TU Delft

Delft University of Technology

CoPEM

Cooperative Perception Error Models for Autonomous Driving

Piazzoni, Andrea ; Cherian, Jim ; Vijay, Roshan ; Chau, Lap-Pui ; Dauwels, Justin

DOI
10.1109/ITSC55140.2022.9921807

Publication date
2022

Document Version
Final published version

Published in
Proceedings of the 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)

Citation (APA)
Piazzoni, A., Cherian, J., Vijay, R., Chau, L-P., & Dauwels, J. (2022). CoPEM: Cooperative Perception Error Models for Autonomous Driving. In *Proceedings of the 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)* (pp. 3934-3939). (IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC; Vol. 2022-October). IEEE . https://doi.org/10.1109/ITSC55140.2022.9921807

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

Green Open Access added to TU Delft Institutional Repository

'You share, we take care!' - Taverne project

https://www.openaccess.nl/en/you-share-we-take-care

Otherwise as indicated in the copyright section: the publisher is the copyright holder of this work and the author uses the Dutch legislation to make this work public.

2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)
October 8-12, 2022, Macau, China

CoPEM: Cooperative Perception Error Models for Autonomous Driving

Andrea Piazzoni[1,2], Jim Cherian[2], Roshan Vijay[2],
Lap-Pui Chau[3], *Fellow, IEEE*, Justin Dauwels[4], *Senior Member, IEEE*

2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC) | 978-1-6654-6880-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/ITSC55140.2022.9921807

***Abstract*— In this paper, we introduce the notion of Cooperative Perception Error Models (coPEMs) towards achieving an effective and efficient integration of V2X solutions within a virtual test environment. We focus our analysis on the occlusion problem in the (onboard) perception of Autonomous Vehicles (AV), which can manifest as misdetection errors on the occluded objects. Cooperative perception (CP) solutions based on Vehicle-to-Everything (V2X) communications aim to avoid such issues by cooperatively leveraging additional points of view for the world around the AV. This approach usually requires many sensors, mainly cameras and LiDARs, to be deployed simultaneously in the environment either as part of the road infrastructure or on other traffic vehicles. However, implementing a large number of sensor models in a virtual simulation pipeline is often prohibitively computationally expensive. Therefore, in this paper, we rely on extending Perception Error Models (PEMs) to efficiently implement such cooperative perception solutions along with the errors and uncertainties associated with them. We demonstrate the approach by comparing the safety achievable by an AV challenged with a traffic scenario where occlusion is the primary cause of a potential collision.**

I. Introduction

AVs need to perceive the environment around them to drive safely. In a typical configuration, an AV relies only on its onboard Sensing and Perception subsystem (S&P) to achieve this goal. This solution, while independent, is intrinsically limiting, as it implies a single Point of View (PoV); that of the ego-vehicle itself. This may cause occlusion problems, leading to perception errors, e.g., misdetection, and potential safety hazards. For example, on March 24th 2017, a human driver hit an occluded, oncoming AV at an intersection [1].

Cooperative Perception (CP), i.e., third-party external perception information communicated to the AV via V2X protocols, provides an avenue to overcome this limitation, as it enables the AV to leverage multiple PoVs. Specifically, various infrastructure-based (V2I) roadside perception units (RSPU) or even other connected vehicles (V2V), that communicate with an Autonomous Vehicle (AV) using V2X through Roadside Units (RSU) and Onboard Units (OBU), may be collaboratively combined with its onboard perception (OBP) output. This study focuses on the effects on AV safety given augmented perception information provided by such third-party sources.

In recent years, virtual testing has been well-established as an indispensable tool for validating the safety of AVs. However, achieving an effective integration of V2X-assisted CP within a simulation environment remains challenging and computationally expensive, more so when considering high fidelity simulations of multiple infrastructure and AV S&P instances. The AV under test may itself have many types of complex sensors. Alternatively, one can run virtual tests without S&P by relying on the ground truth, which is readily available from the simulation tool. This solution may provide results that do not necessarily correspond to the real world as perception errors are often ignored.

In our previous works [2], we have introduced the notion of Perception Error Models (PEM) as a convenient and efficient solution to incorporate perception errors into a simulation pipeline. PEMs directly model such errors and apply them on the ground truth, effectively injecting errors in-the-loop with the benefit of not needing to generate or simulate synthetic sensor data. If simulating just a single high-fidelity AV with a few sensors seems computationally expensive, simulation of V2X CP for large deployment areas, requiring simultaneous simulation of multiple S&P instances, can quickly become infeasible. In this paper, we aim to expand the PEM approach by including V2X use cases such as CP and model the additional perception errors and uncertainties involved.

In this paper, we make the following contributions:

- We define cooperative PEMs (coPEMs) as the extension of PEMs for V2X-based CP applications.
- We show how to model V2X-relevant perception errors, implement them as coPEMs, and integrate them into a simulation pipeline.
- We demonstrate the approach by comparing the safety achievable by an AV when challenged by a traffic scenario where occlusion can lead to a potential collision.

II. Related works

The field of V2X communication for road safety-related services has been well studied in the past two decades. The communication aspects and some typical use cases have been standardized by organizations such as SAE International and ETSI. Examples include the SAE J2735 for DSRC message format [3] and the ETSI TR 103 562 [4] describing the CPM message format for Collective Perception Service (CPS). These standards focus on the information that can be exchanged to provide a safer and more efficient use of

This work was supported in part by the Centre of Excellence for Testing & Research of AVs - NTU (CETRAN), under the Connected Smart Mobility (COSMO) programme.

[1]ERI@N, Interdisciplinary Graduate Programme, Singapore andrea006@ntu.edu.sg

[2]Centre of Excellence for Testing & Research of AVs, Nanyang Technological University, Singapore jcherian@ntu.edu.sg, rvijay@ntu.edu.sg

[3]School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore elpchau@ntu.edu.sg

[4]TU Delft, Dept. of Microelectronics, Fac. EEMCS, Mekelweg 4 2628 CD, Delft. j.h.g.dauwels@tudelft.nl

Authorized licensed use limited to: TU Delft Library. Downloaded on November 03,2022 at 10:53:22 UTC from IEEE Xplore. Restrictions apply.

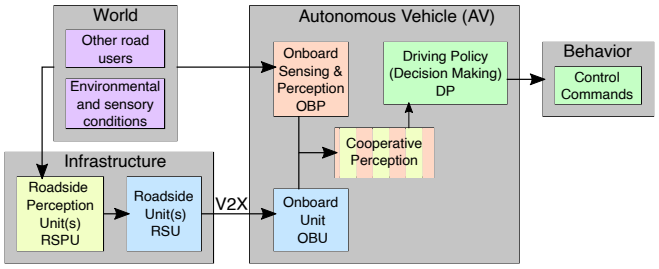


Fig. 1: Conceptual overview of AV Perception (both Onboard and Cooperative), in a V2X-based deployment.

the transportation network, such as road layout, safety alerts, signal phase and timing, and data on other road users and vehicles. Evidently, the communication component of V2X requires in-depth testing, analysis [5], and simulation [6], as bandwidth and latency can play a major role in the overall quality of the system [7].

Within the context of this paper, we focus on the task of V2X-enabled CP, i.e., multiple perception-capable units such as On-board Perception (OBP) and Roadside Perception Units (RSPU) augment their perception via V2X communication [8]. Identifying the location of RSPUs and their sensors is a major problem in this context, as their optimal placement can drastically improve safety and reduce cost (e.g., procurement, installation, and maintenance). Many studies aim to optimize coverage [9], [10]. To achieve this, some authors consider probabilistic coverage, introducing probabilistic sensor models [11]–[13].

Recent works are also studying synthetic data generation for CP [14], [15]. For example, the open-source CARLA simulator [16] is used to simulate the traffic environment and generate synthetic data.

Instead of RSU/RSPU placement or communication protocols, in this study, we look at V2X from the perspective of AV safety. Virtual simulation is a safe and convenient approach [17] for validation of AV safety. Shan et al. demonstrated CP in a simulation environment relying on synthetic data [18]. This approach is limited by the bottleneck of computational cost in the case of more than a few RSPUs or OBUs. Thus, we propose a framework for AV safety testing that includes perception errors and supports V2X solutions, i.e., an indefinite number of OBPs, OBUs, RSUs, and RSPUs.

III. Approach

In this section, we introduce Cooperative PEM (coPEM). First, we provide a summary background on the notation we employ (see Figure 1. We abstract an AV as a composition of a Sensing and Perception module (S&P) and the Driving Policy (DP; the decision-making module) [2]. The S&P has the task of observing the surrounding objects o_j, included in the surrounding world $\mathcal{W}$, to generate the perceived world $\tilde{\mathcal{W}}$, i.e., the object list describing the perceived surrounding obstacles $\tilde{o}_j$. The object list $\tilde{\mathcal{W}}$ is then analyzed by DP to determine the AV response.

In the case of CP, let N be the number of the third-party Perception Units (PUs; RSPUs and other OBPUs), each identified by $n \in \{1, \dots, N\}$, with ego-vehicle index 0. We assume that each PU_n perceive the surrounding world with its S&P system: $\tilde{\mathcal{W}}_n = \text{S\&P}_n(\mathcal{W})$, following the same abstraction of AVs. The $N+1$ object list $\tilde{\mathcal{W}}_n$, are then fused together: $\tilde{\mathcal{W}} = f(\tilde{\mathcal{W}}_n)$, where f implements the CP fusion task. We assume the fusion is at feature-level [19], which is suitable for V2X solutions as the communication can be performed at the objects level.

A. Individual PEM

In our previous work [2], we defined Perception Error Model (PEM) for AVs. A PEM is a model that receives the ground truth world $\mathcal{W}$ and returns the perceived world $\tilde{\mathcal{W}}$. Since virtual environments provide direct access to the ground truth world $\mathcal{W}$, a PEM can be integrated into a simulation pipeline as a flexible and efficient surrogate of the S&P. This approach removes the need for synthetic signals and introduces explicit perception errors at the object level. For example, a PEM can alter the perception of an object's position, or remove it from the object list, or more.

We define a PEM as an approximation of the combined function of sensing (S) and a perception (P) subsystems:

$$\text{PEM}(\mathcal{W}) \approx \text{S\&P}(\mathcal{W}) = \tilde{\mathcal{W}} = \mathcal{W} + \mathcal{E}, \tag{1}$$

where $\mathcal{E}$ is the perception error. We do not prescribe any preferred approach for a PEM implementation, but we opt for probabilistic models for a demonstration. In this article, we implement PEM_n as a set of $|C|$ couples:

$$\text{PEM} = \{(d_c, p_c), c \in C\}. \tag{2}$$

where:

- C is the set of conditions that alter the perception quality.
- d_c is the detection model, i.e., the model that determines if an object is detected.
- p_c describes the parameter error ε distribution, where $\varepsilon_j = \tilde{o}_j - o_j$, i.e. the difference between perception output and ground truth.

Generally, we include positional aspects (i.e., Field of View, operational range) in C. This notation affords a flexible implementation, and a single PEM may include different probabilistic models given the condition c. For example, a couple (d, p) can be dedicated to model the frontal area of the PU, with high detection rates and low parameter errors. Conversely, another (d, p) associated with a further away area may have lower detection rates and higher parameter errors.

A PEM_n generates $\tilde{\mathcal{W}}_n$ by processing each object $o_j \in \mathcal{W}$. First, we select the (d, p) to employ, by determining the current condition c, e.g., the relative position. Then, we compute if the object is detected via $d(o_j)$, i.e. if it is included in the $\tilde{\mathcal{W}}$. Lastly, we sample the parameter error ε_j according to the distribution $p(\varepsilon|o_j)$.

B. CoPEMs

The previous definition of PEM does not consider external sources of information, i.e., CP elements:

Authorized licensed use limited to: TU Delft Library. Downloaded on November 03,2022 at 10:53:22 UTC from IEEE Xplore. Restrictions apply.

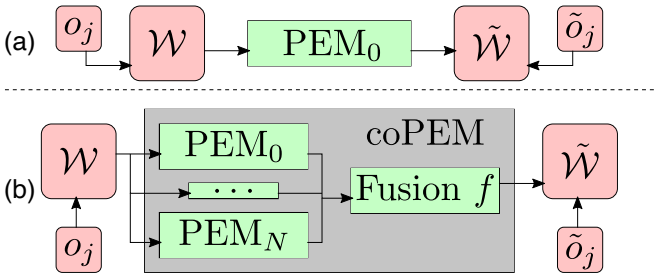


Fig. 2: Illustrations of (a) a PEM and (b) a coPEM as a collection of PEMs. Both the PEM and the coPEM receives $\mathcal{W}$ in input and return $\tilde{\mathcal{W}}$ in output, in fact, they have the same interface. In the coPEM, each PEM_n generates a $\tilde{\mathcal{W}}_n$, i.e. the perception output of each Perception Unit, and through a fusion phase (i.e., CP) are combined into the final $\tilde{\mathcal{W}}$.

- Heterogeneous Units: The S&P deployed in an AV is not self-contained anymore, and the DP does not directly consume its output $\tilde{\mathcal{W}}$.
- Dynamic Relative Position: Third-party RSU and OBU, static (V2I) or dynamic (V2V), have a dynamic relative position w.r.t. the ego-vehicle.
- Trust, Reliability, Latencies, and Uncertainties: Not all units have the same reliability. For example, static units potentially have lower uncertainties, while surrounding vehicles can be less precise or even malicious and not trustworthy [20]. In this study, we limit the analysis to the uncertainties and do not consider security issues.

However, a PEM can model each independent PU since we can assume that each of them has an S&P system. Hence, we define coPEM as a CP fusion function f and a set of PEMs representing the deployed $N+1$ perception units (i.e., the N third-party PUs and the ego vehicle's OBP).

$$\text{coPEM} = (f; \{(\mathbf{X}_n, \text{PEM}_n), 0 \leq n \leq N\}). \tag{3}$$

Where $\mathbf{X}_n$ indicates the pose of the unit. Given the dynamic location of the AV, we need to keep the model flexible and be able to compute the relative positions of all the PU involved in each step. Hence, in each step, $\mathbf{X}_n$ can be updated to keep the models aligned. In Figure 2, we outline the differences and similarities between the structures of PEMs and coPEMs.

C. Ideal Fusion Model

The newly introduced component to PEM is the fusion model f. Once the $\tilde{o}_n$ are generated, we outline two approaches for the fusion step:

- Perception Fusion Algorithm f: Since the coPEM receives $\mathcal{W}$, we can apply each PEM_i to generate all the $\tilde{\mathcal{W}}_i$. Then, we can integrate an actual perception fusion algorithm f to generate the consolidated $\tilde{\mathcal{W}}$. This approach is preferable when the focus of the virtual testing is this fusion algorithm itself.
- Ideal Fusion Model: Alternatively, we can model an optimal function f and directly generate $\tilde{\mathcal{W}}$. This option affords a simpler simulation as we can study other aspect of CP while discounting additional errors due to an not optimal f.

The first solution avoids any assumption, as an actual fusion algorithm is integrated into the pipeline. We propose an Ideal Fusion Model by assuming an optimal algorithm:

- Matching: The first step of merging M units is to find the matching of each object $\tilde{o}_j$ within multiple $\tilde{\mathcal{W}}_m$. An optimal f would perform this task without any error, avoiding false positives.
- Parameter Errors: Despite perfect matching, the objects $\tilde{o}_{jm}$ within each object list $\tilde{\mathcal{W}}_m$ will be affected by an error ε_{jm}, as each PU is affected by errors. The optimal f would minimize the resulting error ε_j, i.e., the error affecting $\tilde{o}_j$ after the fusion.

By leveraging the virtual environment and PEMs, we can implement an optimal f. We can exploit the underlying ground truth $\mathcal{W}$ to craft an optimal matching. Moreover, PEMs provide a probabilistic distribution for the parameter errors, which we can use to weigh the reliability of the PUs.

As a first step, we must determine the M_j units are actively involved in each object detection. Thus, each PEM_n should determine if the unit n detects an object o_j and participates in its parameter estimation. To combine parameter errors, we can formulate a measure combination problem. Each unit $m \in \{1, \ldots, M_j\}$ participating in the detection of an object o_j provides a vector of measurements for each physical property (i.e., the object parameters) with an error ε_{jn}. We model the distribution of each error ε_{jn} in each PEM_n. Thus, we need to determine the posterior distribution of $p(\varepsilon_j|o_j)$.

Bayesian inference can solve this problem, and the joint distribution can be written as:

$$p(\varepsilon_0|o_j) \ldots p(\varepsilon_M|o_j)p(o_j). \tag{4}$$

where $p(\varepsilon_m|o_j)$ is the distribution of the error ε_m given the object o_j. Instead, $p(o_j)$ is the prior of the model, which describes the dynamics of the object.

In the case of Gaussian distributions, the inverse-variance weighted average minimizes the variance of the average [21]. This way, more precise units (i.e., lower variance) have more influence on the outcome, and we imply an optimal combination of the units. We can write the V2X outcome distribution of the error ε as:w

$$p(\varepsilon_j|o_j) \sim \mathcal{N}(\mu, \Sigma), \tag{5}$$

Where mean and covariance are calculated as:

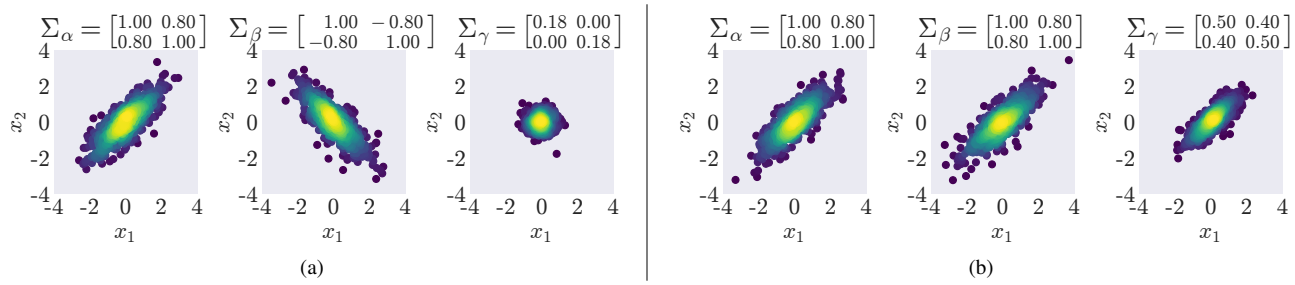
$$\mu = \left(\sum_n \left(\Sigma_n^{o_j}\right)^{-1} \right)^{-1} \sum_n \left(\Sigma_n^{o_j}\right)^{-1} \mu_n^{o_j}. \tag{6}$$

$$\Sigma = \left(\sum_n \left(\Sigma_n^{o_j}\right)^{-1} \right)^{-1}. \tag{7}$$

We note that, in Equations 5, 6, and 7, parameters $\mu_n^{o_j}$ and $\Sigma_n^{o_j}$ are conditioned to o_j via c (see Equation 2).

This approach implies that each parameter is modeled in absolute values. We avoid relative values such as distance and

Authorized licensed use limited to: TU Delft Library. Downloaded on November 03,2022 at 10:53:22 UTC from IEEE Xplore. Restrictions apply.



(a) (b)

Fig. 3: Sampling from 2 Gaussian distributions (α,β) and from their combination obtained by applying Equation 7 (γ). (a) The distributions α and β have different covariance matrices. We observe that Σ_γ has lower values, and its shape roughly resembles the intersection. In (b), instead, the distributions α and β have equal covariance matrices. Hence, their combination Σ_γ has the same shape and slightly smaller values. This approach is the multivariate equivalent of averaging multiple measures to increase accuracy.

azimuth from ego-vehicle. Instead, we model absolute coordinates (e.g., northing and easting). With these assumptions and procedures, we can use $p(\varepsilon|o_j)$ to sample the perception error. In Figure 3, we show two examples with Gaussian distributions.

D. V2X Latency

As previously mentioned, latency can be a significant bottleneck in V2X solutions [7]. Although it is not an issue in a virtual environment, latency can be specifically modeled and introduced to improve realism in virtual V2X testing. PEMs and coPEMs provide a straightforward solution to inject latency. When generating $\tilde{\mathcal{W}}^t$ at time t, instead of applying the model to the current world state $\mathcal{W}^t$, we can instead consider a previous $\mathcal{W}^{t-l}$ where l is the overall latency. The parameter l indicates the time difference between when an event happens and when the AV DP receives the information.

This solution effectively implies a slower reaction time induced by the communication delay. We do not consider, in this study, heterogeneous delays, as we assume a single $\tilde{\mathcal{W}}$ is consolidated each time frame.

IV. Experiments

We conduct a series of experiments under specific configurations to demonstrate our proposed approach. With these experiments, we aim to show the impact of occlusion on safety and how cooperative perception may address it. Each experiment is a set of simulations run identified by two elements: a perception configuration and a test case (i.e., a parametrized scenario). We implement our experiment in the ViSTA framework [22], which interfaces with SVL simulator [23] and Apollo [24] as the AV under test. We choose SVL since it facilitates the co-simulation by providing connection bridges toward independent AV stacks.

A. Perception Configurations

We aim to compare the different behavior induced by the varying perception quality, Thus, we consider three perception configurations, with additional variations for a total of 6 configurations (see Table I):

TABLE I: List of Perception Configurations.

ID	Sensor configuration
GT	Ground Truth based.
PEM	A PEM injects perception errors.
coPEM:0s	coPEM with 0s latency.
coPEM:0.5s	coPEM with 0.5s latency
coPEM:1s	coPEM with 1s latency.
coPEM:1.5s	coPEM with 1.5 latency

- GT: The simulation is based on ground truth, and occlusion is not a factor. The AV can detect anything in the environment with no delay.
- PEM: We replace the AV's S&P with a PEM. Occlusion has a significant role, and the detection rate of the model $d(o_j)$ is equal to the portion of the surface visible from the AV continuously from 0% for a completely occluded object, to 100% for a completely visible one. Moreover, we model an error in object location with a multivariate Gaussian distribution over northing and easting with variance of 1 m in both directions.
- coPEM: The AV's PEM is the same as the previous configuration. However, it is supported by $N = 20$ external V2X units, each modeled by its PEM. We tested four variations with increasing latency (0.5s increments).

With this set of configurations, we can observe the effects occlusion (PEMs) and latency (coPEMs) have on safety, and we can consider GT as the baseline.

B. Test Cases and Preliminary Exploration

The second element of each experiment is the test case. Each test case is defined by an AV starting point, a destination, and scripted actors with pre-defined behavior. In our previous work presented at the IEEE Autonomous Driving AI Test Challenge [22], we have defined and implemented a comprehensive set of Test Cases for testing AVs.

As the first step, we conducted a preliminary exploration phase. We tested each of the ~ 30 scenarios with various perception configurations to identify interesting variations in AV behavior due to occlusion issues.

For a deeper analysis, we selected a scenario inspired by TC_016.d from [22]. This test case was problematic to

Authorized licensed use limited to: TU Delft Library. Downloaded on November 03,2022 at 10:53:22 UTC from IEEE Xplore. Restrictions apply.

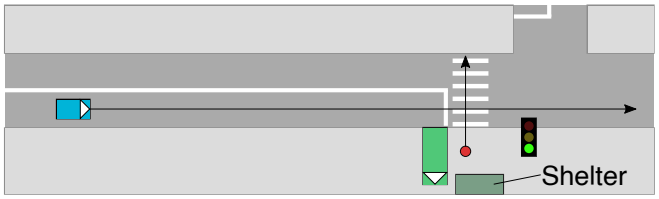


Fig. 4: Illustration of the Test Case. The AV (blue) approaches a pedestrian (red) occluded by a parked truck (green).

implement under the existing setup [22], i.e., using ground truth, as occlusion was not modeled. In this scenario, the AV approaches a signalized intersection on an urban road. A truck parked on the sidewalk occludes a pedestrian near a bus shelter, as illustrated in Figure 4. The pedestrian begins to cross once the AV approaches, even if the AV has the right of way as indicated by the green light.

We considered slight variations of this scenario by changing the AV path, location (i.e., introducing specific road features that can induce occlusion, such as a road bend), or tuning other parameters. We found that Apollo generally employs a conservative driving style and can handle most road features or maneuvers. Thus, we focused on parameter variations, employing GT and PEM configurations to achieve a meaningful test case. We scripted the test case to have the following properties:

- Plausible: The parameters applied (e.g., pedestrian walking speed) should be in a plausible range or, at most, incorporate edge cases.
- Fair: The test case should allow a safe outcome, and we expect a high success rate under GT configuration.
- Challenging: The test case should not be trivial, and we expect failures in cases of significant perception errors.

We found relevant parameters as follows: the AV starts far enough to reach a cruising speed of $\sim 10m/s$, and the pedestrian starts walking at $1m/s$ when the AV is within a $20m$ range. These settings lead to a potential collision $\sim 2s$ after the pedestrian starts walking. Given the occlusion and V2X latency, however, the AV may not be able to detect the danger instantaneously, resulting in a delayed reaction. We note that Apollo does not perform any evasive maneuver. Thus, the collision may only be avoided by braking.

Moreover, we experimented with RSPU positioning, and simplified the experiment set by considering only optimal placements (refer [10] for details) that actually help overcome the occlusion. Without this step, there could be redundant RSPU placements and coPEM configurations that may result in the same or similar AV behavior as with the regular PEM configurations (non-cooperative perception).

C. Results

Due to the page limit, we do not report results for all the explored scenarios and parameter variations. We simulated 500 runs for each perception configuration, i.e., 3000 runs.

During the test case selection and configuration phase, we employed plots of the logged simulation runs Figure 5. These

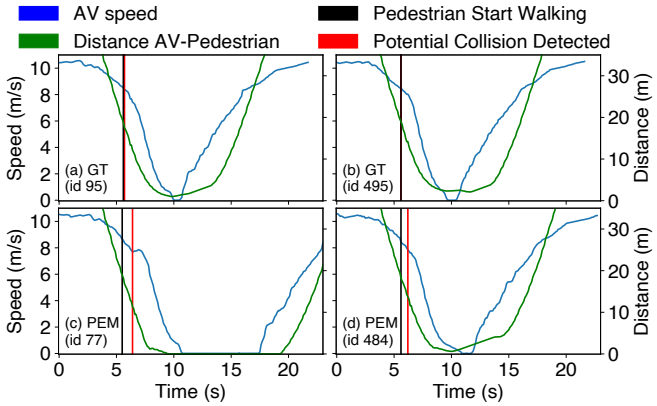


Fig. 5: Illustration of the relevant parameters and events of four example simulation runs.

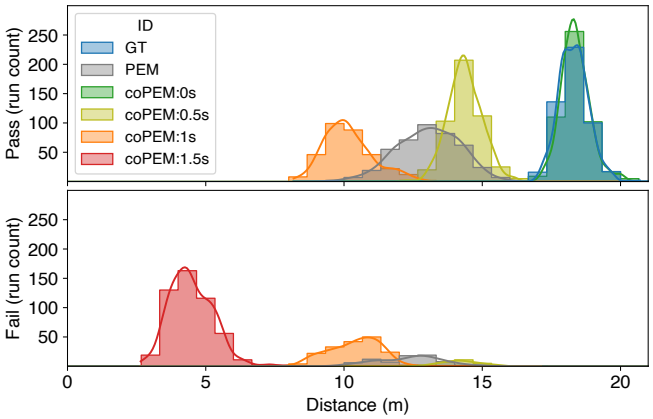


Fig. 6: Summarized results from 3000 runs (500 for each Perception Configuration). For each run, we consider a datapoint, i.e., the distance between the AV and the pedestrian at the moment of detection of the potential collision.

plots facilitate the exploration and parametrization of the test cases. In each plot, we depict the run dynamics, and we report the most relevant parameters and events. We can see how the potential collision is immediately detected under the GT configuration, and the AV can brake accordingly. However, under PEM configuration, the delay of the detection may (Figure 5c) or may not (Figure 5d) lead to a collision.

We categorize the outcome of the runs as Pass or Fail: we consider a run as successful if the AV reaches its destination and maintains a distance $> 0.5m$ from all actors (vehicles and pedestrians). In Figure 6, we summarize all 3000 runs, Pass and Fail, for each perception configuration. We note that the pedestrian was scripted to start walking when the distance is $< 20m$, but latency and occlusion can delay detection. In fact, we can see how the distance between the AV and the pedestrian decreases inversely to the latency introduced in the model. Simulation results clearly show how occlusion may lead to collisions. Moreover, we observe how CP can help to overcome the problem if the latency is kept low. Higher latency values, however, make the approach detrimental. Therefore, one may infer from this study that the

Authorized licensed use limited to: TU Delft Library. Downloaded on November 03,2022 at 10:53:22 UTC from IEEE Xplore. Restrictions apply.

TABLE II: Success Rate under each configuration.

ID	Success Rate		ID	Success Rate	
GT	500/500	100%	coPEM:0.5s	474/500	94.8%
PEM	426/500	85.2%	coPEM:1s	323/500	64.6%
coPEM:0s	499/500	99.8%	coPEM:1.5s	0/500	0.0%

adoption of low-latency communication schemes is essential to a successful deployment of V2X-based CP.

D. Discussion

CoPEMs are designed to be seamlessly integrated into a simulation pipeline. They provide an avenue for testing CP fusion algorithms, their effects on the AV perception, and thus the resulting behavior and safety. CoPEM limitations are mainly two: they require a PEM for each PU (i.e., a statistical model of the perception errors) and do not consider communication issues other than latency.

Moreover, we introduce an Ideal Fusion Model, which determines the optimal outcome of any CP approach while considering perception errors and uncertainties. However, this algorithm assumes normally distributed errors on the object parameters and it only applies in virtual environments since it requires access to the ground truth for the matching step. Nevertheless, it facilitates the testing other aspects of CP, e.g., AV safety, RSPU location, or OBPU critical mass in traffic.

V. Conclusion

In this paper, we gauged the effects of misdetection errors on AV behavior, and we analyzed how cooperative perception (CP) can improve safety. We conducted a series of simulation experiments by testing multiple perception configurations. These configurations included ground truth, PEMs, and coPEMs (PEMs extended for CP). These tools provide a convenient way to tune perception performance in simulation. Moreover, since they abstract the sensor technology, this solution scales well with an increasing number of sensors to support CP applications. We experimented with a rich set of scenarios to identify relevant conditions and parameters to finalize concrete test cases. We conducted the tests in a popular virtual simulation platform (SVL), and we deployed a standard open-source AV software stack (Baidu Apollo) to generate the AV behavior. Simulation results show the effect of perception errors (occlusion and latency) on AV safety in the selected test cases. In the future, we plan to extend the proposed approach to support heterogeneous latencies involved in CP, and integrate it with other simulators. Finally, we also aim to deploy CoPEMs towards testing CP with physical hardware in the loop testing configurations, more complex scenarios, and on-road AV trials.

References

[1] O. Germbek, A. A. Kurzhanskiy, A. Medury, P. Varaiya, M. Yu, and A. Siddiqui, "Safe operation of automated vehicles in intersections," No. UCB-ITS-PRR-2019-03. Institute of Transportation Studies, UC Berkeley, 2018.

[2] A. Piazzoni, J. Cherian, M. Slavik, and J. Dauwels, "Modeling perception errors towards robust decision making in autonomous vehicles," in *Proc. of the 29th International Joint Conference on Artificial Intelligence, IJCAI-20*, 7 2020, pp. 3494–3500, main track.

[3] SAE, "J2735_202007 standard: V2x communications message set dictionary," 2020.

[4] ETSI, "Etsi tr 103 562: Intelligent transport systems (its); vehicular communications; basic set of applications; analysis of the collective perception service (cps); release 2," 2019.

[5] F. A. Schiegg, N. Brahmi, and I. Llatser, "Analytical performance evaluation of the collective perception service in c-v2x mode 4 networks," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 181–188.

[6] E. Zhang and N. Masoud, "V2xsim: A v2x simulator for connected and automated vehicle environment simulation," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–6.

[7] C. Pilz, A. Ulbel, and G. Steinbauer-Wagner, "The components of cooperative perception - a proposal for future works," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2021, pp. 7–14.

[8] H.-J. Gnther, B. Mennenga, O. Trauer, R. Riebl, and L. Wolf, "Realizing collective perception in a vehicle," in *2016 IEEE Vehicular Networking Conference (VNC)*, 2016, pp. 1–8.

[9] F. Geissler and R. Gräfe, "Optimized sensor placement for dependable roadside infrastructures," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 2408–2413.

[10] R. Vijay, J. Cherian, R. Riah, N. De Boer, and A. Choudhury, "Optimal placement of roadside infrastructure sensors towards safer autonomous vehicle deployments," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2021, pp. 2589–2595.

[11] V. Akbarzadeh, C. Gagne, M. Parizeau, M. Argany, and M. A. Mostafavi, "Probabilistic sensing model for sensor placement optimization based on line-of-sight coverage," *IEEE transactions on instrumentation and measurement*, vol. 62, no. 2, pp. 293–303, 2012.

[12] V. Akbarzadeh, J.-C. Lévesque, C. Gagné, and M. Parizeau, "Efficient sensor placement optimization using gradient descent and probabilistic coverage," *Sensors*, vol. 14, no. 8, pp. 15 525–15 552, 2014.

[13] M. Argany, F. Karimipour, F. Mafi, and A. Afghantoloee, "Optimization of wireless sensor networks deployment based on probabilistic sensing models in a complex environment," *Journal of Sensor and Actuator Networks*, vol. 7, no. 2, p. 20, 2018.

[14] E. Arnold, S. Mozaffari, and M. Dianati, "Fast and robust registration of partially overlapping point clouds," *IEEE Robotics and Automation Letters*, pp. 1–8, 2021.

[15] Y. Li, Z. An, Z. Wang, Y. Zhong, S. Chen, and C. Feng, "V2x-sim: A virtual collaborative perception dataset for autonomous driving," *arXiv preprint arXiv:2202.08449*, 2022.

[16] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.

[17] W. Young, A. Sobhani, M. G. Lenné, and M. Sarvi, "Simulation of safety: A review of the state of the art in road safety simulation modelling," *Accident Analysis & Prevention*, vol. 66, pp. 89–103, 2014.

[18] M. Shan, K. Narula, Y. F. Wong, S. Worrall, M. Khan, P. Alexander, and E. Nebot, "Demonstrations of cooperative perception: Safety and robustness in connected and automated vehicle operations," *Sensors*, vol. 21, no. 1, p. 200, 2020.

[19] N. Kaempchen, M. Buehler, and K. Dietmayer, "Feature-level fusion for free-form object tracking using laserscanner and video," in *IEEE Proceedings. Intelligent Vehicles Symposium, 2005.*, 2005, pp. 453–458.

[20] M. Ambrosin, L. L. Yang, X. Liu, M. R. Sastry, and I. J. Alvarez, "Design of a misbehavior detection system for objects based shared perception v2x applications," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 1165–1172.

[21] C. Shalizi, "Advanced data analysis from an elementary point of view," 2013.

[22] A. Piazzoni, J. Cherian, M. Azhar, J. Y. Yap, J. L. W. Shung, and R. Vijay, "ViSTA: a framework for virtual scenario-based testing of autonomous vehicles," in *2021 IEEE International Conference on Artificial Intelligence Testing (AITest)*, 2021, pp. 143–150.

[23] G. Rong, B. H. Shin, H. Tabatabaee, Q. Lu, S. Lemke, M. Možeiko, E. Boise, G. Uhm, M. Gerow, S. Mehta *et al.*, "Lgsvl simulator: A high fidelity simulator for autonomous driving," *arXiv:2005.03778*, 2020.

[24] H. Fan, F. Zhu, C. Liu, L. Zhang, L. Zhuang, D. Li, W. Zhu, J. Hu, H. Li, and Q. Kong, "Baidu Apollo EM Motion Planner," *arXiv: 807.08048*, 2018.

Authorized licensed use limited to: TU Delft Library. Downloaded on November 03,2022 at 10:53:22 UTC from IEEE Xplore. Restrictions apply.