Accepted version

To appear in *Proc. of the 9th International Conference on System Reliability and Safety (ICSRS)*, Turin, Italy, 2025

© 2025 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Modeling and Safety Analysis of Automated Vehicle Teleoperation with Dynamic Fault Trees

Shahriar Hasan*, Matthias Volk†, Nazakat Ali*, Falak Sher‡, Peter Wallin§, Katrin Sjöberg¶
*School of Innovation, Design and Engineering, Mälardalen University, Sweden
†Formal System Analysis, Eindhoven University of Technology, The Netherlands
‡DGB Technologies, United States
§Boliden Mineral AB, Stockholm, Sweden
¶Volvo Autonomous Solutions, Göteborg, Sweden
Corresponding author: S. Hasan (e-mail: shahriar.hasan@mdu.se)

***Abstract*—Teleoperation is emerging as a key enabler for deploying automated vehicles in safety-critical environments where full autonomy is not yet feasible. Ensuring safety in such systems is challenging due to their reliance on, for example, human operators, wireless communication networks, mixed-traffic interactions, and onboard automated systems, as well as the interdependencies among these components that can trigger cascading failures. To address this, we present a case study on the safety analysis of teleoperated driving in underground mines. We model the teleoperation architecture using a Dynamic Fault Tree (DFT), an extension of traditional static fault trees with explicit modeling support for functional dependencies, order-dependent failures, and redundancy mechanisms. The DFT model is rigorously evaluated through probabilistic model checking across a broad range of system-wide metrics, including degradation behavior and criticality of components. The results provide insights into system reliability, the sensitivity of failure probabilities with respect to different component failure rates, and the behavior of the system in fully functional, fail-operational, and degraded modes. Overall, the findings highlight the critical components that dictate system dependability and establish a structured basis for enhancing the safety of teleoperated vehicle systems to comply with industrial safety standards.**

***Index Terms*—Dynamic fault trees, Fault tree analysis, Model checking, Teleoperation, Remote Driving, Automated driving.**

I. Introduction

Fully autonomous vehicles (SAE J3016 Level 5 autonomy [1]) that can operate in all scenarios without human intervention remain a long-term research challenge [2]. Teleoperation serves as a middle ground between autonomous and human-driven vehicles, where remote vehicles are supervised or directly driven by humans from a Remote Operation Station (ROS). In particular, the teleoperation of heavy vehicles is increasingly adopted in confined areas such as mines, quarries, construction zones, forestry, and agriculture. By combining remote driving or tele-supervision with vehicle autonomy, teleoperation enables unmanned operation in hazardous or inaccessible environments. This approach enhances both safety and efficiency but also introduces complex interactions between human operators, communication networks, and automated vehicles [3]. Failures in communication links, perception systems, operator inputs, or vehicle control units may compromise safe operation, potentially leading to hazardous situations.

Ensuring safety in this context is challenging because heavy vehicles must comply not only with performance requirements but also with stringent safety standards for machinery. ISO 12100 [4] provides a general framework for risk assessment and risk reduction in machinery, while ISO 3849-1 [5] extends these principles to functional safety, in alignment with IEC 61508 [6]. Unlike ISO 26262 [7], which is tailored for passenger vehicles, these standards are specifically relevant for off-road and industrial vehicles, where teleoperation and automation are increasingly deployed. A rigorous safety analysis approach is therefore required to provide systematic evidence of compliance and to quantify residual risks in such safety-critical systems.

Fault Tree Analysis (FTA) [8], [9] is widely used for safety evaluations in industrial systems. However, classical (static) fault trees cannot adequately capture the dynamic dependencies typical of teleoperation architectures, such as failures induced by communication latency, fail-operational safety mechanisms, or sequential interactions between operators and vehicles. To address these limitations, *Dynamic Fault Trees (DFTs)* [10] extend FTA with constructs for order-dependent failures, standby components, and functional dependencies. DFTs have demonstrated strong potential for analyzing safety-critical systems in domains such as automated vehicle guidance [11], nuclear power plants [12], and high-speed rail [13]. In teleoperation, although safety assessments have been performed using *System-Theoretic Process Analysis (STPA)* [14], [15], an in-depth safety analysis using DFTs remains missing in the literature. Teleoperation introduces unique challenges, including the interplay between remote operators, wireless communication networks, onboard safety modules, and autonomous driving functions, which require both quantitative safety assurance and explicit consideration of degraded operation modes.

In this work, we present a case study of automated vehicle teleoperation in underground mines and perform a DFT-based safety analysis. The functional architecture of the teleoperation system, including the ROS, human operator, communication

This work was supported by Swedish Knowledge Foundation [Stiftelsen för kunskaps- och kompetensutveckling (KKS)] "TeleDrive: Teleoperated and Autonomous Driving of Unmanned Vehicles in Confined Areas" Project under Grant 20230076.

networks, safety systems, Advanced Driver-Assistance System (ADAS), sensors, and actuators, is systematically modeled and transformed into a multi-layer DFT. To conduct the analysis, we employ the SAFEST tool [16], which supports the modeling and probabilistic model checking of DFTs by automatically transforming them into *Continuous-Time Markov Chains (CTMCs)* for scalable evaluation [17]. Through probabilistic model checking, the resulting models are analyzed to obtain precise quantitative safety and reliability measures. In particular, model checking allows for analyzing a broad range of metrics on the CTMC state space. This facilitates a rigorous assessment of the complete teleoperation system, its degraded performance modes, and various importance measures, which are crucial for meeting the requirements of ISO 12100 and ISO 23849.

The remainder of the paper is structured as follows. Section II introduces the technical background on dynamic fault trees, and Section III reviews relevant studies on DFTs and teleoperation. Section IV then presents the teleoperation system model, while Section V outlines the corresponding DFT developed from this model. Section VI presents the evaluation metrics and analysis of the results. Finally, Section VII summarizes the main conclusions and outlines directions for future work.

II. Background

A. Dynamic fault trees

Fault trees model how component failures propagate through a system and lead to a system failure [9]. *Dynamic fault trees (DFT)* [9], [10] extend (static) fault trees by dynamic gates which model order-dependent failures, functional dependencies, and spare management. Fig. 1 depicts the relevant elements in (dynamic) fault trees. We introduce the DFT elements in the following and refer to [18] for further details.

Basic Events (BE), the leaves of the fault tree, represent atomic components which are not further subdivided. BE fail according to an exponential distribution with a failure rate.

Static gates are the gates of static fault trees and represent Boolean failure conditions over the state of the gate inputs. An AND-gate fails if all its inputs are failed. An OR-gate fails if at least one input is failed. The voting gate VOT_k is the generalization of both gates and fails if at least k out of n inputs are failed.

The *Priority-AND (PAND)* represents order-dependent failures. The PAND-gate fails if all inputs fail in order from left to right. If an input fails out of order, the PAND becomes *fail-safe* and can never fail.

The *Probabilistic dependency (PDEPp)* allows failures to be forwarded. If the first input (the *trigger*) is failed, then, with probability p, all other inputs (the *dependent events*) are immediately rendered failed as well. A PDEP can for instance model common-cause failure.

The *SPARE-gate* models spare management. Initially, the leftmost input is actively used and can fail according to its associated failure rate. All other inputs are not in use and fail according to their *passive* failure rate which is typically lower than the active failure rate—or even zero in case of a cold SPARE-gate. If the currently used input fails, the next input is claimed and starts to be actively used. The SPARE-gate fails if all inputs are failed.

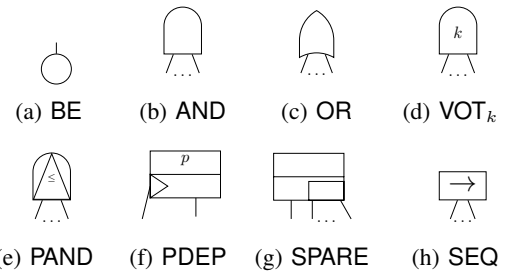


Fig. 1: Node types in static (top) and dynamic (all) fault trees.

The *sequence enforcer (SEQ)* requires that all inputs fail in order from left to right. In contrast to a PAND, the SEQ does not allow failures out of order.

B. Fault tree analysis

Dynamic fault trees can be analyzed via probabilistic model checking [17], [19]. The DFT is translated into a *CTMC* where states represent the current status of DFT elements and transitions correspond to BE failures governed by the corresponding failure rate. The translation from DFT to CTMC makes use of several optimizations, such as exploiting independent modules and symmetries [17].

The CTMC can be effectively analyzed by model checking techniques [20]. The state-based model allows the calculation of metrics expressed in mathematical logic such as *continuous stochastic logic (CSL)* [20]. This expressiveness combined with extensive tool support, e.g., SAFEST [16] and Storm [21], allows for computing a broad range of metrics on a DFT, such as overall system reliability, mean-time-to-failure (MTTF), importance metrics of sub-components, and the probability to encounter degraded states.

III. Related Works

Due to the limitations of static fault trees in modeling capabilities and capturing functional dependencies, safety analysis using DFTs is gaining increasing interest in complex industrial and automotive systems. For instance, Ghadhab et al. [11] applied DFTs to the safety analysis of automated vehicle guidance systems and carried out a quantitative evaluation of various system-wide and degraded metrics. Rao et al. [12] proposed a Monte Carlo simulation–based approach for DFT analysis to overcome the limitations of traditional Markov models, which often suffer from state-space explosion and restrictive assumptions of exponential failure/repair distributions. Their method, validated on nuclear power plant case studies, demonstrated the ability to realistically capture sequence-dependent failures, spare management, and testing effects in large-scale probabilistic safety assessments. In [17], Volk et al. introduced model-checking–based techniques to accelerate DFT analysis. By exploiting structural reductions such as symmetry, partial-order reduction, and don't-care detection, along with partial

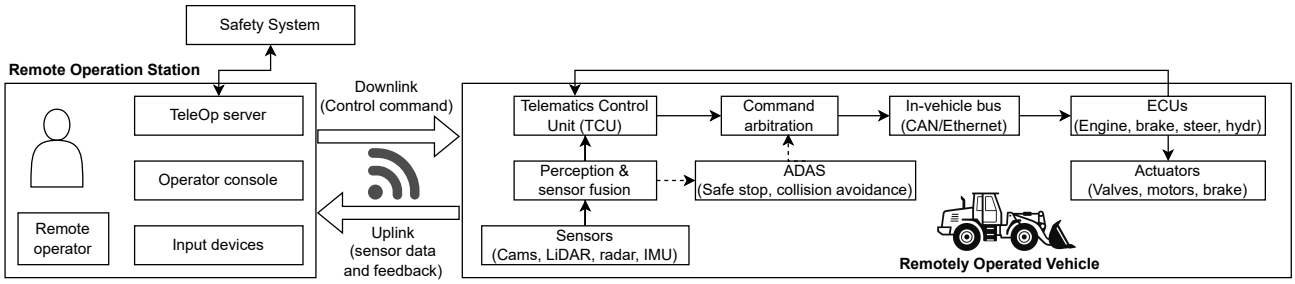


Fig. 2: Functional block diagram of automated vehicle teleoperation.

state-space generation for approximations, they achieved significant performance gains. Wang et al. [13] combined DFTs with Bayesian Networks to address fault diagnosis challenges in the train control and management system of high-speed rail. By transforming DFT models into BNs, their approach enabled probabilistic reasoning under uncertainty and bidirectional inference, supporting both predictive reliability analysis and root-cause diagnosis.

In the context of teleoperation, Hoffmann and Diermeyer [14] employed STPA for the safety assessment of remotely operated road vehicles. Their work systematically analyzed the teleoperation control structure to identify unsafe control actions and their causal factors, identifying a broad range of risks related to human operators, communication networks, and sensory perception faults. Similarly, Kamaraj et al. [15] considered the complexities of control loops between the remote vehicle dispatcher and driver to perform STPA analysis, identifying system-level hazards in automated vehicle teleoperation. They highlighted the lack of situational awareness and the mismatched mental models between drivers and dispatchers as key contributors to unsafe vehicle control. While the studies in [14] and [15] provide valuable hazard identification frameworks for teleoperated driving, they lack executable models, model-checking capabilities, and simulation-based performance evaluation, which limits their applicability for quantitative safety assurance. In contrast, our work develops a comprehensive DFT model of automated vehicle teleoperation and carries out an extensive quantitative evaluation, providing novel insights into system-level behavior, degraded-mode performance, and component importance measures.

IV. Teleoperation System Model

Automated vehicle teleoperation can be broadly classified into *remote driving* and *remote assistance* [2]. In remote driving, a human operator directly controls the vehicle from a *Remote Operation Station (ROS)*, whereas in remote assistance, the vehicle operates autonomously and requests human intervention only when it cannot resolve a situation on its own. In this case study, we focus on the remote driving of heavy machinery in underground mines, where the vehicle operates within the mine while the driver remains at the ROS and has full control of the vehicle.

The ROS is equipped with a complete driving interface, including brake and accelerator pedals, joystick or steering controls, a speedometer, and a console panel for auxiliary functions, as shown in Fig. 2. In addition, the ROS is connected to a teleoperation server that hosts a safety layer, ensuring that all outgoing commands pass through safety checks before being transmitted downstream to the vehicle.

In the uplink direction, the remote vehicle is equipped with multiple sensors such as radar, LiDAR, cameras, and an Inertial Measurement Unit (IMU). These sensors collect a wide range of telemetry data, including video streams, acceleration, position, velocity, orientation, system health, and diagnostic logs. The data is transmitted to the ROS through the Telematics Control Unit (TCU). The communication infrastructure may combine Ethernet with wireless technologies such as Wi-Fi, private 5G networks, or a hybrid solution to balance low latency, high bandwidth, and reliable connectivity.

In addition to teleoperation, the vehicles are equipped with ADAS that can execute fallback maneuvers such as emergency braking or lane keeping in the event of, e.g., a communication outage with the ROS. This ADAS acts as a redundancy layer to ensure minimal safe operation. A command arbitration module selects between teleoperation commands and ADAS commands based on factors such as information freshness and situational context, as depicted in Fig. 2. Final actuation commands are transmitted via the in-vehicle bus to the respective Electronic Control Units (ECUs), which carry out the required actions.

In the teleoperation of heavy machinery, the communication network is a key enabler. Reliable and low-latency data transmission is essential for maintaining situational awareness at the ROS and ensuring timely control of the remote vehicle. Network performance directly influences the freshness of information and, by extension, the safety of real-time remote driving. Additional challenges arise in mixed-traffic scenarios, particularly in narrow mine tunnels where vehicles may need to coordinate at stop points to allow passage. On the ROS side, performance is also constrained by the cognitive load, situational awareness, and skill of the remote driver, which differ considerably from those of an onboard operator. Finally, the security and resilience of the communication network must be taken into account, as cyberattacks or system vulnerabilities could lead to catastrophic outcomes.

V. DFT MODEL FOR TELEOPERATION SYSTEM

The DFT of the teleoperation system in Fig. 2 is shown in Fig. 3, combining both static and dynamic fault tree gates. Overall, the system includes components such as the remote operator, ROS, safety system, communication unit, mixed traffic interaction, and control commands. Failure of the top-level event can either happen due to actuator failures of the remote vehicle, or due to failure of a component and the ADAS system which acts as a redundancy.

In modeling the remote operator failure, factors such as loss of situational awareness due to delays in video feeds and telemetry data, as well as the cognitive load of the operator, are considered, as illustrated in Fig. 3. This loss of situational awareness can lead to command misselection by the operator. The PDEP gate *IncorrectOperatorInput* models how a loss of situational awareness can lead to, e.g., command misselection. This misselection may ultimately result in incorrect control inputs and failures in steering or braking. To capture delayed reactions caused by sensor perception latency, a SEQ gate is employed. The SEQ ensures that first perception latency occurs, then a delay in the decision is possible which ultimately can lead to incorrect control inputs. In addition, incorrect control inputs under poor visibility conditions are modeled using a SPARE gate, where the failure of the leftmost child sequentially activates the subsequent events.

For the ROS, console and software crash failures are represented using VOT_k gates, which can behave as an OR or AND gate depending on the threshold value k. Downstream actuation command loss failures are mitigated by a redundant command channel, modeled using a SPARE gate. Finally, mixed traffic interaction failures are modeled by considering delays in commands transmitted from the ROS to the vehicle, the uncertainties associated with human-driven vehicle maneuvers in the narrow mine tunnels and sensor fusion issues.

VI. PERFORMANCE MEASURES AND RESULTS ANALYSIS

The SAFEST tool [16] is employed for analyzing the DFT modeled for the Teleoperation use case in this paper. This section first defines some of the performance metrics utilized for the analysis and then presents the results analysis. Note that we carry out the analysis employing over 20 different performance metrics, partially based on [11]. It is beyond the scope of this paper to define all performance metrics for the sake of conciseness; hence, we refer to the SAFEST Manual[1] for detailed explanations.

A. Performance Metrics

The metrics are based on two standard queries in CSL which make use of the time-bounded reachability ($F^{\leq t}$) and time-bounded until ($U^{[0,t]}$), c.f. [20].

- *Event Probability within a Time-Bound*: Probability of a specific event occurring within the time t.

$$P_{\text{event}}(t) \;=\; P\big(F^{\leq t}\text{event}\big)$$

- *Time-Bounded Reach-Avoid Probability*: Probability that event_2 occurs within t while event_1 does not occur before.

$$P_{\text{RA}}(t) \;=\; P\big(\neg\text{event}_1\; U_{[0,t]}\; \text{event}_2\big)$$

The performance metrics are formulated using these metrics.

- *Unreliability*: Probability of system failure within time t.

$$P_{\text{unrel}}(t) \;=\; P\big(F^{\leq t}\; \text{sys\_failed}\big)$$

- *Average Failure Probability per Hour (AFH)*: The average probability of system failure per time unit within $[0, t]$; computed as the unreliability at operational lifetime t, normalized by t.

$$\text{AFH}(t) = \frac{1}{t}\, P\big(F^{\leq t}\; \text{sys\_failed}\big)$$

- *Mean Time to Failure (MTTF)*: The expected time to system failure, obtained as the integral of the reliability function over time.

$$\text{MTTF} \;=\; \mathbb{E}[T_{\text{sys\_failed}}] \;=\; \int_0^\infty (1 - P_{\text{unrel}}(t))\, dt$$

The following metrics consider the behavior in degraded mode, where specific components have already failed, but the overall system is still operational.

- *Full Function Availability (FFA)*: Probability that the system remains fully operational within time t, i.e., it is neither failed nor degraded.

$$\text{FFA}(t) \;=\; 1 - P\big(F^{\leq t}\; (\text{sys\_failed} \vee \text{degraded})\big)$$

- *Failure Without Degradation (FWD)*: Probability that the system fails directly without first entering a degraded state within time t.

$$\text{FWD}(t) = P\big((\neg\text{degraded})\; U^{[0,t]}\; (\text{sys\_failed} \wedge \neg\text{degraded})\big)$$

- *Mean Time from Degradation to Failure (MTDF)*: The expected time of system operation upon entering the degraded state before reaching complete failure.

$$\text{MTDF} = \sum_{s \in \text{degraded}} \big(P(\neg\text{degraded}\; U\; s) \cdot \mathbb{E}[T^s_{\text{sys\_failed}}]\big)$$

- *Failure under Limited Operation in Degradation (FLOD)*: FLOD quantifies the probability that the system, after entering a degraded state, fails within a time bound (the drive cycle) while still being in that degraded mode. It is calculated by multiplying the probability of reaching the degraded state and the probability of subsequently reaching the system failure within the drive cycle.

$$\text{FLOD}(t) \;=\; \sum_{s \in \text{degraded}} \Big(P((\neg\text{degraded})\; U^{[0,t]}\; s) \cdot P^s(F^{\leq \text{drive\_cycle}}\; \text{sys\_failed})\Big)$$

- *System Integrity under Limited Fail-Operation (SILFO)*: Probability the system avoids both direct failure (FWD) and failure from a degraded state within a drive cycle (FLOD).

$$\text{SILFO}(t) \;=\; 1 - \big(\text{FWD}(t) + \text{FLOD}(t)\big)$$

[1] https://www.safest.dgbtek.com/src/components/installation/installation.html

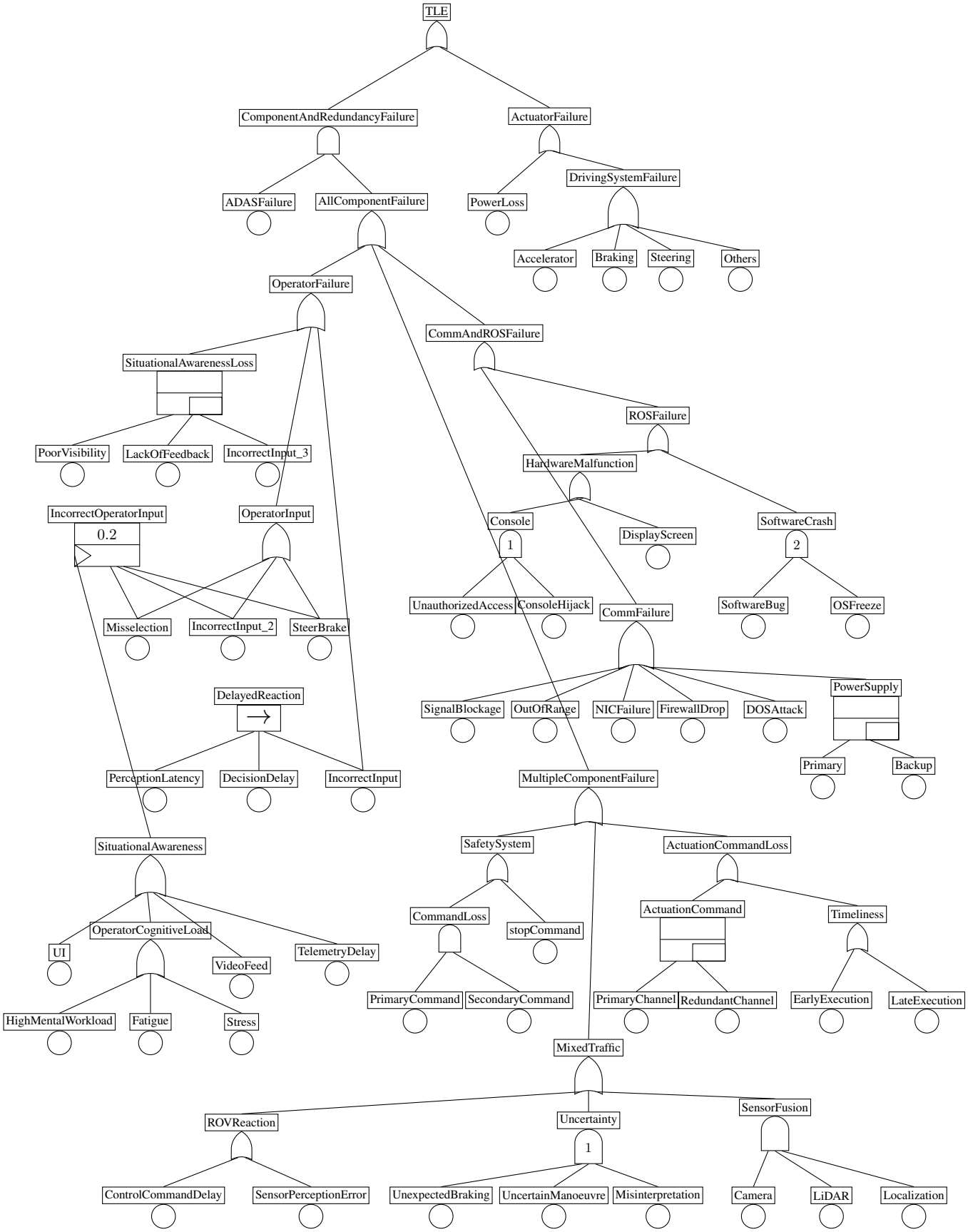


Fig. 3: Dynamic Fault Tree of automated vehicle teleoperation in underground mines.

Importance metrics compute the criticality of specific component failures w.r.t. the overall system failure [22], [23].

- *Birnbaum Importance (BI)*: Sensitivity of the system unreliability to the unreliability of component i, quantifying how critical component i is to the overall reliability.

$$BI_i(t) \;=\; \frac{\partial\, P_{\text{unrel}}(t)}{\partial\, P_{\text{component}_i\text{\_failed}}(t)}$$

- *Risk Achievement Worth (RAW)*: The impact on the system unreliability of making the component always failed.

$$RAW_i(t) \;=\; \frac{P^{\text{component}_i\text{\_failed}}_{\text{unrel}}(t)}{P_{\text{unrel}}(t)}$$

- *Risk Reduction Worth (RRW)*: The impact on the system unreliability of making the component fully reliable.

$$RRW_i(t) \;=\; \frac{P_{\text{unrel}}(t)}{P^{\neg\text{component}_i\text{\_failed}}_{\text{unrel}}(t)}$$

- *BAGT+ (Before-After Guided Test, upper bound)*: The change in the system's MTTF when a component is forced to fail.

$$BAGT_i^{+} \;=\; \left|\, \text{MTTF} - \text{MTTF}^{\text{component}_i\text{\_failed})} \,\right|$$

- *BAGT- (Before-After Guided Test, lower bound)*: The change in the system's MTTF when a component is assumed to be fully reliable.

$$BAGT_i^{-} \;=\; \left|\, \text{MTTF} - \text{MTTF}^{\neg\text{component}_i\text{\_failed}} \,\right|$$

We analyze the performance of the teleoperation system using metrics from three different categories: *system-wide safety performance*, *performance under degradation*, and the *importance of system components* for safety. For instance, the unreliability, AFH, and event probability metrics are used with model checking to capture the system-wide safety performance, while the FFA, FWD, MTDF, FLOD, and SILFO metrics are used to analyze the system under degraded states. We also analyze the importance metrics, such as BI, RAW, and BAGT, to obtain a holistic understanding of the system components contributing to system safety.

B. Results Analysis

This section first presents the system-wide performance, followed by the performance under degraded conditions and the importance measures. To this end, experiments are conducted to evaluate the system's performance across all metrics, considering an operational lifetime of 10,000 hours and a 1-hour driving cycle. Subsequently, selected metrics are analyzed in greater detail through a sensitivity analysis by varying the failure rates. It should be noted that the failure rates of certain basic events are taken from available literature, while the remaining rates are assumed for the purpose of analysis. These assumptions do not necessarily represent the actual failure rates of a real teleoperation system. Nevertheless, the in-depth sensitivity analysis presented here provides valuable insights for industry, both in understanding the impact of different failure rates on system performance and in identifying the maximum allowable failure rates required to achieve a desired level of dependability and safety. The initial failure rates assumed for the ADAS system, hardware, software, human operator, security, communication, and control inputs are 1×10^{-5}, 1×10^{-7}, 3.03×10^{-5}, 1×10^{-3}, 1×10^{-5}, 1×10^{-6}, and 4.86×10^{-5}, respectively.

TABLE I: System-wide performance with 10k h of operational lifetime (#CTMC States: 21610; #Transitions: 178219).

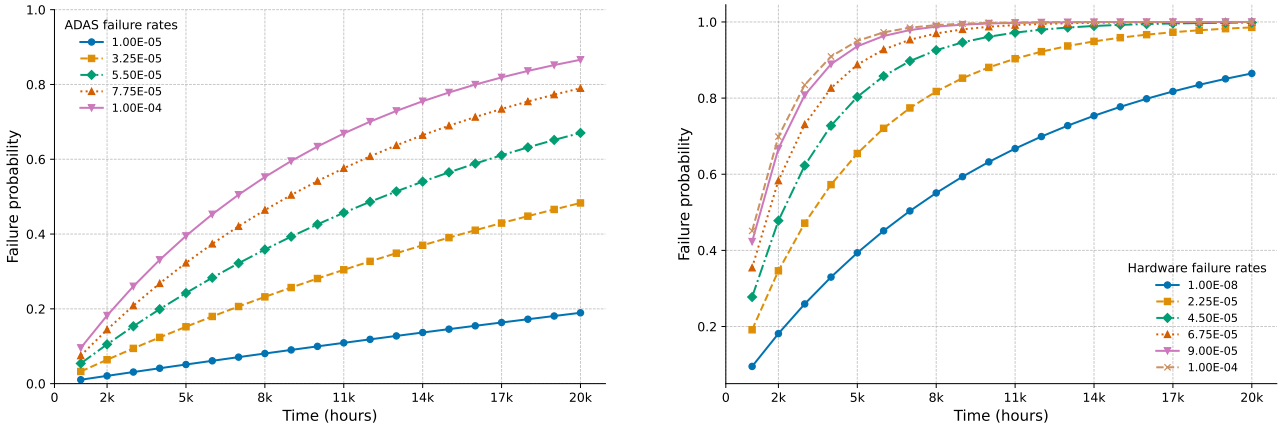
Metrics	**Results**	**Analysis duration (s)**
Unreliability	9.97E-02	3.44
Average failure probability/hour (AFH)	9.97E-06	3.43
Mean time to failure (MTTF)	9.52E+4	3.45
Event probability	1	3.49

TABLE II: Event probability within 20k h time-bound.

Event Name	**Probability**
Remote operator station (ROS) failure	7.10E-01
Permanent communication failure	5.52E-01
Actuator failure	9.95E-03
ADAS Failure	8.65E-01
Safety system failure	3.92E-02
Situational awareness loss due to poor visibility	2.00E-03
Actuation command loss	1.97E-04
Primary command channel	1.98E-02
Emergency stop command lost	1.98E-02
Attacks (DOS, spoofing or jamming)	1.81E-01
Control console failure	3.30E-01
Software system crash due to bugs or OS freeze	2.07E-01
Signal lost due to persistent blockage	3.96E-05
Video feed failed due to outage	3.00E-01
Incorrect control input due to high-latency perception	1.25E-01
Incorrect control input due to poor visibility	2.00E-03
Braking system failure	2.00E-03
Top-level event failure	8.66E-01
Control command delay	1.98E-02
Sensor perception error	2.00E-03
Sensor fusion failure	7.90E-08
Human operator failure	1
All components failure	1
Mixed traffic interaction failure	1

1) Performance of the Complete Teleoperation System: Table I presents the overall system performance for an operational lifetime of 10,000 hours. When performing model checking of the DFT with the system-wide metrics, the SAFEST tool generated a CTMC with 21,610 states and 178,219 transitions. The corresponding analysis durations for each metric are reported in Table I. The results show that the teleoperation system has an unreliability of approximately 10^{-1} within 10,000 hours, with a MTTF of roughly 10^5 hours. While the average failure probability per hour (AFH $\approx 10^{-5}$) is relatively low, it accumulates to a significant value over long operational lifetimes. Furthermore, the event probability of 1 in Table I signifies the eventual system failure probability under unbounded time, reflecting the fact that any system will ultimately fail if operated indefinitely.

We also analyzed the event probabilities at the component



(a) Unreliability with varying ADAS failure rates

(b) Unreliability with varying Hardware failure rates

Fig. 4: Sensitivity analysis of system unreliability with varying failure rates for 20,000 hours of operational lifetime.

or basic event level of the fault tree over a 20,000-hour operational lifetime, as shown in Table II. The higher probabilities can be explained by either the position of the basic events higher up in the fault tree (e.g., ADAS failure), higher assigned failure rates (e.g., human failure in the ROS), or OR gates with a large number of children (e.g., permanent communication failure). By contrast, hardware failures exhibit comparatively lower probabilities due to their lower failure rates, even when located high in the fault tree and in the absence of explicit redundancy. Events such as sensor fusion failure, actuation command loss, or signal loss due to persistent blockage remain comparatively rare, since their propagation to the top event depends on combinations of AND or SPARE gates. From a teleoperation perspective, one of the main dependability bottlenecks arises from the human driver (event probability of 1 in Table II), who is constrained by physiological limitations, as well as from the delivery of video feeds with low latency through the communication system. Another critical challenge is the interaction between remotely driven and human-driven vehicles, which also reaches an event probability of 1 within the 20k-hour time bound. Ultimately, under this time bound, all components fail with probability 1, and the ADAS system serves as the only redundancy prior to top-level failure, playing a crucial role in determining the overall system reliability.

As failures of the ADAS system and hardware are critical contributors to overall system failure, we performed a sensitivity analysis by varying their failure rates. Fig. 4 illustrates the system unreliability for ADAS and hardware systems with different failure rates over a 20k-hour operational lifetime. As expected, higher ADAS failure rates lead to a significantly steeper growth in system unreliability, as shown in Fig. 4a. For the lowest rate (1×10^{-5}), unreliability remains below 0.2, whereas at a failure rate of 1×10^{-4}, system unreliability exceeds 0.85. This demonstrates that even a modest increase in the ADAS failure rate drastically reduces overall system dependability and emphasizes the importance of a reliable ADAS. In comparison, hardware failures produce a much steeper unreliability curve (Fig. 4b), primarily because actuator-related hardware lies high in the fault tree and failures propagate directly to the system level.

TABLE III: Results of degraded metrics with 10k h operational lifetime and 1 h driving cycle (#CTMC States: 85968; #Transitions: 998284).

Metrics	Results	Analysis duration (s)
Full function availability (FFA)	6.31E-01	16.72
Failure without degradation (FWD)	3.69E-01	16.7
Mean time from degradation to failure (MTDF)	7.35E+4	37.88
Minimal degraded reliability (MDR)	0	17.68
Failure under limited operation in degradation (FLOD)	7.98E-02	26.86
System integrity under limited fail-operation (SILFO)	5.52E-01	43.33
Reach-avoid probability	9.89E-01	16.81
Reach-avoid probability (time-bounded)	3.65E-01	16.86

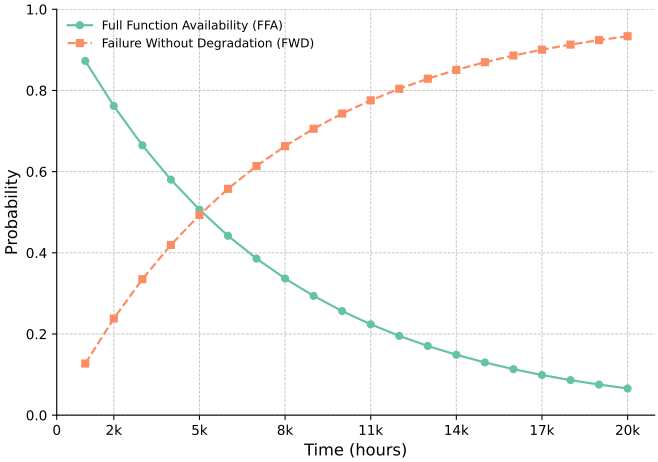
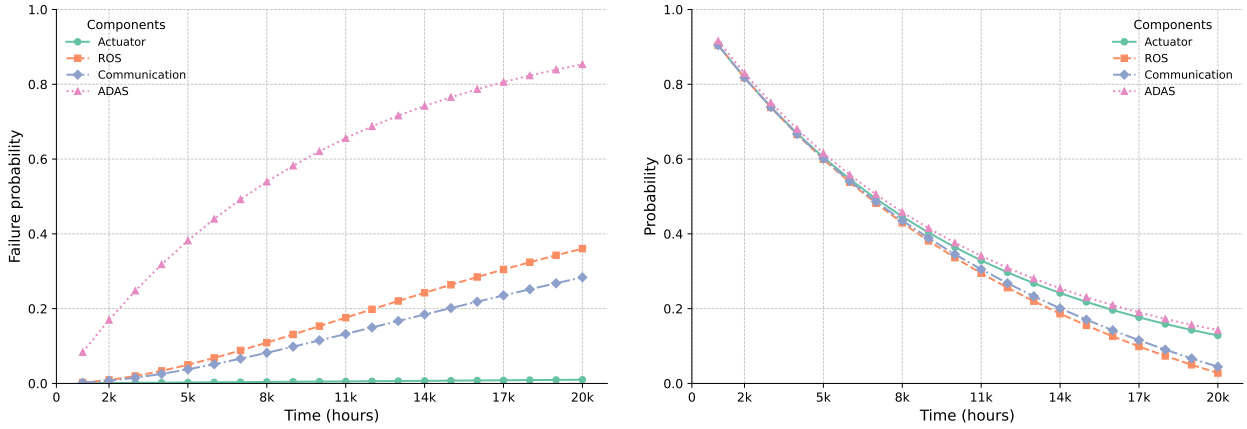


Fig. 5: Full Function Availability (FFA) vs. Failure Without Degradation (FWD) over 20k hours of operational lifetime.

2) System Performance in Degradation: Table III presents the results for the degraded metrics, considering an operational lifetime of 10k hours and a 1-hour driving cycle. In this analysis, 11 states in the DFT (Fig. 3) were treated as degraded



(a) Failure under Limited Operation in Degradation (FLOD) (b) System Integrity under Limited Fail-Operation (SILFO)

Fig. 6: Analysis of FLOD and SILFO metrics assuming various degraded components.

states: camera data loss, primary command channel failure, emergency stop command loss, high-latency perception, poor visibility, video feed loss, telemetry delay, display screen malfunction, communication outage, persistent signal blockage, and ADAS failure.

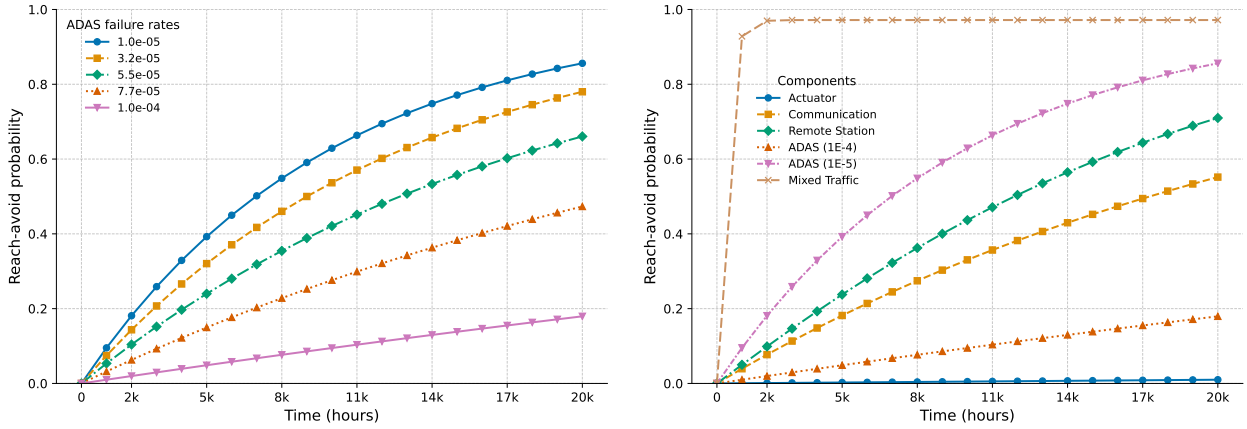
The analysis generated a CTMC with 85,968 states and approximately 1 million transitions. The results show that the system remains fully functional without degrading or failing first (i.e., FFA) in about 63% of the cases, while the FWD metric indicates that nearly 37% of failures occur without the system entering degraded states first. Fig. 5 illustrates the evolution of FFA and FWD over a 20k-hour lifetime. At the start of operation, the system is almost certainly fully functional, and the probability of failure without degradation is negligible. Over time, however, FFA gradually declines while the probability of direct failure (FWD) increases. The two curves intersect at around 5k hours, after which direct failure without degradation becomes more likely than continued full functionality.

The MTDF metric in Table III shows that once the system enters a degraded state, it can on average remain in that state for about 73,500 hours before failing. By contrast, the minimal degraded reliability (MDR = 0) indicates that certain degraded states (e.g., ADAS) cannot sustain operation at all and immediately lead to failure. The probability that the system first reaches a degraded state and then fails within the 1-hour driving cycle is approximately 0.0798 (FLOD). Adding this to the probability of failure without degradation (FWD) yields 0.448, implying that the system avoids both FWD and FLOD, i.e., maintains SILFO, with a probability of 0.552. This means the system remains either fully functional or degraded without failure about 55% of the time.

To better understand FLOD behavior for different teleoperation components, Fig. 6a presents FLOD curves for different degraded components over a 20k-hour operational lifetime and a 1-hour driving cycle. The results show that when a 1-hour time limit is imposed on actuators in degraded mode, the system failure probability remains very low. In contrast, operating ADAS in degraded mode results in significantly higher failure probabilities, especially as the overall operational lifetime increases. Similarly, degraded operation of the remote operator station and communication also substantially increases the likelihood of system failure over time. The SILFO curves in Fig. 6b confirm this trend, showing that the probability of avoiding both FLOD and FWD gradually decreases as the operational time grows.

The unbounded reach–avoid probability in Table III considers 11 different basic events or components that constitute the degraded states as the reference event to be avoided (event_2). The result shows that the likelihood of eventually seeing a system failure without encountering any degraded event first is 0.989 (almost certain). This probability drops sharply to 0.365 when a realistic operational time limit of 10k hours is imposed. When only ADAS is considered as the reference failure component and its failure rates are varied, the resulting reach–avoid probability curves are shown in Fig. 7a. As the ADAS failure rate increases, the probability of system failure occurring before ADAS failure decreases over time, since ADAS is more likely to fail first. At lower ADAS failure rates, the probability of reaching system failure without ADAS failing first becomes much more pronounced. In either case, the results demonstrate that ADAS reliability strongly influences whether failures bypass ADAS or not, making it a critical factor in the teleoperation system.

The reach–avoid probability with other components treated as the reference failure event (event_2) is presented in Fig. 7b. When mixed traffic interaction is used as the reference event, the probability of system failure occurring before mixed traffic interaction is almost certain. This is because mixed traffic interaction lies deeper in the fault tree and its occurrence depends on combinations of AND and VOT_k gates, making it far less likely than other failure pathways. By contrast, components such as communication, ROS, actuators, or ADAS fail with comparatively higher probability, which reduces the



(a) Reach-avoid probability with varying ADAS failure rates

(b) Reach-avoid probability with various component failures

Fig. 7: Time-bounded reach–avoid probability within 20k hours of operational lifetime: (a) probability of occurring system failure without occurring ADAS failure before with varying failure rates, and (b) probability of occurring system failure without occurring various other component failures before.

reach–avoid probability of direct system failure.

3) Importance Evaluation of System Components: Table IV presents the results of various importance metrics for the teleoperation system and ranks the four most critical components. The BI results show that ADAS and actuator failures rank the highest among all components, indicating that the reliability of these two subsystems strongly influences the overall system reliability. This is in line with the previous results which showed especially the influence of ADAS. The CI, RAW, and RRW metrics yield consistent results, with failures such as those of the remote station, communication, human operator, or safety system having only marginal impacts. The rationale is that ADAS serves as redundancy for most components, except actuators; hence, while other teleoperation components are important for system dependability, their failures can often be compensated for by ADAS.

The DIF metric highlights that human operator, mixed traffic interaction, and command loss failures return values close to 1, demonstrating a strong correlation with system failure. ADAS also shows high importance with a DIF value of 0.955, albeit being ranked fourth. Finally, the BAGT$^+$ (upper bound) results indicate that eliminating actuator and ADAS failures can significantly increase the mean time to failure (MTTF $\approx 10^5$ hours). Conversely, the BAGT$^-$ (lower bound) results show that forcing ADAS failure leads to the shortest MTTF ($\approx 1.9 \times 10^6$ hours lost), underscoring its role as a single point of failure.

TABLE IV: Results of importance metrics.

Importance metric	Component	Result	Rank
Birnbaum Index (BI)	ADAS failure	9.95E-01	1
	Actuator failure	9.05E-01	2
	Human operator failure	3.40E-04	3
	Mixed traffic interaction failure	2.15E-04	4
Criticality Importance (CI)	ADAS failure	9.52E-01	1
	Actuator failure	4.74E-02	2
	Remote station failure	1.24E-05	3
	Communication failure	9.63E-06	4
Risk Achievement Worth (RAW)	ADAS failure	1.00E+01	1
	Actuator failure	1.00E+01	2
	Human operator failure	1.00E+00	3
	Mixed traffic interaction failure	1.00E+00	4
Risk Reduction Worth (RRW)	ADAS failure	2.00E+01	1
	Actuator failure	1.05E+00	2
	Human operator failure	1.00E+00	3
	Safety system failure	1.00E+00	4
Diagnostics Importance Factor (DIF)	Human operator failure	1.00E+00	1
	Mixed traffic interaction failure	1.00E+00	2
	Command loss failure	1.00E+00	3
	ADAS failure	9.55E-01	4
Before–After Guided Test, upper bound (BAGT+)	Actuator failure	9.52E+04	1
	ADAS failure	9.51E+04	2
	Mixed traffic interaction failure	1.38E-01	3
	Safety system failure	1.38E-01	4
Before–After Guided Test, lower bound (BAGT–)	ADAS failure	1.90E+06	1
	Actuator failure	4.76E+03	2
	Human operator failure	2.47E-01	3
	Mixed traffic interaction failure	2.00E-01	4

VII. Conclusion and Future Work

This work presents a case study of teleoperation in underground mines by modeling the teleoperation system and constructing a DFT that captures conditional dependencies, order-dependent failures, and redundancies. The DFT is rigorously analyzed using probabilistic model checking across a broad set of system-wide, degradation, and importance metrics. Evaluation results show that the system is highly sensitive to ADAS and actuator failures. Nevertheless, once the system enters a degraded state, it can continue operating for a considerable period before eventual failure. The importance metrics further reveal that components such as ADAS, actuators, communication, the human operator, and the ROS play pivotal roles in determining overall system reliability.

Although the case study is focused on remote driving in underground mines, the proposed model can be generalized to remote assistance of Level-4 automated vehicles and applied beyond mining to road vehicle teleoperation. Future work will focus on more realistic modeling of conditional dependencies introduced by wireless communication outages, enabling a deeper understanding of their impact on system safety.

References

[1] SAE International Recommended Practice, "Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles," *SAE Standard J3016_202104*, Revised April 2021, Issued January 2014.

[2] S. Hasan, K. Sjöberg, and P. Wallin, "Wireless communication in underground mining teleoperation: A systematic review," *IEEE Access*, vol. 13, pp. 72577–72602, 2025.

[3] F. Tener and J. Lanir, "Driving from a distance: Challenges and guidelines for autonomous vehicle teleoperation interfaces," in *CHI*, pp. 250:1–250:13, ACM, 2022.

[4] ISO, "ISO 12100:2010 Safety of machinery–General principles for design–Risk assessment and risk reduction," Edition 1, 2010.

[5] ISO, "ISO 13849-1:2023 Safety of machinery–Safety-related parts of control systems–Part 1: General principles for design," 2023. Edition 4, 2023.

[6] International Electrotechnical Commission (IEC), "IEC 61508:2010 CMV – Functional safety of electrical/electronic/programmable electronic safety-related systems, Parts 1–7," 2010.

[7] ISO, "ISO 26262:2018 Road vehicles–Functional safety," 2018.

[8] C. A. Ericson and C. Ll, "Fault tree analysis," in *System Safety Conference, Orlando, Florida*, vol. 1, pp. 1–9, 1999.

[9] E. Ruijters and M. Stoelinga, "Fault tree analysis: A survey of the state-of-the-art in modeling, analysis and tools," *Comput. Sci. Rev.*, vol. 15, pp. 29–62, 2015.

[10] J. B. Dugan, S. J. Bavuso, and M. A. Boyd, "Fault trees and sequence dependencies," in *Annual Proceedings on Reliability and Maintainability Symposium*, pp. 286–293, IEEE, 1990.

[11] M. Ghadhab, S. Junges, J.-P. Katoen, M. Kuntz, and M. Volk, "Safety analysis for vehicle guidance systems with dynamic fault trees," *Reliab. Eng. Syst. Saf.*, vol. 186, pp. 37–50, 2019.

[12] K. D. Rao, V. Gopika, V. V. S. S. Rao, H. S. Kushwaha, A. K. Verma, and A. Srividya, "Dynamic fault tree analysis using monte carlo simulation in probabilistic safety assessment," *Reliab. Eng. Syst. Saf.*, vol. 94, no. 4, pp. 872–883, 2009.

[13] C. Wang, L. Wang, H. Chen, Y. Yang, and Y. Li, "Fault diagnosis of train network control management system based on dynamic fault tree and bayesian network," *IEEE Access*, vol. 9, pp. 2618–2632, 2021.

[14] S. Hoffmann and F. Diermeyer, "Systems-theoretic safety assessment of teleoperated road vehicles," in *VEHITS*, pp. 446–456, SCITEPRESS, 2021.

[15] A. V. Kamaraj, J. E. Domeyer, and J. D. Lee, "Hazard analysis of action loops for automated vehicle remote operation," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 65, pp. 732–736, SAGE Publications Sage CA: Los Angeles, CA, 2021.

[16] M. Volk, F. Sher, J.-P. Katoen, and M. Stoelinga, "SAFEST: Fault tree analysis via probabilistic model checking," in *2024 Annual Reliability and Maintainability Symposium (RAMS)*, pp. 1–7, IEEE, 2024.

[17] M. Volk, S. Junges, and J.-P. Katoen, "Fast dynamic fault tree analysis by model checking techniques," *IEEE Trans. Ind. Informatics*, vol. 14, no. 1, pp. 370–379, 2018.

[18] S. Junges, J.-P. Katoen, M. Stoelinga, and M. Volk, "One net fits all - A unifying semantics of dynamic fault trees using gspns," in *Petri Nets*, vol. 10877 of *Lecture Notes in Computer Science*, pp. 272–293, Springer, 2018.

[19] H. Boudali, P. Crouzen, and M. Stoelinga, "Dynamic fault tree analysis using input/output interactive markov chains," in *DSN*, pp. 708–717, IEEE Computer Society, 2007.

[20] J.-P. Katoen, "The probabilistic model checking landscape," in *LICS*, pp. 31–45, ACM, 2016.

[21] C. Hensel, S. Junges, J.-P. Katoen, T. Quatmann, and M. Volk, "The probabilistic model checker storm," *Int. J. Softw. Tools Technol. Transf.*, vol. 24, no. 4, pp. 589–610, 2022.

[22] W. Vesely, T. Davis, R. Denning, and N. Saltos, "Measures of risk importance and their applications," tech. rep., Battelle Columbus Labs, 1983.

[23] Y. Dutuit and A. Rauzy, "Efficient algorithms to assess component and gate importance in fault tree analysis," *Reliab. Eng. Syst. Saf.*, vol. 72, no. 2, pp. 213–222, 2001.