Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Multi-Dimensional Failure Modeling for Shared Data in Cooperative Systems

Georg Jäger * **Konstantin Kirchheim** ** **Frank Schrödel** ***
Sebastian Zug *

* *Technische Universität Bergakademie Freiberg, 09599 Freiberg, Germany (e-mail: {georg.jaeger, sebastian.zug}@informatik.tu-freiberg.de).*
** *Otto-von-Guericke Universität Magdeburg, 39104 Magdeburg, Germany (e-mail: konstantin.kirchheim@ovgu.de)*
*** *Development Center Chemnitz/Stollberg, IAV GmbH, 09366 Stollberg, Germany (e-mail: frank.schroedel@iav.de)*

Abstract: Autonomous systems will share data to enrich their environmental model and provide cooperative functionality. However, as shared data might be imprecise or inaccurate, its failure characteristics have to be analyzed by the receiving system before using the data. A corresponding failure model for describing failure characteristics was proposed by Jäger et al. (2018), but is limited to one-dimensional sensory data. In this work, we extend the failure model to support multi-dimensional feature data as well. We exemplary evaluate the approach by modeling the failure characteristics of a lane detection system of a simulated car. By comparing it to state-of-the-art failure modeling techniques, we can show that the model accurately predicts failure amplitudes of previously unseen tracks even when trained on limited data.

Keywords: Failure Modeling, Cooperative Systems, Feature Data, Sensor Data, Shared Data, Failure Semantics, Lane Following, Lane Detection, Cooperative Sensing.

1. INTRODUCTION

Continuously exchanging individual state variables (speed, position) and environmental information (detected lane properties in uncharted areas, location of construction sides) between autonomous cars improves the overall safety and performance significantly. Received information extend the local environment model beyond the range of local sensors and facilitate proactive path planning or cooperative convoying Hobert et al. (2015). From a safety perspective, however, the flexible handling of externally aggregated data is a challenging task. On the one hand, quality information about its failure characteristics becomes available solely at run-time while safety is traditionally analyzed at design-time. On the other hand, current standards, such as EN 302 637-2 V1.4.1, only target driver assistance functionalities rather than autonomous cars and therefore define only coarse quality metrics (e.g. symmetric confidence intervals (ETSI TS 102 894-2 V1.3.1)) for a limited set of exchanged information. Contrarily, Jäger et al. (2016) exemplary identified the need for detailed descriptions of failure characteristics of external data when used in autonomous systems and define abstract *failure semantics*. These can be shared along with external data and matched with a system's fault tolerance during a runtime safety analysis to maintain its safety, e.g. by rejecting insufficiently accurate data. Jäger et al. (2018) refine the idea by postulating that abstract *failure semantics* need to be represented in terms of concrete failure models. Consequently, they develop a *generic (sensor) failure model (GFM)* as a detailed description of failure characteristics, which can be shared along with sensory data at runtime. This enables the receiving system not only to take uncertainties into account while planning, but also to check that the data's failure characteristics will not result in any safety violations. Furthermore, they provide a data-driven processing chain for constructing GFMs.

However, the approach proposed by Jäger et al. (2018) has two drawbacks. Firstly, the failure model and the processing chain, were designed and evaluated regarding one-dimensional sensory data only. Contrarily, cooperative systems will share multi-dimensional feature data as well (e.g. detected lanes). Secondly, the presented failure model uses artificial neural networks, which are black box models and therefore not interpretable. In safety-critical applications, however, interpretability of employed models is mandatory.

Thus, the present work aims at closing these gaps. We extend the GFM and its processing chain to support multi-dimensional feature data. Furthermore, we adapt the employed function approximation scheme to use polynomials, which have a clear interpretation.

The text is structured as follows: The next section will discuss exemplary state-of-the-art approaches on sensor failure modeling. We firstly review general approaches before we introduce the generic failure model of Jäger et al. (2018). In Section 3 we extend the generic failure model and its processing chain to overcome its drawbacks. The extended GFM is evaluated in Section 4 by modeling failures of a lane detection system of a simulated car. We can

show that its modeling performance outperforms state-of-the-art approaches while maintaining interpretability. The last section summarizes the contributions and states possible directions for future work.

2. STATE OF THE ART

In this section we briefly discuss the state of the art on modeling sensor failures. We firstly review general approaches in the next subsection before we introduce the generic failure model of Jäger et al. (2018) in the second subsection.

2.1 Sensor Failure Modeling

A sensor, considered as an individual system, is designed to provide observations of a certain phenomenon. In case of deviations between the provided observation $\hat{o}$ and the true value of the phenomenon o, the sensor fails to adhere to its specification meaning that a sensor failure has occurred.

The possible ways in which such deviations instantiate for a certain sensor is known as its failure characteristics and are commonly described by failure models. They vary in level of detail, which motivated Jäger et al. (2016) to group them in four failure semantics (*Arbitrary Failures*, *Bounded Failures*, *Modeled Failures*, or *None*)

The *None* failure semantics is satisfied only by ideal sensors emitting no measurement failures. It poses the strongest assumption ($\hat{o} = o$) on provided sensor observations and thereby dismisses the need for a failure model.

On the contrary, a sensor providing no information about its failure characteristics satisfies only the weakest failure semantics of *Arbitrary Failures*. For instance, sensors that provide Transducer Electronic Data Sheet (TEDS) (Song and Lee, 2008) to facilitate plug-and-play applications are not obligated to provide any description of their failure characteristics. Thus, they can satisfy only an *Arbitrary Failures* semantics.

The optional parts of TEDS, on the other hand, allow sharing manufacturer-defined data. It can be used to share a minimal description of failure characteristics, e.g. uncertainty margins (Taylor and Kuyatt, 1994) in the form of $\pm f$. These limit the true value o to be within an interval $[\hat{o} - f, \hat{o} + f]$ around the sensor observation $\hat{o}$ at any time and thereby satisfy the *Bounded Failures* semantics. Due to the lack of further information, one has to assume an uniform distribution of sensor failures f in this case.

Therefore, approaches satisfying the *Modeled Failures* semantics aim at providing more complete descriptions, e.g. by explicitly stating the distribution of sensor failures. Elnahrawy and Nath (2003), for instance, assume a Gaussian distribution with zero mean and a fixed standard deviation for failures of one- and multi-dimensional sensors in a sensor network. Adding knowledge about the distribution of sensor failures enables to reason about the validity of sensor readings, which they use to clean noisy sensor readings. Similarly, Rauscher et al. (2016) assume a Gaussian distribution of failures of depth cameras. However, they model the distribution's standard deviation as a function of the measured depth using quadratic polynomials to represent value-correlations of sensor failures.

Fagbemi et al. (2019) take a more complex approach. They consider a Gaussian distribution of failure amplitudes $f \sim \mathcal{N}(\mu, \sigma_D)$ as the *Noise* failure type ($\hat{o}_{Noise} = o + f$), which is one out of nine failure types comprising their failure model. Moreover, they define failure types such as *Scaling* ($\hat{o}_{Scaling} = \sigma \cdot o$), which affects a sensor observation multiplicative, or *Bias* ($\hat{o}_{Bias} = o + \beta$) which has an additive effect. Their work is embedded into a simulation of unmanned air systems which renders the presented failure model application specific and requires experts to manually determine the model's parameters (e.g. σ, β). Nevertheless, failure type based failure models can be found throughout the literature. Heredia et al. (2008) present a failure model comprising five linguistically defined failure types for an fault detection isolation (FDI) system for an autonomous helicopter. Ni et al. (2009) present linguistic definitions of nine failure types for sensor networks and propose different features, e.g. mean, variance, and gradient, for modeling them. Jäger et al. (2014) describe features to detect four failure types (*Outlier*, *Constant Offset*, *Constant Noise*, *Stuck-At-Zero*) using artificial neural networks.

2.2 Generic Sensor Failure Model

Another approach satisfying the *Modeled Failures* semantics is the generic failure model of Jäger et al. (2018). Defining a time series of failure amplitudes over the discrete time index $k \in \mathbb{N}_0$ as $f(k, o_k) = \hat{o_k} - o_k$, they take a data-driven approach. They define the failure model $\mathcal{M}$ as a set of failure types $\mathcal{M} = \{F_1, \ldots, F_N\}$ with $|\mathcal{M}| = N$ where each failure type F_n consists of a state-function $s_n(k, o_k)$ and a failure amplitude $f_n(k, o_k)$. Here, $o_k \in \mathbb{R}^b$ denotes the true value of the observed phenomenon while $\hat{o_k}$ is the (possibly faulty) sensor observation, as before. Note that in Jäger et al. (2018) the number of dimensions is restricted to $b = 1$.

While a failure type's state function describes when a failure type is active ($s_n(k, o_k) = 1$) or inactive ($s_n(k, o_k) = 0$), its failure amplitude $f_n(k, o_k)$ describes the effect the failure type has on the sensor observation. All failure types $F_1, \ldots, F_N$ of a model $\mathcal{M}$ are independent and therefore impose each other additively:

$$f(k, o_k) = \sum_{n=1}^{N} s_n(k, o_k) \cdot f_n(k, o_k) \tag{1}$$

In other words, the failure types decompose an overall failure amplitude $f(k, o_k)$ into several, failure type specific failure amplitudes $f_n(k, o_k) \in \mathbb{R}^b$. For the n-th failure type $f_n(k, o_k)$ is defined as

$$f_n(k, o_k) = p_n((k - k_0)/K_n) \cdot m_n(k, o_k) \tag{2}$$

where k_0 is the time step at which the failure type was activated and K_n is the duration for which it is active. The function $p_n((k - k_0)/K_n) \in [-1, 1]^b$ is a normalized, deterministic failure pattern. It enables the GFM to not only model Noise-like failure characteristics, but also to represent pattern that evolve over time. Hence, dependencies between failure amplitudes at different time steps can be represented. To model the stochastic nature of a failure type, the failure pattern p_n is scaled by $m_n(k, o_k) \in \mathbb{R}^b$. The value of this function follows a *time-*

and value-correlated random distribution (Jäger et al., 2018)

$$m_n(k, o_k) = \sigma(k, o_k) \cdot Q(u) + \mu(k, o_k) \quad (3)$$

The distribution of the function values is represented by its inverse cumulative distribution function (or quantile function) $Q(u) \in \mathbb{R},\ u \in [0, 1]$. In other word, by provisioning an appropriate function $Q(u)$ an uniformly distributed random variable u can be transformed to follow an arbitrary distribution. As this mapping is static, that is, not dependent on time step k or value o_k, the transformed random variable is multiplied by $\sigma(k, o_k) \in \mathbb{R}^b$ and shifted by $\mu(k, o_k) \in \mathbb{R}^b$ to introduce such correlations. Consequently, these functions represent the standard deviation and mean of the transformed distribution. Note that the use of quantile functions renders the GFM to be a generative model.

For representing the functions describing a failure type, Jäger et al. (2018) employ Radial Basis Function (RBF) networks (Kruse et al., 2016), a variant of artificial neural networks, that can be used for universal function approximation (Park and Sandberg, 1991).

To increase the applicability of the failure model, the authors additionally present a processing chain capable of extracting a failure model from a given series of failure amplitudes $f(k, o_k)$.

Jäger et al. (2018) evaluate the processing chain and the generic failure model by representing the failure characteristics of an one-dimensional infra-red distance sensor. They show that the failure model out-performs uniform and normal distributions and performs similar to Multi-Layer Perceptrons(MLP) (Kruse et al., 2016).

3. EXTENSION TO THE MULT-DIMENSIONAL CASE

The generic failure model of Jäger et al. (2018) is limited to one-dimensional data by its use of quantile functions $Q(u)$, see Eq. (3). In general, given a random variable X, quantile functions map a probability $u \in [0, 1]$ to the value $x \in \mathbb{R}$ for which $Q^{-1}(x) = P(X \leq x) = u$ is true. For one-dimensional random variables, the quantile function can be constructed by inverting its cumulative distribution function such that $P(X \leq Q(u)) = u$. In case of a multi-dimensional random variable, however, $x \in \mathbb{R}^b$ is a vector, which implies that $Q(u)$ needs to map from a scalar to a vector ($Q : [0, 1] \Rightarrow \mathbb{R}^b$). As this function is not injective in general, we can not invert the cumulative distribution function for constructing the quantile function.

Hence, for extending the GFM to support multi-dimensional feature data, we firstly discuss different approaches to multi-variate quantile functions in the next subsection. Deeming the approach of *standard construction* as most suitable, we apply it to the GFM for representing multi-variate qunatile functions in Section 3.2. In the last subsection, we briefly discuss extensions that are required for the processing chain to construct multi-dimensional GFMs.

3.1 Multi-Variate Quantile Functions

The main challenge when defining multi-variate quantile functions is that no natural ordering for dimensions greater than one exists (Serfling, 2002; Belzunce et al., 2007).

Several approaches to circumvent this problem where proposed. For instance, Liu et al. (1999) use statistical depth-functions and defined quantiles as their superlevel sets. However, as these are thereby sets of vectors and not unique vectors, they are unsuitable for use in the GFM. In contrast, Chaudhuri (1996) consider a vector $u \in [0, 1]^b$ instead of a scalar probability to realize a unique mapping $[0, 1]^b \Rightarrow \mathbb{R}^b$ based on norm minimization. However, the interpretation of $||u||$ as a probability does not hold for $b > 1$ (Serfling, 2002).

Finally, in the field of simulation, the approach of *standard construction* (Joslin, 2012; Fernandez-Ponce and Suarez-Llorens, 2003) generalizes quantile functions to multi-dimensional data by using the indices of the dimensions as an replacement for the natural ordering. Let $X = [X_1, \ldots, X_b]$ be a random vector in $\mathbb{R}^b$ and $u = [u_1, \ldots, u_b] \in [0, 1]^b$ then we can *sample* a random vector X of distribution $Q(u)$ as follows (Joslin, 2012):

$$\begin{aligned} x_1(u_1) &= Q_{(X_1)}(u_1) \\ x_2(u_1, u_2) &= Q_{(X_2|X_1=x_1(u_1))}(u_2) \\ &\ldots \\ x_b(u_1, \ldots, u_b) &= Q_{(X_n|\cap_{j=1}^{d-1} X_j = x_j(u_1,\ldots,u_j))}(u_b) \end{aligned}$$

Here, $Q_{(X_i|\ldots)}(u)$ denotes the uni-variate quantile function of the marginal distribution of the i-th component given that all components $j < i$ have values $x_1, \cdots, x_j$. Thus, a dependency structure based on the chosen ordering of dimensions is imposed on the actual distribution of the random vector. Note that, opposed to the previous approach, the values $u_1, \ldots, u_b$ retain their interpretation as probabilities.

3.2 Extending the Failure Model

Although the approach based on norm minimization and the *standard construction* approach both facilitate the usage of quantile functions within the GFM, the latter retains a probabilistic interpretation of the elements of u. Therefore, we use this approach to represent the quantile function $Q(u) \in \mathbb{R}^b$ of the failure types of the GFM, see Eq. (3). However, allowing $b \geq 1$ renders the functions $\mu(k, o_k)$ and $\sigma(k, o_k)$ to be vector functions as well. This extension is inline with the interpretation of $\mu(k, o_k) \in \mathbb{R}^b$ which still provides the *time- and value correlated* mean of the distribution. Contrarily, σ would be required to provide a $b \times b$ covariance matrix. This squares the number of values to represent by the σ function. Furthermore, values apart from the diagonal of the covariance matrix represent correlations between dimensions of the random vector. As these are captured by the quantile function $Q(u)$ and its internal dependency structure already, we assume that these values are zero. Consequently, we define the function $\sigma(k, o_k) \in \mathbb{R}^b$ to provide an element-wise scaling of the random vector produced by the quantile function $Q(u)$.

Besides the definition of the failure model, the function approximation scheme employed for representing a failure type's functions is affected by allowing multi-dimensional data as well. It has to support vector functions. While this is the case for RBF networks, as proposed in Jäger et al. (2018), they are considered as black box models and therefore lack a clear interpretation.

To overcome this drawback, we employ multi-variate and multi-response polynomial regression. A polynomial represents the function $G : \mathbb{R}^I \implies \mathbb{R}^R$ where I denotes the number of independent variables and R denotes the number of response variables. Given the order of the polynomial as $D \in \mathbb{N}_{\geq 1}$, each response variable g_r is represented as:

$$g_r = \Phi_0 + \sum_{i=1}^{I} \sum_{d=1}^{D-(i-1)} \sum_{c \in C_i} \Phi_z \cdot \prod_{j \in c} x_j^d \tag{4}$$

with $\hat{I} = \{i | i \in \mathbb{N}_{\geq 1} \wedge i \leq I\}$ being the set of indices of the independent variables and $C_i = \binom{\hat{I}}{i}$ denoting the set of their i-combinations. z is merely an index to distinguish the parameters of the polynomial.

According to Eq. (4) a polynomial with $D = 2,\ I = 2,\ R = 1$ would be:

$$g = \Phi_0 + \Phi_1 x_1 + \Phi_2 x_2 + \Phi_3 x_1^2 + \Phi_4 x_2^2 + \Phi_5 x_1 x_2 \tag{5}$$

As one can see, the response g is a weighted sum of multiplicative combinations of the independent variables and their powered versions. Thus, rearranging the elements in matrices enables the multi-response polynomial to be written as a linear system:

$$G = \hat{X} \cdot \Phi \tag{6}$$

As the independent variables are directly reflected by the $\hat{X}$ matrix and their effect on the response variables are explicitly stated by the Φ parameters, the interpretation of a polynomial is clearer compared to artificial neural networks as they were used by Jäger et al. (2018).

3.3 Extending the Processing Chain

Apart from the failure model, Jäger et al. (2018) introduced a processing chain comprised of three steps to extract a GFM. The inputs to the processing chain are time series of ground truth values o_k and sensor observations $\hat{o}_k$ from which failure amplitudes $f(k, o_k) = \hat{o}_k - o_k$ are calculated. In accordance with the failure model, we shortly review the steps and describe necessary adaptations to support multi-dimensional data.

Identifying Failure Types The goal of the first step is to identify a preliminary set of $\hat{N}$ failure types and their occurrences to facilitate a decomposition of the given failure amplitudes $f(k, o_k)$ according to Eq. (1). In that endeavor, an optimization based pattern recognition algorithm is applied. The algorithm optimizes randomly generated patterns to match the failure amplitudes $f(k, o_k)$ and to occur as frequently as possible. The match between a failure type and the failure amplitudes boils down to calculating the area between both curves:

$$v_1 = \sum_k ||f(k, o_k) - \sum_{n=1}^{\hat{N}} s_n(k, o_k) \cdot f_n(k, o_k)||_1 \tag{7}$$

As the L1-norm can be evaluated for vector functions as well, this step supports multi-dimensional data already. Patterns that not exceed a predefined threshold of v_1 are accepted as failure patterns and form, together with their occurrences, a failure type. The algorithm generates and optimizes failure patterns until v_1 becomes zero. Hence, the set $\mathcal{M}_1$ of $\hat{N}$ failure types allow to completely reconstruct $f(k, o_k)$ according to Eq. (1). The identified failure types are passed to the second step.

Generalizing Failure Types The second step aims at generalizing the obtained failure types. Failure types with similar patterns or similar occurrences are identified and merged into a single failure type. This causes a loss of information. Thus, reconstructing failure amplitudes according to Eq. (1) produces $\hat{f}(k, o_k)$ instead of $f(k, o_k)$ and the loss of information is assessed via:

$$v_2 = \sum_k ||f(k, o_k) - \hat{f}(k, o_k)||_1 \tag{8}$$

To limit the loss of information, the process is repeated until $v_2 \geq \epsilon$, where ϵ is a predefined threshold. Again, extending the processing chain requires to evaluate Eq. (8) for vector functions, which was solved for the previous step already. The output of this step is the set $\mathcal{M}_2$ of N failure types.

Parameterizing Failure Types To obtain the final set of failure types, their functions (see Sec. 2.2) will be represented by polynomials. The corresponding training data for fitting these is extracted from the occurrences of the generalized failure types using the sliding window approach from Jäger et al. (2018), which allows to identify time and value correlations. Furthermore, to increase the robustness of the polynomials, we normalize the training data to have zero mean and a standard deviation of one before we apply Conjugate Gradient (Hestenes and Stiefel, 1952) to solve the linear system of Eq. (6) and obtain the parameter matrix Θ for each polynomial. The output of this step is the failure model $\mathcal{M}$ with N parametrized failure types.

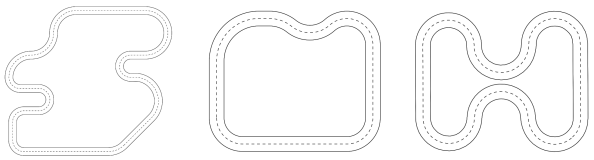
4. USE CASE: FAILURE MODELING OF A LANE DETECTION ALGORITHM

The GFM and its extensions presented in the last section are designed to be used in cooperative systems sharing data about the environment and objects detected therein, see Sec. 1. Specifically, in automotive scenarios, detected lanes might be shared between vehicles to facilitate cooperative sensing or even cooperative maneuvers. However, as abstract representations of detected lanes are generated by complex processing chains operating on inaccurate sensor data, e.g. cameras, they exhibit failures as well (Hillel et al., 2014). Thus, in this section we apply the GFM and other state-of-the-art approaches ranging from uniform- and normal distributions to artificial neural networks to model the failure characteristics of the lane detection system of a simulated, autonomous car.

The next subsection introduces the simulation setup that was used to generate training data for designing the failure models and test data to assess their performance. In subsection 4.3 we present all approaches and their training procedures before we discuss the applied evaluation metric in Subsection 4.4. We present our results in Subsection 4.5.

4.1 Simulation Setup

For modeling failure characteristics, both, ground truth data (o_k) as well as sensor/feature data ($\hat{o}_k$) have to be



(a) Track 1 (b) Track 2 (c) Track 3

Fig. 1. Tracks on which the autonomous car was simulated.

available. To provide both, we simulate an autonomous car on a scale of 1:10 on different tracks (see Fig. 1) using Gazebo (Koenig and Howard, 2004). The software facilitates simulating individual sensors, such as the car's front camera, which provides grayscale images of the road to the lane detection system. The lane detection starts by transforming received images into a birds-eye view, see Fig. 2. To cope with noise and outliers, morphological operations (erosion and dilation) are applied. Continuous white areas in the resulting image are then assumed to be lane markings and extracted as polylines. Finally, clothoids are fitted as the finale representation of the detected lane markings.

Clothoids are favored representations for lane markings due to their use in road constructions (Marzbani et al., 2015). A clothoid is a curve with a linearly changing curvature that can be described by three parameters: c_0 describes the initial curvature of the curve, c_1 describes the curvature change along the curve and Θ describes the orientation of the curve in a 2D coordinate frame. These three parameters are considered as one observation $\hat{o}_k = [c_0, c_1, \Theta]$. Note that we ignore the starting position of the curve as we are focused on the failures affecting the extracted clothoids itself.

For each detected lane in the image, a separate clothoid is produced by the system. Thus, in our scenario the system produces separate observations $\hat{o}_k$ for the left-left lane (LL), left lane (L), and right lane (R).

Despite simulating the scenario, uncertainties inherently present in the lane detection system cause failures in each of the clothoid's parameters, which we aim to model. For instance, as one can see in the upper right corner of Fig. 2, the perspective transformation of the camera image into a birds-eye view causes the width of lane markings to be doubled from 8 pixel to 16 pixels. The blurred lane markings result in uncertain clothoid parameters, which can be seen in Fig. 4a. It shows an excerpt of the failures of the c_0 parameter produced by the lane detection system for Track 1 on the left-left lane.

4.2 Training and Test Data Aggregation

The simulation environment described in the last subsection facilitates the acquisition of ground truth data (i.e. ideal observations o), which is available from the tracks, and observations $\hat{o}$ from the lane detection system. The

Table 1. Parameters of the tracks.

Track	Radii in [m]	Length in [m]
1	{1.12, 2.25, 2.41}	64
2	{1.41, 1.61, 1.91, 2.29}	28
3	{1.41}	34

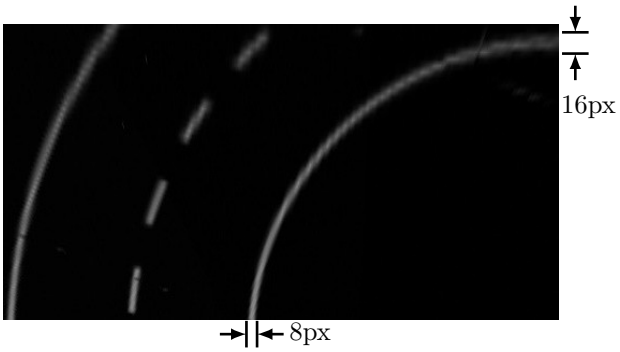


Fig. 2. Exemplary topview projection of the front camera image used during lane detection. While lane markings close to the car are sharp (width of 8 pixel), lane markings in the top of the image are blurred (width of 16 pixel) due to the perspective transformation.

difference between both constitutes the failure amplitudes that we aim to model. We simulated the car on each track for each driving direction (clock-wise, counter clock-wise) for 250 seconds which resulted in approx. 12.000 observations. During the simulation, the velocity of the car was set to 1 ms^{-1}.

In contrast to our simulation setup, constructing failure models for real-world lane detection systems would require training and test data from real-world experiments. In this case, facilitating training of failure models on a minimal set of data while maintaining their generalization capabilities is mandatory for tractable failure modeling approaches. Following the same suit, we use only 1/3 of the available data for training and 2/3 for testing. More specifically, we use data obtained from Track 1 for training the failure models. The data is assumed to contain most information about the failure characteristics of the lane detection algorithm due to its increased length and the number of curves and their varying radii, see Tab. 1. Complementary, data from Track 2 and 3 is used for testing.

4.3 Modeling Approaches

The training data comprised by failure amplitudes $f(k, o_k)$ and ground truth data o_k is used to design five failure models, which are summarized in Tab. 2. Each model represents the failure characteristics of the lane detection system regarding the clothoid parameters c_0, c_1, and Θ. Thus, a single model represents the failure amplitudes of all lanes (LL, L, R).

The first approach is an uniform distribution that models failure characteristics as the minimal and maximal failure amplitude in each dimension. It is similar to the uncertainty margins Taylor and Kuyatt (1994) and therefore satisfies the *Bounded Failures* semantics and serves as a baseline approach. The approach is trained by storing the minimal and maximal failure amplitude of all parameters contained in the training data.

In accordance with Section 2.1, the second approach is a normal distribution which satisfies the *Modeled Failures* semantics. It uses the mean and standard deviation of each parameter to represent a failure characteristics.

Table 2. Number of parameters of each model

Model	Parameters	Time- and Value-Correlated	Temporal Pattern
Uniform	2×3	No	No
Normal	2×3	No	No
ICDF	-	No	No
LSTM	2648	Yes	Implicit
GFM	2653	Yes	Yes

As both of these approaches lack in representing the true distribution of failure amplitudes, the third approach is an inverse cumulative distribution function (ICDF) which is non-parametric but uses the training data directly. It perfectly represents the distribution of failure amplitudes, but can neither represent time- and value correlations nor temporal failure patterns.

Thus, we train a Long Short-Term Memory (LSTM) network, a variant of recurrent neural networks, which proved useful in predicting time series, e.g. for traffic speed (Ma et al., 2015). In contrast to the other approaches, the LSTM network is a discriminative model that directly predicts a failure amplitude $f(k, o_k)$. To match the number of independent parameters of the GFM (see Tab. 2), we designed the network to have 23 LSTM neurons followed by a fully-connected, linear layer providing the three-dimensional output.

Lastly, we applied the processing chain (see Sec. 3.3) to extract a GFM. Starting with polynomials of order $D = 7$, we could reduce D to 5 for polynomials representing failure patterns and $D = 3$ for polynomials representing the scaling functions in Eq. (3) while maintaining the models performance. The processing chain identified 7 failure types with 379 parameters each.

The models are summarized in Tab. 2. As one can see, the uniform and normal distributions require the least amount of parameters to be determined, but have limited power for representing failure characteristics. The ICDF approach, although capable of representing the true distribution, requires no parameters but directly uses the training data. In contrast, only the LSTM and GFM approaches facilitate representing time- and value-correlations as well as (implicit or explicit) temporal patterns. The increased complexity, however, requires an increased number of parameters, which in turn require an appropriate amount of training data.

4.4 Evaluation Metric

We follow the idea of Jäger et al. (2018) to assess the performance of the constructed failure models. Firstly, we apply Monte-Carlo simulations to predict failure amplitudes $f_x(k, o_k)$ with $x \in \{Uniform, Normal, ICDF, LSTM, GFM\}$. In a second step, we compare the predicted failure amplitudes $f_x(k, o_k)$ with the original failure amplitudes $f(k, o_k)$ generated by the lane detection system. The comparison is based on a sliding window approach which defines a time interval $[k_w, k_w + K_w]$ of size K_w. The size of the window was set to $K_w = 120$ to match the time needed for the car to pass a curve and thereby capture value-correlations within the failure amplitudes. To assess them appropriately, we shifted the window with a step size of 12 which produces an overlap of 90% between subsequent windows. In each window we consider the predicted and the original failure amplitudes as two random distributions. Jäger et al. (2018) use the Kolmogorov-Smirnov (KS) statistic for comparing them. However, as the statistic is restricted to one-dimensional data, we employ the Cramer-von-Mises statistic which was extended to multi-dimensional data in Baringhaus and Franz (2004). Applying it to each window of the sliding window approach yields a time series of scalar values. By calculating the mean over all windows, we obtain a separate assessment value for each lane (LL, L, R) and each track (1,2,3), see Tab. 3-5. In general, greater values indicate a greater mismatch between the original failure amplitudes $f(k, o_k)$ and the predicted failure amplitudes $f_x(k, o_k)$. However, one can not assume a linear relationship, that is, a doubled performance value does not directly imply a doubled mismatch between predicted and original failure amplitudes.

Table 3. Results from Track 1

Model	LL	L	R	Σ	Rank
Uniform	49.82	51.7	54.21	311.45	5
Normal	4.11	2.54	3.11	19.52	4
ICDF	2.98	0.68	0.77	8.86	2
LSTM	3.48	1.94	1.25	13.34	3
GFM	1.01	0.66	0.32	3.97	1

Table 4. Results from Track 2

Model	LL	L	R	Σ	Rank
Uniform	49.44	52.35	55.49	314.57	5
Normal	3.26	2.37	3.15	17.57	4
ICDF	2.05	0.39	0.49	5.87	2
LSTM	2.7	1.55	1.26	11.03	3
GFM	1.2	0.33	0.22	3.5	1

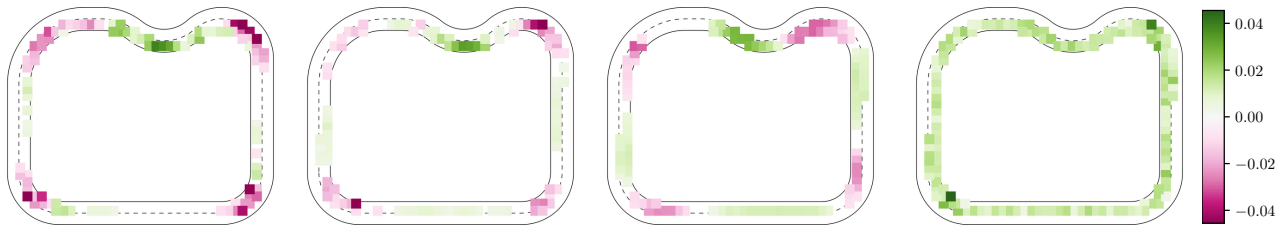
Table 5. Results from Track 3

Model	LL	L	R	Σ	Rank
Uniform	47.38	47.52	53.44	296.7	5
Normal	4.56	1.61	3.12	18.58	4
ICDF	4.43	0.82	0.83	12.16	2
LSTM	3.94	2.19	2.09	16.44	3
GFM	2.01	0.88	0.91	7.59	1

4.5 Results

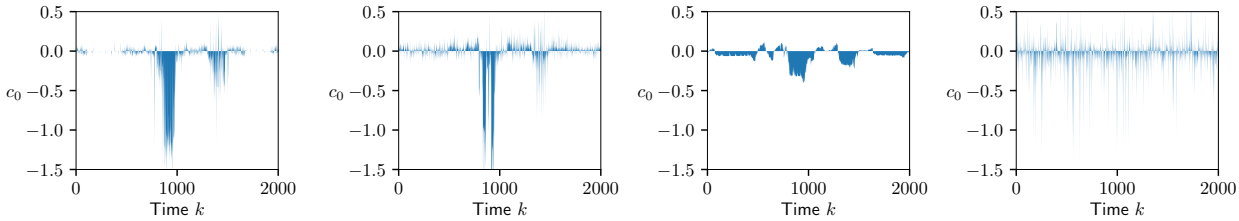
The results are summarized in Tab. 3-5. Tab. 3 assess the models performance regarding their training data while Tab. 4 and 5 assess their performance when applied to previously unseen scenarios.

As one can see, the uniform distribution achieves the worst performance. This matches the state-of-the-art discussion where the uniform distribution, as part of the *Bounded Failures* semantics, is considered to carry only limited information about a sensor's failure characteristics. Similarly, the normal distribution assumes a predefined distribution and calculates mean and standard deviation to be the most likely fit. The summed performance values over all three lanes on each track show that this assumption results in much greater performance compared to a uniform distribution. The ICDF follows this trend as it is capable of perfectly representing the true distribution of failure amplitudes. However, comparing its performance



(a) Real (b) GFM (c) LSTM (d) ICDF

Fig. 3. Failures of the Θ parameter of the right lane markings on Track 2 averaged over approximately 6 rounds.



(a) Real (b) GFM (c) LSTM (d) ICDF

Fig. 4. Excerpt of failures amplitudes of the c_0 parameter of the left-left lane markings of Track-1.

values to those achieved by the LSTM and GFM uncovers a disadvantage of the Cramer-von-Mises statistic. The ICDF approach is ranked higher than the LSTM model despite its inability to represent temporal patterns or value correlations. This is underlined in Fig. 4 where excerpts of the failure amplitudes of the c_0 parameter for Track 1 are shown. While the LSTM network as well as the GFM model are capable of representing the negative Spike pattern around $k = 1000$, the ICDF model produces only a Noise pattern. The same behavior can be observed for the Θ parameter when simulating failure amplitudes for Track 2 (see Fig. 3), which results in an almost homogeneous colorization of the track in case of the ICDF approach. In contrast, the LSTM and GFM model represent value correlated failure amplitudes and therefore predict negative failures for Θ in right-hand bends and positive failures in left-hand bends. However, the LSTM network misses to model the noise behavior of the real failure amplitudes, see Fig. 4.

This shows the advantage of the GFM, where a separate failure type captures this behavior and therefore reproduces the failure amplitudes more realistically.

A similar statement can be made about the minimal and maximal failure amplitudes. As shown exemplary in Fig. 3, the GFM can predict outliers to occur in curves resulting in dark red and green spots. The LSTM network, on the other hand, under estimates their magnitude and therefore produces more faint colors. This underlines the ranks in Tab. 3-5, where the GFM is ranked first.

5. CONCLUSIONS AND FUTURE WORK

The present work focuses on extending the generic failure model (GFM) proposed by Jäger et al. (2018) to support multi-dimensional feature data. In that endeavor, we identified the use of quantile functions as the limitation of the GFM. By employing multi-variate quantile functions based on the approach of *standard construction* and adapting the processing chain for constructing GFMs accordingly, we overcame the limitation. Moreover, we introduced polynomials to represent the functions of the GFM, which ensures interpretability of the model. By applying the GFM to model the failure characteristics of a lane detection system of a simulated car, we could show that the GFM outperforms commonly used failure models (uniform distribution, normal distribution) and even achieves a modeling performance comparable to Long Short-term Memory (LSTM) networks. Most importantly, these results were achieved by training the GFM using only data from one track while predicting failure amplitudes for two previously unseen tracks. This underlines the generalization capabilities of the proposed model.

In future work, we aim at exploiting these generalization capabilities. We plan to construct artificial tracks that yield the minimal amount of training data while providing a maximal performance of the trained GFM. Such an artificial track shall further be used as a template for finding suitable tracks for real-world experiments. To facilitate this work, the evaluation metric based on the Cramer-von-Mises statistic has to be refined further to resolve contradictions as raised in the last section.

Another direction for future work will be the run-time safety analysis. For that, the use of polynomials in GFMs enables us to consider confidence bands (Scheffe, 1999) to (i) state the goodness of fit during training the failure model and (ii) as a means to predict upper and lower bounds of failure amplitudes along with the most likely failure amplitude. The thereby increased interpretability of the GFM shall be used during the abstract run-time safety-analysis of Jäger et al. (2016) based on failure semantics to maintain safety in cooperative systems.

ACKNOWLEDGEMENTS

We thank Christian Speich and Moritz Bösenberg for their contribution to setting up the simulation environment and the evaluation scenario.

REFERENCES

Baringhaus, L. and Franz, C. (2004). On a new multivariate two-sample test. *Journal of multivariate analysis*, 88(1), 190–206.

Belzunce, F., Castaño, A., Olvera-Cervantes, A., and Suárez-Llorens, A. (2007). Quantile curves and dependence structure for bivariate distributions. *Computational Statistics & Data Analysis*, 51(10), 5112–5129.

Chaudhuri, P. (1996). On a geometric notion of quantiles for multivariate data. *Journal of the American Statistical Association*, 91(434), 862–872.

Elnahrawy, E. and Nath, B. (2003). Cleaning and querying noisy sensors. In *Proceedings of the 2Nd ACM International Conference on Wireless Sensor Networks and Applications*, WSNA '03, 78–87. ACM, New York, NY, USA.

EN 302 637-2 V1.4.1 (2019). Intelligent transport systems (its); vehicular communications; basic set of applications; part 2: Specification of cooperative awareness basic service. Standard, European Telecommunications Standards Institute (ETSI).

ETSI TS 102 894-2 V1.3.1 (2018). Intelligent transport systems (its); users and applications requirements; part 2: Applications and facilities layer common data dictionary. Technical specification, European Telecommunications Standards Institute (ETSI).

Fagbemi, M., Perhinschi, M.G., and Al-Sinbol, G. (2019). Modeling of upset sensor operation for autonomous unmanned systems applications. *International Journal of Intelligent Unmanned Systems*, 7(1), 19–34.

Fernandez-Ponce, J. and Suarez-Llorens, A. (2003). A multivariate dispersion ordering based on quantiles more widely separated. *Journal of Multivariate Analysis*, 85(1), 40 – 53.

Heredia, G., Ollero, A., Bejar, M., and Mahtani, R. (2008). Sensor and actuator fault detection in small autonomous helicopters. *Mechatronics*, 18(2), 90–99.

Hestenes, M.R. and Stiefel, E. (1952). *Methods of conjugate gradients for solving linear systems*, volume 49. NBS Washington, DC.

Hillel, A.B., Lerner, R., Levi, D., and Raz, G. (2014). Recent progress in road and lane detection: a survey. *Machine vision and applications*, 25(3), 727–745.

Hobert, L., Festag, A., Llatser, I., Altomare, L., Visintainer, F., and Kovacs, A. (2015). Enhancements of v2x communication in support of cooperative autonomous driving. *IEEE Communications Magazine*, 53(12), 64–70.

Jäger, G., Brade, T., and Zug, S. (2016). Using failure semantics to maintain safety for dynamic composed systems. In *ARCS 2016; 29th International Conference on Architecture of Computing Systems*, 1–7.

Joslin, P. (2012). *Multivariate comparisons of random vectors with applications*. Ph.D. thesis, Universidade de Murcia.

Jäger, G., Zug, S., Brade, T., Dietrich, A., Steup, C., Moewes, C., and Cretu, A. (2014). Assessing neural networks for sensor fault detection. In *2014 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, 70–75.

Jäger, G., Zug, S., and Casimiro, A. (2018). Generic sensor failure modeling for cooperative systems. *Sensors*, 18(3).

Koenig, N. and Howard, A. (2004). Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, volume 3, 2149–2154. IEEE.

Kruse, R., Borgelt, C., Braune, C., Mostaghim, S., and Steinbrecher, M. (2016). *Computational intelligence: a methodological introduction*. Springer.

Liu, R.Y., Parelius, J.M., and Singh, K. (1999). Multivariate analysis by data depth: descriptive statistics, graphics and inference, (with discussion and a rejoinder by liu and singh). *Ann. Statist.*, 27(3), 783–858.

Ma, X., Tao, Z., Wang, Y., Yu, H., and Wang, Y. (2015). Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies*, 54, 187–197.

Marzbani, H., Jazar, R.N., and Fard, M. (2015). Better road design using clothoids. In *Sustainable Automotive Technologies 2014*, 25–40. Springer.

Ni, K., Ramanathan, N., Chehade, M.N.H., Balzano, L., Nair, S., Zahedi, S., Kohler, E., Pottie, G., Hansen, M., and Srivastava, M. (2009). Sensor network data fault types. *ACM Trans. Sen. Netw.*, 5(3), 25:1–25:29.

Park, J. and Sandberg, I.W. (1991). Universal approximation using radial-basis-function networks. *Neural computation*, 3(2), 246–257.

Rauscher, G., Dube, D., and Zell, A. (2016). A comparison of 3d sensors for wheeled mobile robots. In *Intelligent Autonomous Systems 13*, 29–41. Springer.

Scheffe, H. (1999). *The analysis of variance*, volume 72. John Wiley & Sons.

Serfling, R. (2002). Quantile functions for multivariate analysis: approaches and applications. *Statistica Neerlandica*, 56(2), 214–232.

Song, E.Y. and Lee, K. (2008). Understanding ieee 1451-networked smart transducer interface standard-what is a smart transducer? *IEEE Instrumentation & Measurement Magazine*, 11(2), 11–17.

Taylor, B.N. and Kuyatt, C.E. (1994). Guidelines for evaluating and expressing the uncertainty of nist measurement results. Technical report, US Department of Commerce, Technology Administration, National Institute of Standards and Technology.