World Academy of Science, Engineering and Technology
International Journal of Industrial and Systems Engineering
Vol:17, No:2, 2023

Diversity for Safety and Security of Autonomous Vehicles against Accidental and Deliberate Faults

Anil Ranjitbhai Patel, Clement John Shaji, Peter Liggesmeyer

Abstract—Safety and security of Autonomous Vehicles (AVs) is a growing concern, first, due to the increased number of safety-critical functions taken over by automotive embedded systems; second, due to the increased exposure of the software-intensive systems to potential attackers; third, due to dynamic interaction in an uncertain and unknown environment at runtime which results in changed functional and non-functional properties of the system. Frequently occurring environmental uncertainties, random component failures, and compromise security of the AVs might result in hazardous events, sometimes even in an accident, if left undetected. Beyond these technical issues, we argue that the safety and security of AVs against accidental and deliberate faults are poorly understood and rarely implemented. One possible way to overcome this is through a well-known diversity approach. As an effective approach to increase safety and security, diversity has been widely used in the aviation, railway, and aerospace industries. Thus, paper proposes fault-tolerance by diversity model taking into consideration the mitigation of accidental and deliberate faults by application of structure and variant redundancy. The model can be used to design the AVs with various types of diversity in hardware and software-based multi-version system. The paper evaluates the presented approach by employing an example from adaptive cruise control, followed by discussing the case study with initial findings.

Keywords—Autonomous vehicles, diversity, fault-tolerance, adaptive cruise control, safety, security.

I. Introduction

DESPITE tremendous research on AVs over the past decades in the area of sensor technology, artificial intelligence, human-machine interaction, legal issues, and machine perception, full control capabilities of AVs are still far behind due to safety and security challenges. Additionally, AVs are equipped with variant sensors and actuators that are subject to sources of degradation, physical defects, and intended security vulnerabilities, which may lead to accidental faults (random faults) or deliberate faults (with malicious intent). AVs are complex safety-critical embedded systems that operate in an uncertain and dynamic environment, therefore, it is challenging to anticipate all potential faults at design time. Under such circumstances, one way to ensure safety and security of the AVs behavior is to introduce diversity principle during design time to handle the hazardous situation at runtime. In ISO 26262 [7], diversity principle is stated as ”The different solutions satisfying the same requirements with the aim of independence.” In IEC 61508 [6], diversity is described as ”Different types of components are used for the diverse channels of a safety-related system.” Diversity principle is making a Fault-Tolerance (FT) system in the form of redundancy to counter random faults and also makes it difficult for an intruder to exploit known security vulnerabilities.

Anil Ranjitbhai Patel, Clement John Shaji, and Peter Liggesmeyer are with the Chair of Software Engineering: Dependability, Technische Universität Kaiserslautern, Kaiserslautern 67663, Germany (e-mail: patel@cs.uni-kl.de; shaji@rhrk.uni-kl.de; liggesmeyer@cs.uni-kl.de).

FT by design diversity provides redundancy with the specification despite faults occurred or occurring [2]. A potential advantage of design diversity is to increase reliability and availability through support from hardware and software diversity, and architectural features [3]. Diversity is generally used for Instrumentation and Control (I&C) Systems for decreasing Common-Cause Fault (CCF) risks in Aviation, Railway, Chemical Process Plant, and Nuclear Power Plant (NPP) [4]. However, there is a gap in understanding how to assess the diversity for safety and security of AVs considering the following [5]:

- Firstly, identification of strategy for sensor diversity and/or redundancy
- Secondly, identification of relevant diverse properties of redundant sensors, i.e., operating spectra, noise sources, environmental limitations, etc. Radar, Lidar, and Ultrasonic sensors detection range and its angular coverage in different vehicle operational speed, geographic and road conditions, and operational parameters.
- Thirdly, quantitative support for overall sensor diversity, redundancy, and fusion strategy providing acceptably capable detection.
- Finally, traceability of sensor diversity and/or redundancy argument to operational environments and operational modes, including degraded vehicle modes.

Therefore, this paper represents research results for safety and security of AVs with design diversity methodology based on a use case study. This article aims to answer research questions: (1) How to define the frameworks to form the application of diversity?, (2) How to choose the types and version of redundancy to assure the safety?, (3) How to design diversity-based complex system like AVs?, and (4) How diversity influences on security-related threats? Another research aspect is to provide an effective risk mitigation strategy through different architectures by introducing design by diversity in AVs. The paper is structured as follows: Section II compares the presented approach to the state-of-the-art approaches to obtain the overall structure of the methodology. Section III describes the details of our approach, classification, and analysis of application of FT by diversity. Section IV is dedicated to the use case study and implementation of diverse architecture for safety and security. Concluding remarks and

future works are presented in Section V.

II. Related Work

Diversity has been devised by many researchers since the late 1970s, to cope with the accidental and deliberate faults. Situations involving accidental and/or deliberate faults are caused by an adverse physical event, software error (bug), or malicious human action [12]. On the one hand, Randell [9] has introduced "Recovery Blocks" as diverse alternative software solutions for FT to limit the impact of unintentionally introduced bug (accidental fault) in the execution environment. On the other hand, Avizienis [8] introduced "N-version Programming" from the same initial specification using a voting mechanism to reduce the number of faults. Later on, both of these techniques were included in the same global framework [18] and used to design multiple versions of firewalls [14]. Furthermore, to understand the software diversity landscape, Baudry and Monperrus [11] have surveyed the classical work about design and data diversity for FT, as well as cybersecurity literature that investigates randomization at different system levels. In their study, it was found that diversity in a natural complex system will be an essential step for software diversification. Similarly, a fundamental review of diversity for safety and security of embedded and cyber-physical systems (CPS) was also conducted by Kharchenko [16] to highlight the applicability and limitation of diversity. To increase safety, security, and survivability within complex CPS, Brezhnev [17] has introduced cyber diversity to build FT and intrusion-tolerant smart substations connected to NPP. Garcia et al. [15] conducted a study on the impact of diverse operating systems to build an intrusion-tolerant system. Apart from this, Hiltunen et al. [10] propose the "Cactus Mechanism" to tolerate unpredictable events such as bugs or malicious attacks on the system. It also discusses how to switch between different components and different security and FT solutions to achieve higher resilience and availability. Looking at all of these studies, we realize that the implementation of diversity in AVs has a very huge potential to discover the phenomena of multiple effects. To assure safety and security and to design FT and intrusion-tolerant architecture within the system, diversity through D3 (Defense-in-Depth and Diversity) principle is used [20].

In the automotive industry, it is not common to install dormant spare units for cost reasons. Therefore, this huge industry domain has not experienced implementing the application of diversity for safety-critical embedded systems. On the one hand, diversification of FT architecture has the potential to improve vehicle safety and reliability without excessive redundancy [19]. On the other hand, there are many open questions regarding what type (types) of diversity should be used, how to take into account dependencies of types of diversity, and to search regularized set of decisions, and how to assess diversity which should be analyzed before implementing diversity application [20]. However, as mentioned in the previous section, the standard ISO 26262 [7] and IEC 61508 [6] advocate the application of diversity in on-board vehicle system but does not contain any requirements and/or recommendation concerning actual diversity assessment [13].

III. Methodology

One of the most fascinating trends in automotive domain is to make AVs safe against internal component failures and external threats. Therefore, AVs should be modeled as a socio-technical system to keep it safe and reliable in case of an accidental fault within the system or a deliberate attack on the system. Faults can be classified into three different classes proposed by Avizienis et al. in [1]: Physical faults, Design faults, and Interaction faults. Physical faults are hardware faults caused by random natural phenomena, for example, sensor deviation. Design faults are introduced during the design phase of the system, for example, incorrect design algorithms whereas, interaction faults are caused by incorrect interaction between the system and its environment. It can be further divided into the accidental faults and deliberate faults which can occur without or with malicious intent. In this paper, as an effective approach to improve safety and security of AVs, FT by diversity is introduced. The question of how to recognize and mitigate such faults at runtime becomes important when an accidental (without malicious intent) or deliberate faults (with malicious intent) have consequences that lead to injuries, or loss of human lives. One possibility to guarantee safety and security in AVs is to deploy diversity techniques that aim to tolerate some kind of design faults. It can be implemented at different levels [12]:

- At the level of users or operators
- At the application of software level
- At the hardware or operating system level
- At the execution level
- At the human-machine interface.

In this paper, as per the standard ISO 26262 [7] and IEC 61508 [6], we have implemented diversity at hardware (HW) and software (SW) level to counter accidental and deliberate fault as FT mechanism. Failure free and failure independence are the critical assumptions when designing a safety-critical system like AVs using diversity principle. In this approach, we have ensured that both these assumptions are satisfied among diverse solutions provided or build within the system. AVs are equipped with a variety of sensors to sense the surrounding environment and send the information to control algorithm to make tactical decisions (e.g., steering, acceleration, deceleration, and braking). The safety-critical control algorithm allows achieving full autonomy which enables AVs to automatic steering, braking, and maintained speed regulation as per the physical road conditions. As human lives are at stake, in the context of AVs design, it is of utmost importance to make AVs safe and secure against any face of fault or attack. A general overview of the FT by diversity is presented in Fig. 1 which has four segments: Accidental fault in HW and SW, and deliberate fault in HW and SW. In this paper, we discuss the diversity-by-design to counter faults in AVs in a holistic manner, by utilizing D3-principle. In the next subsection, important aspects of design diversity for accidental and deliberate faults are reviewed at HW and SW levels.

A. Fault-Tolerance by Diversity Approach

FT by diversity means that a system fulfills a specified function, even if a system is facing failure due to malfunctioning sensors or attacks on the system. Implementing diversity along with FT mechanism is commonly seen as plain redundancy, but diversity is a form of redundancy with each unit being different from the other unit in one way or the other. For example, in a car, if a brake-by-wire system fails, the hydraulic brakes help out. An example of AVs to collect camera-based pedestrian recognition can be performed by number of methods, i.e, feature selection, pre-segmentation method, and stereo information are examples of a diverse software program. In diverse operating conditions, where dynamic environmental behavior can impact the system´s behavior, diversity can be implemented either directly by deploying an array of cameras or indirectly by measuring angular resolution, degree of vignetting, and contrast at different points in time. In order to cope with systematic unknown random failures and residual uncertainties, safety principles such as redundancy and diversity are applied to design [21]. Design criteria to obtain an optimal FT against accidental and deliberate faults within the system are as follows:

- Maximum diversity at each level of a subsystem against accidental and deliberate faults.
- Determining a set of versions that replicate the same tasks with existing diversity.
- Restraining from using identical copies to avoid identical errors.
- Cost versus performance of the same function for required degree of FT.
- Consistency and attributes of diversity (e.g. HW, SW, data, process etc.) in all the diverse algorithms.

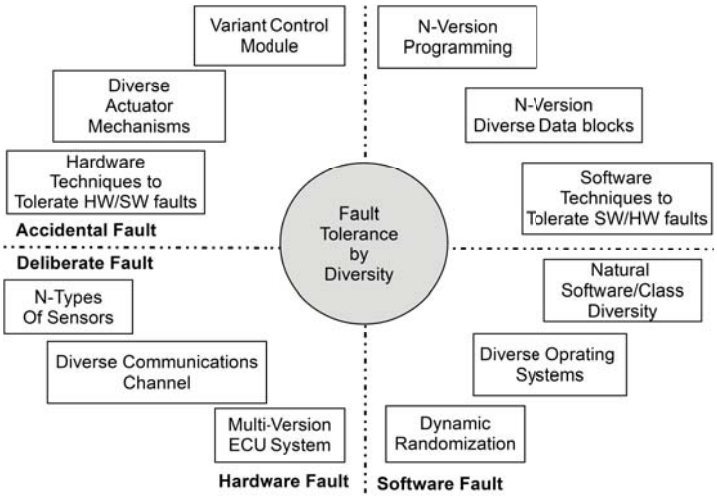

Fig. 1 Fault Tolerance in AVs for Accidental and Deliberate Fault

The proposed FT by diversity based on different solutions is shown in Fig. 1. It comprises approaches that are suitable to detect and mitigate the elusive faults at runtime through design diversity. The potential advantage of design diversity is in their multiple diverse architectures to completely avoid physical fault. Multiple hardware components and software programs that can support functional diversity are an extension of existing FT mechanism. Error detection and system recovery can be handled by combining such approaches as N-version programming, randomization, and diverse actuator mechanism. The proposed architecture described here have provided a proof-of-concept and shown the feasibility of design diversity in AVs using different solution as a complement to FT. Besides design diversity, depending on the redundancy degree, different combinations of synchronizer, voting, and randomization mechanism are considered providing the final output.

B. Accidental Faults

Accidental faults occur due to unknown faults (bug, vulnerabilities, undiscovered sensor failure) leading to an unacceptable system behavior [1]. It is usually considered that a bad decision (unintended) taken by the developer(s) or operator(s) was a fault. However, in AVs, we are considering that such faults can only occur if HW or SW fails at runtime accidentally without malicious objectives. HW and SW diversity and their degree of redundancy enable the system to become FT and may decrease CCF.

1) Diversity at Hardware: In case of the HW fault occur accidentally in AVs, the hardware techniques, diverse actuator mechanism, and variant control module can avoid and tolerate the transient or permanent hardware faults, and single event upset. The architecture shown in Fig. 2 contains a diverse microcontroller and an error correction code in the event of SW fault along with various actuator mechanisms with synchronizer and voter mechanism.

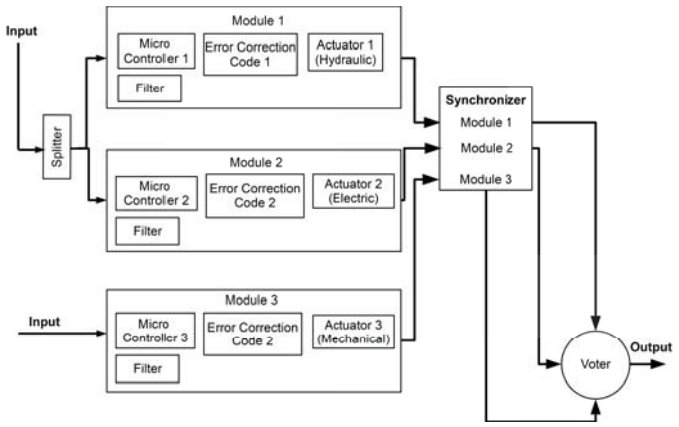

Fig. 2 Structure of diversity at HW for accidental fault

– Hardware Techniques to Tolerate HW/SW Faults: To mitigate the fault, signal integrity, power supply decoupling, component derating, protection of Input/Output circuits, and diverse circuit board designs from different platform suppliers can be applied.
– Diverse Actuator Mechanisms: Diverse actuators combined with different purposes, functionality with different specifications, diverse control logic, and actuator means can be applied for the same functionality (e.g., hydraulic, electric, and mechanical actuator).

– Variant Control Module: A variant microcontroller for each module with an in-built error correction code, different sequencing of operations, and bus architecture can be utilized.

2) Diversity at Software: In case of an accidental SW fault occurrence in AVs, N-Version Programming, N-version diverse data block, and SW techniques to tolerate HW/SW faults can be applied. The architecture shown in Fig. 3 contains N-Version programming, a comparator to compute each value coming from the different programs, diverse data blocks to handle unstable programming algorithms, and a voter mechanism to trigger the correct computed value.

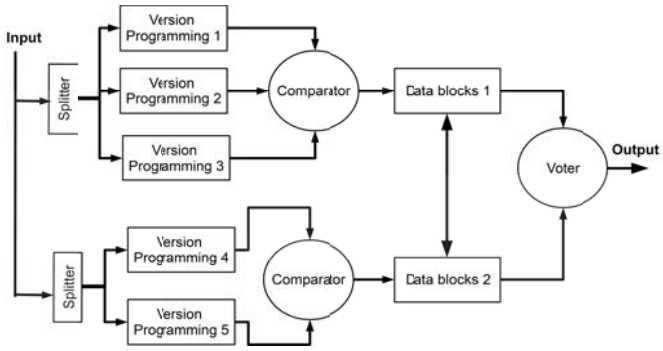

Fig. 3 Structure of diversity at SW for accidental fault

– N-Version Programming: It deploys N-version programming using different tools and compilers build by different teams with the same requirements.
– N-Version Diverse Data Blocks: In order to cope with the unstable algorithm of N-Version programming, diverse data retry blocks should be deployed to re-express the data. For example, image size of a camera can be resized based on the computation time.
– Software Techniques to Tolerate HW/SW Faults: To avoid SW fault accidentally, different types of watchdogs, memory protection unit, and HW schedulers can be applied to assure data integrity at runtime.

C. Deliberate Faults

Deliberate faults occur due to malicious intent to harm the system which leads to an unacceptable system behavior [1]. It is usually considered that intentionally bad decisions are made by persons with an objective to harm the system during runtime. In AVs, we are considering that such deliberate faults can only occur in HW or SW at runtime with malicious objectives. Such faults can be tolerated by following these steps: identify the objectives and requirements of the system, estimate the capabilities of the adversaries, design control feature to compensate the security threats, and assess the sensitivity of the system [22]. The malicious attack on AVs can exploit its enhanced connectivity through jamming, signal spoofing, physical access, code modification, doe injection, packet sniffing, and remote access [24]. Therefore, we have proposed randomization and comparator mechanism in our architecture to counter the different threats at runtime using multiple copies of the same system to create trouble for the attacker to defeat the system defense.

1) Diversity at Hardware: In case of a deliberate attack on the HW of AVs, variant types of sensors, diverse communication channels, and multi-version of Electronic Control Units (ECU) can avoid such malicious intention at runtime. The architecture shown in Fig. 4 contains a diverse set of sensors used for the same functionality, variant communication channel, and multi-version ECUs with voter mechanism.

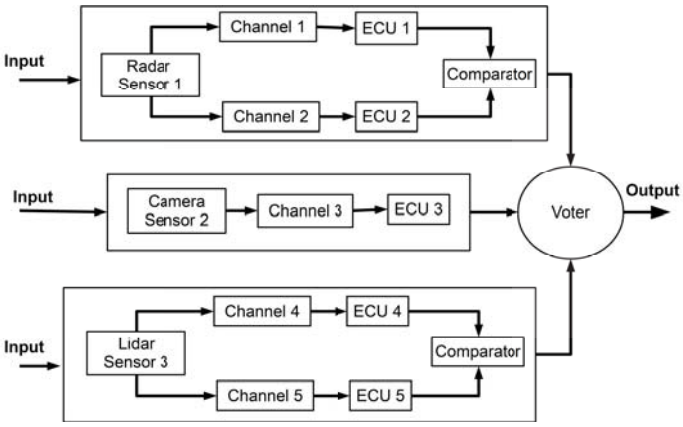

Fig. 4 Structure of diversity at HW for deliberate fault

– N-Types of Sensors: Similar sensors sensing the same parameters as a redundant set can be applied; for example, object detection can be done by radar, camera, etc.
– Diverse Communication channel: Diverse data-flow and logic-processing channels can be applied for different communication channels used in an on-board aviation system.
– Mutli-Version ECU system: Variant timing and sequence of operation with multiple ECU can minimize the complete blackout of the functionality in the event of an attack.

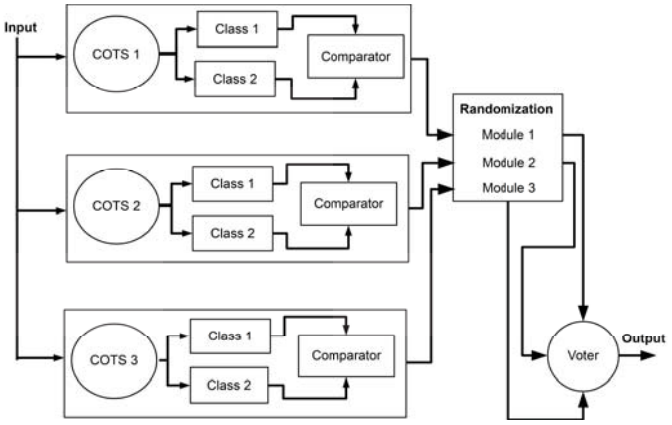

Fig. 5 Structure of diversity at SW for deliberate fault

2) Diversity at Software: Variant types of software solutions that provide similar functionalities, variant forms of natural software diversity, and diverse operating systems along with

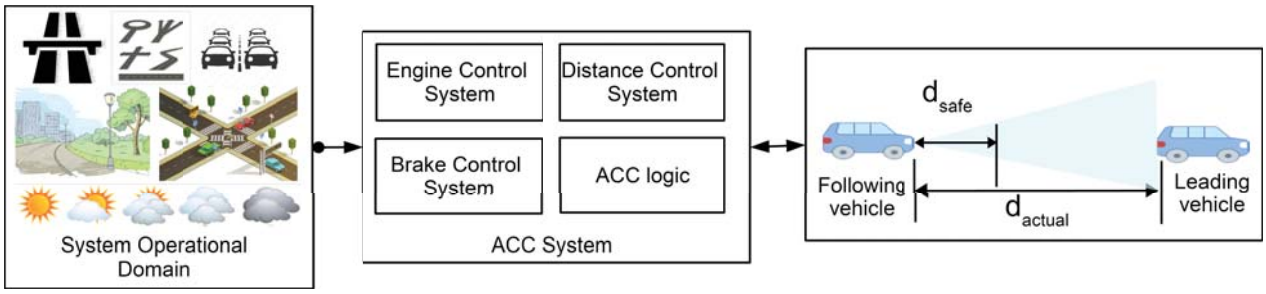

Fig. 6 Adaptive Cruise Control System

dynamic randomization have used to counter intentional attack on AVs software system. The architecture shown in Fig. 5 contains natural class diversity with variant Commercial-off-the-shelf (COTS) operating systems combined with dynamic randomization at runtime.

- Natural Software/Class Diversity: A natural software or class that can be modified through several parameters can have different outputs. For example, to calculate the object to distance with the same class program with different parameters can have different calculation methods.
- Diverse Operating Systems: Three different operating systems with variant algorithms and diverse software packages combined with different languages can be applied to increase the diversity in the network of systems.
- Dynamic Randomization: Through dynamic randomization, one can create randomization between the different modules for the same functionality to make it difficult for an attacker. This technique changes the instruction set so that unauthorized code will not run and decrease chances of attack at runtime.

IV. Simulation Study: Adaptive Cruise Control

In order to study some of the discussed design diversity on AVs, we have implemented the Adaptive Cruise Control (ACC) system functionality in Carla open-source simulator [24]. The functionality of ACC consists of engine control system, distance control system, brake control system, and ACC logic controller as shown in Fig. 6. We have used Carla which provided vehicle physics control for the longitudinal movement of the vehicles and the sensor suits that are modified according to our implementation strategy to test the vehicle system stability under variant traffic scenarios. We present simulation results of FT by diversity on AVs with different HW and SW facing accidental and deliberate faults at runtime.

A. Case Study I: Accidental Hardware Fault

In this case study, we have implemented structural diversity at HW as shown in Fig. 2, in which the diversity is applied to the distance control system to detect an object in the longitudinal direction to maintain the speed and safe distance to the leading vehicle. The behavior of this architecture was investigated with the exposure of random fault (omission) that occurred in the radar sensor during acceleration operation as shown in Fig. 7, which illustrates the tolerance of the system for random fault due to variant module version and error correction code with a voting mechanism. Module 1 and module 2 are connected with the radar sensor values and module 3 is connected with the lidar sensor values. This fault is simulated by preventing the sensors from receiving signals for a few seconds and then resuming its operation. Both the leading and following vehicles are driving at the same speed while maintaining a safe distance from each other even if there is a failure in one of the sensors.

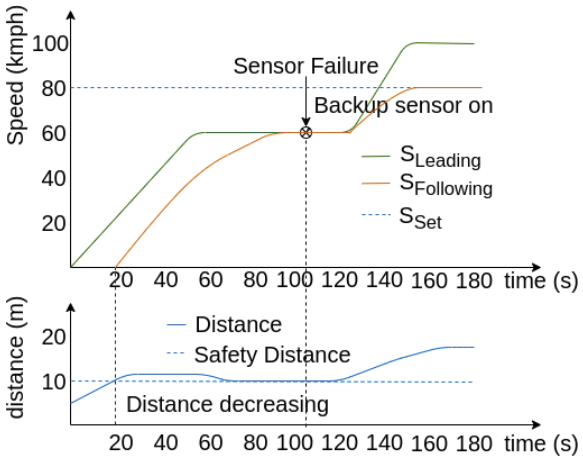

Fig. 7 Accidental Hardware Fault

B. Case Study II: Accidental Software Fault

In this case study, we have implemented a structural diversity at SW to counter accidental fault as shown in Fig. 3, in which the diversity is applied to the engine control system in order to acquire cruising speed of the vehicle during acceleration operation. The engine RPM, throttle pedal position, and gear status are used to calculate the desired engine torque. The behavior of this architecture was investigated with the exposure of random fault (wrong value) that occurred in the engine torque calculator by SW during acceleration operation shown in Fig. 8, which illustrates the tolerance of the system for random fault due to N-Version programming and comparator mechanism to overcome such fault at runtime. The variant data blocks also helped the structure to cross-check the desired engine torque value in present scenario.

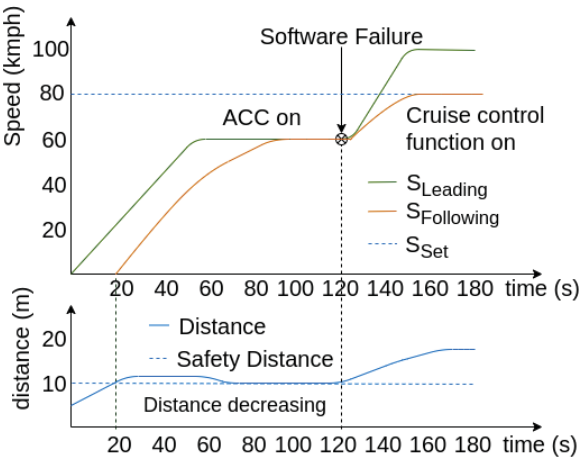

Fig. 8 Accidental Software Fault

C. Case Study III: Deliberate Hardware Fault

In this case study, to counter deliberate fault in HW, we have implemented structural diversity at HW as shown in Fig. 4, in which the diversity is applied to the ACC control module to calculate the throttle pedal and brake pedal angle for acceleration and braking operation. Variant ECUs were implemented through different feedback control, reverse dynamics, feed-forward control to compute the pedal angle using an engine and torque map. Multiple sensors were also implemented to feed the distance to an object into the control module. For simulating the deliberate faults, we have implemented a scenario where an attacker jams the sensor values intentionally to fail the vehicle operation. The behavior of this architecture was investigated with the exposure of deliberate fault (Jamming) that occurred in the multiple sensors at runtime shown in Fig. 9. After all the three safety-critical sensors were compromised, vehicle was came to an halt on hard-shoulder and stop its operation. The higher tolerance to attacks showed by diversity may be justified by its characteristics to operate in such a dynamic environment under attack.

Fig. 9 Deliberate Hardware Fault

D. Case Study IV: Deliberate Software Fault

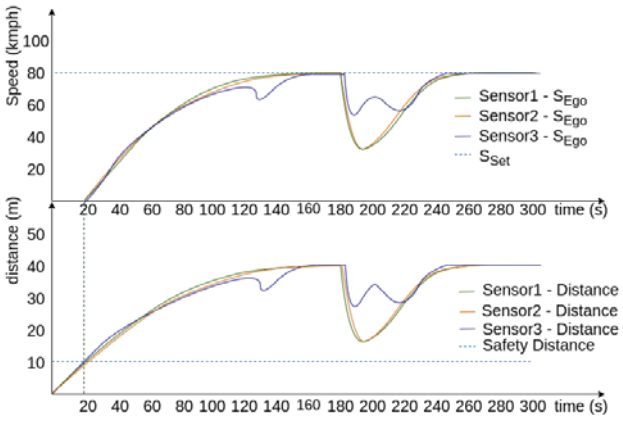

Fig. 10 Deliberate SW fault under attack

In this case study, we have implemented a structural diversity at SW to cope with the deliberate fault as shown in Fig. 5, in which the diversity is applied to brake control system to calculate the required braking force in the event of sudden deceleration or sudden braking from the leading vehicle. Variant types of SW solutions have been integrated to calculate the braking force and brake pedal angle to counter the tampering attack (code modification) on the system. As shown in Fig. 10, one of the speed sensors erroneously measures different speed than the actual speed. It can also crash the ACC control logic if such speed values parse to the ACC logic, which leads to hazardous situations. Due to the dynamic randomization, robust comparator, and voting mechanism, we can prevent such attacks on the AVs at runtime as shown in Fig. 11. Randomization can also help to prevent the crash of the ACC control logic because of the unique executions for the very same program. Dynamic randomization between different operating systems and class diversity can also mitigate the large memory error and is also considered as one of the strongest obfuscation mechanisms [25].

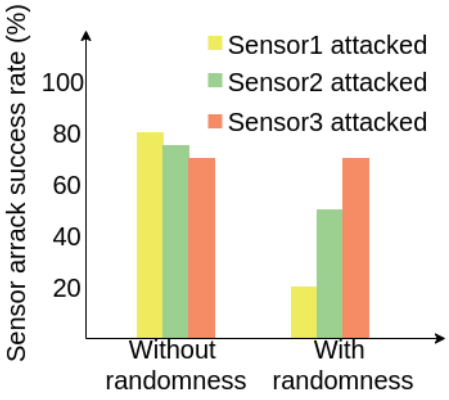

Fig. 11 Attacks success rate with and without randomness in sensor selection

V. Conclusion

We presented a framework to introduce diversity in AVs design and development to make a FT system against

accidental and deliberate fault. HW and SW program-controlled redundancy not only increase the reliability and safety but also increase security which is very important when it comes to AVs. In the automotive industry, it is not common to install dormant spare units for cost reasons. Therefore, this huge industry domain has not experienced implementing application of diversity for safety-critical embedded systems. However, the implementation of D3-principle with calculated actual diversity level must be assessed using a qualitative and quantitative way by AVs developer and regulator bodies to formulate the framework. We argued the issue by modeling an ACC in a virtual environment to validate our proposed approach by viewing from the perspective of a socio-technical system.

A holistic and systematically introduced diversity in AVs for safety and security should be considered as an advantage over accidental and deliberate fault at runtime. Parallel, identification of technical challenges to enhance safety and security in AVs were also discussed. Analysis of ACC use case and different case studies allows that implementation of diversity in AVs requires a high level of system background knowledge and their requirements. The paper also describes the introduction of diversity techniques for safety-critical algorithms and how to prevent AVs from cyber-attack at runtime, which is easy to scale, modify, with compliance with the requirements of the standard. Future research could be related to the implementation of diversity for complex CPS and adaptation of variant types of diversity at different levels of safety-critical application domains.

ACKNOWLEDGMENT

This work has been funded by the Ministry of Education, Science, Continuing Education and Culture of Rheinland-Palatinate (Ministerium f¨ur Wissenschaft, Weiterbildung und Kultur Rheinland-Pfalz) through the Virtual Engineering of Smart Embedded System (ViSE) Project (15412- Tgb.Nr.:3172/18).

REFERENCES

[1] Avizienis, Algirdas, J-C. Laprie, Brian Randell, and Carl Landwehr. ”Basic concepts and taxonomy of dependable and secure computing.” IEEE transactions on dependable and secure computing 1, no. 1 (2004): 11-33.

[2] Avizienis, Algirdas, and J-C. Laprie. ”Dependable computing: From concepts to design diversity.” Proceedings of the IEEE 74, no. 5 (1986): 629-638.

[3] Avizienis, Algirdas, and John PJ Kelly. ”Fault tolerance by design diversity: Concepts and experiments.” Computer 8 (1984): 67-80.

[4] Wood, Richard Thomas, Randy Belles, Mustafa Sacit Cetiner, David Eugene Holcomb, Kofi Korsah, Andy Loebl, Gary T. Mays et al. Diversity strategies for nuclear power plant instrumentation and control systems. No. ORNL/TM–2009/302. Oak Ridge National Laboratory (United States). Funding organisation: ORNL work for others (United States), 2010.

[5] Koopman, Phil. ”An Overview of Draft UL 4600: Standard for Safety for the Evaluation of Autonomous Products.” Medium, Jun (2019).

[6] IEC: Functional safety of electrical/electronic/programmable electronic safety-related systems, IEC 61508:2010.

[7] ISO: Road Vehicles – Functional Safety ISO 26262:2018.

[8] Avizienis, Algirdas. ”The N-version approach to fault-tolerant software.” IEEE Transactions on software engineering 12 (1985): 1491-1501.

[9] Randell, Brian. ”Reliable computing systems.” In Operating systems, pp. 282-391. Springer, Berlin, Heidelberg, 1978.

[10] Hiltunen, Matti A., Richard D. Schlichting, Carlos A. Ugarte, and Gary T. Wong. ”Survivability through customization and adaptability: The cactus approach.” In Proceedings DARPA Information Survivability Conference and Exposition. DISCEX’00, vol. 1, pp. 294-307. IEEE, 2000.

[11] Baudry, Benoit, and Martin Monperrus. ”The multiple facets of software diversity: Recent developments in year 2000 and beyond.” ACM Computing Surveys (CSUR) 48, no. 1 (2015): 1-26.

[12] Deswarte, Yves, Karama Kanoun, and J-C. Laprie. ”Diversity against accidental and deliberate faults.” In Proceedings Computer Security, Dependability, and Assurance: From Needs to Solutions (Cat. No. 98EX358), pp. 171-181. IEEE, 1998.

[13] Kharchenko, V. ”Diversity for Safety of Systems and Software in Context of the Standard ISO/IEC26262.” In 13th Workshop on Automotive on Software and Systems. 2015.

[14] Liu, Alex X., and Mohamed G. Gouda. ”Diverse firewall design.” IEEE Transactions on Parallel and Distributed Systems 19, no. 9 (2008): 1237-1251.

[15] Garcia, Miguel, Alysson Bessani, Ilir Gashi, Nuno Neves, and Rafael Obelheiro. ”Analysis of operating system diversity for intrusion tolerance.” Software: Practice and Experience 44, no. 6 (2014): 735-770.

[16] Kharchenko, Vyacheslav. ”Diversity for safety and security of embedded and cyber physical systems: Fundamentals review and industrial cases.” In 2016 15th Biennial Baltic Electronics Conference (BEC), pp. 17-26. IEEE, 2016.

[17] Brezhnev, E., V. Kharchenko, A. Boyarchuk, and J. Vain. ”Cyber diversity for security of digital substations under uncertainties: assurance and assessment.” In Proceedings of the 19th Internation Conference on Conference on Circuits, Systems, Communications and Computers, CSCC2015. 2015.

[18] Avizienis, Algirdas. ”The methodology of n-version programming.” Software fault tolerance 3 (1995): 23-46.

[19] Hayama, Ryouhei, Masayasu Higashi, Sadahiro Kawahara, Shirou Nakano, and Hiromitsu Kumamoto. ”Fault-tolerant automobile steering based on diversity of steer-by-wire, braking and acceleration.” Reliability Engineering and System Safety 95, no. 1 (2010): 10-17.

[20] Yastrebenetsky, Michael, ed. Nuclear power plant instrumentation and control systems for safety and security. IGI Global, 2014.

[21] Kumamoto, Hiromitsu. Satisfying safety goals by probabilistic risk assessment. Springer Science and Business Media, 2007.

[22] Chattopadhyay, Anupam, Kwok-Yan Lam, and Yaswanth Tavva. ”Autonomous vehicle: Security by design.” IEEE Transactions on Intelligent Transportation Systems (2020).

[23] Thing, Vrizlynn LL, and Jiaxi Wu. ”Autonomous vehicle security: A taxonomy of attacks and defences.” In 2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), pp. 164-170. IEEE, 2016.

[24] Dosovitskiy, Alexey, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. ”CARLA: An open urban driving simulator.” arXiv preprint arXiv:1711.03938 (2017).

[25] Mavrogiannopoulos, Nikos, Nessim Kisserli, and Bart Preneel. ”A taxonomy of self-modifying code for obfuscation.” Computers and Security 30, no. 8 (2011): 679-691.