Runtime Safety Assurance of Autonomous Vehicles

A. Forrai
Research and Technology Development
Siemens Industry Software Netherlands B.V.
Helmond, The Netherlands
alexandru.forrai@siemens.com

V. Neelgundmath, K.K. Unni
Automotive Technology
Eindhoven University of Technology
Eindhoven, The Netherlands
{v.neelgundmath, k.k.unni}@student.tue.nl

I. Barosan
Computer Science Department
Eindhoven University of Technology
Eindhoven, The Netherlands
i.barosan@tue.nl

***Abstract*—The paper deals with runtime safety assurance of autonomous vehicles (AVs). First, we closely examine the current state of AV technology, highlighting the incredible progress made by industry leaders in ensuring minimal risks in their vehicles. Our analysis of reported disengagements in AVs reveals interesting insights, showing that most disengagements are initiated by human test drivers rather than by the AV systems themselves and that they often occur on streets and freeways. Next, to ensure AVs' safety, we explored various safety architectures, including *1oo2*, *1oo2D*, and *2oo3*, and thoroughly evaluated their reliability. We also introduce the concept of runtime safety assurance (RTA), a system that closely monitors the AV's state and employs emergency measures through a backup controller if the safety conditions are breached. Such a safety architecture - having separate safety and performance loops - has several benefits, which are highlighted in this paper. Furthermore, we emphasize the crucial role of robust sensing and perception systems in AVs, showing that multi-modal sensing and sensor fusion can effectively enhance the system reliability. We validated the reliability of these systems through extensive software testing and comparisons with ground-truth data. Finally, we explore fault-detection techniques for AVs by employing simulations to detect and handle faults in the control system.**

***Index Terms*—Autonomous vehicles, runtime safety assurance, safety architectures, sensing and perception, fault detection.**

I. Introduction

Autonomous driving technology has enormous potential for providing safer, more convenient, and more enjoyable commutes [1]. Autonomous vehicles (AV) rely on cutting-edge technological advancements to safely guide them from one point to another while handling obstacles and adapting to traffic situations without human intervention [2]. By introducing benefits such as reduced driver stress, increased productivity, increased fuel efficiency, diminished demand for parking at destinations, and improved accessibility for many people with disabilities [1], AVs can change the traditional transportation landscape.

The benefits of autonomous driving technology (ADS) extend beyond increased convenience and comfort levels. The most valuable aspect of ADS is its ability to enhance safety by eliminating humans, who are often responsible for causing accidents, from the equation. While still technologically advanced and not yet readily available to consumers in their vehicles, skyrocketing levels of investment in ADS are accelerating advancements like never before. Human mistakes account for 94% of all motor vehicle accidents according to the data provided by the NHTSA. Fortunately, AVs can decrease the number of crashes resulting from driver errors by up to approximately 90% [3]. This reduction leads to potential cost savings of nearly 190 billion [3].

Despite the potential benefits of self-driving technology, it faces several constraints in terms of safety and dependability. As the technology is still in its early phases, issues such as cyber-attacks, system failures, and data privacy concerns may have serious consequences [4], [5]. Consequently, it is critical to create a solid safety architecture capable of ensuring the safe functioning of autonomous cars.

The objective of this study is to ensure runtime safety in autonomous cars. The capacity to make assurances about the reliability, safety, and behavior of autonomous vehicles while operating on the road is referred to as "runtime assurance." This study explores the difficulties and potential solutions for achieving runtime assurance in autonomous vehicles by considering factors such as fault tolerance, fault detection, system redundancy, monitoring techniques, and failure recovery strategies.

This study intends to contribute to the advancement of safe and dependable autonomous driving technologies by addressing issues of runtime safety assurance and disengagement. The conclusions and suggestions from this study have the potential to direct future research efforts, enhance the dependability and safety of autonomous cars, and inspire public trust in their widespread deployment.

II. Current state of Autonomous Vehicles technology

Autonomous vehicles rely heavily on fault tolerance and detection mechanisms to ensure safe and reliable operations. Three major players in the autonomous vehicle industry–Apollo Baidu, Waymo, and Cruise–have developed sophisticated systems to achieve minimal risk conditions in their vehicles.

Apollo Baidu, a Chinese autonomous vehicle company, uses three different strategies to ensure fault tolerance in its vehicles. First, it uses redundancy in critical components, such as sensors, actuators, and controllers, to mitigate the risk of system failures. Second, it employs a monitoring system that continuously tracks the vehicle performance and detects

This work has received funding from the European Union's Horizon Europe research and innovation program under grant agreement No 101076754 - AIthena project.

anomalies that may indicate a fault. Third, Baidu's system uses fault diagnosis algorithms to identify the root cause of a fault and provides recommendations for addressing it [6].

Waymo, a subsidiary of Alphabet Inc., also employs a multifaceted approach to fault tolerance and detection. It uses redundancy in key systems, such as sensors, actuators, and power supplies, as well as a comprehensive fault detection system that monitors vehicle performance in real time. In addition, Waymo uses a simulation software to test its vehicles under various conditions, including simulated faults, to identify potential issues before they occur on the road [7].

Cruise, a subsidiary of General Motors, uses a unique approach to fault tolerance that relies heavily on its sensor fusion system. It uses data from multiple sensors to create a holistic view of the vehicle's surroundings, allowing it to detect and respond to faults in real-time. Additionally, the Cruise system is designed to continue operating in the event of a fault, allowing it to safely navigate to a location where repairs can be made [8].

A. *Disengagements in Autonomous Vehicles*

Disengagement in autonomous vehicles offers insightful information on the state of AV technology. Disengagements occur when the transfer of control from autonomous mode to manual mode is initiated by human drivers or the ADS itself. The limitations and difficulties experienced by autonomous vehicles in real-world driving situations can be revealed by investigating the reasons for disengagement as well as their frequency. AV manufacturers are required to report disengagements annually thereby providing valuable insights into the current state of AV technology [9].

Waymo LLC reported the highest number of miles travelled by AVs, covering over 2.9 million miles. They also reported 170 disengagements, resulting in miles per disengagement of 17,060. Cruise LLC reported the lowest number of disengagements at just nine, with over 863,000 miles travelled, resulting in miles per disengagement of 95,901. Apple Inc. reported a significantly higher number of disengagements than other companies, with 5,982 disengagements reported for 125,096 miles travelled, resulting in miles per disengagement of only 21.

Disengagements were initiated by both the AV system and test driver. Test drivers initiated the majority of disengagements (89%) compared to those initiated by the AV system (11%). Most disengagements occurred on streets (58%), followed by freeways (31%), highways (10%), and urban areas (1%). Developing reliable and accurate AV technologies that can minimize errors is crucial for improving the safety of AVs on public roads.

The reports also provided a breakdown of the causes of disengagements. The most common cause was ADS prediction error, leading to 1,507 disengagements. Other causes included ADS perception, motion planning, location, map, hardware, and other errors. From this data, it is evident that various attributes related to ADS such as prediction, perception, and planning, need to be made more reliable. In an effort to do

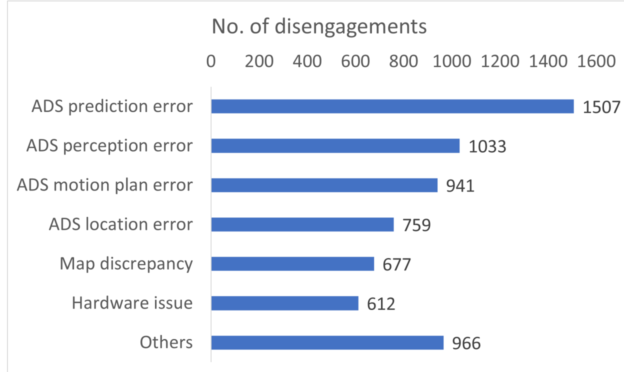

Fig. 1: Cause of disengagements

the same, a safety loop is applied consisting of various safety architectures, which are discussed in the next section.

III. Safety Architectures for Safety Critical Systems

For safety-critical systems used in autonomous vehicles, it is important that the system be capable of detecting and mitigating faults to ensure the safety of passengers and their surroundings. This is achieved by incorporating specific redundancies into the system to ensure that failures are detected, and in the case of a failure, the vehicle is brought into a safe state.

A. *Safety Architectures*

The *1oo2* architecture uses two independent modules for redundancy, as illustrated in Fig. 2. If one of the modules fails, it can be detected by the voter and the system is brought into a safe state.

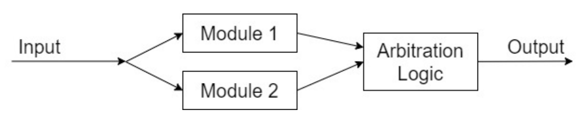

Fig. 2: 1 out of 2 architecture.

The *1oo2D* architecture is similar to the 1oo2 architecture - see Fig. 3 - where the additional diagnostics mechanisms monitors each module and this capability is quantified by a parameter called diagnostics coverage factor.

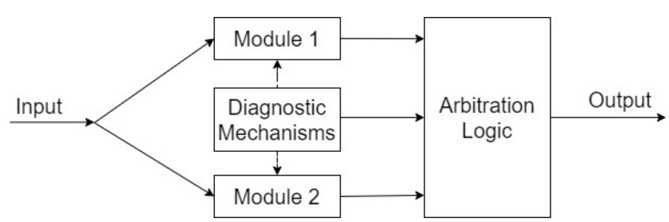

Fig. 3: 1 out of 2 architecture with diagnostics.

The *2oo3* architecture uses three modules in parallel and one voter, as shown in Fig. 4. This architecture is also called

Triple Modular Redundancy *TMR*. The outputs from the three modules were compared by using a majority voter. This architecture can handle one failure and continue to work as long as the two modules function properly.

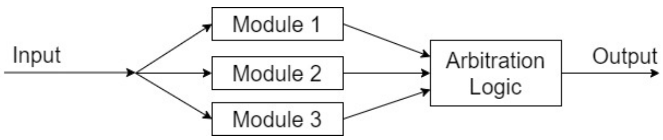

Fig. 4: 2 out of 3 architecture.

B. Reliability Assessment

The reliability of a system at time t is the probability that the system operates without failure in the interval $[0, t]$, provided the system was functioning properly at time $t = 0$ [10]. In safety-related systems, high reliability is generally required for operations without interruptions. Next, the reliability of the previously presented architectures is calculated.

Let us denote by $f(t)$ the probability density function of a failure, then:

$$F(t) = Prob(T \leq t) = \int_0^t f(t)dt \tag{1}$$

is the probability that a component fails at or before time t. The reliability of a component is (survive at least until time t)

$$R(t) = Prob(T > t) = 1 - F(t) \tag{2}$$

The conditional probability (or failure rate) is defined as the probability that a good component at time t will fail within the next short duration dt [11].

$$\lambda(t) = \frac{f(t)}{1 - F(t)} = \frac{1}{R(t)} \frac{dR(t)}{dt} \tag{3}$$

If we assume that λ is a constant value and $\lambda t << 1$ we obtain: $R(t) = e^{-\lambda t} \approx 1 - \lambda t$ as well as $F(t) = 1 - e^{-\lambda t} \approx \lambda t$ where λ denotes failure rate.

1) 1oo2 architecture: Considering that both modules should fail for the system to fail, the probability of failure of the system is given by:

$$F_s = (1 - e^{-\lambda t})(1 - e^{-\lambda t}) \tag{4}$$

Reliability is the complement of the probability to failure (see equation (4)), which is given by $R_s = 1 - F_s$. Therefore, we obtain:

$$R_s = R_a[1 - \left(1 - e^{-\lambda t}\right)\left(1 - e^{-\lambda t}\right)] \tag{5}$$

where R_a denotes the reliability of the arbitration logic Assuming an ideal arbitration logic (R_a = 1),

$$R_s = 2e^{-\lambda t} - e^{-2\lambda t} \tag{6}$$

2) 1oo2D architecture: This architecture is considered reliable when both modules are working, or either one module is working. The reliability of the system is given by:

$$R_s = R_a[R^2 + 2cR(1 - R)] \tag{7}$$

where c denotes the diagnostic coverage factor, which is defined as the probability that a faulty processor is correctly diagnosed, identified, and disconnected [11]. Furthermore, R_a is the reliability of the arbitration logic and R denotes the reliability of each module. Considering $R = e^{-\lambda t}$ and an ideal arbitration logic R_a = 1 we obtain:

$$R_s = e^{-2\lambda t} + 2ce^{-\lambda t}(1 - e^{-\lambda t}) \tag{8}$$

3) 2oo3 architecture: The reliability of a system that consists of N modules and needs at least K of them for proper operation $(K - of - N)$ according to [10] is:

$$R_s = \sum_{i=K}^{N} \frac{N!}{i!\,(N-i)!}\, e^{-\lambda i t}\left(1 - e^{-\lambda t}\right)^{N-i} \tag{9}$$

The best-known example of this type of system is the $2oo3$ system. By substituting the values of $N = 3$ and $K = 2$, we obtain:

$$R_s = R_a\left[\frac{3!}{2!} e^{-2\lambda t}\left(1 - e^{-\lambda t}\right) + e^{-3\lambda t}\right] \tag{10}$$

where R_a denotes the reliability of the arbitration logic. Considering an ideal arbitration logic, R_a=1, the reliability of this system is given by:

$$R_s = 3e^{-2\lambda t} - 2e^{-3\lambda t} \tag{11}$$

The reliabilities of the three architectures are summarized in Table I.

TABLE I: Reliability comparison.

No.	*Architecture*	*Reliability*
1	1oo2	$2e^{-\lambda t} - e^{-2\lambda t}$
2	1oo2D	$e^{-2\lambda t} + 2ce^{-\lambda t}(1 - e^{-\lambda t})$
3	2oo3	$3e^{-2\lambda t} - 2e^{-3\lambda t}$

In addition to reliability, another important factor to be considered while selecting the safety architecture is its complexity, which can be assessed based on the number of components.

Considering reliability and complexity, along with the capability of fault detection, a variant of $1oo2D$ architecture shown in Fig. 5 is selected an later on discussed in the runtime safety assurance section.

This architecture, dynamic redundancy with hot standby, comprises a primary module (performance loop) that is capable of performing the necessary functions and a backup module (safety loop), which is responsible only for safety-related functions that are performed in the event of failure of the primary module. In this study, the backup module on the sensor side has a $2oo3$ architecture, and the control logic and actuator side have a $1oo2D$ architecture.

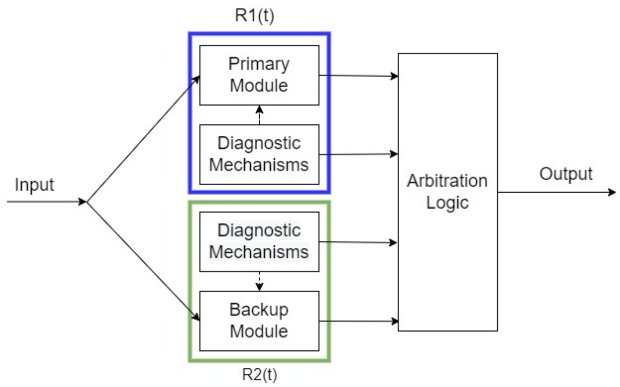

Fig. 5: Dynamic redundancy with hot standby.

IV. Runtime Safety Assurance for Autonomous Vehicles

Runtime safety assurance (RTA) is the dominant approach for enforcing safety in real-world autonomous and semi-autonomous systems. Runtime assurance systems guarantee the safety of increasingly complex and intelligent control systems by monitoring the state of the system and intervening when necessary [12], [13]. A critical feature of RTA systems is their ability to explicitly alter unsafe control inputs to ensure safety [14]–[16]. The RTA system monitors the state of the system and the output of the primary controller, the backup (safety) controller replaces and modifies the control input if the safety condition is violated (see Fig. 6). In this case, the autonomous vehicle executes an emergency maneuver under the supervision of the backup controller.

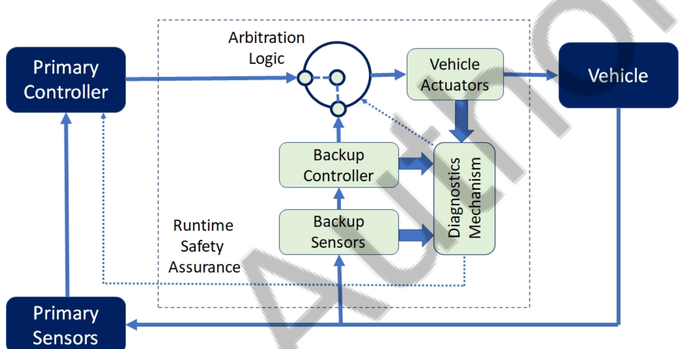

Fig. 6: Runtime safety assurance.

The overall block diagram resembles a safety architecture with dynamic redundancy with hot standby, where the safety loop is designed to be a dual-channel system (although not shown explicitly) with multi-modal sensing (diversity) and diagnostics. If the safety loop, which consists of backup sensors, backup controller, vehicle actuators, and diagnostic mechanism, fails, the failure is detected by the diagnostic mechanism, and the primary controller is instructed to execute an emergency maneuver (see the dotted line from the diagnostics mechanism to the primary controller).

The emergency maneuver can be described as follows:

- an emergency stop in the same lane
- a lane change to the emergency lane and an emergency stop in the lane
- a lane change to the emergency lane and operation with low speed to the nearest garage.

One of the main benefits of an RTA safety mechanism is the decoupled design and verification of the backup (safety) controller from the primary controller. This decoupling allows the RTA to focus on safety, whereas the primary controller is optimized for performance. The practical advantages of this RTA approach are that it provides a means of testing new control algorithms on existing hardware platforms without compromising safety and without requiring additional certification [12]. Furthermore, as the RTA system is generally simpler than a performance-based controller, its verification, validation, and certification are simpler, faster, and cheaper. The idea of separating the safety loop from the control loop is not new in the industry; however, a rigorous mathematical description and formal proof of how the primary and backup controllers interact very often is not in place.

In this paper, the verification, validation, certification, and assurance are defined/described as follows. *Verification* is an activity that determines whether a system meets the requirements, answering the question: “Did we build the system right?” *Validation* is assessing if the system meets the end user needs, answering the question: “Did we build the right system”. In contrast, *model validation* is evaluating how well the model represents reality. *Assurance* is justified confidence that the system functions as intended. *Certification* determines whether a system conforms to a set of criteria or standards [12].

V. Vehicle Model Identification and Validation

This section focuses on how to build reliable models. The model building has a natural process flow. System identification is a mature technical field with numerous commercially available software tools that allow efficient model building and validation using experimental data [17], [18].

We will discuss how to build and validate a simple mathematical model of the vehicle, which can be used to verify and validate the backup control algorithm as well as runtime safety assurance.

The host vehicle kinematic model - considering the center of the rear axle as a reference - can be written in the following form (see Fig. 7):

$$\begin{cases} \dot{x}_h &= v_h \cos\theta_h \\ \dot{y}_h &= v_h \sin\theta_h \\ \dot{\theta}_h &= \omega_h = v_h/R = (v_h/L)\tan\delta_h \end{cases} \tag{12}$$

where x_h and y_h are the vehicle position in inertial frame, θ_h is heading angle. Furthermore, L is the vehicle wheelbase, CG is the centre of gravity and O is the Instantaneous Centre of Rotation (ICR), R is the rotation radius. When v_h is the host vehicle speed and δ_h is steering angle, we observe that: $\tan\delta_h = L/R$.

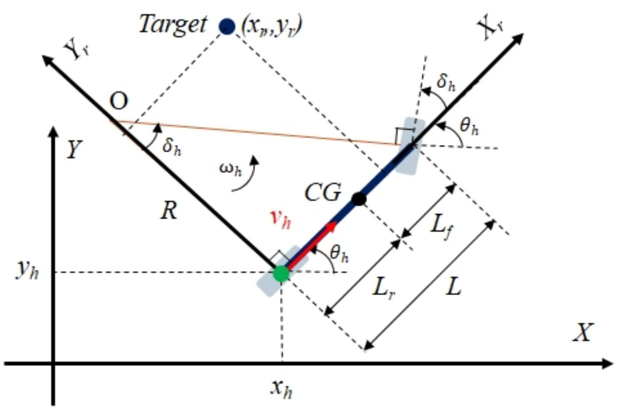

Fig. 7: Host vehicle kinematic model.

Our use-case considers urban driving, where the host vehicle speed shall not exceed $50[km/h]$. Hence, a simplified vehicle dynamics model that is valid at low speeds is derived. Next, the longitudinal and lateral dynamics of the vehicle are identified. One of the requirements imposed on the input signal – applied during system identification experiments – is that it should be persistently exciting of a certain order (e.g., it should contain sufficiently distinct frequencies) [17], [19].

Therefore, a pseudo-random binary signal (PRBS), which is a deterministic signal with white-noise properties, is generated by linear feedback shift registers. As a remark, in the case of non-linear systems, a so-called multi-level random signal is generated and is used in the identification of the system [18].

The longitudinal and lateral dynamics (as shown in Fig. 8) of the vehicle are identified as follows:

- Longitudinal dynamics: from the reference acceleration to vehicle acceleration
- Lateral dynamics: from reference steering angle to measured steering angle (steering actuator dynamics) and from the measured steering angle to vehicle heading.

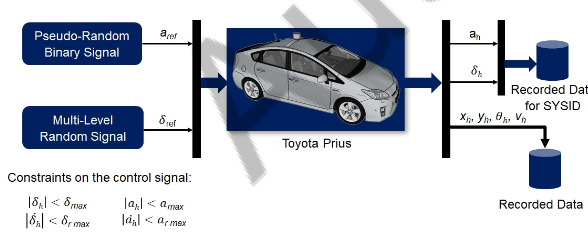

Fig. 8: System identification of the vehicle: input and output signals.

Next, the input-output signals and a comparison between the measured output and identified model output are shown in the case of longitudinal and lateral dynamics identification (see Fig. 9 and Fig. 10), where only the steering actuator dynamics is shown wherein the measured signals have been filtered and the time delay compensated.

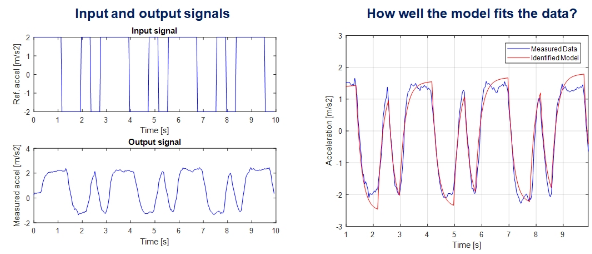

Fig. 9: Longitudinal dynamics identification.

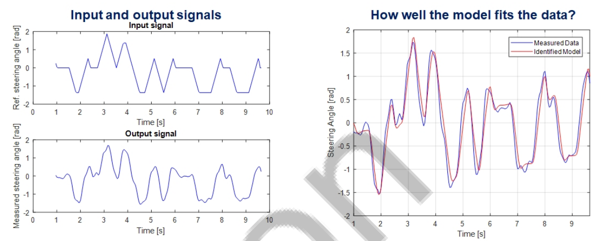

Fig. 10: Steering actuator dynamics.

At low vehicle speeds, in both cases, the powertrain and the steering actuator can be modeled as a linear system with time delays.

The longitudinal dynamics is modeled as a first-order system with time delay, as follows:

$$\frac{a_h(s)}{a_{ref}(s)} = \frac{1}{0.2s+1} e^{-0.3s} \tag{13}$$

The system's time delay $T_d = 0.3[s]$ can negatively impact performance and stability during closed-loop control. A Smith predictor is implemented to effectively control the system owing to the input time delay [20]. Therefore, in later discussions, the time delay is omitted.

VI. Runtime Safety Assurance - Implicit Approach

In this paper, safety constraints are defined using inequality constraints on the state. For example, $\varphi_i : X \to \mathbb{R}$ for $i \in 1, ..., M$ where M denotes the number of safety constraints with $\varphi(x) \geq 0, \forall i$.

The set of states that satisfies all the safety constraints is referred to as a constraint set.

$$C_A := \{x \in X | \varphi_i(x) \geq 0; i \in \{1, ..., M\}\} \tag{14}$$

The constraint set for collision avoidance between the host and target vehicle over the states

$$x = [x_h, y_h, \dot{x}_h, \dot{y}_h, x_t, y_t, \dot{x}_t, \dot{y}_t] \tag{15}$$

is given by:

$$C_A := \left\{x \in \mathbb{R}^4 | (x_t - x_h)^2 + (y_t - y_h)^2 - d_s^2 \geq 0\right\} \tag{16}$$

where d_s is the safety distance.

It is important to note that there may exist states in C_A that satisfy the safety constraints at a given time, but will lead to violations in the future. A meaningful definition of safety must contain additional information on whether the safety constraints will continue to be satisfied for all times with a particular control law, subject to particular dynamic and actuation constraints [**?**].

A system is safe if the state belongs to C_A for all times; in other words, the state lies in a forward invariant subset of the constraint set, that is:

$$x(t_0) \in C_S \Rightarrow x(t) \in C_S; \forall t \geq t_0 \tag{17}$$

When $C_S \subseteq C_A$, C_S is said to be a safe set.

It is important to note that forward invariance, and by extension, safety, is a property of the closed-loop system and is not defined in the absence of a controller.

Explicit identification of the forward invariant subset is typically obtained only at the expense of conservatism. However, we can implicitly define C_S in terms of closed-loop trajectories under the control law. For example, consider a backup control law $u_b : X \rightarrow U$ and let $\phi^{u_b}(x;t)$ represent the state reached after starting at $x \in X$ and applying u_b for t units of time. In this case, the set:

$$C_S = \{x \in X | \forall t \geq 0, \phi^{u_b}(x;t) \in C_A\} \tag{18}$$

is an invariant (safe) set under u_b and is entirely constrained in C_A.

By integrating the dynamic forward, it is possible to check whether the individual states are safe. Moreover, while the minimum can be solved in a closed form, the solution can be used to define an explicit safe set.

A. *Collision avoidance - emergency stop*

Let us consider two vehicles (host and target) located in the same lane traveling on a straight road in the same direction (see Fig. 11).

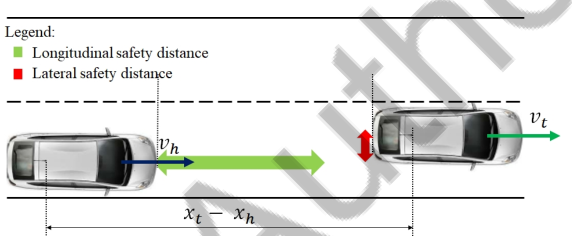

Fig. 11: Safety distances during collision avoidance.

The host vehicle dynamics is described as follows:

$$\begin{bmatrix} \dot{x}_h \\ \dot{v}_h \\ \dot{a}_h \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -b_h \end{bmatrix} \begin{bmatrix} x_h \\ v_h \\ a_h \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ b_h \end{bmatrix} u_h \tag{19}$$

where $b_h = 1/0.2 = 5$ according to the identified model and u_h is the reference acceleration a_{ref}. Similarly, the target vehicle is described as:

$$\begin{bmatrix} \dot{x}_t \\ \dot{v}_t \\ \dot{a}_t \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -b_t \end{bmatrix} \begin{bmatrix} x_t \\ v_t \\ a_t \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ b_t \end{bmatrix} u_t \tag{20}$$

The safety constraint is collision avoidance $\varphi(x) = x_t - x_h$ and the constraint set is $C_A = \{x \in \mathbb{R} | \varphi(x) - d_s \geq 0\}$, where d_s is the minimum admissible longitudinal safety distance.

The safety distance is defined according to [21], as:

$$\begin{aligned} d_s^{lon} &= d_{min}^{lon} + v_h^{lon} T_R + \frac{1}{2} a_{max,accel,h}^{lon} T_R^2 + \\ &+ \frac{1}{2} \frac{(v_h^{lon} + a_{max,accel,h}^{lon} T_R)^2}{a_{min,brake,h}^{lon}} - \frac{1}{2} \frac{(v_t^{lon})^2}{a_{max,brake,t}^{lon}} \end{aligned} \tag{21}$$

where T_R is the reaction time of the host vehicle, and it is assumed that during the reaction time, the host vehicle accelerates with maximum acceleration $a_{max,accel,h}^{lon}$ and then brakes with minimum deceleration $a_{min,brake,h}^{lon}$ - worst condition. On the other hand, the target vehicle brakes with maximum deceleration $a_{max,brake,t}^{lon}$. The upper index lon indicates that the speed and acceleration are related to the longitudinal speed and acceleration of vehicles [21].

A similar equation can be derived for the lateral safety distance. In this case, the worst condition is that the host vehicle accelerates during the reaction time $a_{max,accel,h}^{lat}$ and then brakes with minimum deceleration $a_{min,brake,h}^{lat}$ and the target vehicle accelerates during the reaction time $a_{max,accel,t}^{lat}$ and then brakes with minimum deceleration $a_{min,brake,t}^{lat}$ [21].

$$\begin{aligned} d_s^{lat} = d_{min}^{lat} &+ \frac{v_h^{lat} + v_1}{2} T_R - \frac{v_t^{lat} + v_2}{2} T_R \\ &\frac{v_1^2}{2a_{min,brake,h}^{lat}} + \frac{v_2^2}{2a_{min,brake,t}^{lat}} \end{aligned} \tag{22}$$

where

$$\begin{aligned} v_1 &= v_h^{lat} + a_{max,accel,h}^{lat} T_R \\ v_2 &= v_t^{lat} - a_{max,accel,t}^{lat} T_R \end{aligned} \tag{23}$$

where v_h^{lat} is positive and v_t^{lat} is negative. The upper index lat indicates that the speeds and accelerations are related to the lateral speed and acceleration of the vehicles.

As a remark, safety is violated when both longitudinal and lateral safety distances are violated; in our case, the lateral safety distance is violated by default because the vehicles are in the same lane.

By neglecting the reaction time T_R, the longitudinal safety distance can be written in a simpler form, where the upper index lon has been omitted:

$$d_s = d_{min} + \frac{v_h^2}{2a_h} - \frac{v_t^2}{2a_t} \tag{24}$$

The back-up controller $u_b = u_h = -1$ and $b_h = 5$ is integrated over a 5-s horizon, where Fig. 12 shows the simulation results of simulation with: $b_t = 5$ and $u_t = -2$, the target brakes stronger than the host.

In Fig. 13 the actions of the primary and back-up controllers are shown. The safety distance is constantly monitored by the runtime assurance system, and when the safety distance is violated, the back-up controller takes over the control and starts to brake the vehicle; thus, collision is avoided. A similar runtime safety assurance mechanism is applied in the case of failure of sensing and perception system, as well as in the case of localization failure, where the related fault detection will be discussed in the next sections.

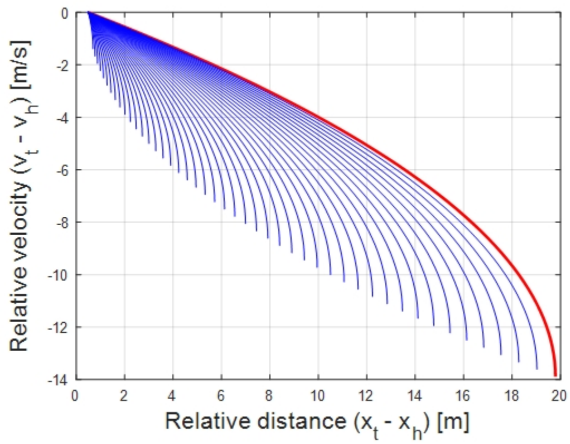

Fig. 12: Safety set - collision avoidance.

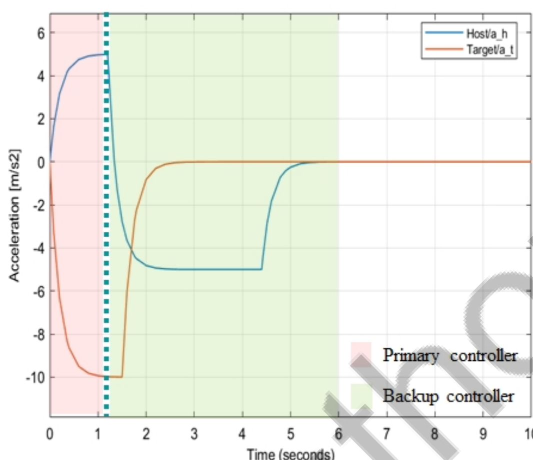

Fig. 13: Primary and back-up controllers' actions.

B. Fail operational - emergency maneuver with lane change

In this subsection, the emergency maneuver with lane change is discussed. When the primary controller fails, the vehicle executes an emergency maneuver under the supervision of the backup controller, as shown in Fig. 14.

The host vehicle mathematical model is described by equation (25), where the steering actuator dynamics are neglected.

$$\begin{cases} \dot{x}_h &= v_h \cos\theta_h \\ \dot{y}_h &= v_h \sin\theta_h \\ \dot{v}_h &= a_h \\ \dot{a}_h &= -b_h a_h + b_h u_h \\ \dot{\theta}_h &= (v_h/L)\tan\delta_h \end{cases} \tag{25}$$

In the equation above, $u_h = u_h(t)$ is the reference acceleration and $\delta_h = \delta_h(t)$ is the reference steering angle. This equation can be solved in real-time over a finite time horizon.

The reference path during the emergency lane change is defined using a 3rd-order parametric Bezier curve, defined as follows:

$$\begin{aligned} B_x(t) &= x_1(1-t)^3 + 3x_2 t(1-t)^2 + 3x_3 t^2(1-t) + x_4 t^3 \\ B_y(t) &= y_1(1-t)^3 + 3y_2 t(1-t)^2 + 3y_3 t^2(1-t) + y_4 t^3 \end{aligned} \tag{26}$$

where $P_1 = (x_1, y_1), P_2 = (x_2, y_2), P_3 = (x_3, y_3), P_4 = (x_4, y_4)$ for $0 \leq t \leq 1$.

Hereby, if we assume a lateral acceleration comfort threshold $a_y = 1.6[m/s^2]$ and if we denote with r the turning radii along the path, we can find out the maximum allowable speed: $v_{max} \leq \sqrt{a_y r}$

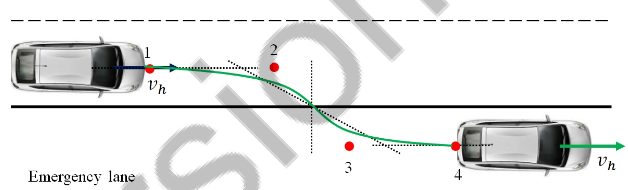

Fig. 14: Emergency maneuver with lane change.

The backup controller design and verification are not detailed in this paper; however, we mention that the controller is designed systematically, where the closed-loop system's equations are written in the polar coordinate frame and the stability condition of the closed-loop system is established.

In the following sections, the diagnostics mechanisms associated with sensing and perception system failure (see also [22], [23]), as well as vehicle localization failure, are discussed. In these cases, the backup controller can no longer execute the emergency maneuver, which implies a lane change, so an emergency stop is activated.

VII. Reliability of the Sensing and Perception System

Autonomous vehicles rely on multi-modal sensing, sensors such as camera, radar, and LiDAR in combination with sensor fusion and object tracking are used to reliably detect, classify and localize objects (see Fig. 15).

Cameras are more efficient in determining the features of an object, and are hence employed for functionalities such as understanding traffic signs. Radar is capable of capturing the motion characteristics of an object with high resolution, whereas LiDAR has wide coverage for detection along with superior ranging performance. Cameras are vulnerable to varying illumination conditions, and LiDAR is erroneous under extreme weather conditions, whereas radar is more robust under adverse weather, environmental, and illumination conditions [24]. The advantages of each type of sensor are utilized by using the concept of sensor fusion. Sensor fusion collectively processes inputs from various sensors and derives an interpretation of the environment surrounding the vehicle with a level of certainty.

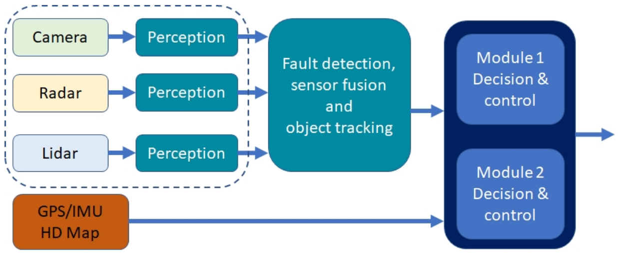

Fig. 15: Multi-modal sensing and perception system.

The reliability of the sensing and perception system, which consists of YOLOv3 is tested in the Simcenter Prescan simulation environment (see Fig. 16) [25]. YOLOv3 was trained on the COCO dataset and was used for object detection and classification [26]. The accuracy of the data from YOLOv3 was validated by a frame-wise comparison with the reference ground truth data. The objects detected by YOLOv3 need to be matched with objects in the ground truth, which is an ordered list. A similar comparison can be made between datasets coming from two different real sensors, such as radar and camera.

Fig. 16: Simcenter Prescan simulation environment.

The association of objects is based on the object labels and the proximity of the center points of the object bounding boxes. Once the objects are associated, the objects are classified into three classes: "Easy," "Moderate" and "Difficult" with respect to ease of detection. This classification is based on the size of the bounding box, object truncation, and object occlusion.

Object occlusion is a measure of the degree of obstruction in perceiving an object varying between 0 (fully visible), 1 (partly occluded), 2 (largely occluded), and 3 (fully occluded). Object truncation is a measure of whether the object is leaving the camera's field of view and is varying continuously between 0 (non-truncated) and 1 (truncated).

Each object occlusion value has its own difficulty categorization. For a detected object, the first difficulty value is obtained from the occlusion and the second difficulty value is obtained from the difficulty classification based on the truncation and size ratio shown in Fig.17. Finally, the higher difficulty among the two was selected.

		Truncation	Occlusion			
			0	1	2	3
Object size (ratio)	> 0.05 Large	< 0.2	Easy	Easy	Easy	Difficult
		[0.2, 0.5]	Easy	Easy	Moderate	Difficult
		> 0.5	Easy	Moderate	Difficult	Difficult
	[0.025, 0.05] Medium	< 0.2	Easy	Easy	Moderate	Difficult
		[0.2, 0.5]	Easy	Moderate	Difficult	Difficult
		> 0.5	Moderate	Difficult	Difficult	Difficult
	< 0.025 Small	< 0.2	Easy	Moderate	Difficult	Difficult
		[0.2, 0.5]	Moderate	Difficult	Difficult	Difficult
		> 0.5	Difficult	Difficult	Difficult	Difficult

Fig. 17: Detection difficulty classes.

Objects matched in both data sets (YOLOv3 and ground truth) are marked as "True Positive" (TP). Any object that is not detected by YOLOv3 has been labeled as "False Negative" (FN) and objects that have been detected wrongly by YOLOv3 are labeled as "False Positive" (FP). The validation accuracy of YOLOv3 object detection is calculated as a ratio of the number of correct predictions to the total number of predictions.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{27}$$

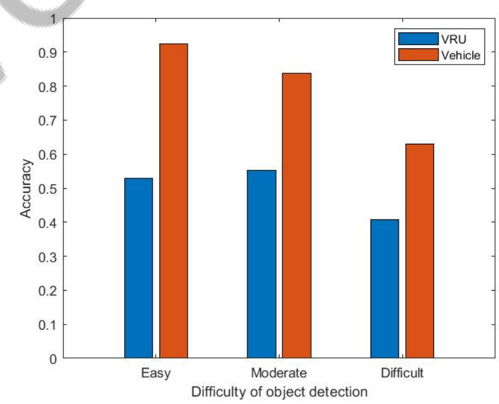

Fig. 18: Validation accuracy vs detection difficulty.

Based on the difficulty of the detection, the calculated accuracy values are grouped and the mean values for each class are calculated - the obtained results are shown in Fig. 18. From the figure, it can be observed that the accuracy of detecting vehicles is higher compared to that of vulnerable road users (VRU) owing to their larger size. It can also be inferred that the accuracy of detecting objects decreases with an increase in the difficulty of detection.

Any instance of a "False Negative" or "False Positive" is considered a misdetection and is classified as a failure. A failure in the sensing and perception system shall trigger an emergency maneuver, such as moving to the emergency lane and going to the nearest garage or stop, depending on the safety architecture used.

VIII. Fault Detection in Autonomous Vehicle Systems

In autonomous vehicle systems, fault detection plays a crucial role in ensuring vehicle safety and reliability. The methodology employed in this study aims to simulate the addition of noise to the vehicle positioning system and analyze its impact on the accuracy of position measurements. Potential faults in the system were identified by analyzing the Euclidean errors and using a limit-checking fault detection method. The following steps were followed to achieve these objectives:

Noise simulation: To simulate the noise in the positioning system, a mathematical model was developed based on known sources of interference and disturbances. This model allowed the generation of noise that closely mimicked real-world conditions. Simulated noise was then added to the recorded position data to introduce variability and perturbations to the system.

Assumptions and considerations: During the simulation process, several assumptions and considerations were made. First, it was assumed that the noise introduced into the system followed a random distribution, with characteristics similar to those observed in practical scenarios. Additionally, noise was assumed to affect both the x- and y-coordinates of vehicle position. These assumptions allow for a realistic representation of noise in the positioning system.

Euclidean error analysis: The Euclidean error was computed to evaluate the impact of noise on position accuracy. The Euclidean error represents the straight-line distance between the measured and reference positions in two-dimensional space. By quantifying the magnitude of the differences between these positions, the Euclidean error provides insights into positioning accuracy under different noise conditions. The analysis of Euclidean errors allowed for the identification of patterns, trends, and outliers, providing a comprehensive understanding of the system's performance. The Euclidean error plot plays a crucial role in analyzing the accuracy of Euclidean distance measurements. For instance, Fig. 19 represents the Euclidean error plot for the ideal situation (blue line) and a noise variance of 0.1 (green line). These plots enabled the observation of the deviations of the measured distances from the ideal values, helping to assess the impact of noise on positioning accuracy.

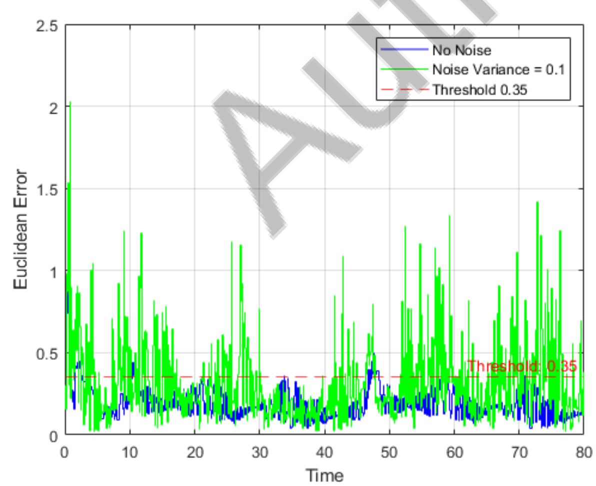

Fig. 19: Euclidean error plot with noise variance of 0.1.

A. Threshold determination using empirical statistical distribution

The threshold was determined using an empirical statistical distribution [27], [28] to detect faults based on Euclidean error distance. The following steps were followed:

1) Statistical histogram: A statistical histogram of the Euclidean error distance was constructed to visualize the distribution of the errors (see Fig. 20).
2) Cumulative distribution function: The cumulative distribution function of the Euclidean error distance was calculated as shown in Fig. 21. This function provides insights into the probability distribution of the errors.
3) Threshold determination: Based on the cumulative distribution function, percentiles were determined. In this study, the 90th percentile was selected as the threshold. The threshold value was found to be 0.35 [m] for the given noise variances of 0.1 and 0.01 (see Fig. 19 and Fig. 21).

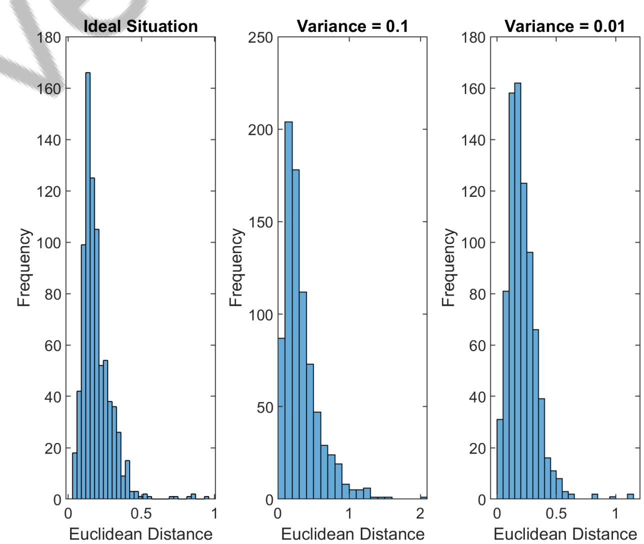

Fig. 20: Statistical histogram of the Euclidean error distance.

Conclusion

The paper explored various aspects of autonomous vehicle technology, with a focus on safety and reliability. The disengagement data analysis highlighted the significance of creating reliable mechanisms to reduce errors and guarantee the safety of autonomous vehicles on public roads. The analysis of the safety architecture highlighted the importance of including redundancy and fault-detection methods to reduce hazards.

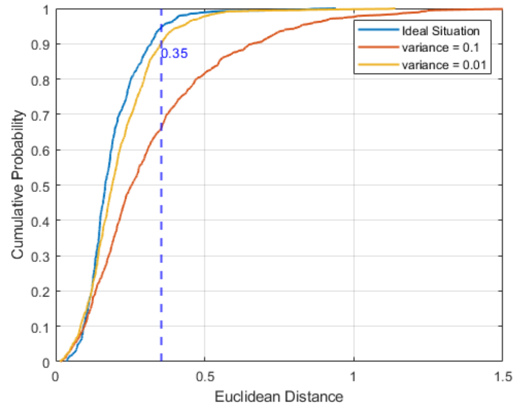

Fig. 21: Euclidean error - cumulative distribution function.

Multiple safety architectures were assessed in terms of reliability and the most suitable architecture was selected to be employed to ensure runtime safety.

This paper also emphasizes the main benefits of the RTA safety mechanism, which decouples the design and verification of the backup (safety) controller from the primary controller. This decoupling allows the RTA to focus on safety, while the primary controller is optimized for performance.

This implies that RTA allows the testing of new control algorithms on existing hardware platforms without compromising safety and without requiring additional certification [**?**]. Furthermore, since the RTA system is generally simpler than a performance-based controller, its verification, validation, and certification are simpler, faster, and cheaper.

In the sensing and perception system, the fault detection mechanism is based on the detection of ”False Positive” or ”False Negative” values, comparing camera sensor data with the ground truth. In practice, this can be easily extended to a multi-modal sensing architecture that contains camera, radar, and LiDAR sensors. In addition, the variation in the validation accuracy among different object classes with respect to the difficulty of object detection was studied.

The sensing and perception subsystem - part of the runtime assurance - has a *2oo3* architecture if we consider camera, radar, and LiDAR as sensors. The decision, control, and actuation subsystem - part of the runtime assurance subsystem–has a *1oo2D* architecture.

Related to control system failure, the impact of localization error on control failure has been studied, and a simple fault detection method is proposed.

Finally, it is highlighted that a failure in the sensing and perception system or control system - part of the runtime safety assurance–shall trigger an emergency maneuver for the autonomous vehicle.

Our comprehensive study aims to contribute to the advancement of AV technology, safety, and reliability, paving the way for a future in which autonomous vehicles have become an everyday reality while ensuring utmost safety and efficiency.

References

[1] The future of autonomous vehicles (AV), McKinsey. Available: https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/autonomous-drivings-future-convenient-and-connected

[2] Automated vehicles for safety, NHTSA. Available: https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety

[3] Autonomous vehicles factsheet, Center for Sustainable Systems, University of Michigan. 2022. Pub. No. CSS16-18. Available: https://css.umich.edu/publications/factsheets/mobility/autonomous-vehicles-factsheet

[4] Disruptive trends that will transform the auto industry, McKinsey. Available: https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/disruptive-trends-that-will-transform-the-auto-industry

[5] Ethics guidelines for trustworthy AI, EU-Act, 2019. Available: https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai

[6] Apollo 5.5 software architecture Available: https://daobook.github.io/apollo-doc-old/docs/specs/Apollo_5.5_Software_Architecture.html.

[7] Waymo safety. Available: https://waymo.com/safety/

[8] Cruise safety commitment, approach to safety, driverless car safety FAQs. Available: https://www.getcruise.com/safety/

[9] Disengagement reports, 2022. Available: https://www.dmv.ca.gov/portal/vehicle-industry-services/autonomous-vehicles/disengagement-reports/

[10] C. E. Ebeling, An introduction to reliability and maintainability engineering, Waveland Press, 2019.

[11] I. Koren, and C. M. Krishna, “Fault tolerant systems,” Elsevier, 2007.

[12] K. L. Hobbs, M. L. Mote, M. C. L. Abate, S. D. Coogan, and E. M. Feron. ”Runtime assurance for safety-critical systems: An introduction to safety filtering approaches for complex control systems,” IEEE Control Systems Magazine, vol. 43, no. 2, pp. 28-65, April 2023, doi: 10.1109/MCS.2023.3234380.

[13] K. L. Hobbs, J. Davis, L. Wagner and E. Feron, ”Formal Specification and Analysis of Spacecraft Collision Avoidance Run Time Assurance Requirements,” 2021 IEEE Aerospace Conference (50100), Big Sky, MT, USA, 2021, pp. 1-16, doi: 10.1109/AERO50100.2021.9438135.

[14] L. Sha et al., “Using simplicity to control complexity,” IEEE Software, vol. 18, no. 4, pp. 20–28, 2001.

[15] U. Mehmood, S. Bak, S. A. Smolka, and S. D. Stoller, “Safe cps from unsafe controllers,” arXiv preprint arXiv:2102.12981, 2021.

[16] S. Chen, Y. Sun, D. Li, Q.Wang, Q. Hao and J. Sifakis, “Runtime Safety Assurance for Learning-enabled Control of Autonomous Driving Vehicles,” arXiv preprint arXiv:2109.13446, 2021.

[17] L. Ljung, ”System identification - Theory for the user,” New York: Prentice-Hall, 1987.

[18] R. Haber, and L. Keviczky. ”Nonlinear system identification,” Springer, 1999.

[19] A. Forrai, ”Embedded control system design - A model based approach,” Springer Verlag, 2012.

[20] M. Araki, and H. Taguchi. ”Two-degree-of-freedom PID controllers,” Engineering International Journal of Control Automation and Systems, 2003.

[21] S. Shalev-Shwartz, S. Shammah, and A. Shashua.” On a formal model of safe and scalable self-driving cars,” arXiv:1708.06374, 2017.

[22] N. Carlini, and D. Wagner. ”Towards evaluating the robustness of neural networks,” IEEE Symposium on Security and Privacy (SP), IEEE, 2017, pp. 39-57.

[23] C. Szegedy, et al. ”Intriguing properties of neural networks,” arXiv preprint arXiv:1312.6199, 2013.

[24] A. Pandharipande, et al. ”Sensing and machine learning for automotive perception: a review,” IEEE Sensors Journal, vol. 23, no. 11, pp. 11097-11115, June 1, 2023, doi: 10.1109/JSEN.2023.3262134.

[25] Simcenter Prescan, 2023. Available: https://plm.sw.siemens.com/en-US/simcenter/autonomous-vehicle-solutions/prescan

[26] J. Redmon, et al. ”YOLOv3: An incremental improvement,” arXiv, cs.CV, 1804.02767, 2018.

[27] D. Miljković, ”Fault detection methods: A literature survey,” 2011 Proceedings of the 34th International Convention MIPRO, Opatija, Croatia, 2011, pp. 750-755.

[28] D. Miljković. ”Fault detection using limit checking: A brief introductory review,” 44th International Convention on Information, Communication and Electronic Technology (MIPRO), Opatija, Croatia, 2021, pp. 908-913, doi: 10.23919/MIPRO52101.2021.9596786.