

Real-Time Distance Estimation and Filtering of Vehicle Headways for Smoothing of Traffic Waves*

Rahul Bhadani
The University of Arizona
Tucson, Arizona
rahulbhadani@email.arizona.edu

Matthew Bunting
The University of Arizona
Tucson, Arizona
mosfet@email.arizona.edu

Benjamin Seibold
Temple University
Philadelphia, Pennsylvania
seibold@temple.edu

Raphael Stern
University of Illinois
Urbana-Champaign, Illinois
stern5@illinois.edu

Shumo Cui
Temple University
Philadelphia, Pennsylvania
shumo.cui@temple.edu

Jonathan Sprinkle
The University of Arizona
Tucson, Arizona
sprinkjm@email.arizona.edu

Benedetto Piccoli
Rutgers University
Camden, New Jersey
piccoli@camden.rutgers.edu

Daniel B. Work
Vanderbilt University
Nashville, Tennessee
dan.work@vanderbilt.edu

ABSTRACT

In this paper we describe an experience report and field deployment of real-time filtering algorithms used with a robotic vehicle to smooth emergent traffic waves. When smoothing these waves in simulation, a common approach is to implement controllers that utilize space gap, relative velocity and even acceleration from smooth ground truth information, rather than from realistic data. As a result, many results may be limited in their impact when considering the dynamics of the vehicle under control and the discretized nature of the laser data as well as its periodic arrival. Our approach discusses trade-offs in estimation accuracy to provide both distance and velocity estimates, with ground-truth hardware-in-the-loop tests with a robotic car. The contribution of the work enabled an experiment with 21 vehicles, including the robotic car closing the loop at up to 8.0 m/s with the filtered estimates, stressing the importance of an algorithm that can deliver real-time results with acceptable accuracy for the safety of the drivers in the experiment.

CCS CONCEPTS

• **Computer systems organization** → **Robotic control**; **Robotic autonomy**; *Real-time systems*; • **Applied computing** → *Engineering*;

*This research is supported by the National Science Foundation, awards CNS-1446435, 1446690, 1446702, 1446715.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
ICCCPS '19, April 16–18, 2019, Montreal, QC, Canada
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6285-6/19/04...\$15.00
<https://doi.org/10.1145/3302509.3314026>

KEYWORDS

Autonomous vehicles, traffic, connected vehicles, simulation, sensors, real-time applications, digital filter, Cyber-Physical Systems

ACM Reference Format:

Rahul Bhadani, Matthew Bunting, Benjamin Seibold, Raphael Stern, Shumo Cui, Jonathan Sprinkle, Benedetto Piccoli, and Daniel B. Work. 2019. Real-Time Distance Estimation and Filtering of Vehicle Headways for Smoothing of Traffic Waves. In *Proceedings of The 10th ACM/IEEE International Conference on Cyber-Physical Systems (ICCCPS 2019)*. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 INTRODUCTION

As Connected Autonomous Vehicles (CAVs) enter the marketplace and traffic flow, controllers are being designed to implement myriad behaviors. Since penetrations of up to 100% of the traffic flow are not expected for some years, it is necessary for sensors to acquire state data where ground-truths (or near enough) are unavailable.

A particular application we have studied is emergent traffic waves, which occur naturally in human-driven traffic due to driver dynamics composed with the dynamics of the physical systems (vehicles) they are controlling. The work of Sugiyama et al. [25] demonstrated these phantom traffic jams (also called jamitons) using full-scale vehicles with drivers in dramatic fashion, wherein a single-lane flow of 22 cars driving on a ring road exhibited traffic waves although there were no infrastructure bottlenecks.

In [24] we were the first to experimentally show that it was possible to dampen these traffic waves by controlling only one of the vehicles in the flow. The work required significant experimental innovation in order to overcome what were perceived as simple engineering problems. During the execution and deployment of this large-scale cyber-physical systems (CPS) experiment we considered many trade-offs that we believe are of use to the community.

The real-time considerations of the work are significant: the control algorithms depend on sensor data when determining the velocity of the vehicle, and estimating the relative velocity of others. As the technology for LiDAR and other sensors matures and prices

fall, algorithms to interpret these data in real-time will be used by vehicles (or by infrastructure) in order to aid in the control. The frequency at which reliable data are available, coupled with the real-time constraints of the physical platform, constrains algorithms in a non-intuitive way.

The principal inputs for the controller are the distance and relative velocity of the immediately preceding vehicle, with expected use at velocities of up to 8.0 m/s (average driving speed in stop-and-go traffic). As we will show in this paper, the estimation of relative velocity at the high end of this range is susceptible to errors that are on the order of decision points for the need for braking: given the objective of traffic smoothing, we must design these controllers and companion sensor filters in a way that does not exacerbate the problem of unnecessary braking.

Contribution

This work describes the real-time design, analysis, and constraints of the CPS controller and sensor used for the experiment [24] on dissipation of emergent traffic waves. We describe how the error qualities of high-rate distance sensors require filtering that is cognizant of the controller in use. In short, the paper goes beyond existing results in object-tracking using laser sensors due to the complexity of the cyber-physical scenario: a full-scale passenger vehicle moving at up to 8.0 m/s alongside other vehicles.

The results include derivation and trade-off discussions on how to filter these data, depending on the frequency and dynamics of the controller, plant, and sampling rate of sensors. The theoretical results of these trade-offs are used to develop real-time implementations of the filtering algorithms, which are informed by the controller and experiment. Experimental results are given (including initial experiments that determined these algorithms were required) and comparison of the results to offline estimates of the space gap (also referred to as a headway or range), and velocity of each vehicle is compared to the real-time estimates. Discussion of the upper bounds of the sensors and control inputs are provided, which provide insights into when the controllers could be used effectively with these algorithms.

The insights provided in the paper will be valuable to developers of CAV algorithms, networking models, and infrastructure designers, and they provide an important context of an experiment that required real-time sensing and control, which resulted in a significant scientific and engineering result.

2 BACKGROUND

Our approach to the use of sensors is constrained due to the real-time requirements of the application, and the potential for dangerous outcomes if the algorithms fail to provide valid answers. Note in particular that we depart from a traditional use of the laser rangefinder in robotics to build an obstacle grid or similar navigation map: instead, the goal is to use it to close the control loop in real-time with a companion controller.

Although a typical distance resolution of many distance scanners is 0.1 m, such a resolution may be unsuitable for use in a cyber-physical system's controller if the controller's dynamics are such that the system is susceptible to noise, or that the software delays would result in instability. In this section we examine how distance

sensors have been used in car-like robot applications, such as target tracking, obstacle-avoidance, etc. in a real-time setting, with a goal of understanding the context of the contribution of this work.

Real-time object tracking for stationary objects has been explored based on Kalman and particle filters, as well as through object classification. The results in those works provide an answer in real-time but with limitations for our intended purpose: (i) the obstacles are moving, but not the laser sensor [2], (ii) the obstacles are fairly uniform (persons) and typically occlude in a uniform way [1], and (iii) the velocity of each detected object is utilized for disambiguation and trajectory tracking of that object, not as an input to a controller [12, 17]. On the contrary, autonomous driving requires temporal tracking because of continuous change of background information and movement of sensors relative to the target.

During the DARPA urban challenge, several research teams including Stanford [18] and MIT [15] developed object tracking models to utilize a number of LiDAR sensors for obstacle detection and tracking of moving objects.

As described in [18] and [20], the Stanford team developed a general purpose tracking algorithm based on geometric and dynamic properties of a vehicle to be tracked. Recent improvements have used a learning-based model to estimate the pose of the vehicles to be tracked, though as is typical in machine learning these results are dependent on the training dataset [6, 14, 26]. For the results presented in this paper, we relax the need to estimate the pose of the vehicle in front as such estimates generally inform high-level driving models that switch between driving modes, rather than the velocity controllers described in this work.

In our experiment, we applied a real-time controller called the *Followerstopper* [4, 24], while driving on a ring road with a minor offset to another car for safety. We define safety metrics and discuss the measurement model. Using them, we formalize safety aspects of vehicle-following which are used in developing an understanding of the filter-design process and its effect on controller design. At the end we demonstrate, using a very simple filter, how sensor characteristics and filter-design process affect safety metrics.

3 FORMULATION

In-motion cyber-physical systems require a trade-off between delay (introduced by computational elements) and the accuracy that those computational elements can provide. [3, 11]. We assume that the control of our moving cyber-physical system is affected by a delay of δ seconds, due to delay from filtering, computation delay, actuation delay, actuator, and plant dynamics, among others. Some of these delays can be mitigated through design, while others are implicit to the system. In addition, there are constraints from the overall system design goals: to dampen traffic waves.

In this section, we first describe the model dynamics and the two-vehicle scenario considered while designing the controller with the two-fold goals of dissipating traffic waves as well as maintaining a safe distance to the vehicle ahead.

3.1 Controller design and constraints

The *Followerstopper* controller is briefly reintroduced here for completeness and to provide context for the contribution of the real-time

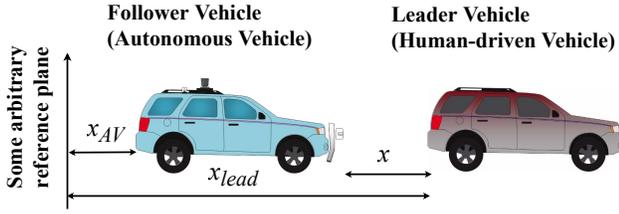


Figure 1: Configuration of the vehicles under consideration, where the AV follows a leader vehicle under human control.

requirements of the CPS. A detailed examination of the design can be found in [4, 24].

A typical scenario in mixed traffic where an AV follows a human-driven vehicle referred to as the leader vehicle (alternatively referred to as the preceding vehicle or the target) is shown in Figure 1.

Let the AV be described by its state (x_{AV}, v_{AV}) where x_{AV} is the position of the AV on the path, and v_{AV} is the velocity of the vehicle along the path. Similarly, the leader vehicle is described by its state (x_{lead}, v_{lead}) where x_{lead} is the position of the leader vehicle, and v_{lead} is the velocity of the leader vehicle. The space gap (distance between the front bumper of the follower vehicle and the rear bumper of the leader vehicle) between two vehicles under consideration is

$$x = x_{lead} - x_{AV} - L. \quad (1)$$

Here L is the length of each vehicle, which we assume to be of the same length. Similarly, the relative velocity of the follower AV with respect to the leader is

$$\dot{x} = v_{lead} - v_{AV}. \quad (2)$$

Through our controller design, we would like to control the velocity of the AV, v_{AV} , through the control input u as shown in Figure 2.

The controller commands the AV's desired velocity, based on the space gap x , the leader vehicle velocity v_{lead} , the AV velocity v_{AV} , and a reference velocity r :

$$u = F(x, v_{lead}, v_{AV}, r). \quad (3)$$

The reference velocity is determined from a nominal controller that sets r to achieve traffic wave smoothing, but which may not consider safety requirements of the AV. For the work presented in this paper, a user input is used to generate the reference velocity r desired for wave smoothing. Clearly the control input to the AV determined by F must prioritize safety constraints, and follow the reference velocity only when it can do so safely.

The Followerstopper controller is designed to achieve the dual goals of wave smoothing while maintaining a safe distance. It defines F as a piecewise controller (having 4 modes) that provides the control input u as a weighted average of the leader vehicle's velocity v_{lead} and the reference velocity r based on the current state of the AV, as determined from the $x\dot{x}$ -phasespace diagram (also known as a state-space diagram or phase-portrait, see [13, Chapter 3], [8]) as shown in Figure 3.

We now explain how this multi-variate function F is chosen. In doing so, with a slight abuse of notation, we use the symbols x and v_{lead} , v_{AV} , and r to denote the arguments of this function.

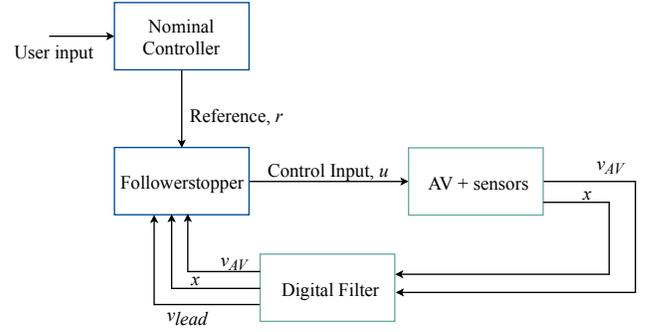


Figure 2: A schematic diagram of the Followerstopper controller and other components used for AV's velocity control. Note that some outputs of the Digital Filter represent estimated signals.

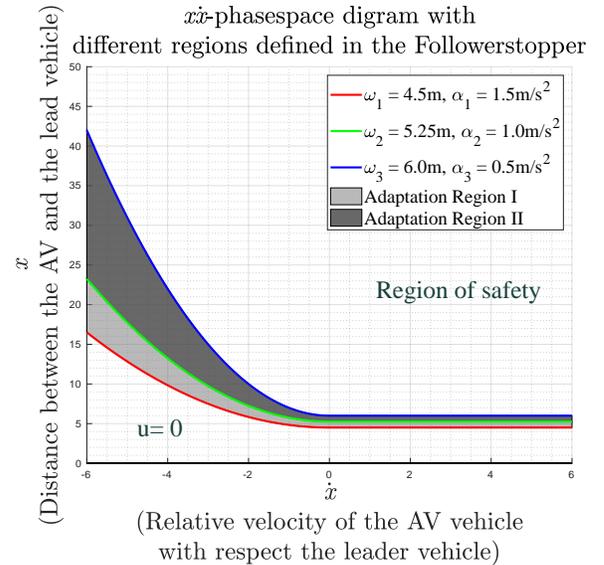


Figure 3: $x\dot{x}$ -phasespace with quadratic bands that describe the Followerstopper controller.

The reason for this notation is that, when used in practice, all of the inputs may be functions of time, which induces the control output u to be a function of time as well. Given parameters α_j and ω_j , $j = 1, 2, 3$, three curves $\xi_j(\dot{x})$ are defined in the phase plane as follows:

$$\xi_j(\dot{x}) = \omega_j + \frac{1}{2\alpha_j}(\dot{x}^*)^2 \text{ for } j = 1, 2, 3. \quad (4)$$

where $\dot{x}^* = \min(\dot{x}, 0)$ is the negative arm of the velocity difference, i.e. the case of the AV falling behind ($\dot{x} > 0$) is treated just like the case of $v_{AV} = v_{lead}$. Note that the mode separators in the function F are chosen as parabolas in the $x\dot{x}$ -phase plane. The motivation for this choice is that, in the situation that the leader vehicle drives at a constant velocity, a constant deceleration of the

AV will result in it describing precisely such parabolas in the $x\dot{x}$ -phase plane. Through the choices of parameters α_j , ω_j , the curves $\xi_j(\dot{x})$ in the phase plane, are defined by (4).

With the above discussion, and noting that $\xi_j = \xi_j(\dot{x})$ in a mild abuse of notation for brevity, the piecewise velocity controller is:

$$u = \begin{cases} 0, & \text{if } x \leq \xi_1 \\ v \frac{x - \xi_1}{\xi_2 - \xi_1}, & \text{if } \xi_1 < x \leq \xi_2 \\ v + (r - v) \frac{x - \xi_2}{\xi_3 - \xi_2}, & \text{if } \xi_2 < x \leq \xi_3 \\ r, & \text{if } \xi_3 < x \end{cases} \quad (5)$$

where $v = \min(\max(v_{lead}, 0), r)$, which is the leader vehicle's velocity (if positive) or the nominal velocity r , whichever is smaller. We call attention to the fact that, when used in a real-time application, all signals appearing in (5) become functions of time.

Some intuition regarding behavior in these modes will emerge from brief discussion. The most straightforward is the final mode, with $\xi_3 < x$, where the control input to the AV is the reference velocity.

Similarly, the first mode, with $x \leq \xi_1$ describes the case where the AV is commanded $u = 0$. Recalling that the function $\xi_1 = \xi_1(\dot{x})$ is defined by a maximum deceleration parameter (α_1) which should fully stop at least $\omega_1 m$ behind the leader vehicle when executed, and will obey the dynamics of the physical system in doing so.

Since frequent maximum deceleration would be unlikely to dampen traffic waves, additional modes are defined which utilize different deceleration parameters (e.g., α_2 and α_3) in Adaptation Regions I and II, respectively. The controller thus will transition between modes as the distance and relative velocity cross the velocity-dependent curves ξ_j defined in the phase plane.

The values of ω_j and α_j are selected based on the desired behavior; for the purpose of the experiment presented in this paper, the chosen values were based on data collected from a pilot experiment to observe human driving (with no vehicles under autonomous control). There were no attempts made to optimize these values, though this may be of interest for future research.

The values used of $(\omega_3, \alpha_3) = (6.0 m, 0.5 m/s^2)$ and the values of $(\omega_1, \alpha_1) = (4.5 m, 1.5 m/s^2)$: this is when the controller fully overrides the reference input to prevent collision, thereby stopping with a maximum deceleration. (ω_2, α_2) is the average of the (ω_1, α_1) and (ω_3, α_3) to provide a smoother transition whenever there is requirement to switch modes.

The overall constraints of the experiment that motivated this controller are: (i) the need to avoid frequent deceleration—as this would potentially cause traffic waves for following vehicles; and (ii) the need to follow a reference velocity as defined by the leader vehicle, as a function of the velocity and relative distance of the vehicles. The accuracy of estimating the distance between vehicles, and the rate at which realistic sensors plus computing time can perform those estimates, motivate the exploration of the problem formation.

3.2 Problem formulation

In the context of an AV, one safety metric is the expected separation distance, d_{min} :

$$d_{min} = \theta_d(v_{AV}, v_{lead}, \delta, x), \quad (6)$$

which is a function of the velocity of the vehicles (v_{lead} , v_{AV}), the system delay δ , and the current distance x that is the space gap. [9, 27]. The system delay is the sum of all delays including delay caused by the nominal controller, the Followerstopper, actuation delay by the AV and delay introduced due to the filtering process, plus any other delay due to some unknown cause. However, we have the most control during design over the delay due the filtering process, which will be clear in later sections.

Let ε be the maximum mean squared error of estimated velocity of the leader vehicle that is acceptable for the traffic waves experiment, which should be calculated based on the velocity distribution \mathcal{V} , at which traffic would be moving. One such velocity distribution, captured during a pilot experiment, is shown in Figure 4 and represents speed variability in oscillatory traffic. The function θ_v should map from velocity distribution \mathcal{V} to velocity-estimate-error:

$$\varepsilon = \theta_v(\mathcal{V}) \quad (7)$$

which we later quantify in terms of mean-squared-error (MSE).

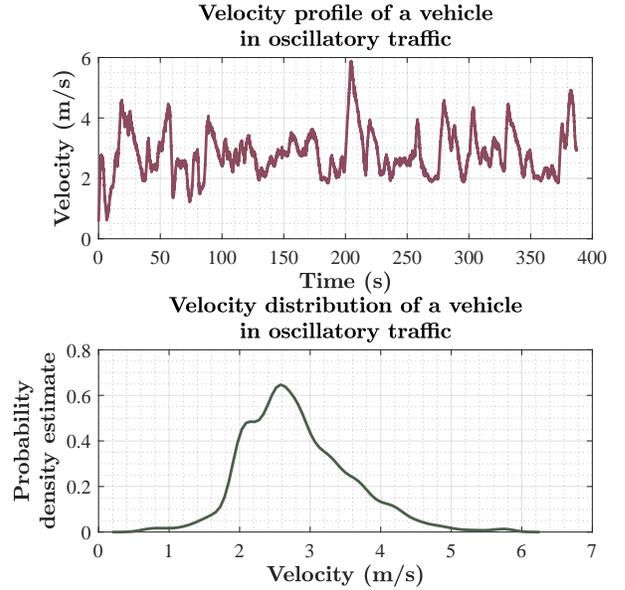


Figure 4: A representative velocity profile of a vehicle and its velocity distribution in oscillatory traffic.

Let filters f_d and f_v estimate the distance and velocity to the leader vehicle, respectively, with sampling frequency F_s .

Problem statement

- (i) Derive or design functions θ_v , θ_d
- (ii) Design f_d , f_v to meet constraints for ε with a given filter frequency
- (iii) With given f_d , f_v , apply θ_d to compare whether the d_{min} exceeds space gaps seen in the traffic flow experiment
- (iv) Evaluate (iii) with various parameters, to explore impact on sensors of other frequencies and accuracies

Problem (i) is solved in Sec. 4.2 and Sec. 4.3; (ii) is addressed in Sec. 5.2; (iii) in Sec. 5.4; and (iv) in Sec. 6.1.

4 APPROACH

A robust control law to follow a human-driven vehicle requires estimation of the relative velocity of the two vehicles within an acceptable signal to noise ratio. However, in realistic terrain and within urban surroundings a laser sensor frequently returns data with discontinuities. In such situations, it is desirable to devise an algorithm so that the time derivative of distance does not present jittery behavior due to discontinuities in the data (as described, f_d). In this section, we detail our approach within underlying assumptions of motion with respect to stop-and-go vehicle traffic.

For the controller in (5), which utilizes relative velocity as part of its feedback law, noise in relative distance or velocity can be amplified without proper real-time filtering. We use the traces collected from the hardware in use to characterize the sensor noise and develop an understanding of filter design technique useful for accomplishing the goal of providing a smooth velocity estimate and hence a stable control input to the AV for the vehicle following.

4.1 Measurement model

Let the sampling rate of the sensor be F_s , and v_{AV} be the velocity of an autonomous vehicle in the traffic flow which is sampled at a rate F_{AV} . The AV is equipped with a sensor (e.g. LiDAR, Camera, Sonar, Radar, etc.) on the vehicle front, returning the relative distance between the AV and the target (e.g. a wall, a moving vehicle, a pedestrian, etc.). We assume the velocity of the AV can be accurately measured. We also assume that the target is not capable of communicating with the AV; hence the target's relative distance is detected only by the sensor mounted on the AV. The distance information is used in estimating the relative velocity of the target and in turn its absolute velocity. Let \hat{x} be the measured relative velocity which is a noisy estimate of the true relative velocity \dot{x} . Further, let the actual velocity of the target be v_{lead} , and \hat{v}_{lead} be the estimated velocity of the target, that is,

$$\hat{v}_{lead} = v_{AV} + \hat{x} \quad (8)$$

Noise present in the sensor and methods used to estimate the relative velocity results in noisy estimate \hat{v}_{lead} which is different from ground truth v_{lead} . As a result, we may see abrupt changes such as short-term fluctuations and discrete jumps in the estimated samples. We are required to use noise removal filters to get smooth data for good control behavior. However, there is always a penalty in terms of delay when using a digital filter. We assume that the filtering algorithm introduces some delay δ_f . The filter delay δ_f is crucial for designing the filter according to the latency constraints, and therefore designing the control algorithm. The measurement noise can be reduced by the virtue of a more complicated filter at the expense of introducing larger delay (e.g. a Kalman Filter). Considering that the digital filter used in the process operates in time-domain, an estimated relative velocity will be obtained at the same sampling rate of the sensor, i.e. F_s . Thus, the number of samples that are delayed due to the filtering process is

$$N = \delta_f F_s \quad (9)$$

As the velocity of the AV is sampled at rate F_{AV} , the estimated velocity of the target is obtained using (8) at a rate of

$$F_v = \max(F_s, F_{AV}). \quad (10)$$

Note that since two quantities sampled at different frequencies are being added, an estimate of the target will be available at the maximum of the rate of two quantities.

As a result, the controller receives the estimated velocity of the target delayed by $\frac{N}{F_v}$ seconds. If we introduce the fixed delay δ_r that can be set during design to take into account friction, actuator dynamics, controller update rate, etc., then the total delay will be

$$\delta = \frac{N}{F_v} + \delta_r = \frac{\delta_f F_s}{F_v} + \delta_r \quad (11)$$

Now we have developed a measurement model, we are ready to formalize a safety model for vehicle-following in a scenario where the AV lacks ground truth about the target and must estimate the state of the target.

4.2 Design of θ_v

The interface for $\varepsilon = \theta_v(\mathcal{V})$ includes as an input only the velocity distribution, \mathcal{V} . Thus the design of θ_v should be based on the application in question, and an additional structure may be derived from the controller, thus taking into account the dynamics of the plant in (5). From that structure (visually depicted in Figure 3) small perturbations in velocity could result in the controller leaving the region of safety. For example, instantaneous errors of the relative velocity of -2 m/s would result in braking for any distance less than 10 m, which could result in a hard braking event for following vehicles. To see this in Figure 3, note that the intersection point $(x, \dot{x}) = (10 \text{ m}, -2 \text{ m/s})$ is in the region of safety for the controller; however, if that value were to suddenly be $(10 \text{ m}, -4 \text{ m/s})$ due to an instantaneous error, then the vehicle would brake at the maximum deceleration rate (see $(10 \text{ m}, -4 \text{ m/s})$ on the figure).

In future work, we may define more sophisticated models, though here it is sufficient to examine the inverse relationship of the expectation E of the velocity distribution, and the frequency at which updates may be made.

$$\varepsilon \propto \min\left(\frac{1}{E(\mathcal{V})}, \frac{|\dot{v}_{min}|}{F_s}\right) \quad (12)$$

For large speeds, accuracy is necessary (the first argument in (12)), though at any speed more accuracy is needed with the large sampling frequency (the second argument in (12)), in order to prevent burning out controllers. For the latter, this is tempered by the ability of the system to actually stop, so at very low speeds and low frequencies, an erroneous value can be corrected in the next timestep, if $|\dot{v}_{min}|$ is sufficiently large.

4.3 Design of θ_d

In the experiment, vehicles are expected to drive at the desired speed, unless they must slow down for the vehicle in front; in this case, the current velocity of each vehicle, the space gap, delays, and possible deceleration rates of the vehicles are very important.

Consider the case with the AV traveling at a velocity v_{AV} following a target driving at the speed v_{lead} , separated by a space gap x . We introduce a safety metric θ_d as mentioned in Sec. 3.2 referred to as the expected separation distance d_{min} . The expected separation distance indicates if a collision is expected to occur based on the leader velocity, the follower velocity, deceleration rates, and

the system delay of the AV. The metric is computed as:

$$d_{min} = x + (v_{lead} - v_{AV}) \left(\frac{\delta_f F_s}{F_v} + \delta_r \right) + \frac{1}{2} (a_{lead} - a_{AV}) \left(\frac{\delta_f F_s}{F_v} + \delta_r \right)^2, \quad (13)$$

with $d_{min} > 0$ indicating expected avoidance of collision. The derivation is provided in the Appendix. Note that the parameters a_{lead} and a_{AV} control the degree of conservativeness of d_{min} . For example, a stringent metric can be derived setting a_{lead} at an acceleration rate corresponding to emergency braking, while assuming the acceleration rate of the AV is corresponds to the maximum acceleration rate of the vehicle. A less conservative measure can be designed by setting a modest deceleration rate of the leader and assuming the follower maintains a constant speed during the delay. Other measures (e.g., do not encroach on a minimum buffer behind the leader vehicle) could easily be designed as well.

5 RESULTS

5.1 Sensor configuration and data acquisition

A typical laser rangefinder operates using the principle of the time of laser light pulse. A pulsed laser beam is emitted and—if reflected—the time between transmission and reception is proportional to the distance between the laser and the object. We used a SICK LMS 291 laser rangefinder attached to the front bumper of the University of Arizona’s CAT Vehicle, our autonomous vehicle (AV), for high-frequency distance measurement of the lead vehicle. The unit was configured in RS422 mode which enabled us to acquire data at 75Hz at a resolution of 1° (0.0175 radians), with distance estimates from [0.1, 81] meters. The LiDAR unit was mounted in level with the vehicle frame, just above the front bumper of the CAT Vehicle. The device scans in a counterclockwise motion by sending laser pulses and measuring the time difference of a return of that pulse. The first angle scanned is at $-\frac{\pi}{2} + \frac{\pi}{360}$ and the sensor measures a total of 180 points until it reaches the angle $\frac{\pi}{2} - \frac{\pi}{360}$. The resulting semicircular scan is an array of values that represent the distance of the nearest object at that angle, or the value 81.0 m if no return is received. The entire scan is stamped with the same time, which is obtained from the ROS master node [5, 21].

5.2 Sensor characterization, and design of f_d , f_v

We begin with a brief description of how data are acquired for error characterizations of the sensor in use. The AV is stationed such that the sensor is approximately 17.9 m away from a stationary target. The vehicle is then driven toward, and then away from, a stationary target in order to calibrate filtering algorithms used to determine distance and velocity of the target in the front. Use of a stationary target ensures ground truth velocity of the target to be known (namely, 0 m/s). We point out that characterizing sensor noise with a moving-target is significantly harder without sharing ground truth velocity of the target (in the absence of inter-vehicle communication).

To formalize the characterization, let μ_x be the expected value of the distance and σ_x^2 be the sample variance of the distance measured. For nearly 80 s when the AV, sensor and the target were stationary,

we collected sample data to characterize stochastic noise. The mean of the signal during this period was found to be $\mu_x = 17.91288$ m with standard deviation of $\sigma_x = 0.01439$ m.

Assuming the expected value of the signal during this period as the true value, we calculated error in the distance estimation shown as a histogram in Figure 5. A portion of the raw sensor data is shown in the Appendix in Figure 13.

Consider estimation of velocity through a first order finite difference method

$$\dot{x}_k = (x_k - x_{k-1})F_s \quad (14)$$

where k denotes k -th sample, provides $\mu_{\dot{x}} = 0.00084268$ m/s and $\sigma_{\dot{x}} = 1.6847$ m/s of the stationary object, at $F_s = 75$ Hz. It is common to experience increased noise in this approach as the differentiation acts as a high pass filter. As a result, low frequency motion data in the passband is attenuated while stochastic noise such as white noise in the higher frequencies is amplified [19]. This results in lower signal-to-noise ratio (SNR). The corresponding histogram can be seen in Figure 6. The mean squared error (MSE) (or sample variance $\sigma_{\dot{x}}^2$)¹ obtained from this method is 2.8377 (m/s)². As shown in the Appendix, Figure 13 shows short term fluctuation in the distance data due to sensor noise, which will typically result in significant variance in the velocity estimation.

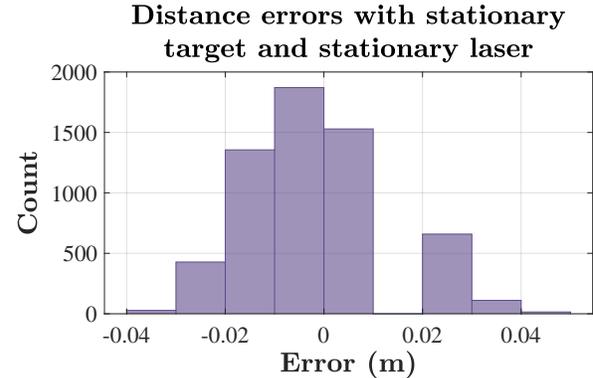


Figure 5: Histograms for errors in distance with expected value of the signal as true value.

As the laser sensor moves, the velocities may exhibit additional vibrations due to the dynamics of the moving vehicle, the detection of different features of the object in non-flat terrain, or other various reasons. An additional concern is discrete jumps in distance, based on detecting features of the lead vehicle that are on the order of tens of centimeters apart. For example, the rear bumper at time t_k and then at time t_{k+1} detecting the rear tire. Such a discrete jump of a few tens of cm in a single time-step could result in instantaneous velocity estimates of up to 20 m/s—such an estimate could result in unnecessary braking that might cause damage or serious injury. As an example, consider the velocity estimates shown in Figure 7, which exhibits chattering of the velocity at 123 s and 132 s. The presence of these high-frequency signals could drive the controller

¹Statistically speaking, MSE and sample variance σ^2 are similar. See [7] for more details.

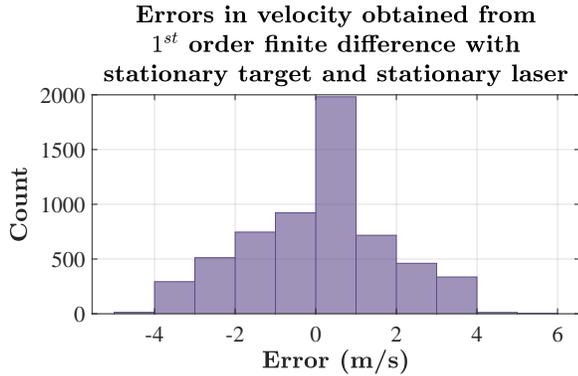


Figure 6: Histograms for errors in the velocity estimate from (14) when target and sensor are stationary. MSE of relative velocity is $2.8377 (m/s)^2$

unstable or cause riding discomfort. Such jumps must be detected, but unnecessary delay in detecting (or resuming filtering) could have detrimental effects on the ability of the controller to dampen traffic waves.

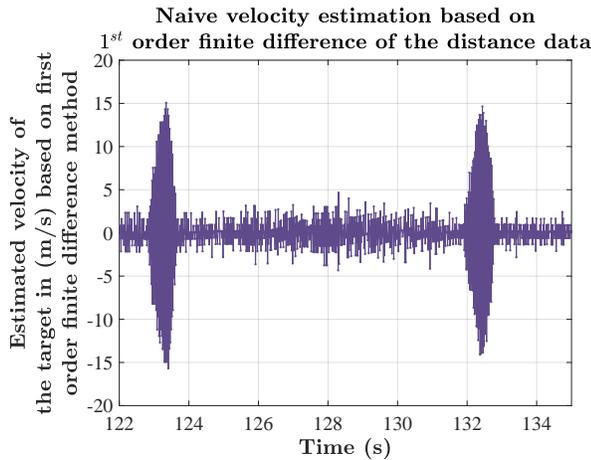


Figure 7: Estimated relative velocity using (8) and (14) exhibits chattering of velocity at times 123 s and 132 s, where the sensor returns discontinuous data points due to non-flat terrain. Note that we are characterizing error and noise with a stationary target, hence we expect signals to be close to zero.

This leads to the need to filter the estimated velocity before it can be used for any control algorithm. Real-time considerations require a fixed horizon for the filter, with a width for the filter based on the system response time. Thus leaving some of the filter parameters as variables, the structure of the filter is to be designed in such a way that control loop used in the vehicle control algorithm remain faster than the rate of filtered signal.

For noise smoothing operations, we are interested in a causal filter with real-time properties. There are myriad objectives one can aim for while designing a filter such as maximizing SNR, minimizing

signal distortion or minimizing filter delay. For our purpose, we were interested in a filter to minimize delay while maintaining the necessary accuracy ϵ which we quantify in terms of σ_x^2 . There are two main types of digital filter we consider: *finite impulse response* (FIR) filters and *infinite impulse response* (IIR) filter. For ease of design and intuitive realization we used a moving average filter, which is an FIR filter. Moving average filters have low-pass filter properties and offer constant group delay because of their linear phase. A moving average filter estimate of \dot{x} is

$$\hat{x}_k = \sum_{m=k-N}^k \frac{\dot{x}_m - \dot{x}_{m-1}}{N}, \quad 2 < N < k \quad (15)$$

We applied a moving average filter to the velocity estimation obtained from (14). To determine the order of the moving average filter we utilized residual analysis and calculated the MSE of the filtered data by varying the window size (i.e. the order for the moving average filter), as shown in Figure 8. MSE is $\approx 0.01(m/s)^2$ for the window size of 10 or greater: however, a larger window size comes with the price of larger delay and risk of oversmoothing. Note that there is negligible improvement in MSE with window size larger than 20.

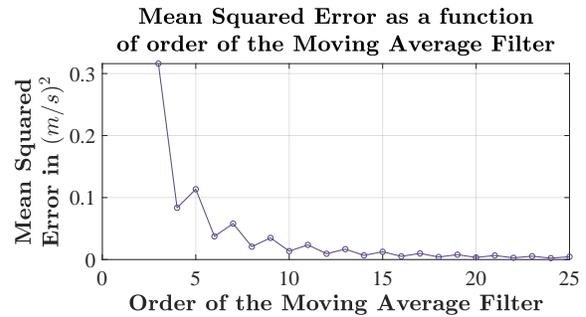


Figure 8: We see that MSE approaches zero: $0.013623(m/s)^2$ for window size of 10 and $0.003391(m/s)^2$ for window size of 20. However higher order introduces a larger delay.

Suppose we select a window size of 20, introducing the delay of $N = 20/2 = 10$ samples since delay introduced due to moving average filter is half its window size. As discussed in Sec. 5.1, the sampling rate of our sensor is $F_s = 75 \text{ Hz}$, hence the filter causes the delay of $\delta_f = 10/75 = 0.133 \text{ s}$. We can further examine the performance of the filter when the AV is in motion. Figure 9 compares the filter in (15) with the first order approach of (14). The top graph is the residual of the relative velocity estimate calculated as MSE from (14); the lower graph is the residual of filtered velocity using (15) with $N = 20$. As the target remains stationary during this period and the AV with sensor mounted on it moves relative to the target, the relative velocity of the target with respect to the AV is equal to but opposite in sign of the AV velocity. In this way we are able to calculate MSE by considering the fact that negative of the AV velocity is the true value of the stationary target's relative velocity. The MSE for unfiltered relative velocity of the target during motion was $2.704824 (m/s)^2$; after filtering it was reduced to $0.0087039 (m/s)^2$, which is the same order as seen while stationary.

Hence we see that a better filter design provides a more reliable estimate of the relative velocity.

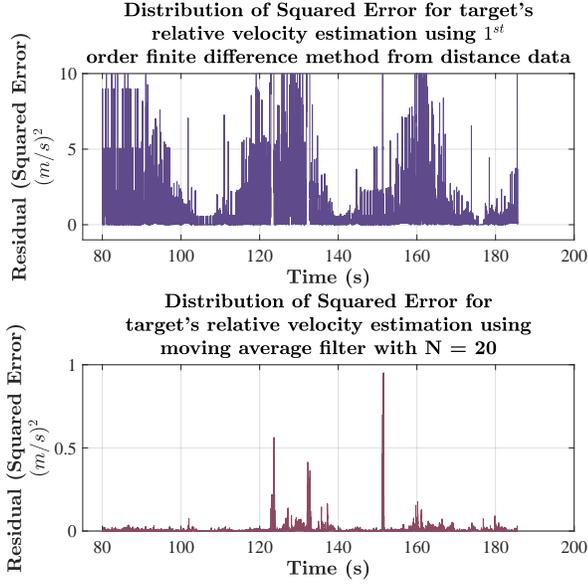


Figure 9: We observe that filtered relative velocity of the target matches fairly well with the AV's velocity. Notice the y-scale of lower plot.

5.3 Simulation of filter designs

We performed a simulation of (13) described in Sec. 4.3 using MATLAB and the Robotic System Toolbox. Since we can only estimate the speed of the target, v_{lead} in (13) will be replaced by an estimate \hat{v}_{lead} . We generated a synthetic dataset for the velocity profile of the AV and distance estimation of the target based on an empirical distribution of realistic datasets gathered using the CAT Vehicle. We performed simulations for (14), and (15) with both $N = 5$ and $N = 20$, and for each of these we checked whether $d_{min} > 0$. Note that due to the noisy estimate of the velocity of the leader vehicle, d_{min} violated the constraint of $d_{min} > 0$ which could result in a collision. The result is summarized in Figure 10.

Although Sec. 4.3 formalizes an expected separation distance metric for autonomous vehicle operation, we would still require a supervisory controller in addition to the proper filter design to provide safe behavior in autonomous operation. Nevertheless, this provides a suitable framework for designing a supervisory controller for enforcing safety.

5.4 Application of filtering in traffic wave dissipation

From the analysis, we knew to choose a sensor with relatively fast sampling rate to return the estimated results to the velocity controller for the dissipation of traffic-waves. Analysis of signal traces obtained from the ring road experiment proves our hypothesis that a sensor with faster sampling provides smoother estimation with relatively little lag or delay. To validate the overall experimental

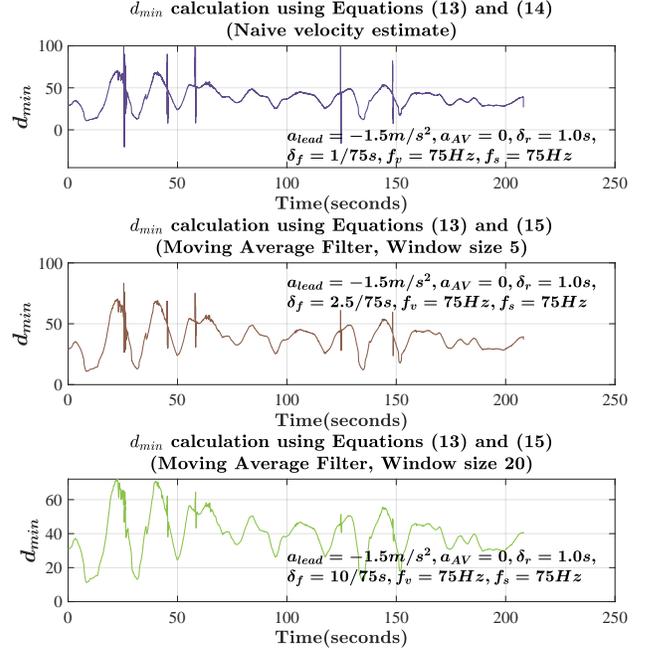


Figure 10: Simulation study of the effect of filter design using equations (13), (14) and (15), with acceleration values a_{lead} , a_{AV} representing potential reactions during steady-state driving. Some design choices provide poor ϵ and lead to violation of the expected separation distance, i.e. $d_{min} < 0$.

results, we placed a 360° camera at the center of the ring; in addition, there was a LiDAR mounted on the front bumper of the AV. The sampling rate of 360° camera is 30 Hz; Had we used such a camera for vehicle space gap and relative velocity estimation, the AV would not have been able to close the gap on time, resulting in either a collision or unexpectedly large space gap.

The application of a moving average filter from (14) with $N = 20$, using information from camera data, as well as from LiDAR data obtained from the experiment, is shown in Figure 11. It should be noted that the shapes of the signal obtained from two different sensors differ in the amplitude of peaks and amount of noise. This is attributed to the varying placement of sensors and their characteristics but the overall trend is similar. The camera introduces a delay of $\delta_f = \frac{10}{30} s = 0.333 s$, whereas LiDAR introduces a delay of $\delta_f = \frac{10}{75} s = 0.133 s$. Without ground truth data, we are unable to validate ϵ , but our empirical result, in this case, is representative of the safety framework discussed earlier. However, the velocity controller discussed in Sec. 3.1 satisfied the safety metric when driven autonomously during the ring road experiment as shown in Figure 12.

6 CONCLUSION

In this paper, we presented a real-time distance and velocity estimation method for following a leader vehicle in urban stop-and-go traffic, for the purpose of dampening traffic waves. We validated estimation techniques according to safety constraints, which were

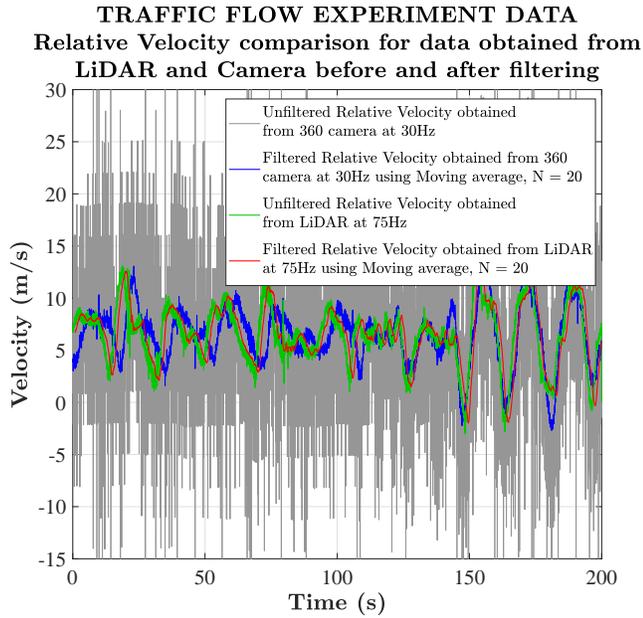


Figure 11: Use of moving average filter of order $N = 20$ for sensors with different frequency: Use of slower sensor (camera) produces data with low SNR. Results show poor smoothing in camera data running at 30Hz. The camera introduces more delay during the filtering processes as well, as we observe that the velocity curve for filtered camera data is time-shifted to the right with respect to the one obtained from LiDAR. For clarity, we have only compared two sensors for only window size i.e. $N = 20$, but comparative results are similar for $N = 5$.

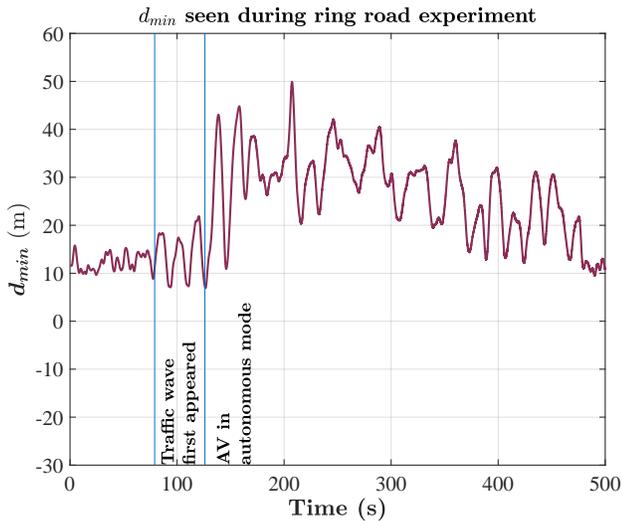


Figure 12: Ring road experiment results: d_{min} calculated using (13) and (15). During the period of autonomous control, we see no violations with respect to d_{min} .

jointly considered with accuracy and real-time metrics for the controller in use. As a result, our method gives insights into the required frequency of sensors in use, in order to operate at velocities at which stop-and-go traffic is expected (in this case, 8 m/s). The work gives a means by which designers can consider how smooth the estimation of the leader vehicle’s velocity must be, as a function of the tolerance for error in velocity estimation that is driven by the vehicle controller. We emphasize that the sensor characterization discussed here is in the domain of stop-and-go traffic, but a similar kind of approach may be taken for high-speed scenarios.

In our discussion, we find that output rate of measurement devices, desired accuracy, and choices of sensors are factors that should be given consideration by AV application designers and engineers when considering how vehicles may be used to implement low-level control that could positively impact the larger traffic system, through control of vehicles in the flow. This may ensure a high degree of safety without sacrificing efficiency with conservative design choices, such as driving large safe distances. Our discussion here may lead to developing a closed-form mathematical model establishing a relationship among safety metrics, sensor frequency, and desired accuracy, thus enabling more inexpensive sensors (such as cameras) to be used, rather than laser scanners or radars, for such control.

6.1 Future work with other sensors and speeds

The problem may be extended to other sensing frequencies and modalities, though there are limitations. As an example, our vehicle-following solution would differ in the context of new radar technology. With FMCW radar [10, 16, 23], it is possible to directly measure the distance and velocity of the target simultaneously using the Doppler effect, so using distance data to estimate velocity is not required. However, filtering of the velocity might be required based on measurement noise and interference caused by other radio services (24 GHz) or due to signal attenuation (77 GHz) [23]. For $f_c = 24 \text{ GHz}$ FMCW Radar, if chirp frequency is $f_p = 5 \text{ KHz}$ (i.e. period 0.0002 s), then the maximum unambiguous velocity of the target is given by²

$$v_{max} = \frac{(3e8)f_p}{4f_c} = \frac{(3e8)(5e3)}{(4)(24e9)} = 15.625 \text{ m/s} \quad (16)$$

Hence, we see that the design of velocity control is affected by the choice of sensors in use. As an example, Bosch LRR3 77 GHz Radar offers accuracy of 0.1 m for distance estimation and 0.12 m/s for velocity estimation at the sampling rate of 12.5 Hz³.

With improved accuracy, but lower frequency, we expect that a similar kind of digital filter discussed in this paper may have a similar expected separation distance. The situation may worsen if we desire lower MSE for the given application.

With other perception units such as a stereocamera (10 – 15 Hz operation) and the Velodyne 3D LiDAR (10 – 20 Hz operation), the need to filter results will further delay action, especially if the sensor accuracy is similar to that of a line-scanning laser such as

²Interested readers may refer to ‘Fundamentals of Radar Signal Processing’[22] for more details.

³https://www.bosch-engineering.jp/media/jp/pdfs_3/einsatzgebiete_4/produkttdatenblaetter_2/120903_LRR3_EN_V05_final.pdf

that considered in this work. Finally we note that in the experiments presented here, the leader vehicle is always observed in the range of the sensor. At higher speeds and at a larger space gap, the limited range of the sensor will also be an important design concern.

REFERENCES

- [1] Ahmed Abdel-Hadi. 2010. Real-time object tracking using color-based Kalman particle filter. In *Computer Engineering and Systems (ICCES), 2010 International Conference on*. IEEE, 337–341.
- [2] António Almeida, Jorge Almeida, and Rui Araújo. 2005. Real-time tracking of multiple moving objects using particle filters and probabilistic data association. *AUTOMATIKA-ZAGREB* 46, 1/2 (2005), 39.
- [3] Amir Aminifar, Petru Eles, Zebo Peng, Anton Cervin, and Karl-Erik Årzén. 2017. Control-Quality Driven Design of Embedded Control Systems with Stability Guarantees. *IEEE Design & Test* (2017).
- [4] Rahul Bhadani, Benedetto Piccoli, Benjamin Seibold, Jonathan Sprinkle, and Daniel B. Work. 2018. Dissipation of Emergent Traffic Waves in Stop-and-Go Traffic Using a Supervisory Controller. *57th IEEE Conference on Decision and Control* 57 (2018), 3628–3633. <https://doi.org/10.1109/CDC.2018.8619700>
- [5] Rahul Bhadani, Jonathan Sprinkle, and Matthew Bunting. 2018. The CAT Vehicle Testbed: A Simulator with Hardware in the Loop for Autonomous Vehicle Applications. *Proceedings 2nd International Workshop on Safe Control of Autonomous Vehicles (SCAV 2018), Porto, Portugal, 10th April 2018, Electronic Proceedings in Theoretical Computer Science* 269, pp. 32–47 (2018).
- [6] Ashwin Carvalho, Alek Williams, Stéphanie Lefevre, and Francesco Borrelli. 2016. Autonomous cruise control with cut-in target vehicle detection. In *Advanced Vehicle Control: Proceedings of the 13th International Symposium on Advanced Vehicle Control (AVEC'16), September 13-16, 2016, Munich, Germany*. CRC Press, 93.
- [7] George Casella and Roger L Berger. 2002. *Statistical inference*. Vol. 2. Duxbury Pacific Grove, CA.
- [8] P Fancher and Z Bareket. 1994. Evaluating headway control using range versus range-rate relationships. *Vehicle System Dynamics* 23, 1 (1994), 575–596.
- [9] Douglas Gettman and Larry Head. 2003. Surrogate safety measures from traffic simulation models. *Transportation Research Record: Journal of the Transportation Research Board* 1840 (2003), 104–115.
- [10] Jürgen Hasch, Eray Topak, Raik Schnabel, Thomas Zwick, Robert Weigel, and Christian Waldschmidt. 2012. Millimeter-wave technology for automotive radar sensors in the 77 GHz frequency band. *IEEE Transactions on Microwave Theory and Techniques* 60, 3 (2012), 845–860.
- [11] Fei Hu, Yu Lu, Athanasios V Vasilakos, Qi Hao, Rui Ma, Yogendra Patil, Ting Zhang, Jiang Lu, Xin Li, and Neal N Xiong. 2016. Robust cyber-physical systems: concept, models, and implementation. *Future generation computer systems* 56 (2016), 449–475.
- [12] Irene Anindaputri Iswanto and Bin Li. 2017. Visual Object Tracking Based on Mean-shift and Particle-Kalman Filter. *Procedia Computer Science* 116 (2017), 587–595. <https://doi.org/10.1016/j.procs.2017.10.010> Discovery and innovation of computer science technology in artificial intelligence era: The 2nd International Conference on Computer Science and Computational Intelligence (ICCS2017).
- [13] Holger Kantz and Thomas Schreiber. 2004. *Nonlinear time series analysis*. Vol. 7. Cambridge university press.
- [14] Stéphanie Lefevre, Ashwin Carvalho, and Francesco Borrelli. 2016. A learning-based framework for velocity control in autonomous driving. *IEEE Transactions on Automation Science and Engineering* 13, 1 (2016), 32–42.
- [15] John Leonard, Jonathan How, Seth Teller, Mitch Berger, Stefan Campbell, Gaston Fiore, Luke Fletcher, Emilio Frazzoli, Albert Huang, Sertac Karaman, et al. 2008. A perception-driven autonomous urban vehicle. *Journal of Field Robotics* 25, 10 (2008), 727–774.
- [16] Pavlo Molchanov, Shalini Gupta, Kihwan Kim, and Kari Pulli. 2015. Short-range FMCW monopulse radar for hand-gesture sensing. In *Radar Conference (Radar-Con), 2015 IEEE*. IEEE, 1491–1496.
- [17] V. M. Monica and K. G. J. Nigel. 2017. Object tracking based on Kalman filter and gait feature extraction. In *2017 International Conference on Inventive Systems and Control (ICISC)*. 1–5. <https://doi.org/10.1109/ICISC.2017.8068677>
- [18] Michael Montemerlo, Jan Becker, Suhrid Bhat, Hendrik Dahlkamp, Dmitri Dolgov, Scott Ettinger, Dirk Haehnel, Tim Hilden, Gabe Hoffmann, Burkhard Huhneke, et al. 2008. Junior: The stanford entry in the urban challenge. *Journal of field Robotics* 25, 9 (2008), 569–597.
- [19] Alan V Oppenheim and Ronald W Schaffer. 2014. *Discrete-time signal processing*. Pearson Education.
- [20] Anna Petrovskaya and Sebastian Thrun. 2009. Model based vehicle detection and tracking for autonomous urban driving. *Autonomous Robots* 26, 2-3 (2009), 123–139.
- [21] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. 2009. ROS: an open-source Robot Operating System. In *ICRA workshop on open source software*, Vol. 3. Kobe, Japan, 5.
- [22] Mark A Richards. 2005. *Fundamentals of radar signal processing*. Tata McGraw-Hill Education.
- [23] Josef Steinbaeck, Christian Steger, Gerald Holweg, and Norbert Druml. 2017. Next generation radar sensors in automotive sensor fusion systems. In *Sensor Data Fusion: Trends, Solutions, Applications (SDF), 2017*. IEEE, 1–6.
- [24] Raphael E. Stern, Shumo Cui, Maria Laura Delle Monache, Rahul Bhadani, Matt Bunting, Miles Churchill, Nathaniel Hamilton, R’mani Haulcy, Hannah Pohlmann, Fangyu Wu, Benedetto Piccoli, Benjamin Seibold, Jonathan Sprinkle, and Daniel B. Work. 2018. Dissipation of stop-and-go waves via control of autonomous vehicles: Field experiments. *Transportation Research Part C: Emerging Technologies* 89 (2018), 205 – 221. <https://doi.org/10.1016/j.trc.2018.02.005>
- [25] Yuki Sugiyama, Minoru Fukui, Macoto Kikuchi, Katsuya Hasebe, Akihiro Nakayama, Katsuhiko Nishinari, Shin-ichi Tadaki, and Satoshi Yukawa. 2008. Traffic jams without bottlenecks - experimental evidence for the physical mechanism of the formation of a jam. *New journal of physics* 10, 3 (2008), 033001.
- [26] Wenshuo Wang, Chang Liu, and Ding Zhao. 2017. How much data are enough? A statistical approach with case study on longitudinal driving behavior. *IEEE Transactions on Intelligent Vehicles* 2, 2 (2017), 85–98.
- [27] William Young, Amir Sobhani, Michael G Lenné, and Majid Sarvi. 2014. Simulation of safety: A review of the state of the art in road safety simulation modelling. *Accident Analysis & Prevention* 66 (2014), 89–103.

A APPENDIX

A.1 Derivation of d_{min}

Here we derive the expected separation distance d_{min} indicating if a collision is expected to occur between the leader vehicle and the AV during the time delay:

$$d_{min} = x + (v_{lead} - v_{AV}) \left(\frac{\delta_f F_s}{F_v} + \delta_r \right) + \frac{1}{2} (a_{lead} - a_{AV}) \left(\frac{\delta_f F_s}{F_v} + \delta_r \right)^2 \quad (17)$$

recalling x is the current distance between the AV and the leader vehicle (i.e., the space gap), and a_{lead} and a_{AV} are assumed acceleration rates of the leader vehicle and the AV respectively during over the period δ . Note $d_{min} > 0$ indicates expected avoidance of collision.

PROOF. In δ seconds, the leader vehicle travels

$$d_{lead} = v_{lead} \left(\frac{\delta_f F_s}{F_v} + \delta_r \right) + \frac{1}{2} a_{lead} \left(\frac{\delta_f F_s}{F_v} + \delta_r \right)^2, \quad (18)$$

assuming it is initially traveling at v_{lead} and is accelerating at a rate a_{lead} over the time interval.

The distance moved by the AV over the same time interval δ seconds, which is currently traveling at v_{lead} and assuming it follows a constant acceleration a_{lead} is

$$d_{AV} = v_{AV} \left(\frac{\delta_f F_s}{F_v} + \delta_r \right) + \frac{1}{2} a_{AV} \left(\frac{\delta_f F_s}{F_v} + \delta_r \right)^2. \quad (19)$$

Using $d_{min} = x + d_{lead} - d_{AV}$, we can arrive at (17). \square

A.2 Example signal traces for noisy sensor data

Figure 13 visualizes how the laser sensor is returning data for a stationary target approximately 17.9 m away from the sensor, for a period of approximately 10 s.

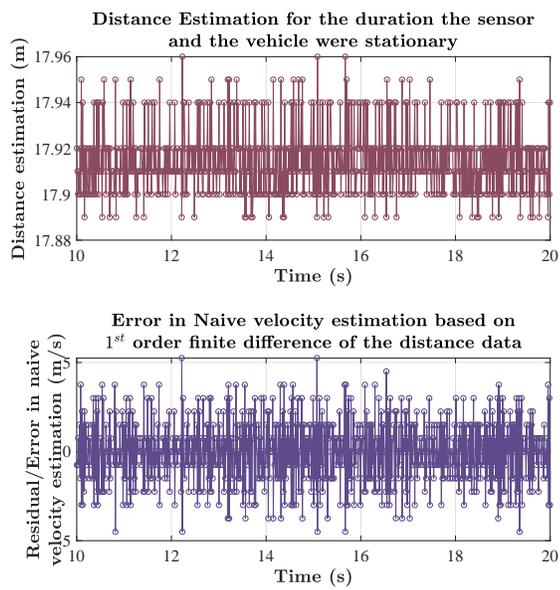


Figure 13: Signal traces for 10 s when target and sensor are stationary. The trace shows presence of short term fluctuation within a longer trend.