Identification of Risk Significant Automotive Scenarios Under Hardware Failures

Mohammad Hejase, Arda Kurt, Umit Ozguner
Department of Electrical and Computer Engineering
The Ohio State University
Columbus, Ohio, USA
hejase.1@osu.edu, ozguner.1@osu.edu

Tunc Aldemir
Department of Mechanical and Aerospace Engineering
The Ohio State University
Columbus, Ohio, USA
aldemir.1@osu.edu

The level of autonomous functions in vehicular control systems has been on a steady rise. This rise makes it more challenging for control system engineers to ensure a high level of safety, especially against unexpected failures such as stochastic hardware failures. A generic Backtracking Process Algorithm (BPA) based on a deductive implementation of the Markov/Cell-to-Cell Mapping technique is proposed for the identification of critical scenarios leading to the violation of safety goals. A discretized state-space representation of the system allows tracing of fault propagation throughout the system, and the quantification of probabilistic system evolution in time. A case study of a Hybrid State Control System for an autonomous vehicle prone to a brake-by-wire failure is constructed. The hazard of interest is collision with a stationary vehicle. The BPA is implemented to identify the risk significant scenarios leading to the hazard of interest.

1 Introduction

Emerging cars in todays markets have tens of interconnected Electronic Control Units (ECUs) that have to realize possibly thousands of features [10]. As the level of autonomous functions in cars keep increasing, the need for alternatives to physical testing for ensuring safe operation of these functions increases. Ensuring safe operation of an engineered system is accomplished by inferring conditions and causes that could lead to the violation of safety goals (safety analysis). Johansson [21] discusses a method that ensures completeness of safety goals definition through the definition of hazardous events. The method reduces the problem of providing an assurance case that supports controller compliance to safety goals to providing proof that contributions of modeled uncertainties and behaviors only lead to hazardous events within an acceptable risk. Quantitative analysis methods are typically used for estimating likelihoods of reaching hazardous events, or violating safety goals under certain system failures. Among the most common methods for quantitative analysis in the automotive industry are quantitative Failure Mode and Effect Analysis (FMEA), quantitative Fault Tree Analysis (FTA), Markov Models, and Reliability Block Diagrams [27, 36].

Over the past few years, research towards the development of tools and methods that provide compliant quantitative assurance cases for autonomous vehicle features have intensified. Takeichi et al. [35] describe a priority FTA calculation approach for latent faults. Das and Taylor [12] demonstrate a structured and systematic quantitative FTA which shows various techniques for the calculation of fault tree metrics.

M. Gleirscher, S. Kugele, S. Linker (Eds.): 2nd International
Workshop on Safe Control of Autonomous Vehicles (SCAV 2018)
EPTCS 269, 2018, pp. 59–73, doi:10.4204/EPTCS.269.6

Zhang et al. [40] present a case study for applying combinations of FTA and FMEA techniques for thorough model based hazard analysis of autonomous systems. Cherfi et al. [11] use Markov chains to model behaviors of a large class of automotive systems protected by safety mechanisms. Hoffman and Scharfenberg [19] show, via an example, compliance of a standard cell balancing circuit with requirements set by industry standards with respect to random failures.

Traditionally, reachability analysis has been a widely used analytical tool for verification of automated vehicle safety using simulation techniques [1, 6, 23, 33]. Reachability analysis works by computing the set of all reachable states when sensor measurements, disturbances, and initial vehicle states are uncertain. Safety is ensured by computationally confirming that none of the reachable states violate a safety goal. Authors of [1, 6, 23, 33], utilize reachability analysis as a proof to compliance with safety goals for various scenarios and case studies. Limiting features to this type of analysis, however, as noticed in the aforementioned work, are typically challenges associated with using high fidelity nonlinear models which lead to long computation times. It is also challenging to develop a generic approach based on reachability analysis that can be used on a wide spectrum of scenarios.

Hybrid system analysis techniques have been proposed for the verification of control functions in cyber-physical systems. Such methods are powerful tools as they can provide formal verification of large-scale systems. Loos et al. [24] developed a formal model of a distributed car control system in which an arbitrary number of vehicles sharing a highway use adaptive cruise control. The authors performed a full verification of the system by utilizing a modular proof structure. Mitsch et al. [26] also use hybrid state analysis to formally prove safety of robot vehicles under sensor uncertainty and actuator perturbation. Such techniques are ideal use in systems with known dynamics and behaviors. However, these techniques have challenges when incorporating random hardware failures or using high-fidelity simulators that have dynamics without explicit analytic forms, such as look-up tables.

Currently, software development in the automotive industry follows the V process [37]. In the V process, testing is left to the latter stages of development. New standards are being developed that emphasize on testing in the earlier stages of design.

One common way of testing in the early design stages is done by assigning specifications to Model Based Designs (MBDs) of autonomous features, and testing using simulation techniques, including fault simulation [7, 25, 30, 38]. The use of MBDs allows for system testing via accurate simulation. This approach eliminates the high costs of testing over extensive distances in various environments and locations.

In this paper, a generic Backtracking Process Algorithm (BPA) algorithm is proposed for the determination of quantitative metrics that probabilistically rank scenarios leading to user specified Top Events (e.g. hazardous events) by risk significance [39]. Within the context of this paper, a risk significant scenario is defined as a sequence of events that lead to an undesirable consequence with probability higher than a user-specified threshold. An event is defined as a change in system dynamic behavior and configuration that occurs over time. The BPA is a deductive and memory efficient implementation of a Markov Cell-to-Cell Mapping Technique (CCMT) [5] that is used for risk informed identification and quantification of critical scenarios leading to undesirable consequences (Top Events). Markov/CCMT allows for the global analysis of dynamic systems under both epistemic and aleatory uncertainties. Probabilistic system evolution is quantified in time, and fault propagation is traced throughout the system. Markov/CCMT has mostly been used in literature for the failure analysis and diagnostics of process control systems under uncertainties [2, 3, 4, 5, 9, 8, 13, 39]. More specifically, BPA is proposed to solve the problem of tracing fault propagation in systems with complex dynamics and varying configuration, such as random hardware failures of system components. Generally speaking, existing approaches either have challenges with accurately capturing high-fidelity system dynamics, or when incorporating possible random component failures and configuration changes. The algorithm has already been used in a validation and

verification framework development for Unmanned Aircraft Systems (UAS) as part of the System-Wide Safety Assurance Technologies (SSAT) initiative taken by the National Aeronautics and Space Administration (NASA) which the authors conducted jointly with ASCA, Inc. [14, 15, 16, 17].
Sections 2 and 3 of the paper provide, respectively, overviews of the autonomous ground vehicle controller design framework the analysis in this paper is based on. Section 3 describes Markov/CCMT, and the BPA. Section 4 presents the case study under consideration in this paper, the definition of the different possible types of brake failures, and the hazard of interest. Section 5 illustrates the use of the BPA in identifying the risk significant scenarios. Section 6 provides a discussion on the identified challenges and future directions. Section 7 gives the conclusions of the study.

2 Overview of Control System Design Framework

Based on the model based validation and verification framework described in [14, 15] , a similar framework was constructed for the validation and verification of an autonomous ground vehicle controller. The framework, depicted in Fig. 1, is made up of six main elements,

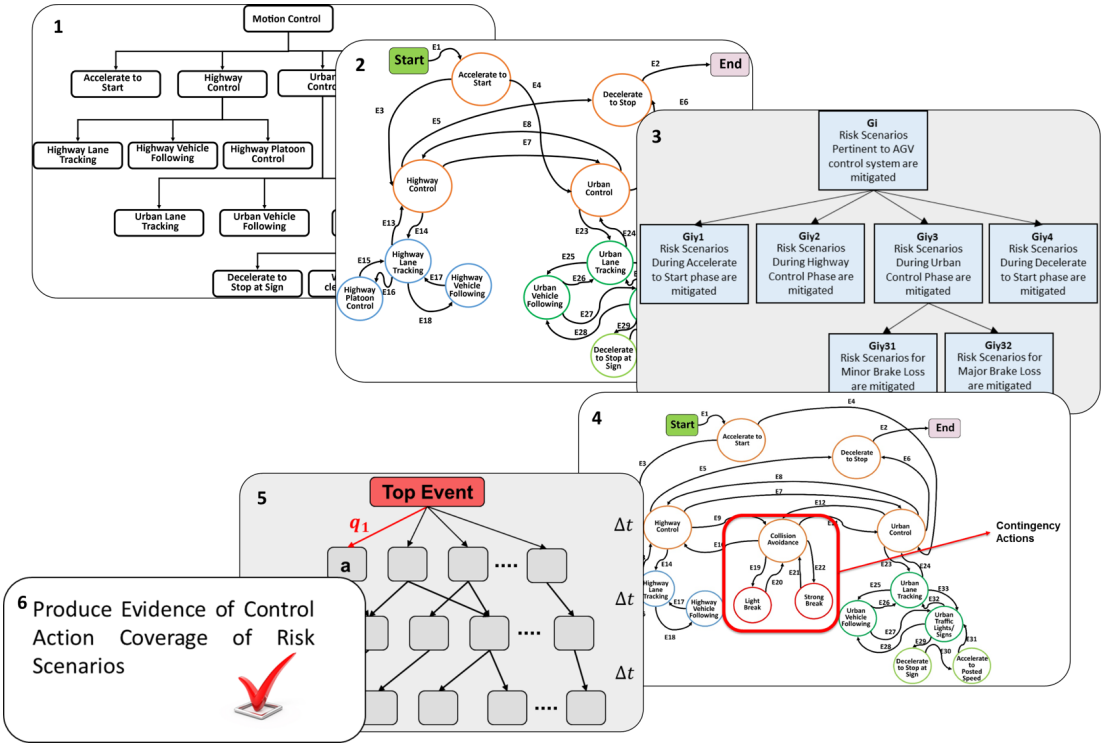

Figure 1: Model Based Validation and Verification Framework (adapted from [14, 15])

1) *Definition of Control System Functional Hierarchy*: The control system functional hierarchy is designed based on the work of Ozguner [31]. This allows for the initial definition of system and mission requirements and specifications, and the subsequent decomposition into mission phase specific functionality.

2) *Design of a Finite State Machine*: A Finite State Machine (FSM) representation of the high level

mission controller is designed based on the control system functional hierarchy. Each state of the FSM corresponds to a different phase of the mission, and transitions between those states are determined via the definition of event based rules.

3) *Development of Risk Prioritized Scenarios*: The system top level safety goals are first determined. A safety case goal and evidence tree model reflecting risk prioritized scenarios encompassing nominal, contingency, and emergency conditions and actions is developed using Goal Structure Notation (GSN) [22]. This model allows breaking up each of the top level safety goals into a set of hazards, where collectively avoiding these hazards represent the safety goals.

4) *FSM Augmentation with Emergency and Contingency Actions*: Based on the determined hazards, contingency and emergency actions are defined and incorporated into the FSM.

5) *Expansion of Risk Prioritized Scenarios for Details Analysis*: Each of the specified hazards, which can be thought of as the consequence of a risk significant scenario, is expanded upon with Markov/CCMT for detailed analysis under relevant hardware failures. The aim of this analysis is to provide an assurance case for system compliance with a target probability metric for hardware failures. Physical motion of the system is represented in this step via the use of a high-fidelity simulator.

6) *Produce Evidence of Control Action Coverage of Risk Scenarios*: The results produce evidence of control action coverage of prioritized operational and risk scenarios, supporting a controller assurance case.

3 Markov Cell to Cell Mapping Technique

Section 3.1 presents an overview of Markov/CCMT and the required assumptions, along with the required assumptions. In Section 3.2, BPA is illustrated and described in detail.

3.1 Overview and Assumptions

Markov/CCMT is a logic tool used to provide quantified metrics for system reliability and safety [39, 5, 3, 9, 8, 2, 4, 13]. Theoretical basis of BPA is presented in the work of Yang and Aldemir [39]. System evolution in time is represented through a series of discrete-time transitions among computational cells that partition the system state-space in a manner similar to finite element or finite difference methods. Each cell can be regarded as accounting for the uncertainty in the system location at a given point in time. A transition probability from one system cell to another is determined via system dynamics, controller behavior, or system constituent malfunction. Such transitions reflect a probabilistic mapping of the system state-space onto itself, including system hardware normal or faulted states, over a user defined time-step Δt.

Two assumptions are placed on the system of interest in order to employ Markov/CCMT:

1. The system components configurations are fixed over $[t,\ t+\Delta t)$, but can change at $t+\Delta t$.
2. Transitions among cells or hardware states do not depend on system history.

The first assumption means that the system components can only fail or change their mode of operation once during the interval Δt. Through proper selection of the time-step Δt, the system configuration changes and the probabilities of those changed can be realistically modeled and captured. The second assumption leads to the system having Markov property. However, the second assumption can be relaxed via the use of sufficient number of auxiliary state variables.

3.2 BPA

BPA is depicted in Fig. 2. The system continuous state-space is first discretized, and system components/configurations are defined. The combination of the discretized state-space, and the system configurations form the complete space of the system. Using a simulator, a cell-to-cell mapping of the complete space is constructed under a user-specified time-step. A Top Event of interest is specified, and sequential paths of risk significance leading to the Top Event for a user-specified search depth are identified. Probabilities are associated to each of the identified sequences.

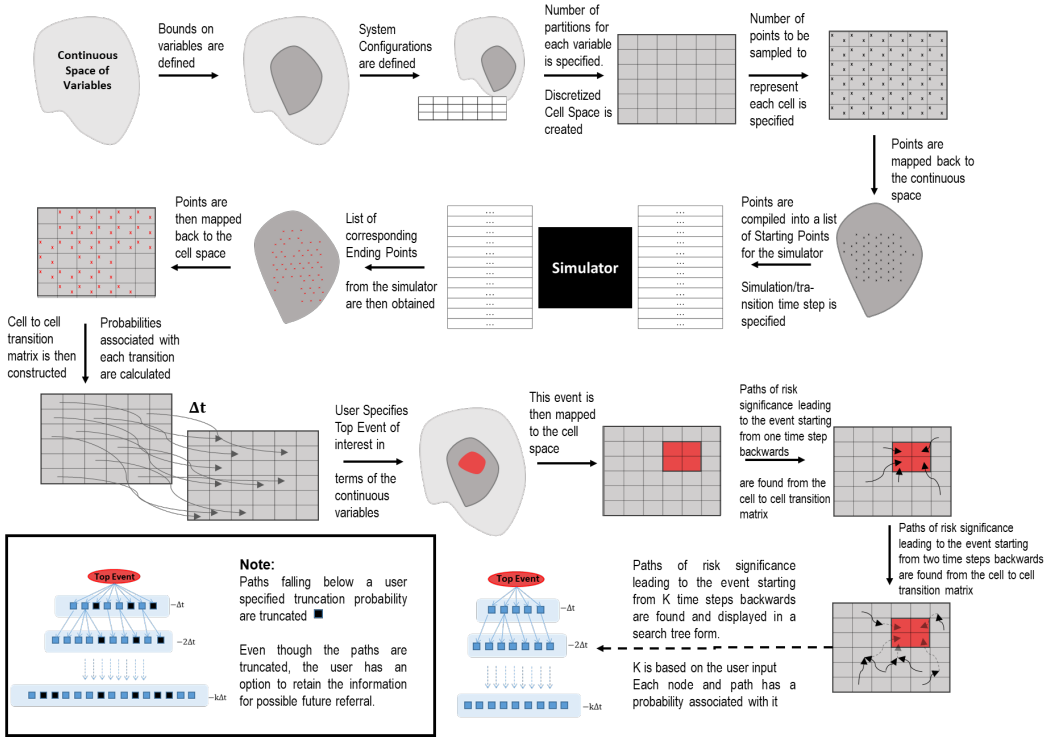

Figure 2: BPA flowchart (adapted from [17])

In Section 3.2.1, System discretization into a cell space is described. Section 3.2.2 contains the method of cell-to-cell transition probability calculation. An equal weight quadrature scheme is included in Section 3.2.3. Section 3.2.4 describes the process for the identification of risk significant event sequences.

3.2.1 System Discretization

The continuous L dimensional state space is represented by $\mathcal{X} \triangleq \mathbb{R}^L$. The M dimensional discrete state space of the system components is represented by $\mathcal{N} \triangleq \mathbb{Z}^M$. The space $\mathcal{X} \triangleq \mathbb{R}^L$is discretized by partitioning each continuous variable x_l (l= 1,…, L) into intervals of J_l partitions and considering combinations of those partitions to form the cells. Knowledge of the state-space upper bounds $\overline{x}$, and lower bounds $\underline{x}$ is required for the partitioning. The cells can be regarded as means to accommodate epistemic uncertainties (such as model uncertainties) or aleatory uncertainties (such as process noise and minor environmental disturbances).

The possible states of each hardware component M of interest are then defined (e.g. operational, degraded, failed), with each component m, having N_m possible states, each denoted by n_m ($m = 1,\ldots, M$).

The unique combinations of the partitioned $\mathcal{X} \triangleq \mathbb{R}^L$ along with the discrete system component configurations forms the complete state-space of the system, denoted by $\mathcal{V}$. Each cell in the cell space is represented by an (L+M) dimensional vector $[\mathbf{j}\ \mathbf{n}] \equiv [j_1, \ldots, j_l, \cdots, j_L, n_1, \ldots, n_m, \cdots, n_M]$, where ($j_l = 1,2,\ldots, J_l; l = 1,\ldots,L$) enumerate the partitioning of the interval $\underline{x_l} \leq x_l < \overline{x_l}$, and n_m represents the state of component m ($n_m = 1,\ldots,N_m$; $m = 1, \ldots, M$). The cell space $\mathcal{V}$ is composed of $J \times N$ unique cells with $J = J_1 \times \cdots \times J_L$ and $N = N_1 \times \cdots \times N_M$.

Let $\mathcal{V}_\mathcal{X} \triangleq \mathbb{Z}^L$ be a subspace of $\mathcal{V}$ containing the vectors $\mathbf{j}$. Let $\mathcal{V}_\mathcal{N} \triangleq \mathbb{Z}^M$ be a subspace of $\mathcal{V}$ containing the vectors $\mathbf{n}$. Note that $\mathcal{V}_\mathcal{X} \cup \mathcal{V}_\mathcal{N} = \mathcal{V}$.

The discretized system, along with the relevant notations is illustrated in Fig. 3.

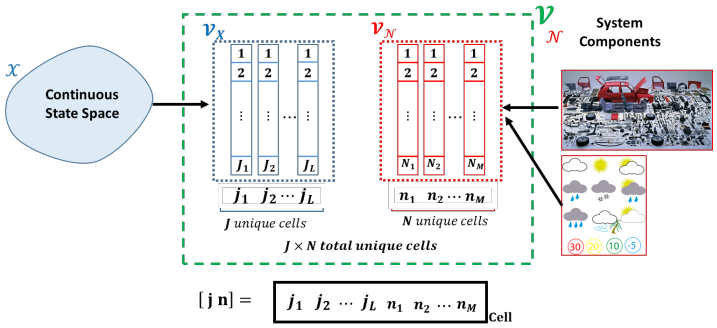

Figure 3: Illustration of Discretized System

3.2.2 Cell to Cell Transition Probability Calculation

Using the Markov property, and as derived in [20], the cell-to-cell probabilities over a single time-step transition Δt can be calculated from

$$q(j, \mathbf{n}|\ \mathbf{j'}, \mathbf{n'}, \Delta t) = h(\mathbf{n}|\mathbf{n'}, \mathbf{j'} \rightarrow \mathbf{j}, \Delta t) \times g(\mathbf{j}|\mathbf{j'}, \mathbf{n'}, \Delta t) \tag{1}$$

where $g(\mathbf{j}|\mathbf{j'}, \mathbf{n'}, \Delta t)$represents the transition probability from cell $\mathbf{j'}$ to $\mathbf{j}$ over Δt under configuration $\mathbf{n'}$, and $h(\mathbf{n}|\mathbf{n'}, \mathbf{j'} \rightarrow \mathbf{j}, \Delta t)$ quantifies the system configuration transition probabilities over Δt.

For each component of interest m, a component state transition probability matrix H_{n_m} is constructed. Contents of this matrix represent the probability of component state transitions over Δt. These probabilities can be based on hardware component data, such as failure rates, or expert opinion in the absence of reliable data. An example of such a matrix can be seen in Table 1 where $\lambda_{n'_m, n_m}$ denotes the transition rate from n'_m to n_m.

Using the Chapman-Kolmogorov equation under the assumptions stated earlier, the system cell-to-cell state transition probabilities $g(\mathbf{j}|\mathbf{j'}, \mathbf{n'}, \Delta t)$ over a single time-step can be found from [3]

$$g(\mathbf{j}|\mathbf{j'}, \mathbf{n'}, \Delta t) = \frac{1}{v_{j'}} \int_{v_{j'}} u_\mathbf{j}[\mathbf{x}(\mathbf{x'}, \mathbf{n'}, \Delta t)]\, dx' \tag{2}$$

$$u_\mathbf{j}[\mathbf{x}(\mathbf{x'}, \mathbf{n'}, \Delta t)] = \begin{cases} 1 & if\ \mathrm{x} \in v_\mathbf{j} \\ 0 & otherwise \end{cases} \tag{3}$$

Table 1: Sample System Configuration Transition Matrix H_{n_m}

		Final System Configuration State			
		Normal State (N)	Failure State 1 (F_1)	...	FailureState N (F_N)
Initial System Configuration State	Normal State (N)	$\lambda_{N,N}\Delta t$	$\lambda_{N,F_1}\Delta t$	...	$\lambda_{N,F_N}\Delta t$
	Fail 1 State (F_1)	$\lambda_{F_1,N}\Delta t$	$\lambda_{F_1,F_1}\Delta t$	...	$\lambda_{F_1,F_N}\Delta t$
	⋮	⋮	⋮	-	⋮
	Fail N State (F_N)	$\lambda_{F_N,N}\Delta t$	$\lambda_{F_N,F_1}\Delta t$	...	$\lambda_{F_N,F_N}\Delta t$

where $v_{\mathbf{j}}$ is the volume of the cell $\mathbf{j}$,

$$\mathbf{x}(\mathbf{x'},\mathbf{n'},\Delta t)= \int_t^{t+\Delta t} f(\mathbf{x}(t'),\ \mathbf{n'})\boldsymbol{dt'}+\mathbf{x'} \tag{4}$$

and $f(\mathbf{x}(t'),\ \mathbf{n'})$ represents the equations describing system dynamics.

3.2.3 Equal Weight Quadrature Approximation Scheme

When it is not practical or possible to evaluate (4), an equal weight quadrature approximation scheme can employed via the use of a high fidelity simulator. System location in the state space is assumed to be uniformly distributed within each cell. Multiple points are sampled to represent each cell, and are passed to the simulator to compute transitions over Δt. Then (2) can be approximated as

$$g(\mathbf{j}|\mathbf{j'},\mathbf{n'},\ \Delta t)=\frac{\#\ of\ sampled\ points\ in\ \text{cell}\ \mathbf{j'}\ arriving\ in\ cell\ \mathbf{J}\ over\ \Delta t}{\#\ of\ points\ sampled\ from\ \ \text{cell}\ \mathbf{j'}} \tag{5}$$

3.2.4 Identification of Risk Significant Event Sequences

While, in principle, backtracking can be accomplished through

$$\mathbf{P^k}=\left[\mathbf{Q^T Q}\right]^{-1}\mathbf{Q^T P^{k+1}} \tag{6}$$

The BPA avoids the challenges associated with (6) by using the search tree that is obtained from a probabilistic map of the system state-space onto itself. This search tree structure is achieved by recursive enumeration of sub-trees emanating from Top Event in decreasing time and the traversal of possible paths through a branching process. In order to avoid a numerical catastrophe, only risk significant scenarios with probabilities above a user-specified cut-off value are identified.

In this study, an undesirable event $\mathcal{E}$ is assumed to be defined through the specification of event upper bounds $\overline{e}$, event lower bounds $\underline{e}$ in terms of the continuous variables in the state-space, and the set of event system configurations e_s as stated in (7)-(9):

$$\overline{e}=\{[\overline{e_1},\cdots,\overline{e}_l,\ \cdots\overline{e_L}]\ \big|\ \overline{e_l}\leq\overline{x_l},\ \overline{e_l}>\underline{x_l}\} \tag{7}$$

$$\underline{e}=\left\{\left[\underline{e_1},\cdots,\underline{e_l},\ \cdots,\ \underline{e_L}\right]\ |\ \underline{e_l}\geq\underline{x_l},\ \underline{e_l}<\overline{x_l}\right\} \tag{8}$$

$$e_n=\{[e_{n_1},\ldots,e_{n_M}]\ |\ [e_{n_1},\ldots,e_{n_M}]\subset\mathcal{V}_N\} \tag{9}$$

However, event definition can include a specific system configuration as well, in general. Sequential paths with non-zero transition probabilities that span backwards by a search depth of k time-steps from the event of interest to cells contained in the cell space are then identified. Fig. 4 graphically illustrates an algorithm for the BPA. Only the paths with probabilities greater than a user-specified probability truncation value ϵ are kept in the *Prune Out* process.

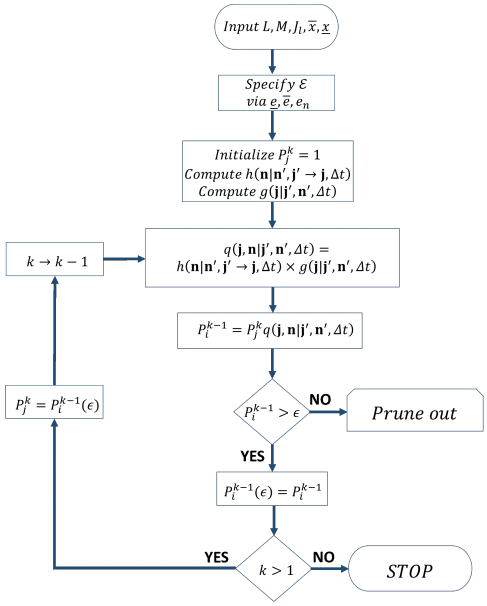

Figure 4: Algorithm for Path Probability Calculation (adapted from [39])

4 Case Study

A high-fidelity simulator for an Autonomous Ground Vehicle (AGV) based on the full 4-wheel model from the work of Ozguner et al. [32] was constructed in Matlab/Simulink environment. The states of the vehicle used in the analysis are the forward velocity, sideward velocity, yaw rate, yaw, x-position, and the y-position. The control surfaces that affect the vehicle are the engine traction force, the braking force, and the steering angle. Vehicle parameters were taken to be those of the 2009 Lincoln MKS.

A Hybrid State Control System was used for the decision making and control of the AGV. Two environments are taken into consideration, an urban environment, and a highway environment, with various phases modeled for each of the two environments. The design procedure found in [18] was used for the design and construction of the Hybrid State Control System. Low-level controllers were designed using LQR controllers to control the body rates, and a PI controller to control Euler positions and angles. A FSM that serves as a high-level controller which guides the AGV through the different phases of possible scenarios was constructed.

The hazard of interest (i.e. Top Event) is taken to be a collision with a stationary vehicle in an urban environment. This hazard was selected and modeled based on a list of selected pre-crash scenarios published by the U.S. Department of Transportation [28]. For the sake of simplicity in illustrating BPA, it is assumed that in such a scenario the brake-by-wire is the only system component that is prone to random hardware failures. Based on the hazard of interest, the FSM was augmented with emergency and

contingency actions for collision avoidance. The resulting FSM can be seen in Fig. 5. The contingency actions that were augmented to the FSM are based on the work of [33].

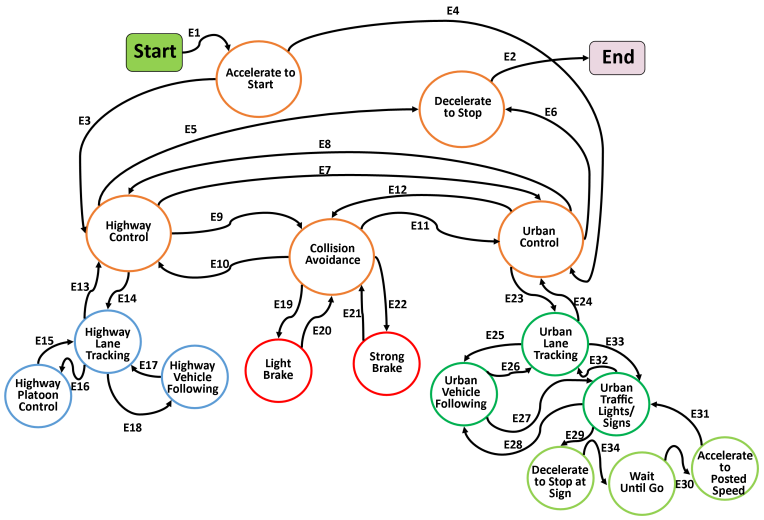

Figure 5: Autonomous Ground Vehicle Finite State Machine for High-Level Decision Making

Three 'brake condition states' were defined in this case study as seen in Table 2. The first is Brake Normal, in this state the brake operates normally. The second state is 'Minor Brake Fault', in this state the braking system delivers 50% of what the controller asks of it. The third state is a 'Major Brake Fault', in this state the braking system only delivers 25% of what the controller asks of it. Each of the failed brake states is assigned a probability of $\lambda = (2\times10^{-7})/h$ which is fairly consistent with some of component failure probabilities in literature [34]. The failures are assumed to be permanent ones.

Table 2: Brake States Transition Probabilities

		Final Brake State		
		Normal State	Minor Fault	Major Fault
Initial Brake State	Normal State (N)	≈ 1	2 x 10^{-7} /h	2 x 10^{-7} /h
	Minor Fault	0	1	0
	Major Fault	0	0	1

The host AGV is initially assumed to be on the road at a positon of (0, 0) in a single lane urban environment with a posted speed limit of 15m/s (≈35mph). It is also assumed that the stationary target vehicle, which the AGV has to avoid a collision with, is at a position of x=500m at all times. It is assumed that the vehicle is equipped with a sensor that allows it to sense and detect other vehicles that are within a range of 100m ahead.

The AGV is initially in the urban lane tracking state, and then encounters a target vehicle within a range of 100m. The host vehicle then switches to the urban vehicle following state and aims to follow the target vehicle at a desired time-gap of 1.3s, such that a desired clearance is obtained from $c_{des} = t_{gapdes} \times v_{host}$, and is 20m. Once the time-gap from the target vehicle is less than the desired time-gap (i.e. $c_{h-t} < c_{des}$), the vehicle switches to collision avoidance and comfortably brakes at -0.3g. If the time-gap from the target is within less than half the desired time-gap (i.e. $c_{h-t} < \frac{1}{2}c_{des}$), the vehicle applies a strong brake at -0.8g.

The event, or hazard of interest, is when the AGV reaches an x-position of 500m or greater. This would indicate that a collision took place with the target vehicle. An illustration of the constructed scenario with the hazard of interest can be seen in Fig. 6.

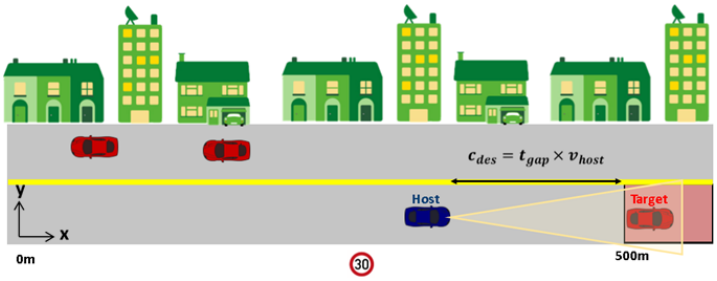

Figure 6: Illustration of Autonomous Vehicle Scenario in the Proposed Case Study

The aim of this analysis is to identify the risk significant scenarios that lead to a hazard, or a violation of the safety goal. Emergency and contingency actions can then be modified based on the identified scenarios. This process can then be iteratively used to modify contingency actions, until results ensure that scenarios only lead to the violation of a safety goal within acceptably low probabilities.

5 Simulation

Results based on user inputs from Table III to BPA over a search depth of $2\Delta t$ with a truncation of scenarios occurring with probabilities $< 10^{-8}$ can be seen in Fig. 7. Each time step was taken to be 2/3 seconds in length. Note that a search step of two time steps was selected since they amount to 1.3 seconds, this is the time-gap at which the contingency actions begin. Another search tree was also constructed in Fig. 8 to illustrate truncation of scenarios occurring with probabilities $< 3\times10^{-7}$. The truncation probability is used to remove risk insignificant values from the search tree. The probabilities displayed in the search tree are used as probabilistic metrics that rank risk significance of scenarios in comparison to one another. Recall from Section 4 that system at hand has six continuous states (i.e., forward velocity, sideward velocity, yaw rate, yaw, x-position, and the y-position, and one system hardware configuration (i.e. brake-by-wire). This means that each cell in the discretized space is represented by 7 integers. The first 6 integers represent the segment number of the partitioned continuous variables based on the system discretization, and the 7^{th} integer represents the brake condition. Each node in the tree of Figs. 7-8 contains 7 integers and an associated probability.

Taking the first sequence from the left in Fig. 7 as an example sequence to interpret results from branch the search tree,

$$[4\ 1\ 1\ 122\ 1\ 1\ \mathbf{1}]_{P=2\times10^{-7}} \rightarrow [3\ 1\ 1\ 124\ 1\ 1\ \mathbf{2}]_{P=0.5} \rightarrow Collision \tag{10}$$

we can make the following observations:

1. [4 1 1 122 1 1 **1**]– The AGV initially has a forward velocity of 12 to 16 m/s, a sideward velocity of -0.5 to 0.5m/s, a yaw rate of -0.5 to 0.5 rad/s, an x-position of 488-492, an y-Position of -6 to 6m, and a Yaw angle of $-\pi/3$ to $\pi/3$. The brake state was Normal.
2. [3 1 1 124 1 1 **2**]– One time step later, the AGV had a forward velocity of 20 to 25 m/s, a sideward velocity of -0.5 to 0.5m/s, a yaw rate of -0.5 to 0.5 rad/s, an x-position of 496 – 500m, an y-Position of -6 to 6m, and a Yaw angle of $-\pi/3$ to $\pi/3$. The brake experienced a Minor Brake Fault.

Table 3: User Input to BPA

Variable Name	Notation	Value
numProcessVariables	L	6
processVariablesNames		[”Fwd Vel.”, ”Side Vel.”, Yaw Rate, x-Pos, y-Pos, Yaw]
numSystemComponents	M	1
systemComponentNames		[”Brake State”]
systemComponentStates	N_i	[3]
systemComponentStateNames		[Normal, Minor Brake Fault, Major Brake Fault]
variableUpperBounds	$\overline{x}$	[20,5,0.5,600,6,pi/3]
variableLowerBounds	$\underline{x}$	[0,-5,-0.5,0,-6, 0-pi/3]
numberOfCells	J_i	[5,1,1,150,1,1,3]
sysConfTransProb	H_{n_m}	$\begin{bmatrix} \approx 1 & 2e-7 & 2e-7 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
eventUpperBounds	$\overline{e}$	[20,0.5,0.5,600,6,pi/3,3]
eventLowerBounds	$\underline{e}$	[0,-0.5,-0.5,500,-6, 0-pi/3,1]

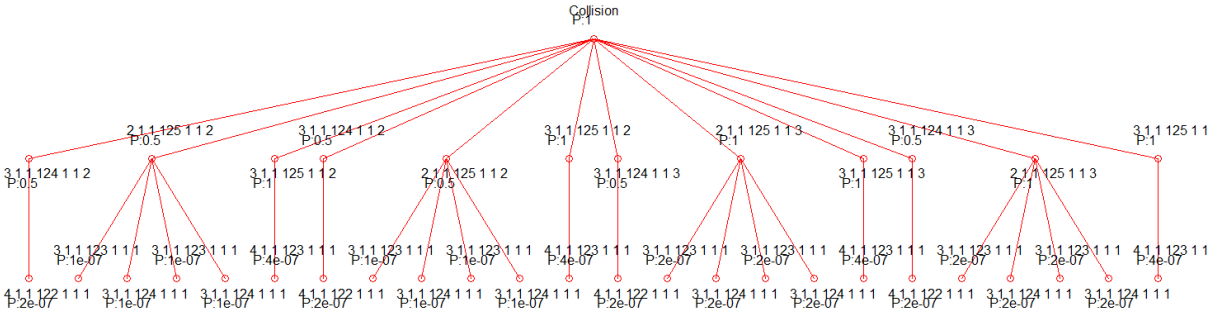

Figure 7: BPA results with truncation at 1e-8

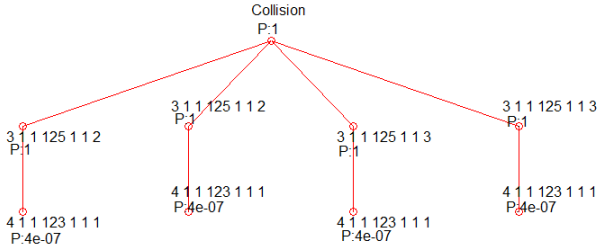

Figure 8: BPA results with truncation at 3e-7

3. **Collision** – One time step later the AGV collides with the stationary vehicle located at an x-position that is greater than 500m, leading to a violation of the safety goal.

Upon investigation of the sequences in the search tree of Fig. 7, it is also observed that the safety goal is violated once a brake failure occurs. In an effort to address this observation, the contingency actions were modified. The time-gap was changed to 2s, such that a Light Brake contingency action is employed once the host vehicle is within 30m clearance from the target vehicle, rather than 20m. The Strong Brake contingency action is employed once the host vehicle is within 15m of the target vehicle, rather than 10m. Based on these modifications in the contingency actions, with all other parameters from Table 3 kept the same. The BPA was run again over a search depth of $3\Delta t$ and truncation of branches with probabilities $< 10^{-8}$. A search depth of $3\Delta t$ was selected since this amounts to 2 seconds, the time-gap at which contingency actions begin. No paths of risk significance leading to the Top Event were identified, which means that the proposed contingency actions managed to bring the system to a safe state within an acceptable risk level, even under the occurrence of the assumed hardware failures. This example also illustrates how BPA can be used towards a safer design.

6 Challenges Faced in Automotive Scenarios, and Potential Solutions

The ultimate goal of the described methodology is the design of a generic quantitative risk assessment scheme that is capable of providing information on risk-significant sequences of events that violate safety goals. Safety assurance of control systems being developed for automotive scenarios has multiple challenges. Two main challenges are identified by the authors: 1) Large-scale scenarios that involve high levels of autonomy and many hardware components do not typically have a single domain expert that is able to accurately set-up BPA parameters for the overall scenario. 2) For autonomous systems with a large state-space, such as platoons of vehicles, combinatorial and computational issues are prone to appear.

Future work of BPA is directed towards solving the identified challenges. A possible solution to Challenge 1 is running phase-specific implementations of BPA, and integrating results of analysis obtained from the multiple phases. The authors are already in the process of developing a generalized scheme for such a solution, with a preliminary approach described in [16]. The nature of BPA equips it with tools that can naturally help alleviate problems faced due to Challenge 2. Through selection of larger cell sizes (a more coarse partitioning scheme), and sampling more cells from each cell, the size of the system cell-to-cell map can be reduced. Noting that the reduction in size is compensated by sampling more points from each cell. This in turn reduces the computations needed to identify risk significant path sequences. Additionally, careful and intelligent selection of the truncation value parameter can lead to reduced wastage of computational resources on risk insignificant event sequences.

7 Conclusion

The need for generic and well defined procedures and methods for the assurance of autonomous ground vehicle functions with respect to safety goals in the early design stage is of vital importance. In this paper, the BPA approach based on a deductive implementation of Markov/CCMT has been proposed for the identification of scenarios that lead to safety goal violations. The scenarios were ranked by risk significance via probabilistic quantification of the scenarios that violate the safety goals. A case study of a hybrid state autonomous vehicle prone to random hardware failures in the braking system was taken into consideration. Simulation results displayed the risk significant scenarios leading to collisions with

a static vehicle under possible brake failure in a search tree format. The simulated scenarios indicated that even though the contingency actions work as required under nominal brake conditions, they did not adequately avoid the risk of collision under sub-nominal brake conditions. It was also shown that, based on the modification of the contingency actions, no paths of risk significance leading to the Top Event were identified. Future work will involve the definition of a broader and more realistic set of contingency actions and hardware failures for various phases of the scenario. Future work will also investigate the applicability of the BPA to current standards such as ISO 26262 [20]. As a final note, it should be indicated that while BPA, in principle, may lead to combinatorial increase in the number of scenarios to be investigated, an upper limit can be imposed on this number through the specification of probability bounds in defining what is regarded as risk significant. This probability bound can be relaxed in new BPA runs once the initially identified scenarios of high risk significance are mitigated.

Acknowledgement

The work is partially funded by the National Science Foundation (NSF) Cyber-Physical Systems (CPS) project under contract 60046665. An application BPA to Unmanned Aircraft Systems (UAS) was developed with ASCA Inc. as part of a project funded by the NASA Ames Research Center (ARC). Discussions in that project with Drs. Sergio Guarro, and Michael Yau from ASCA Inc., and Dr. Matt Knudson from NASA ARC are gratefully acknowledged.

References

[1] Assad Alam, Ather Gattami, Karl H Johansson & Claire J Tomlin (2014): *Guaranteeing safety for heavy duty vehicle platooning: Safe set computations and experimental evaluations*. Control Engineering Practice 24, pp. 33–41, doi:10.1016/j.conengprac.2013.11.003.

[2] T Aldemir & P Wang (1999): *The Use of the Cell-to-Cell Mapping Technique as a Model-Based Diagnostic Tool*.

[3] Tunc Aldemir (1987): *Computer-assisted Markov failure modeling of process control systems*. IEEE Transactions on reliability 36(1), pp. 133–144, doi:10.1109/TR.1987.5222318.

[4] Tunc Aldemir, Mohamed Belhadj & Laurian Dinca (1996): *Process reliability and safety under uncertainties*. Reliability Engineering & System Safety 52(3), pp. 211–225, doi:10.1016/0951-8320(95)00133-6.

[5] Tunc Aldemir, Sergio Guarro, Diego Mandelli, Jason Kirschenbaum, L Anthony Mangan, Paolo Bucci, Michael Yau, E Ekici, DW Miller, X Sun et al. (2010): *Probabilistic risk assessment modeling of digital instrumentation and control systems using two dynamic methodologies*. Reliability Engineering & System Safety 95(10), pp. 1011–1039, doi:10.1016/j.ress.2010.04.011.

[6] Matthias Althoff & John M Dolan (2014): *Online verification of automated road vehicles using reachability analysis*. IEEE Transactions on Robotics 30(4), pp. 903–918, doi:10.1109/TRO.2014.2312453.

[7] Sanket Amberkar, Joseph G D'Ambrosio, Brian T Murray, Joseph Wysocki & Barbara J Czerny (2000): *A system-safety process for by-wire automotive systems*. Technical Report, SAE Technical Paper, doi:10.4271/2000-01-1056.

[8] M Belhadj & T Aldemir (1991): *Probabilistic analysis of asymptotic reactor dynamics and the cell-to-cell mapping technique*. Transactions of the American Nuclear Society;(United States) 63(CONF-911107–).

[9] M Belhadj & T Aldemir (1995): *The Cell to Cell Mapping technique and Chapman-Kolmogorov representation of system dynamics*. Journal of sound and vibration 181(4), pp. 687–707, doi:10.1006/jsvi.1995.0166.

[10] Manfred Broy (2006): *Challenges in automotive software engineering*. In: Proceedings of the 28th international conference on Software engineering, ACM, pp. 33–42, doi:10.1145/1134285.1134292.

[11] Abraham Cherfi, Michel Leeman, Florent Meurville & Antoine Rauzy (2014): *Modeling automotive safety mechanisms: A Markovian approach*. Reliability Engineering & System Safety 130, pp. 42–49, doi:10.1016/j.ress.2014.04.013.

[12] Nabarun Das & William Taylor (2016): *Quantified fault tree techniques for calculating hardware fault metrics according to ISO 26262*. In: Product Compliance Engineering Proceedings (ISPCE), 2016 IEEE Symposium on, IEEE, pp. 1–8, doi:10.1109/ISPCE.2016.7492848.

[13] Laurian Dinca, Tunc Aldemir & Giorgio Rizzoni (1999): *Fault detection and identification in dynamic systems with noisy data and parameter/modeling uncertainties*. Reliability Engineering & System Safety 65(1), pp. 17–28, doi:10.1016/S0951-8320(98)00077-5.

[14] Sergio B Guarro, Michael K Yau, Umit Ozguner, Tunc Aldemir, Arda Kurt, Mohammad Hejase & Matt D Knudson (2017): *Formal Framework and Models for Validation and Verification of Software-Intensive Aerospace Systems*. In: AIAA Information Systems-AIAA Infotech@ Aerospace, p. 0418, doi:10.2514/6.2017-0418.

[15] Sergio B Guarro, Michael K Yau, Umit Ozguner, Tunc Aldemir, Arda Kurt, Mohammad Hejase & Matt D Knudson (2017): *Risk Informed Safety Case Framework for Unmanned Aircraft System Flight Software Certification*. SYSTEM 10(11), p. 12, doi:10.2514/6.2017-0910.

[16] Mohammad Hejase, Arda Kurt, Tunc Aldemir, Umit Ozguner, Sergio Guarro, Michael K Yau & Matt Knudson (2018): *Dynamic Probabilistic Risk Assessment of Unmanned Aircraft Adaptive Flight Control Systems*. In: 2018 AIAA Information Systems-AIAA Infotech@ Aerospace, p. 1982, doi:10.2514/6.2018-1982.

[17] Mohammad Hejase, Arda Kurt, Tunc Aldemir, Umit Ozguner, Sergio B Guarro, Michael K Yau & Matt D Knudson (2017): *Quantitative and Risk-Based Framework for Unmanned Aircraft Control System Assurance*. Journal of Aerospace Information Systems, pp. 1–15, doi:10.2514/1.I010583.

[18] Mohammad Hejase, Abdullah Ersan Oguz, Arda Kurt, Umit Ozguner & Keith Redmill (2016): *A Hierarchical Hybrid State System Based Controller Design Approach for an Autonomous UAS Mission*. In: 16th AIAA Aviation Technology, Integration, and Operations Conference, p. 3294, doi:10.2514/6.2016-3294.

[19] Gerhard Hofmann & Georg Scharfenberg (2015): *Random Hardware failure compliance of a cell balancing circuit with the requirements of automotive functional safety*. In: Applied Electronics (AE), 2015 International Conference on, IEEE, pp. 61–66.

[20] ISO26262 ISO (2011): *26262: Road vehicles-Functional safety*. International Standard ISO/FDIS 26262.

[21] Rolf Johansson (2015): *The Importance of Active Choices in Hazard Analysis and Risk Assessment*. In: CARS 2015-Critical Automotive applications: Robustness & Safety.

[22] Tim Kelly & Rob Weaver (2004): *The goal structuring notation–a safety argument notation*. In: Proceedings of the dependable systems and networks 2004 workshop on assurance cases, Citeseer.

[23] Andreas Lawitzky, Anselm Nicklas, Dirk Wollherr & Martin Buss (2014): *Determining states of inevitable collision using reachability analysis*. In: Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on, IEEE, pp. 4142–4147, doi:10.1109/IROS.2014.6943146.

[24] Sarah M Loos, André Platzer & Ligia Nistor (2011): *Adaptive cruise control: Hybrid, distributed, and now formally verified*. In: International Symposium on Formal Methods, Springer, pp. 42–56, doi:10.1007/978-3-642-21437-0_6.

[25] Caroline Lu, Jean-Charles Fabre & Marc-Olivier Killijian (2009): *An approach for improving fault-tolerance in automotive modular embedded software*. In: 17th International Conference on Real-Time and Network Systems, pp. 132–147.

[26] Stefan Mitsch, Khalil Ghorbal, David Vogelbacher & André Platzer (2017): *Formal verification of obstacle avoidance and navigation of ground robots*. The International Journal of Robotics Research 36(12), pp. 1312–1340, doi:10.1177/0278364917733549.

[27] Mohammad Modarres, Mark P Kaminskiy & Vasiliy Krivtsov (2016): *System Reliability Analysis*. In: Reliability Engineering and Risk Analysis: A Practical Guide, CRC Press, pp. 173–242.

[28] Wassim G Najm, John D Smith & Mikio Yanagisawa (2007): *Pre-crash scenario typology for crash avoidance research*. In: *DOT HS*, Citeseer.

[29] Jonas Nilsson, Jonas Fredriksson & Anders CE Ödblom (2014): *Verification of collision avoidance systems using reachability analysis*. *IFAC Proceedings Volumes* 47(3), pp. 10676–10681, doi:10.3182/20140824-6-ZA-1003.01567.

[30] J-H Oetjens, Nico Bannow, Markus Becker, Oliver Bringmann, Andreas Burger, Moomen Chaari, Samarjit Chakraborty, Rolf Drechsler, Wolfgang Ecker, Kim Grüttner et al. (2014): *Safety evaluation of automotive electronics using virtual prototypes: State of the art and research challenges*. In: *Design Automation Conference (DAC), 2014 51st ACM/EDAC/IEEE*, IEEE, pp. 1–6, doi:10.1145/2593069.2602976.

[31] U Ozguner (1990): *Coordination of hierarchical systems*. In: *Intelligent Control, 1990. Proceedings., 5th IEEE International Symposium on*, IEEE, pp. 2–7, doi:10.1109/ISIC.1990.128431.

[32] Ümit Özgüner, Tankut Acarman & Keith Alan Redmill (2011): *Autonomous ground vehicles*. Artech House.

[33] Jaeyong Park, Arda Kurt & Ümit Özgüner (2014): *Hybrid Systems Modeling and Reachability-Based Controller Design Methods for Vehicular Automation*. *Unmanned Systems* 2(02), pp. 101–119, doi:10.1142/S2301385014500071.

[34] Purnendu Sinha (2011): *Architectural design and reliability analysis of a fail-operational brake-by-wire system from ISO 26262 perspectives*. *Reliability Engineering & System Safety* 96(10), pp. 1349–1359, doi:10.1016/j.ress.2011.03.013.

[35] Masahiko Takeichi, Yoshinobu Sato, Koichi Suyama & Takuya Kawahara (2011): *Failure rate calculation with priority FTA method for functional safety of complex automotive subsystems*. In: *Quality, Reliability, Risk, Maintenance, and Safety Engineering (ICQR2MSE), 2011 International Conference on*, IEEE, pp. 55–58, doi:10.1109/ICQR2MSE.2011.5976568.

[36] Ajit Kumar Verma, Srividya Ajit & Durga Rao Karanki (2016): *Probabilistic Safety Assessment*. In: *Reliability and Safety Engineering*, Springer, pp. 333–372, doi:10.1007/978-1-4471-6269-8_10.

[37] Julian Weber (2009): *Automotive development processes*. 303, Springer, doi:10.1007/978-3-642-01253-2.

[38] Cédric Wilwert, Françoise Simonot-Lion, Yeqiong Song & Françoise Simonot (2005): *Quantitative Evaluation of the Safety of X-by-Wire Architecture subject to EMI Perturbations*. In: *Emerging Technologies and Factory Automation, 2005. ETFA 2005. 10th IEEE Conference on*, 1, IEEE, pp. 8–pp, doi:10.1109/ETFA.2005.1612601.

[39] Jun Yang & Tunc Aldemir (2016): *An algorithm for the computationally efficient deductive implementation of the Markov/Cell-to-Cell-Mapping Technique for risk significant scenario identification*. *Reliability Engineering & System Safety* 145, pp. 1–8, doi:10.1016/j.ress.2015.08.013.

[40] Hongkun Zhang, Wenjun Li & Wei Chen (2010): *Model-based hazard analysis method on automotive programmable electronic system*. In: *Biomedical Engineering and Informatics (BMEI), 2010 3rd International Conference on*, 7, IEEE, pp. 2658–2661, doi:10.1109/BMEI.2010.5639860.