

## QUANTIFYING ERROR PROPAGATION IN MULTI-STAGE PERCEPTION SYSTEM OF AUTONOMOUS VEHICLES VIA PHYSICS-BASED SIMULATION

Fenglian Pan, Yinwei Zhang, Larry Head, Jian Liu

Maria Elli, Ignacio Alvarez

Department of Systems and Industrial Engineering  
University of Arizona  
1127 E. James E. Rogers Way  
Tucson, AZ 85721, USA

Intel Labs  
2111 NE 25th Ave  
Hillsboro, OR 97124, USA

### ABSTRACT

Ensuring the safety of autonomous vehicle (AV) relies on accurate prediction of error occurrences in its perception system. Due to the inter-stage functional dependence, the error occurred at a certain stage may be propagated to the following stage and generate extra errors. To quantify the error propagation, this paper adopts the physics-based simulation, which enables fault injection at different stages of an AV perception system to generate error event data for error propagation modeling. A multi-stage Hawkes process (MSHP) is proposed to predict the error occurrences in each stage, with error propagation represented as a latent triggering mechanism. With explicitly considering the error propagation mechanism, the proposed outperforms benchmark methods in predicting error occurrence in a physics-based simulation of a multistage AV perception system. The proposed two-step likelihood-based algorithm accurately estimates the model coefficients in a numerical simulation case study.

### 1 INTRODUCTION

Autonomous vehicle (AV) technology has been making a prominent appearance in emerging smart cities, providing a promise of improvements in convenient, efficient, and reliable vehicular travel (Jha et al. 2018). One of the major motivations that accelerate the advancements of AV technologies is that AV technologies provide significant driving performance improvements compared to human drivers who are prone to speeding, distracted driving or driving under the influence.

A typical AV consists of five main systems: Perception, Localization and Mapping, Path Planning, Decision Making, and Vehicle Control (Brummelena et al. 2018). The perception system uses a variety of sensors to continuously monitor the surrounding environment and the state of the AV, and to support other systems to ensure that the AV can navigate through complex environments while maintaining a safety envelope (Erlien 2015). The safety of AV driving depends upon the performance of the perception system. The failure of the perception system, such as missing the existence, or miscalculating the measurements, of a pertinent object (e.g., a road user or an inanimate obstacle), could cause a potentially catastrophic accident. A perception system generally is comprised of multiple sensor-perception subsystems (e.g., camera, LiDAR, RADAR) that are integrated into a serial-parallel structure. Specifically, each sensor modality fulfills perception requirements with a pipeline of multiple inter-connected stages, each of which provides one of a series of interrelated functionalities. For instance, a camera perception pipeline may consist of multiple stages, as shown in Figure 1. The hardware camera sensors adopted at the *data acquisition* stage collect environment videos, which are fed to the *object detection/identification* stage to detect and identify the vehicles and/or pedestrians whose distance from the ego vehicle will be given in the *distance measurement* stage (Campbell et al. 2018).

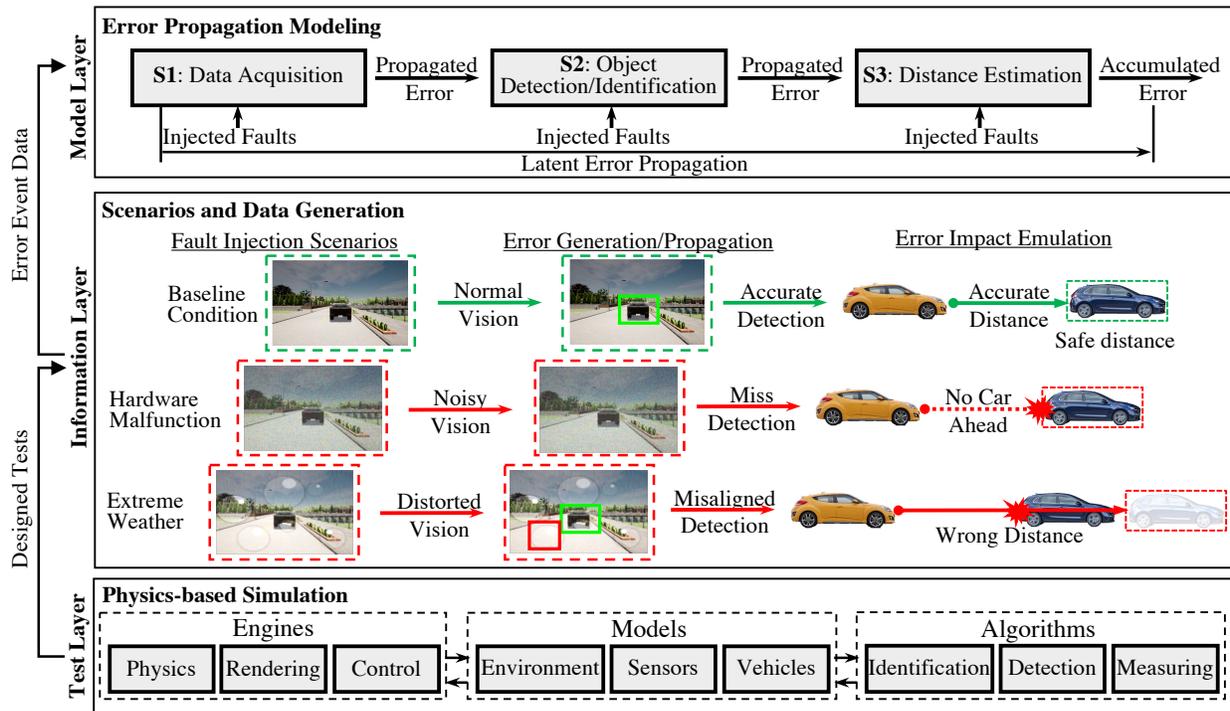


Figure 1: The methodological framework of the simulation enabled error-propagation modeling.

The functioning of the perception system is subject to a variety of faults, errors, and failures that are induced by external environmental conditions or internal hardware, software, and algorithmic imperfections. As a result, its performance may degrade and ultimately affect the safety of the overall AV system. In this paper, a *fault* is defined as an abnormal condition that may cause a reduction in, or loss of, the capability of a functional stage to perform its required function. The manifestation of faults may lead to an *error*, which is defined as a deviation from the required operation of the functionality. The accumulation of errors will eventually result in a *failure* of a perception modality to perform its designated functionality in accordance with pre-specified requirements (Farooq et al. 2012). For instance, the degradation of CCD (i.e., a fault) in camera sensors may generate excessive white noise in collected video/images (i.e., an error), which may lead to the miss detection of a pedestrian on the road (i.e., a propagated error) and a failure to brake and avoid an accident (i.e., a failure).

It is critical to understand and evaluate the impacts of such faults, errors, and failures on the performance of the AV perception system and AV system safety. Statistical reliability analysis can be conducted to model and predict the errors and failures from the data of their random occurrence (Gnedenko et al. 1999). However, the complexity of the AV perception system, especially the serial-parallel multistage structure and the functional inter-stage dependences, makes such modeling and prediction extremely challenging. The faults introduced at a certain stage of a sensor-perception subsystem may lead to errors that are transmitted to the following stage and generate more errors, which will be propagated further to downstream stages. For instance, the severe weather with heavy rainfall leaves excessive water droplets on the lens of the camera sensor at the data acquisition stage and may generate distorted images. Such low-quality images transmitted to the object detection stage may increase the errors in target detection and localization, which may be further propagated to the distance measurement stage, causing the miss-calculation of the actual distance to the leading vehicle, as shown in Figure 1. In other words, the error that occurred at a certain stage may be caused by the faults locally injected at the same stage, or be caused by the errors propagated from an

upstream stage. Without explicitly considering and quantifying this error propagation, the modeling and prediction of the error occurrence will be inaccurate and thus, misleading.

The statistical learning, quantification, and validation of the fault accumulation and error propagation rely on data of error occurrence during the operation of the multi-stage perception system. The data can be collected from real-world operations and testings of AVs on public roads. For instance, The California Department of Motor Vehicles (DMV) reports that between 2014 and 2017, manufacturers tested 144 AVs, driving a cumulative 1,116,605 autonomous miles, and reported 5,328 disengagements and 42 accidents involving AVs on public roads (Banerjee et al. 2018). These on-road testing provide invaluable AV performance data in actual situations and scenarios, from which a probabilistic error occurrence model can be built and the model parameters can be accurately estimated. However, it is economically prohibitive, if not impossible, to run safety tests on an exhaustive list of scenarios and road situations. This is especially true for some extremely low-probability scenarios (Dosovitskiy et al. 2017). The liability concern of testing immature AVs on public roads is another factor that makes real-world tests unfavorable since their safety has not been fully verified (Wakabayashi 2018).

Simulation has been serving as an indispensable alternative option for AV tests to address the cost and safety concerns (Pomerleau 1989). One of the necessitous abilities of simulators is to generate “billions of miles” for AV performance verification (Fremont et al. 2020). In addition, numerous simulation engines and physics-based models have been developed to provide not only the high-fidelity environment and realistic scenarios (Richter et al. 2016), but also the traffic-related agent, including vehicles, pedestrians, traffic rules, and even sensors. This combination enables the injection of various faults to the autonomous driving environment, as well as hardware and software of AV systems (Jha et al. 2018). In this way, the AV safety can be tested in situations that might otherwise rarely occur in reality.

Increasingly powerful physics-based simulation engines and models, together with the fault injection mechanism, create the unprecedented potential for researchers and practitioners to virtually test AV safety, especially the faulty accumulation and error propagation mechanism. Bayesian network approach has been adopted in one of such platforms, *DriveFI* (Jha et al. 2019), to represent the impacts of injected faults on AV safety in the simulated scenarios through causal and counterfactual reasoning. Without a mechanism that explicitly links the errors generated across modalities and functional stages, such a non-parametric approach has limited potential for error prediction and safety evaluation. This research proposes a three-layer framework built upon the integration of physics-based simulation, error-propagation modeling, and statistical inference. The *test layer* combines the capability of simulation engines with the physics-based models and algorithms applied in AV perception systems to emulate the interconnected functionalities of the multi-stage perception modalities. The *information layer* collects all the errors simulated in the test layer along the perception stages and represents them as correlated random events, based on which the *model layer* builds a new MSHP model to explicitly quantify the cross-stage error propagation as with an event-triggering process and estimate the model coefficients with a new statistical inference algorithm.

The rest of the paper is organized as follows: Section 2 introduces the methodology. In Section 3, the case study based on numerical simulation data and the physics-based simulation data will be presented, respectively. The conclusion will be given in Section 4.

## 2 METHODOLOGY

The proposed error-propagation modeling methodology is enabled by the high-fidelity physics-based simulation, which is capable of virtually injecting various types of faults into different stages of the AV perception system, generating errors along stages, and logging error event data. A generic recursive model is formulated to represent the multi-stage error propagation by explicitly differentiating the impacts of the faults injected in the current stage, and that of the latent errors transmitted from an upstream stage. The model coefficients are estimated from the simulated data by a two-step likelihood-based approach with enhanced computational efficiency. The model efficiency is evaluated based on the estimated parameters.

## 2.1 Physics-based Simulation

Physics-based simulation pipeline is adopted in this research to cost-effectively generate sufficient error events along stages of the perception modalities to support error propagation modeling. In this research, the simulation platform is chosen as an integration of a variety of engines, models, and algorithms, as shown in Figure 1. To ensure the provision of high-fidelity simulated data, *physics engine* is employed to bridge the gap between the virtual- and the real-world scenarios, simulating not only the physical phenomena, such as gravity, friction, and collision, but also vehicle properties, such as mass and acceleration. The rendering engine provides a realistic visual presentation of the simulated driving environment, especially effects related to weather, such as precipitation, light condition, etc. The visual effects can be observed by the sensors of the AV perception system. Different simulation settings lead to different rendering environments, which may affect the performance of the perception system and the AV safety. Control engines execute vehicle control actions determined according to the perception.

A variety of physics models build the foundation for the engines to simulate different types of faults that will be injected into the scenarios. Specifically, an injected fault can be categorized as an external fault (corresponding to the environment) or an internal fault (corresponding to the hardware or software of a perception modality). The environment models support different scenarios with various weather, infrastructures, and traffic conditions, while the sensor models support the operations and performance of numerous sensors, such as camera, LiDAR, and GPS, which are used in the AV perception systems to sense the rendered environment and support the decisions based on the perceptions. A fault is injected by making a significant change in one of these models, such as the change of the weather or the performance of a degrading camera. The faulty observations may have detrimental effects on the functionalities of the perception system, which are implemented by a series of algorithms, such as object detection and identification. Thus, the latent generation and propagation of the errors will be manifested in the performance of those algorithms.

## 2.2 Error Propagation Modeling

The physics-based simulation platform provides fault injection in a multi-stage perception system. The injected faults may lead to errors at the same stage, defined as random *local error* events, which may be propagated to the following stages and cause additional errors, defined as *propagated error* events. This error propagation is conceptually existing due to the interrelated functionalities between stages. However, this inter-stage relationship is usually latent and cannot be directly quantified based on the observed error event data collected from stages. Without loss of generality, this paper considers one type of error at each one of the  $S$  stages in a perception system. The event data from stage  $s$  is a series of  $n_s$  error occurrence timestamps, i.e.,  $\mathbf{t}^s = \{t_i^s\}_{i=1}^{n_s}$ , where  $t_1^s < t_2^s < \dots < t_{n_s}^s$ ,  $s = 1, 2, \dots, S$ ,  $S$  is the total number of stages, and  $n_s$  denotes the total number of errors occurred at stage  $s$ .  $\mathbf{t}^s$ 's collected from  $S$  stages can be treated as instances of a multivariate point process in a temporal space (Thompson 2012). For a general point process, the expected instantaneous occurrence probability of an event at time  $t$  can be modeled by a conditional intensity function (CIF, “*intensity*” hereafter) given a set of historical event data,  $\mathcal{H}(t)$ , during  $[0, t)$ . It can be defined as:  $\lambda(t|\mathcal{H}(t)) = \lim_{dt \rightarrow 0} E[N([t, t+dt))|\mathcal{H}(t)]/dt$ , where  $N(A)$  is the number of events occurring at time  $t \in A$ , and  $E[N([t, t+dt))|\mathcal{H}(t)]$  is the expected number of events occurring in the time interval  $[t, t+dt)$  given  $\mathcal{H}(t)$ . This intensity function can be used to model the random error occurrence at a given stage of the perception system and is the building block of the proposed error propagation model. To model the latent error propagation in a multi-stage perception system, this paper proposes a modified MSHP with recursive definition of error intensity at each stage. Specifically, the intensity function is comprised of two components. One is caused by the accumulation of faults locally injected at a stage, and the other one is caused by the error propagated from an upstream stage.

Mathematically, the error propagation in an  $S$ -stage perception system can be represented by a series of recursively defined stage-level intensity functions, as illustrated in Figure 2. Consider stages  $s-1$  and  $s$ , the injected faults at stage  $s$  may cause the error locally at the same stage, whose occurrence randomness

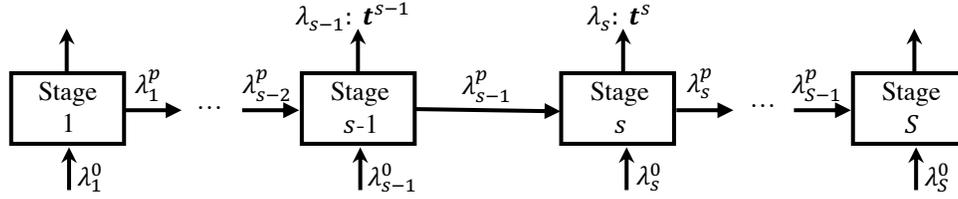


Figure 2: Mathematical representation of error propagation of multistage perception system.

is modeled by a baseline intensity  $\lambda_s^0$  (the superscript “0” denotes the baseline intensity). The occurred error events  $\mathbf{t}^{s-1}$  at stage  $s - 1$  may be propagated to stage  $s$  and trigger extra errors in addition to the local errors caused by the faults injected at stage  $s$ . The occurrence randomness of such triggered errors can be modeled by an intensity function,  $\lambda_{s-1}^p$  (the superscript “p” denotes the triggered intensity), which is superimposed to the baseline intensity function,  $\lambda_s^0$ . Specifically, the error occurrence intensity at stage  $s$  is comprised of the local baseline intensity,  $\lambda_s^0$ , and the propagated triggered intensity  $\lambda_{s-1}^p$ , i.e.,

$$\lambda_s(t|\mathcal{H}_{s-1}(t), \mathcal{H}_s(t)) = \lambda_s^0(t|\mathcal{H}_s(t)) + \lambda_{s-1}^p(t|\mathcal{H}_{s-1}(t)), s = 1, 2, \dots, S, \quad (1)$$

where,  $\mathcal{H}_{s-1}(t)$  and  $\mathcal{H}_s(t)$  represent the historical error events occurred at stage  $s - 1$  and stage  $s$  during  $[0, t)$ , respectively.  $\lambda_s^0(t|\mathcal{H}_s(t))$  is the *baseline intensity* for *local error* at stage  $s$  and  $\lambda_{s-1}^p(t)$  is the *triggered intensity* for the *propagated error* from stage  $s - 1$ . Model (1) follows the same form of a Hawkes process model (Rizoiu et al. 2017), which decomposes an intensity function into a baseline intensity and a triggered intensity to explicitly model the event triggering mechanism.

This paper uses the homogeneous Poisson process as the baseline process for stage  $s$ ,  $\lambda_s^0(t)$ , to model stable performance of perception stages without error propagation. The *triggered intensity*  $\lambda_{s-1}^p(t|\mathcal{H}_{s-1}(t))$  is defined as:

$$\lambda_{s-1}^p(t|\mathcal{H}_{s-1}(t)) = \sum_{i:t_i^{s-1} < t} M_{s-1} \cdot \omega_{s-1} \cdot \exp[-\omega_{s-1} \cdot (t - t_i^{s-1})], s = 1, 2, \dots, S, \quad (2)$$

where,  $\{t_i^{s-1}\}_{i:t_i^{s-1} < t}$  are the timestamps of error event occurred at stage  $s - 1$  before  $t$  in  $\mathcal{H}_{s-1}(t)$ . An exponential kernel with parameter  $\omega_{s-1}$  is chosen to describe that the impact of the propagated error will decay *exponentially* over time. While  $M_{s-1}$  determines the expected number of errors, at stage  $s$ , triggered by the errors propagated from stage  $s - 1$ ,  $\omega_{s-1}$  sets the decaying speed. Basically, both  $M_{s-1}$  and  $\omega_{s-1}$  model the impacts of the erroneous outcomes propagated from stage  $s - 1$  on the functional performance of stage  $s$ , in terms of error intensity. It is worth noting that different types of injected faults at stage  $s - 1$  will result in different  $M_{s-1}$  and  $\omega_{s-1}$ .

### 2.3 Model Estimation

The parameters  $\theta$ ,  $\theta = \{\lambda_s^0, M_{s-1}, \omega_{s-1}\}$ , in Equations (1) and (2) jointly define the latent error propagation mechanism by explicitly setting two separate parameter groups, the baseline parameter  $\lambda_s^0$  and the triggering parameters  $\{M_{s-1}, \omega_{s-1}\}$ . It is time-consuming to estimate these parameters simultaneously by using traditional estimation methods, such as numerical routine, expectation-maximization (EM) algorithm. This is especially true when the number of the parameters is large. However, the two-group setting makes it possible to propose a likelihood-based approach to estimate the parameters, sequentially, in two steps. This is based on two facts about the model: (i) the baseline intensity is assumed as a constant and the error event data without error propagation can be collected from our proposed physics-based simulation framework where the fault is not injected in upstream stages. Thus, the baseline parameter  $\lambda_s^0$  can be estimated by using standard estimation algorithms, such as maximum likelihood estimation. (ii) the triggering parameters begin to play their roles only after the faults are injected in the upstream stage. Thus, a two-step parameters estimation procedure is proposed:

*Step 1: estimation of baseline parameter  $\lambda_s^0$ .* This step is based on the error event data collected at stage  $s$  without any faults injected at stage  $s - 1$ . To differentiate the notation, let  $\boldsymbol{\tau}^s = \{\tau_i^s\}_{i=1}^{N_s}$  denote the error event timestamps at stage  $s$  where there is no error propagated from stage  $s - 1$ . Since  $\boldsymbol{\tau}^s$  does not involve triggering parameters and follows a homogeneous Poisson process, the baseline parameter  $\lambda_s^0$  can be estimated by Maximum Likelihood Estimation (MLE), as

$$\hat{\lambda}_s^0 = \frac{1}{N_s} \sum_{i=1}^{N_s} \tau_i^s. \quad (3)$$

*Step 2: estimation of triggering parameters  $\{M_{s-1}, \omega_{s-1}\}$ .* The error event data  $\mathbf{t}^s = \{t_i^s\}_{i=1}^{n_s}$  at stage  $s$  are collected when the faults are injected at stage  $s - 1$  during time interval  $[0, T]$ . Given  $\{t_i^s\}_{i=1}^{n_s}$  and  $\hat{\lambda}_s^0$  estimated from (3), the log-likelihood function for the  $s$ -th stage can be derived as:

$$l_s(M_{s-1}, \omega_{s-1} | \hat{\lambda}_s^0, \{t_i^s\}_{i=1}^{n_s}) = \sum_{i=1}^{n_s} (\log(\lambda_s(t_i^s | \hat{\lambda}_s^0))) - \int_0^T \lambda_s(t | \hat{\lambda}_s^0) dt. \quad (4)$$

The error event triggering relationship is latent, i.e., it is unknown whether an error event occurred at stage  $s$  is actually triggered by an error event at stage  $s - 1$ . This latency makes it impossible to estimate the triggering parameters by maximizing the log-likelihood function in (4) directly. Thus, an EM algorithm is proposed to estimate the triggering parameters by treating the unknown triggering relation as missing values. In *E-step*, a matrix  $\mathbf{P}$ ,  $\mathbf{P} = \{p_{ij}\}$ ,  $i = 1, \dots, n_{s-1}$ ,  $j = 1, \dots, n_s$ , is introduced to represent whether an error occurred at stage  $s$  is a local error, or is propagated from the error at stage  $s - 1$ . Specifically, the element  $p_{ij}$  represents the probability that the  $j$ -th error at stage  $s$  is propagated from the  $i$ -th error at stage  $s - 1$ , when  $i < n_{s-1} + 1$ . Otherwise, the element  $p_{(n_{s-1}+1)j}$  represents the  $j$ -th error at stage  $s$  is a local error. Thus,  $p_{ij}$  in E-step can be defined as :

$$p_{ij} = \begin{cases} \frac{M_{s-1} \cdot \omega_{s-1} \cdot \exp(-\omega_{s-1} \cdot (t_j^s - t_i^{s-1}))}{\hat{\lambda}_s^0 + \sum_{I: t_i^{s-1} < t_j^s} M_{s-1} \cdot \omega_{s-1} \cdot \exp(-\omega_{s-1} \cdot (t_j^s - t_i^{s-1}))}, & \text{if the } j\text{-th error at stage } s \text{ is a propagated error;} \\ \frac{\hat{\lambda}_s^0}{\hat{\lambda}_s^0 + \sum_{I: t_i^{s-1} < t_j^s} M_{s-1} \cdot \omega_{s-1} \cdot \exp(-\omega_{s-1} \cdot (t_j^s - t_i^{s-1}))}, & \text{if the } j\text{-th error at stage } s \text{ is a local error.} \end{cases} \quad (5)$$

By plugging  $\mathbf{P}$  computed from (5) into the log-likelihood function in (4), the expected log-likelihood function can be obtained with the details in Appendix A.2. In *M-step*, the expected log-likelihood function can be maximized by setting the partial derivative with respect to the  $M_{s-1}$  and  $\omega_{s-1}$ , equal to 0, respectively. Thus, given  $\mathbf{P}$  in E-step, M-step consists of updating the parameters between  $k$ -th iteration and  $(k + 1)$ -th iteration as follows:

$$\hat{M}_{s-1}^{k+1} = \frac{\sum_{j=1}^{n_s} \sum_{i=1}^{n_{s-1}} p_{ij}}{\sum_{t_i^{s-1} < t} (1 - \exp(-\hat{\omega}_{s-1}^k (T - t_i^{s-1})))}, \quad (6)$$

$$\hat{\omega}_{s-1}^{k+1} = \frac{\sum_{j=1}^{n_s} \sum_{i=1}^{n_{s-1}} p_{ij}}{\sum_{t_i^{s-1} < t} \hat{M}_{s-1}^{k+1} \exp(-\hat{\omega}_{s-1}^k (T - t_i^{s-1})) (T - t_i^{s-1}) + \sum_{j=1}^{n_s} \sum_{i=1}^{n_{s-1}} p_{ij} (t_j^s - t_i^{s-1})}. \quad (7)$$

Given initial values for the parameters,  $\{M_{s-1}, \omega_{s-1}\}$ , the EM algorithm iterates between updating the probabilities in Equation (5) for E-step and updating the parameters using Equations (6) and (7) in M-step. The optimization problem can be solved and the parameters can be estimated under certain convergence criterion. The estimated baseline parameter  $\hat{\lambda}_s^0$ , as well as the estimated triggering parameters  $\{\hat{M}_{s-1}, \hat{\omega}_{s-1}\}$  will be used to derive the estimated error event intensity at stage  $s$  at a given  $t$ , as  $\hat{\lambda}_s(t | \hat{\lambda}_s^0, \hat{M}_{s-1}, \hat{\omega}_{s-1})$ .

## 2.4 Model Evaluation

The effectiveness of the model estimation and the accuracy of the estimated model coefficients can be evaluated by the discrepancy between the count of the actually occurred random error events and that of the occurrence predicted by the model. Suppose observations of error events that occurred at  $S$  stages have been collected up to time  $t^*$ . Based on these observations, the estimated parameters  $\hat{\theta}^*$  and the predictive intensity  $\hat{\lambda}_s^*(t)$  can be obtained with Equations (3) - (7) in Section 2.3. Given  $\hat{\lambda}_s^*(t)$ , the number of error event occurrence at stage  $s$  during time interval  $(t^*, t^* + L]$  can be predicted as (Lieshout and NM 2012)

$$\hat{N}_{s|(t^*:t^*+L]} = \int_{t^*}^{t^*+L} \hat{\lambda}_s^*(t) dt. \quad (8)$$

Physics-based simulation will be used to generate the actual number of error event occurrence for the same time interval, denoted as  $N_{s|(t^*:t^*+L]}$ , which will be compared to its predicted counterpart in (8). This comparison can be conducted in  $R$  replications of the simulation. Denote, for replication  $r$ , the predicted and the actual number of error event occurred during  $(t^* : t^* + L]$  as  $\hat{N}_{s|(t^*:t^*+L]}^r$  and  $N_{s|(t^*:t^*+L]}^r$ , respectively, where  $r = 1, \dots, R$ . The Mean Absolute Error (MAE) can be used to evaluate the performance of the proposed model at stage  $s$  during  $(t^* : t^* + L]$ , i.e.,

$$MAE_{s|(t^*:t^*+L]} = \frac{\sum_{r=1}^R |N_{s|(t^*:t^*+L]}^r - \hat{N}_{s|(t^*:t^*+L]}^r|}{R}. \quad (9)$$

## 3 CASE STUDY

Both the numerical simulation case study and physics-based simulation case study are used to evaluate the performance of the proposed method. Specifically, this paper benchmarks the performance against the Poisson process approach which does not take the error propagation mechanism into account. In Section 3.1, a two-stage HP is simulated to illustrate the efficiency of the proposed estimation approach. In Section 3.2, a two-stage (data acquisition and object detection) camera perception system is considered and the error event data for each stage is collected from a physics-based simulation platform. The results and findings demonstrate the efficiency of considering the error propagation in the proposed model.

### 3.1 Numerical Simulation Case Study

The process to generate the data from an MSHP is based on the thinning simulation algorithm (Ogata 1981). The procedure of data simulation is outlined in Algorithm 1 in Appendix A.1. Specifically, this paper illustrates the effectiveness of the proposed estimation procedure in the presence of a two-stage model (i.e.,  $S = 2$ ). The procedure of conducting the experiment is illustrated in Algorithm 2.

Table 1 shows the mean and standard deviation of the fifty estimations for the scalar parameters  $\lambda_2^0$ ,  $M_1$ , and  $\omega_1$ . It can be observed that the mean of estimations for all three parameters are close to their true values. However, the variance of  $\omega_1$  is larger than that of  $M_1$  and  $\lambda_2^0$ . This may be due to the limited number of simulated events in each Hawkes process which downgrades the results estimated by MLE.

Table 1: True and estimated model parameters.

	$\lambda_2^0$	$M_1$	$\omega_1$
True values	3	1	2
Mean estimates	2.9921	0.9789	1.9142
Standard deviation	0.0486	0.0913	0.2971

This paper also investigates the MAE between the predicted event counts and the observed event counts in  $(t^*, t^* + L]$ . Specifically, we use the observations in  $[0, 9]$  (i.e.,  $t^* = 9$ ) to predict the event counts at time interval  $(9, 9 + L]$ . The MAEs of using the Poisson process and the proposed method are treated as the function of prediction window  $L$  which are shown in Figure 3.

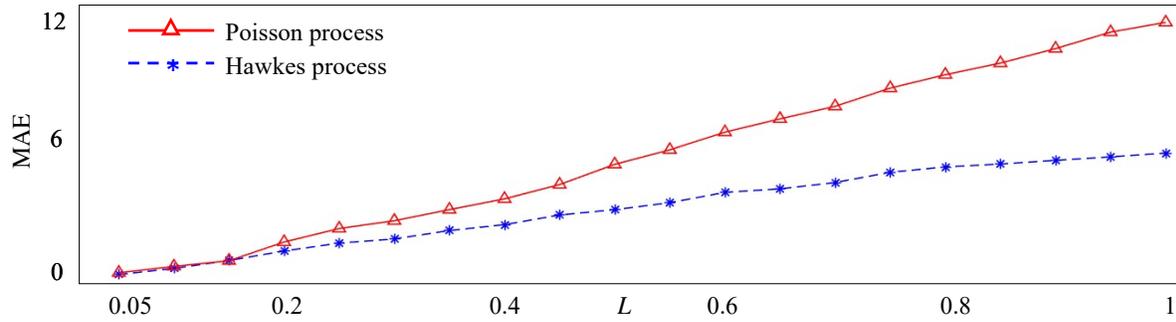


Figure 3: MAE of the predicted event counts during  $(t^*, t^* + L]$ .

There are two main findings from Figure 3: (i) the MAEs of both methods increase monotonically as prediction window length,  $L$ , increases. However, the MAE of Poisson process increases significantly faster than that of the proposed method. This is because the Poisson process does not consider the triggering mechanism and always underestimates the event counts which leads to the accumulation of MAE along the time; (ii) the MAE of the proposed method is consistently lower than the Poisson process approach.

### 3.2 Physics-based Simulation Case Study

Physics-based simulation is conducted to study the performance of the camera modality in an AV perception system. The error propagation between the data acquisition stage and the object detection stage is considered. In the data acquisition stage, the camera sensor generates streams of image of the driving scenarios, which are then transmitted to the object detection stage as input data. The object detection stage detects objects, such as leading vehicles, pedestrian and obstacles on the road, etc. The information of the detected objects, such as their locations, will be sent to proceeding functional stages for further processing to support the decisions on control actions.

In this research, a CARLA-Autoware simulation platform is built to provide high-fidelity simulation and error event data. Specifically, CARLA offers the models, physics and rendering engines for the simulation, while Autoware provides the control engine with a number of algorithms that process the data collected from the perception systems. The combination of these two modules enables the high-fidelity simulation, which is capable of not only injecting various types of faulty inputs to and generating erroneous outputs from different stages, but also connecting the functionalities along the multistage pipeline of the AV perception system.

In this case study, three types of erroneous outputs from the data acquisition stage will be generated by the injected faults. Severe weather leaves many rainfall droplets on the lens of the camera sensor, resulting in *distorted images*. Degrading CCD (charge-coupled device) of the camera sensor may create *noisy images* and even *damaged images*. These erroneous outputs from the acquisition stage will cause extra miss detection errors at the object detection stage.

Two different environment settings are built in CARLA-Autoware simulation platform to support error propagation modeling. In setting I, no fault is injected to the data acquisition stage to estimate baseline intensity of the object detection stage without error propagation. Error event data for the object detection stage are collected as  $\mathbf{t}^2$ , with the timestamps when a miss detection of objects is found. In setting II, the error propagation scenarios with injected faults in data acquisition stage are simulated. In addition to  $\mathbf{t}^2$  of miss detection event timestamps, event data from data acquisition stage are collected as  $\mathbf{t}^1$ , which includes the timestamps when an erroneous image is generated.

By employing the estimation method proposed in Section 2.3, the model parameters and the intensities at different stages are estimated by applying (3)-(7). Thus, the count of future miss detection events can be predicted by (8). This count is used to compare the performance of both the proposed method, which

explicitly considers error propagation, and the Poisson process approach, which does not take the error propagation into account. In addition, this paper compares both approaches in terms of the MAE between the predicted count and the ground truth count. Following the training-testing procedure, the error event data in the first 9 time units, i.e.,  $[0, 9]$  ( $t^* = 9$ ), is used to estimate model parameters and predicts the count of miss detection events in the time interval  $(9, 10]$ . The whole procedure is repeated thirty times and the predictive performances of both methods are illustrated in Figure 4.

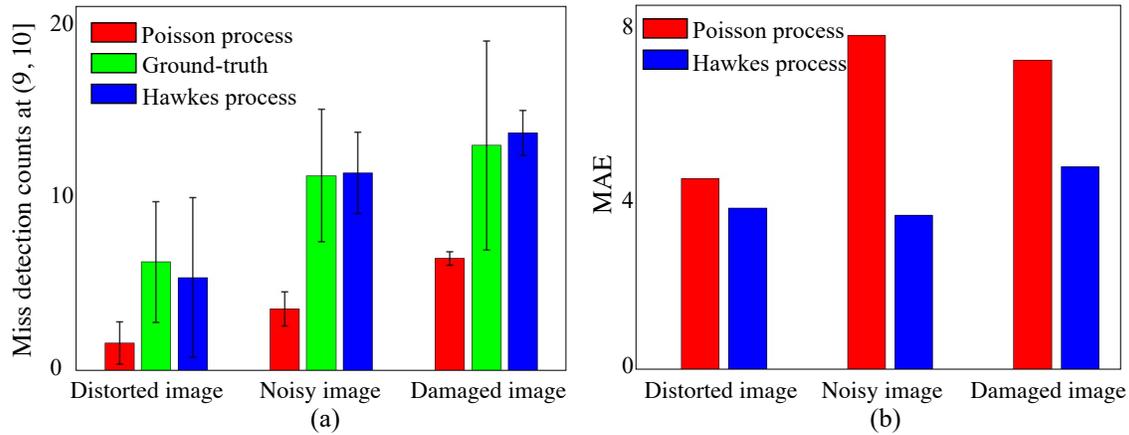


Figure 4: Method performance comparison based on physics-based simulation.

In Figure 4 (a), the green bar represents the ground truth counts of miss detection events in time interval  $(9, 10]$  when various types of faults were injected into the driving environment. These ground truth counts of error occurrence show that (i) the damaged images collected from the data acquisition stage will trigger the highest number of miss detection, (ii) the noisy images trigger comparatively less errors, and (iii) the distorted images trigger the lowest number of miss detection. The red bars and the blue bars represent the predicted counts in time interval  $(9, 10]$  by using Poisson process model and the proposed multi-stage Hawkes process model, respectively. It can be seen that the results predicted by Poisson process (red bars) are significantly different from the ground truth for all the simulated scenarios. This is because the Poisson process fails to take the propagation mechanism into consideration. While the results predicted by the proposed method are close to the ground truth, indicating its better performance with explicitly considering the error propagation mechanism.

The performance is also compared with respect to the MAE of the counts of miss detection event, as defined in (9). Figure 4 (b) shows that the MAE of the event occurrence counts in time interval  $(9, 10]$ . The MAE of the proposed method is significantly lower than that of the Poisson process approach. In terms of MAE of error prediction, the proposed method outperforms the Poisson process approach by 15.3%, 53.5%, and 34.2% with propagated errors of distorted image, noisy image, and damaged image, respectively. This indeed highlights the importance of explicitly considering the error propagation in the modeling for perception system evaluation.

#### 4 CONCLUSIONS

In this work, a high-fidelity physics-based simulation is adopted to provide fault injection in different stages of the AV perception system, which generates the error event data along different stages of an AV perception system for modeling the error propagation. Based on the error event data, this paper presents an MSHP to explicitly consider the latent error propagation for error occurrences prediction. The model parameters can be accurately and efficiently estimated with the proposed two-step estimation algorithm. The proposed model is applied in both the numerical case study and physics-based simulation case study,

and the results show its superiority by comparing it with the benchmark methods. This proposed general framework of physics-simulation based error propagation methodology has great potential to be applied in AV safety validation and improvement.

## ACKNOWLEDGMENTS

This material is based in part upon the work supported by the Arizona Commerce Authority under Agreement No. 2021-37. The authors would like to thank the collaborators who provided insight and expertise that greatly assisted the research.

## A APPENDICES

### A.1 Appendix I: Two-stage HP simulation algorithm

The simulation of two-stage HP with exponential kernel function on a fixed time interval is outlined in Algorithm 1.

---

**Algorithm 1** Simulate a two-stage HP with exponential kernel function on fixed time interval  $[0, T]$ .

---

```

1: Initialize the simulated point accept time  $d = 0$ ,  $\mathcal{T}^2 = \emptyset$ ,  $j = 0$ . Set the  $T = 20$ , the number of stages
    $S = 2$ , the injected event data in the first stage  $\mathcal{T}^1 = \{t_i^1\}_{i=1}^{n_1}$ , which is specified in a time interval  $[5, 10]$ 
   with 0.05 timestamp, the baseline parameters  $\lambda_2^0 = 3$ , and the triggering parameters  $\{M_1 = 1, \omega_1 = 2\}$ .
   Simulate  $\mathcal{T}^2 = \{t_j^2\}_{j=1}^{n_2}$ .
2: while  $t < T$ . do
3:   if  $\mathcal{T}^2 = \emptyset$  then
4:     Generate the first instance for the second stage
5:     Generate  $v \sim Uniform[0, 1]$  and set  $d = -\ln v / \lambda_2^0$ 
6:      $t_1^2 = d$ 
7:   else
8:     Set  $\bar{\lambda} = \lambda(d^+) = \lambda_2^0 + \sum_{t_i^1 < d} M_1 \times \omega_1 \times \exp(-\omega_1(d - t_i^1))$ 
9:     Generate  $u \sim Uniform[0, 1]$ 
10:    Generate  $w = -\ln u / \bar{\lambda}$  as the interarrival of the candidate point.
11:    Set  $d = d + w$  as the new candidate point.
12:    Generate  $D \sim Uniform[0, 1]$ 
13:    if  $D \leq \lambda(d) / \bar{\lambda}$  then
14:       $j = j + 1$ ;  $t_j^2 = d$ ;  $\mathcal{T}^2 = \mathcal{T}^2 \cup \{t_j^2\}$ 
15:    else
16:      Continue
17:    end if
18:  end if
19: end while
20: return  $\mathcal{T}^2$ 

```

---

The procedure of conducting the experiment is illustrated in Algorithm 2.

### A.2 Appendix II: Log-likelihood function of EM algorithm

**Estimation of triggering parameters:** By plugging in the  $P_{ij}$  obtained from Equation (5) in E-step to the log-likelihood function in Equation (4), the log-likelihood function can be rewritten by:

---

**Algorithm 2** Procedure for simulated two-stage HP experiments

---

- 1: Specify the number of replications  $N = 50$ .
  - 2: **for**  $n$  in  $1 : N$  **do**
  - 3:   Use Algorithm 1 to simulate the second stage HP data
  - 4:   Employ two-step estimation approach provided in Section 2.3
  - 5:   Save  $\{\hat{\lambda}_2^0, \hat{M}_1, \hat{\omega}_1\}$
  - 6: **end for**
  - 7: Report the mean and the standard deviation of estimation
- 

$$l_s(M_{s-1}, \omega_{s-1} | \hat{\lambda}_s^0, \{t_j^s\}_{j=1}^{n_s}) = \sum_{j=1}^{n_s} \sum_{i=1}^{n_{s-1}} p_{ij} \cdot \log(M_{s-1} \cdot \omega_{s-1} \cdot \exp(-\omega_{s-1} \cdot (t_j^s - t_i^{s-1}))) \\ + \sum_{j=1}^{n_s} P_{(n_{s-1}+1)j} \cdot \log(\hat{\lambda}_s^0) - \int_0^T \lambda_s(t) dt.$$

Taking the first partial derivative with respect to  $M_{s-1}, \omega_{s-1}$  and setting them equal to 0, the updating parameters in Equations (6) and (7) are obtained.

## REFERENCES

- Banerjee, S. S., S. Jha, J. Cyriac, Z. T. Kalbarczyk, and R. K. Iyer. 2018. "Hands Off the Wheel in Autonomous Vehicles?: A Systems Perspective on over a Million Miles of Field Data". *48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*:586–597.
- Brummelena, J. V., M. O'Brien, D. Gruyer, and H. Najjaran. 2018. "Autonomous Vehicle Perception: The Technology of Today and Tomorrow". *Transportation Research Part C: Emerging Technologies* 89:384–406.
- Campbell, S., N. O'Mahony, L. Kralcova, D. Riordan, J. Walsh, A. Murphy, and C. Ryan. 2018. "Sensor Technology in Autonomous Vehicles: A Review". *29th Irish Signals and Systems Conference (ISSC)*:1–4.
- Dosovitskiy, A., G. Ros, F. Codevilla, A. Lopez, and V. Koltun. 2017. "CARLA: An Open Urban Driving Simulator". *Conference on Robot Learning*:1–16.
- Erlien, S. M. 2015. *Shared Vehicle Control Using Safe Driving Envelopes for Obstacle Avoidance and Stability*. Stanford University.
- Farooq, S. U., S. Quadri, and N. Ahmad. 2012. "Metrics, Models and Measurements in Software Reliability". *IEEE 10th International Symposium on Applied Machine Intelligence and Informatics (SAMII)*:441–449.
- Fremont, D. J., E. Kim, Y. V. Pant, S. A. Seshia, A. Acharya, X. Brusio, P. Wells, S. Lemke, Q. Lu, and S. Mehta. 2020. "Formal Scenario-based Testing of Autonomous Vehicles: From Simulation to the Real World". *IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*:1–8.
- Gnedenko, B., I. V. Pavlov, I. A. Ushakov, and S. Chakravart. 1999. *Statistical Reliability Engineering*. John Wiley & Sons.
- Jha, S., S. S. Banerjee, J. Cyriac, Z. T. Kalbarczyk, and R. K. Iyer. 2018. "Avfi: Fault Injection for Autonomous Vehicles". *48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops*:55–56.
- Jha, S., S. S. Banerjee, T. Tsai, S. K. S. Hari, M. B. Sullivan, Z. T. Kalbarczyk, S. W. Keckler, and R. K. Iyer. 2019. "ML-based Fault Injection for Autonomous Vehicles: A Case for Bayesian Fault Injection". *CoRR* abs/1907.01051.
- Lieshout, V., and M.-C. NM. 2012. "On Estimation of the Intensity Function of a Point Process". *Methodology and Computing in Applied Probability* 14(3):567–578.
- Ogata, Y. 1981. "On Lewis' Simulation Method for Point Processes". *IEEE Transactions on Information Theory* 27(1):23–31.
- Pomerleau, D. 1989. "An autonomous land vehicle in a neural network".
- Richter, S. R., V. Vineet, S. Roth, and V. Koltun. 2016. "Playing for Data: Ground Truth from Computer Games". *European Conference on Computer Vision*:102–118.
- Rizoiu, M.-A., Y. Lee, S. Mishra, and L. Xie. 2017. "A Tutorial on Hawkes Processes for Events in Social Media". *arXiv preprint arXiv:1708.06401*.
- Thompson, W. 2012. *Point Process Models with Applications to Safety and Reliability*. Springer Science & Business Media.
- Wakabayashi, D. 2018, March. "Self-Driving Uber Car Kills Pedestrian in Arizona, Where Robots Roam". *The New York Times*.

## **AUTHOR BIOGRAPHIES**

**FENGLIAN PAN** is a Ph.D. student of the Department of Systems and Industrial Engineering, University of Arizona. Her research interests include quality engineering, data analytics, machine learning and optimization methodologies. Her e-mail address is [fenglianpan@email.arizona.edu](mailto:fenglianpan@email.arizona.edu).

**YINWEI ZHANG** is a Ph.D. candidate of the Department of Systems and Industrial Engineering, at the University of Arizona. His research interests include machine learning, statistical modeling and analysis, optimization and image processing. His e-mail address is [zhangyinwei@email.arizona.edu](mailto:zhangyinwei@email.arizona.edu).

**LARRY HEAD** is a Professor of Systems and Industrial Engineering at the University of Arizona. His research interests focus on cyber-physical systems, intelligent systems, connected and autonomous vehicles, urban traffic operations, transportation modeling, and intelligent Transportation Systems. His e-mail address is [klhead@arizona.edu](mailto:klhead@arizona.edu). His website is <https://sie.engineering.arizona.edu/faculty-staff/faculty/larry-head>.

**JIAN LIU** is an Associate Professor of the Department of Systems and Industrial Engineering, University of Arizona. His research interests focus on information fusion for high-dimensional spatial-temporal data analysis, which has been applied in computer vision, image and video processing, quality, reliability, and productivity improvement for manufacturing, and service processes.. His e-mail address is [jianliu@arizona.edu](mailto:jianliu@arizona.edu). His website is <https://sie.engineering.arizona.edu/faculty-staff/faculty/jian-liu>.

**MARIA ELLI** is a Data Scientist at Intel. She obtained her M. Sc. in Data Science from Indiana University Bloomington in 2017, with focus on machine learning and computer vision applications. Her research focuses on Automated Vehicles safety analysis using simulation and Safety Standards. Her e-mail address is [maria.elli@intel.com](mailto:maria.elli@intel.com).

**IGNACIO ALVAREZ** is a research scientist at Intel. He received his International Ph.D. in Computer Science from the University of the Basque Country (Spain) and Clemson University (USA) in 2011. Ignacio's research is focused on the development of intelligent, connected, automated vehicles that augment human mobility with safer and more enjoyable experiences. His e-mail address is [ignacio.j.alvarez@intel.com](mailto:ignacio.j.alvarez@intel.com).