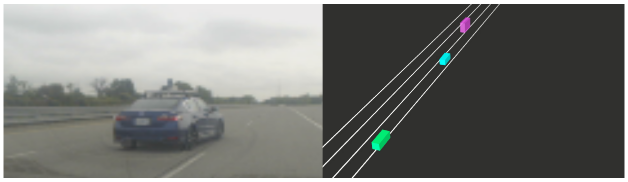


Fig. 9: An example detection on the real vehicle. The image on the left is from a camera in the L4 vehicle (not the camera used for detection). On the right, the detected vehicle is seen in pink in front of the L4 (in blue) and the L2 (in green).

$z_{t-\tau}^{\text{L2}-\text{L4}}$ with a standard deviation of 0.02m. For the ground truth location, we use measurements from an RTK on the L2.

We evaluate our performance with the cross-track error, which is the error in the y direction in the local frame of the L2 vehicle. The right histogram in Figure 10 shows that RMS errors are reduced when we update measurements with information from the L4 vehicle. As there is a lot of noise in the on-track direction in GPS-INS data, the error of the updated estimates is large. However, the cross-track error is small enough, which is more important.

VII. DISCUSSION AND CONCLUSION

We presented approaches for cooperative perception and localization between two vehicles, capable of combining and filtering measurements from both while dealing with latency and drops in communication. This approach was demonstrated successfully both in simulation and on data from real vehicles as part of a full cooperative driving system.

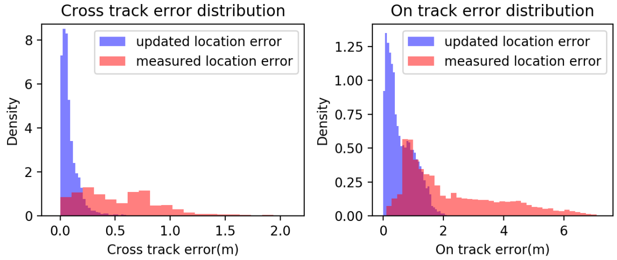


Fig. 10: Distributions of the cross-track error and on-track error of real localization data. Measured location error is the error of measurements from the L2 vehicle's GPS-INS. Updated location error is the error of fused estimates.

Possible issues with this perception approach include tracks that are missed due to occlusion in crowded areas with high sensor uncertainty; because we model the probability of false negatives and the sensor uncertainty as uniform over space, and independent of occlusions, this can cause challenges in matching the observations from the L2 and the L4 correctly. Future work could use a more sophisticated model for p_{FN}, or a more complex sensor model, which accounts for this and makes the matching more robust. Similarly, future work could investigate the use of non-Gaussian distributions to model the errors in both perception and localization more accurately. This work was also focused on highway driving with one L2 vehicle following one L4 vehicle; different scenarios and sets of vehicles are potential extensions.

ACKNOWLEDGEMENTS

This research was funded by Honda R&D Americas, Inc. The contents of this paper reflect the views of the authors, who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views of Honda R&D Americas, Inc.

References

[1] J. Ploeg, E. Semsar-Kazerooni, A. I. M. Medina, J. F. C. M. de Jongh, J. van de Sluis, A. Voronov, C. Englund, R. J. Bril, H. Salunkhe, Á. Arrúe, A. Ruano, L. García-Sol, E. van Nunen, and N. van de Wouw, "Cooperative Automated Maneuvering at the 2016 Grand Cooperative Driving Challenge," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 4, pp. 1213–1226, Apr. 2018.

[2] W. Liu, S. W. Kim, Z. J. Chong, X. T. Shen, and M. H. Ang, "Motion planning using cooperative perception on urban road," in *2013 6th IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, Nov. 2013, pp. 130–137.

[3] S. Saxena, I. K. Isukapati, S. F. Smith, and J. M. Dolan, "Multiagent Sensor Fusion for Connected & Autonomous Vehicles to Enhance Navigation Safety," p. 13.

[4] F. Camarda, F. Davoine, and V. Cherfaoui, "Fusion of evidential occupancy grids for cooperative perception," in *2018 13th Annual Conference on System of Systems Engineering (SoSE)*, June 2018, pp. 284–290.

[5] S. Kim, B. Qin, Z. J. Chong, X. Shen, W. Liu, M. H. Ang, E. Frazzoli, and D. Rus, "Multivehicle Cooperative Driving Using Cooperative Perception: Design and Experimental Validation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 663–680, Apr. 2015.

[6] J. Gan, M. Vasic, and A. Martinoli, "Cooperative multiple dynamic object tracking on moving vehicles based on Sequential Monte Carlo Probability Hypothesis Density filter," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Nov. 2016, pp. 2163–2170.

[7] A. Rauch, S. Maier, F. Klanner, and K. Dietmayer, "Inter-vehicle object association for cooperative perception systems," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, Oct. 2013, pp. 893–898.

[8] S. Lynen, M. W. Achtelik, S. Weiss, M. Chli, and R. Siegwart, "A robust and modular multi-sensor fusion approach applied to MAV navigation," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov. 2013, pp. 3923–3929.

[9] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[10] R. Jonker and T. Volgenant, "Improving the Hungarian assignment algorithm," *Operations Research Letters*, vol. 5, no. 4, pp. 171–175, Oct. 1986.

[11] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE International Conference on Image Processing (ICIP)*, Sept. 2016, pp. 3464–3468.