Analysis of Autonomous Driving Software to Low-Level Sensor Cyber Attacks

Andrew Roberts*, Mohsen Malayjerdi†, Mauro Bellone*, Raivo Sell†,
Olaf Maennel‡, Mohammad Hamad§, Sebastian Steinhorst§
* FinEst Centre for Smart Cities, Tallinn University of Technology
† Department of Mechanical and Industrial Engineering, Tallinn University of Technology
‡School of Computer and Mathematical Sciences, The University of Adelaide
§Department of Computer Engineering, Technical University of Munich

***Abstract*—Autonomous Vehicle (AV) architectures fuse legacy, electromechanical components with advanced sensor technology and digital controllers, governed by software. An open challenge for the design of AVs are cyber threats such as Electromagnetic Injection (EMI) attacks, to the low-level layer, comprising electromechanical components, which can propagate through to the higher-level, intelligent control, affecting decision-making and the safety of the vehicle. This study analyses the robustness of the design of the software stack of a real-world AV to attacks on the low-level actuation using the example of an EMI attack on the steering angle sensor. To achieve this, we create a hybrid testbed which combines the mathematical model of the low-level sensor with the high-fidelity, intelligent control. We further develop safety and performance metrics measured at the high-level, which we use to generate a detailed view on the safety and system performance of the software. We conduct diverse EMI attacks on the target AV, within 3 diverse critical driving scenarios, consisting of $>$ 1000 simulations. The results indicate a correlation between an increase in attack noise with an increase in safety violations and failures to complete the mission of the AV. Our results highlight the importance for AV software developers of testing under diverse attack and driving scenarios, as each scenario within our experimentation exhibits different behaviour of the system and correlations to differing safety and system performance indicators.**

***Index Terms*—Security, Autonomous Driving**

I. Introduction

Cyber attacks that manipulate input to physical processes in cyber-physical systems present a fundamental challenge to secure system design [1]. Within the domain of automotive systems, transformation of legacy, analog architectures to digitally connected and autonomous driving (AD) technologies present new challenges. Legacy, analog automotive systems were designed based on a principle of contained, isolated system boundaries, restricting the flow of data within an analog system and sub-system [2]. The AD system architecture transforms this design, requiring the lower-level, analog control of actuation processes (steering control, braking, acceleration, etc.) to be open and connected to digital controllers so their process signals can be translated to digital input for the higher-level decision control [3].

There have been numerous real-world examples of semi-autonomous control architectures enacting unsafe decisions from erroneous sensing data from low-level actuation sensors [4] [5]. The 2018 SmartLynx Airline incident demonstrated that a physical disturbance from a maintenance activity on the horizontal stabilising sensor caused the sensing input to send erroneous data which propagated through to the control systems for flight planning, stabilisation and safety. The control systems initiated multiple concurrent actuation decisions (horizontal stabilisation, acceleration, etc.) which

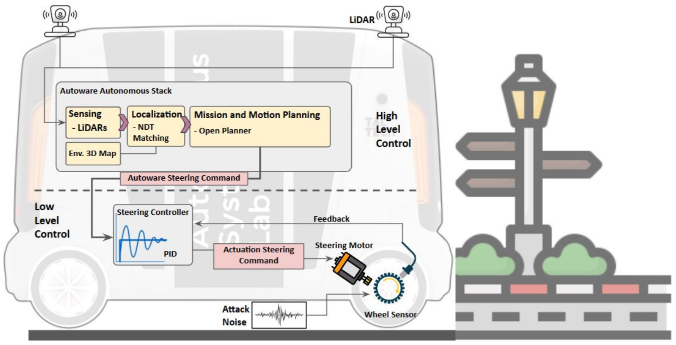

Fig. 1: High-level architecture of Steering Angle Sensor Manipulation within AD System.

affected the safe operation of the flight [5]. Ultimately, manual intervention to override the autonomous control resolved the unsafe state of the flight.

Within the context of cyber threats, numerous studies have proven the vulnerability of microelectronic sensors to electromagnetic interference (EMI) [6], [7], [8], acoustic sensor [9] [10] and data manipulation attacks [11], [12], [13], [14]. Furthermore, the network that exchanges actuation signals, the Control Area Network (CAN) Bus network, has been shown to be inherently vulnerable to a diversity of man-in-the-middle [15], [16] attacks. Yet, there is a lack of practical investigation that extends this analysis of the propagation of malicious data input within an AD system, where physical processes are software-controlled and manual, and human intervention is not available.

Our study is motivated to investigate how cyber attacks to electromechanical components, in our case, a steering-angle sensor, propagate through the AV system, affecting higher-level decision-making. The aim of the study is to analyse the design of a real-world AD vehicular system and assess mechanisms to enhance the design of the architecture of AD systems to be more robust and resilient. To achieve this, we, investigate a real-world AV software ecosystem, analysing the integration between the lower-level control, characterised by electromechanical components, and the high-level control, characterised by digital systems which support algorithmic decision-making. We then investigate how malicious input propagates within this ecosystem. Finally, we determine mechanisms for enhancing secure design.

To guide this research, we focus on the following research questions:

RQ1 How does a manipulation to the electromechanical component propagate through the AD software stack?

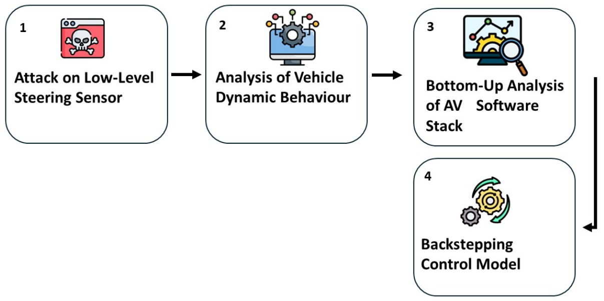
1
Attack on Low-Level Steering Sensor
2
Analysis of Vehicle Dynamic Behaviour
3
Bottom-Up Analysis of AV Software Stack
4
Backstepping Control Model

Fig. 2: Conceptualization of our approach, from attack to backstepping.

RQ2 What dependencies exist between the AD control algorithm and low-level control?

RQ3 Where in the architecture of the autonomous vehicle can defensive mechanisms be placed to defend against control invariants?

Contribution:

- This is the first study, to the best of our knowledge, that conducts an analysis of a production, full-autonomy (Level 4-5), AV system to cyber attacks to the lower-level control.
- Amidst the pressing need to address the design of secure digital critical-infrastructure systems in transportation, we analyse the relationship between low-level, electromechanical systems and high-level control software within AV systems.
- The study provides recommendations for enhancing robustness and resiliency in the design of AV software.

II. Approach Overview

Our approach (see Fig. 2) is to, *first*, implement the sensor interference attack model in our custom high-fidelity AD test-bed environment. The test-bed environment contains the software stack of our real-world vehicle and configurations consistent with the real-world kinematics of the vehicle.

Second, from the results of the experiments, we assess the impact of the cyber attacks utilising defined safety criteria. Furthermore, we conduct a sensitivity analysis of the vehicle's dynamic parameters to identify the behavioural effect of the malicious input and assist in pinpointing critical areas of the AV software which are affected by the attack.

Thirdly, we conduct a bottom-up analysis, to ascertain what happens to the high-level, decision-control, when malicious data is injected into the low-level. The bottom-up analysis details the relationship between inputs and outputs in the AV software stack.

Fourth, the previous analysis enables backstepping at a conceptual level to stabilize elements of the control model that are susceptible to the sensor interference attack.

We justify the use of this approach as it enables us to take an architectural view of the AV software stack. Existing studies use methods that view the problem of manipulation of low-level sensor input either within the context of a PID control [6] [7] issue or solely focus on the autonomous control [12]. We believe that taking an architectural perspective, where the interconnections and dependencies of the system are encountered, enables the designer/s of the AV to gain more insight into the functioning of the system under attack.

III. Autonomous Vehicle Software Architecture

The aim of this section is to describe the entire software stack of the autonomous vehicle platform from a bottom-up perspective. Each layer of the stack will be detailed as to there purpose/task and how they communicate.

A. Low-Level Control

Low-level control is at the base of our software stack, having the task of giving actuators the right commands to generate a desired behaviour. Analog controllers have the function to follow a specific reference signal. It is clear that such signals are measured by transducers and applied to actuators as current or voltage signals to apply a torque to a motor at the low level. The most common and well known analog controller in automotive is the Engine Control Unit (ECU), which regulates injection, speed, and other engine parameters. Brake control modules are also very common and control various aspects of the braking system, such as anti-lock braking (ABS), electronic stability control (ESC), traction control, and brake force distribution. Now, assuming that our goal is to keep any value of cruise speed, a velocity regulator works by receiving a measure of the current speed, comparing it to the reference, and generating a control signal to accelerate or brake accordingly. Low-level controllers typically work on a simple control feedback loop involving some type of linear system model (or a linearized one). The most common, state-of-the-art, and well-established controller in automotive is the PID (Proportional-Integral-Derivative) controller. They are widespread in automotive for their simplicity, robustness, usability, and real-time capabilities. A PID controller continuously calculates an error signal based on the difference between a desired setpoint and the measured process variable and then adjusts the control output accordingly. They use proportional, integral, and derivative actions to regulate a vehicle's actions. The underlying equation is relatively easy, involving three constants, proportional, integral, and derivative constants, typically indicated as K_p, K_i, and K_d, to weigh each action respectively. With reference to Fig. 3, PID controllers are at the base of the "drive controller" and "steering controller" block.

Control theory provides a very stable mathematical theory about analysis and synthesis of the controllers, thus how disturbance might affect the controller is, in principle, well known. This work aims to provide insights on how the behaviour of the controller might affect the decision-making blocks in a real-world, operational AV.

B. Intermediate Layer/Master Controller

The role of the Master Controller is to parse analog input to the digital network of the vehicle.

The Master Controller communicates with the low-level control through the CAN bus. The low-level control section in Fig. 3 shows all the basic components in our system, which are connected to the master controller by three different CAN busses:

- CAN1 is used to connect all safety-critical components, such as brake systems and electric motors.

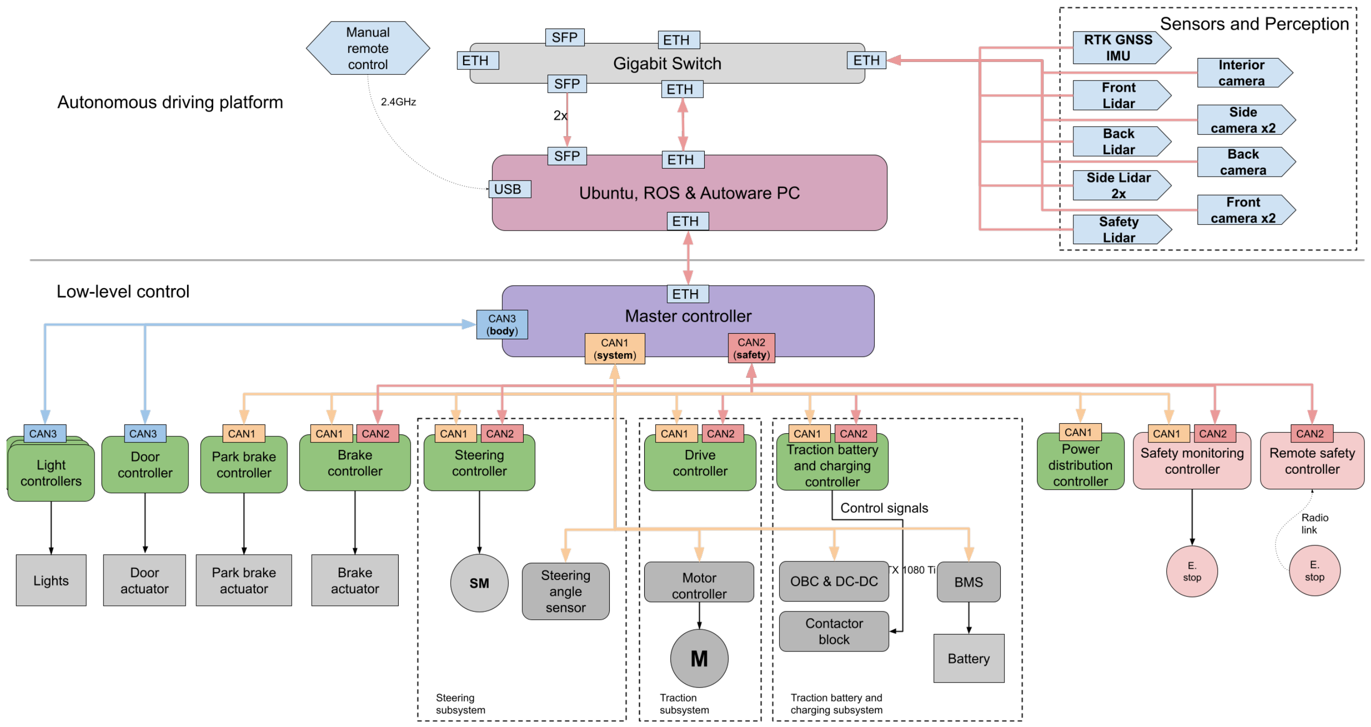

Fig. 3: Autonomous vehicle high-level functional architecture.

- CAN2 is used for redundancy over all the safety-critical components.
- CAN3 is dedicated to low-priority body-related functions such as door automation and lighting.

The master controller receives data from the low level via CAN bus and forwards to the upper-layers via ethernet. Then receives processed signals from the intelligent blocks (the upper-layers) and generate the control commands for the actuators, parsed via CAN bus. Basic data from low-level sensors are processed here and forwarded to the upper layer, this includes speed, acceleration, encoder positioning, voltage and currents.

What this means is that an anomaly in the PID controller, or in its feedback loop, is read here and propagate to the upper layer as limited computation (ARM 32-bit Cortex M7 CPU) can be done at this level. The master controller directly communicates with the upper levels (i.e. Autonomous Driving Software) via ROS topics flowing over the ethernet connected to the Ubuntu-based Autoware PC.

C. Autonomous Driving Platform

The real-world AD platform is fully implemented on a PC featuring an AMD Ryzen Threadripper 1950X (16-core/32-thread) CPU, 2 NVIDIA GeForce GTX 1080 Ti graphic cards, and 64 GB of memory. The driving platform runs the Autoware stack [17], over a ROS environment in the Ubuntu 18.04 OS. Here is where the ROS master is run, handling all topics/subscribers from/to the low level and from the sensing level, but also to each algorithmic component running on concurrent threads and consuming the incoming data flow while providing interpreted information. The intelligent driving software stack includes several modules interpreting information from bottom-up and top-down, referring to high-level perception and low-level control. The main task is to coordinate concurrent processes of sensing, localization, and planning.

The important task of planning and decision-making is assigned to the Open-Planner algorithm [18], which generates a waypoint according to the information coming from the sensors and forward this information to an intelligent control block that generates the actual trajectory in consideration of kinematic/dynamic model of our system. The planning algorithm also generates alternative paths referred to as ”rollouts” which serve as possible drivable trajectories. Each rollout is a kinematically compliant path on which the vehicle could be driving. The change from one rollout to another might normally occur in case of maneuvers (such as overtaking, obstacle avoidance, and intersection crossing), to optimize energy efficiency, or in case of violation of any safety criteria.

This unit is ultimately responsible for the intelligent control in our autonomous vehicle, from localization to planning, perception and control.

Differently, with respect to the low-level controllers (such as PIDs) that are limited to follow a reference point, the intelligent controller decides the reference point to be sent to the low level, and resets it at run time according to the real behaviour of the vehicle. This block can be interpreted as a wider feedback loop acting on a macro scale receiving information from the environment and from the low level. It is clear that, on a micro-scale, the PID controllers are effective, and provide mathematical guarantees of convergence and robustness to noise, while on a macro scale intelligent controllers cannot provide similar guarantees. In case of attack, misleading information, or any source of uncertainty, intelli-

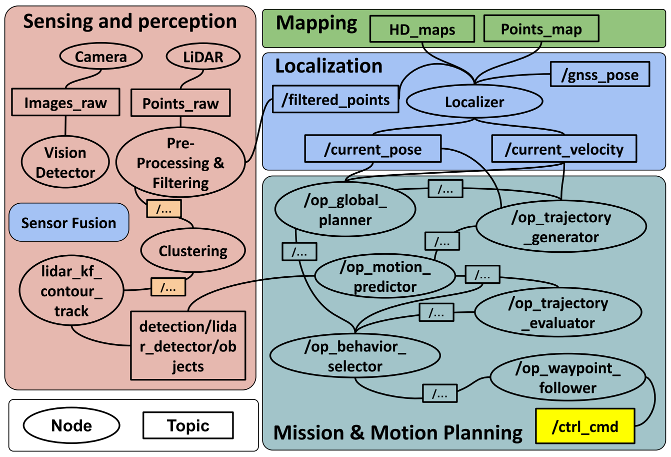

Fig. 4: Intelligent driving software stack structure showing ROS nodes/topics communication between essential elements.

TABLE I: Autonomous Vehicle Sensors.

Sensor	Model
3D lidar (front)	Velodyne Ultra puck VLP 32
3D lidar (rear)	Velodyne VLP-16
2xSide lidar	Robosense Bpearl
Safety lidar	Ouster OS0-90 (Safety)
3xCamera	Flir
GNSS	Trimble BX992
Radar	TI

gent controllers can generate catastrophic decisions, thus our goal is to better understand how uncertainties from the low-level can propagate to the high level generating wrong driving actions.

D. Intelligent driving software stack

A part of the ROS nodes/topics running on the vehicle are represented in Fig. 4. The software stack is mainly composed of the following main components, sensing and perception, mapping, localization and motion planning. Perception modules runs AI-based modules for detection, segmentation and interpretation of traffic scenes. Localization and mission planning receive constant feedback from vehicle and global positioning to generate new control commands. We expect that our attack, though not directly carried out on those modules, would generate errors that propagate from the low level to the localizer and trajectory-generator blocks.

1) Sensors: Sensors are connected to the AD platform running AI-based models for identification, detection and segmentation of objects and environmental information through a Gigabit ethernet switch. Data flow is managed and synchronized directly in the Autoware stack, sending data as ROS topics to concurrent threads (nodes) running inference over the AI-based deployed modules. Sensing information are used for perception-related functionalities such as object detection, segmentation and sensor fusion. Table I provides the list of available sensors.

IV. Adversarial Model

The objective of the attacker is to cause the AV to take unsafe driving actions resulting from manipulation of the steering angle sensor.

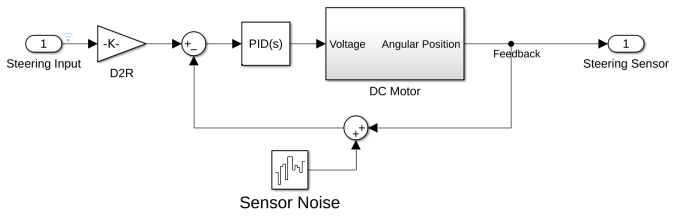

Fig. 5: Steering angle sensor attack.

We assume the attackers cannot directly access the digitised sensor readings. Instead, we assume that the attacker can exploit vulnerabilities in the steering angle sensor using proven techniques such as EMI, to affect the integrity of the sensor data (analog signals on the signal conditioning path before being digitised).

We assume that the attackers can physically place an EMI device near the steering angle sensor and are capable of crafting and transmitting interference to the sensor during the navigation of the AV and thus transform the waveform of the sensor output. We further assume that the attackers do not possess an in-depth understanding of the voltage levels of the steering sensor and therefore focus on injecting incremental noise into the sensor.

We assume that the attackers can observe the operation of the AV and control the attack in terms of initiation and cessation of the attack during varied time periods or within the frames of a critical driving manoeuvre.

V. Attack Model

The attack is conducted in the measurement of the input and output of the PID controller for the steering angle (Fig. 5). The key parameters that affect the success rate of the attack are: *duration*, *noise*, *attack trigger action*.

Within our attack model, attacks are conducted with differing sensitivity levels of the steering angle sensor and durations and are triggered at targeted points of the AV mission. We have chosen a range of sensor attack noise levels (0.01, 0.05, 0.1, 0.2), rather than a specific target point. The study of Pöllny et al. [6], which conducted EMI attacks on a sensor used in an automotive electronic control unit (ECU), indicated that an attacker does not need to set a specific value for the steering angle attack, but simply to find the sufficiently high level of noise that would alter system behaviour to the attacker goal.

Whilst, EMI attacks have been proven successful against microelectronic components in [19] [12] [6] [11] [7] [8] [10], the attacks are applied to the stand-alone sensor hardware and application use-cases such as microphones, temperature sensors, drones. The novelty of the attack model in our study is the implementation of the attack to a fully-autonomous vehicle that integrates low-level actuation with high-level AD decision-making. This enables the ability to assess the affect of the attack on the entire AV software stack. Furthermore, the attack is conducted utilising scenario-specific testing. This is of critical importance, as it is widely understood that the performance of the AD decision-making layer differs based on scenario-specific behaviour [12]. For the AD algorithms may be better optimised for specific driving manoeuvres such as overtaking, or operational driving domains (ODD) such as busy intersections. Our attack is conducted in a simulation test environment, as attacks at the physical, hardware-level are proven, the gap in existing research, is how these inputs propagate within the system and affect the decsion-making within an autonomous system.

VI. Experiment

A. Experimental Setup

To conduct the attack and analyse the subsequent effects, we developed an experimental test environment.This environment consists of a simulation platform that fuses the low-level actuation, simulated in MATLAB, with a high-fidelity simulation of the AV software of our real-world vehicle, simulated in CARLA. The simulation test environment provides an optimal platform as it uses the same mathematical model of the steering actuation sensor and the same software stack as the real-world vehicle. Furthermore, the simulation environment enables attack testing to be conducted in an agile manner, whilst, removing the safety risk factors of testing the AV in the physical road environment.

B. Attack Implementation

We chose to conduct the low-level attack on three diverse scenarios (see Fig. 6): 1) Straight-line, 2) Overtaking manoeuvre and, 3) Left-turning maneuver at intersection. These scenarios were chosen as they are consistent with the most-popularly tested driving scenarios according to the survey of test methods and practices in [20]. As shown in Fig. 6, the high-fidelity simulation view for the three scenarios is conducted. The Straight-Line scenario shows that the EMI attack is initiated after the vehicle traveled 20 meters, with two different attack durations: 10 and 20 meters. For the overtaking manoeuvre, the attack begins during the cut-in process and lasts for 10 meters. Finally, in the intersection scenario, the attack is launched as the vehicle enters the intersection and persists for a distance of 10 meters.

To conduct our experiments, firstly, we conduct the scenario with *no-attack* for 100 runs. This establishes a baseline of the performance of the AV without attacks. From there, each of the attacks with different noise levels and duration are run 100 times. Overall, approx. 1900 simulation runs are recorded, and as the high-fidelity simulation uses GPU and CPU resources, this is a time-consuming process. Figure 7 presents the scenario flow used to integrate the attack into the mission in CARLA. It outlines the sequence of behaviors from the vehicle's initialization and driving towards the goal to executing an attack or stopping based on a distance trigger. The attack is enabled based on a predefined condition. This structured flow allows for precise control over when and how the attack occurs during the scenario, ensuring consistent testing of the AV's response to disturbances.

C. Evaluation Criteria

Table II and III detail the safety and performance criteria applied in our experiments, respectively. As we have diverse scenarios which involve scenarios with ego vehicles, certain criteria is only applicable to their corresponding scenario. In this analysis, mission failure (NotF) and safety violations (SafetyV) are distinct evaluation criteria used to assess the performance and safety of the AV during the scenarios.

Mission failure (NotF) refers to instances where the vehicle was unable to complete the mission. This typically occurs

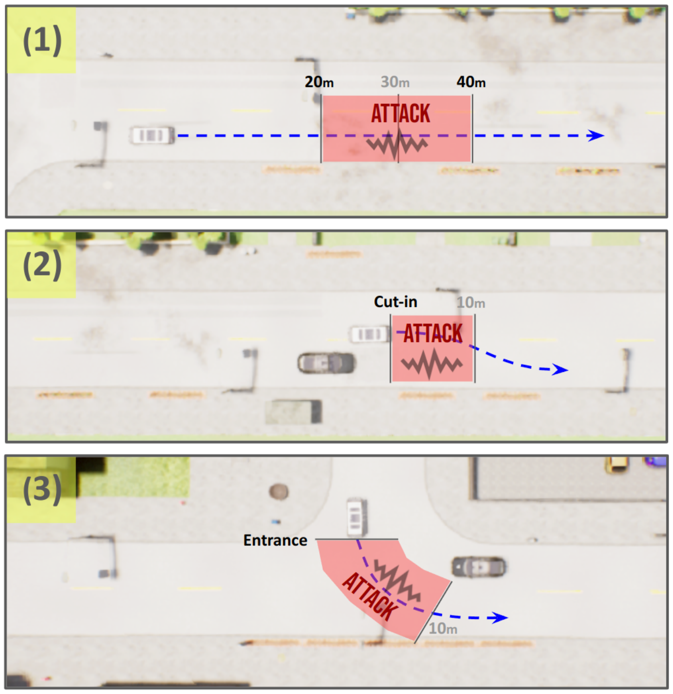

Fig. 6: Game-engine view of three simulated scenarios representing the attack occurrence place during the mission; 1) Straight-line 2) Overtake 3) Intersection.

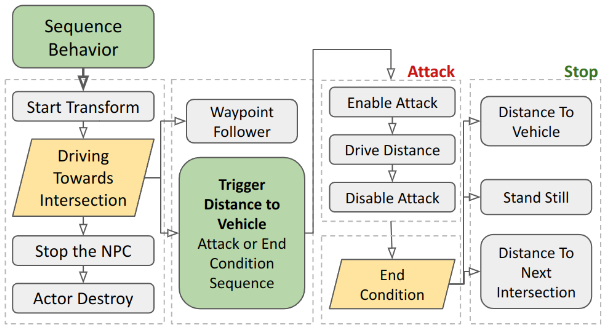

Fig. 7: Flow graph of how each scenario is processed in the simulation platform.

due to critical events that prevent the AV from finishing its task, such as collisions (V_{Col}), localization loss (V_{NDTLs}), or sidewalk incursions (V_{SiIn}). These violations are severe enough to terminate the mission.

Safety violations (SafetyV), on the other hand, refer to any breaches of safety that occur during the mission but do not necessarily prevent the vehicle from completing it. A mission may still be considered successful even if multiple safety violations are recorded. Examples of these include deviation to the center lane (V_{DTL}), sharp braking (V_{BrD}), localization loss (V_{NDTLs}), collisions (V_{Col}), and violations of distance to collision (V_{DTC}VDTC). In these cases, while the AV may exhibit unsafe behaviors or suboptimal performance, it is still able to complete the mission.

Two critical safety metrics are sidewalk incursions (V_{SiIn}) and collisions (V_{Col}), both representing severe safety hazards. A sidewalk incursion indicates where the AV veered off its

TABLE II: Safety Evaluation Criteria.

Safety Condition	**Data Label**	**Description**	**Metric**
Not Finished	*NotF*	Failure to finish the mission	Pass/Fail
Sidewalk Incursion	*SiIn*	AV deviation into pedestrian zone	Pass/Fail
Collision	*Col*	AV collides with NPC	Pass/Fail
Distance-to -Collision	*DTC*	Violation of the safe distance between AV and NPC	AV within 0.5m of other vehicle
Distance-to -Centre Lane	*DTL*	Violation of the safe distance between AV and Centre Lane	AV within 0.4m of centre lane
Break on Driving Lane	*BrD*	AV initiates emergency break on driving lane	Pass/Fail
Localization	*NDTLs*	Localization Loss	NDTerror > 1.0
Violation	*V*	Safety Violation	

TABLE III: Performance Evaluation Criteria.

Performance Criteria	**Data Label**	**Description**	**Metric**
Lane Transition	*RlOut*	AV executes multiple roll-out transition	Pass/Fail
Localization	*NDT*	Localization Performance	AV localization matching
Localization	*NDTer*	Mean localization pose error	Localization error margin
Duration	Dur	Duration in seconds	
Max NDT score	MxNDTSr	Max NDT score during a mission	Smaller = Better
Path Deviation	Dev2Ref	Sum of deviation to the reference path in sampled points	Smaller = Better
Max Lat Deviation	MxLaDev	Max lateral deviation from original path	Smaller = Better

intended path and encroached into pedestrian zones, potentially endangering people on sidewalks. Similarly, a collision signifies an event where the AV collided with a nearby non-player character (NPC) vehicle.

Another key performance indicator is the deviation to the reference path (Dev2Ref), which measures how far the AV strayed from its intended trajectory. It is important to note that Dev2Ref is not the deviation at a single point; rather, it represents the summation of the deviations at several reference points along the planned path to the actual route traveled by the AV. This cumulative nature of the metric results in larger values, especially when the AV frequently deviates from the intended trajectory.

VII. Results

For each of the scenario's, the results, as expressed in Tables. IV, V, and VI demonstrate that increasing level of noise and duration of the EMI attack impact the safety and performance of the AV.

The manipulation of the steering sensor input at higher noise levels affects the feedback loop for the calculation of localisation, which results in the AV experiencing loss and jumps of localisation. The NDT algorithm, used in the localisation algorithm, exhibits weakness in holding the position of the AV during sensor manipulation, which is demonstrated by loss of localisation, in attempting to re-correct, it incurs jumps. The loss and jumps of the localization affect the displacement of the AV. As such, the cost-based algorithm used by the mission and motion planning module recalculates the trajectories and chooses a new roll-out. The choice of a new trajectory for the AV disrupts the flow of critical maneuvers within the scenario, such as the cut-in process of overtaking,

TABLE IV: Summary of the Safety and Performance Evaluation - Straight Line Scenario. The first line is our baseline path, where no attack was applied.

SAFETY								
Length	Noise	NotF	SafetyV	V_{SiIn}	V_{DTL}	V_{NDTLs}	V_{BrD}
-	baseline	0%	0%	0%	0%	0%	0%
10 m	0.01	0%	0%	0%	0%	0%	0%
10 m	0.05	10%	10%	0%	10%	0%	0%
10 m	0.1	12%	12%	0%	6%	6%	0%
10 m	0.2	30%	30%	2%	26%	12%	8%
20 m	0.01	0%	0%	0%	0%	0%	0%
20 m	0.05	34%	34%	2%	30%	8%	2%
20 m	0.1	34%	36%	4%	28%	18%	6%
20 m	0.2	42%	42%	6%	38%	14%	2%

PERFORMANCE					
Length	Noise	Dur	RlOut	MxLaDev	MxNDTSr
-	baseline	57.6s	0	0.1m	11.9
10 m	0.01	59.9s	0	0.2m	11.9
10 m	0.05	61.5s	0.16	1.6m	12.0
10 m	0.1	65.5s	0.3	1.5m	12.5
10 m	0.2	71.8s	1.18	8.3m	25.5
20 m	0.01	70.2s	0	0.3m	14.2
20 m	0.05	75.6s	0.94	1.7m	25.5
20 m	0.1	82.6s	1.36	8.2m	46.9
20 m	0.2	85.6s	1.64	8.2m	35.0

smoothing of trajectory in keeping straight-line and turning at the intersection.

A. Scenario 1: Straight-Line

Within the Straight-Line Scenario Safety Results (Table. IV), safety violations begin to occur when 0.05 noise is introduced into the sensor input, marking the threshold where the AV system starts to struggle with maintaining safety. At this noise level, a 10% safety violation rate provided by lateral deviation violations was observed. As the noise level and attack duration increase, the AV experiences a progressive degradation in performance, culminating in the highest noise level (0.2) and the longest attack duration (20 meters), which results in a 42% safety violation rate and 38% lateral deviation violation.

A key characteristic of the AV's behavior in this scenario is the Deviation-to-Centre-Lane. The noise is injected into the steering sensor, and abrupt changes in the steering actuation cause the vehicle's control system to oscillate between making corrections and following the desired path. Autoware's motion planner attempts to rectify the vehicle's course, but the corrections are often sub-optimal, resulting in the AV veering to a dangerous proximity to the center line. This behavior indicates a weakness in the resilience of the AV's planning algorithm when recovering from anomalous inputs, as the system fails to regain optimal performance after the attack.

A more extreme example of dangerous trajectories, is where the EMI injection causes the AV to lose localisation which, cascades to affect the decision-making of the planning algorithm. The attack localization loss, as indicated by the NDT Error Value and NDT Score increasing, and the sharp variances between autoware and simulator. This behaviour results in the AV veering into the adjacent lane and hitting the side curb, a behaviour characteristic of 6% of the runs within the maximum noise and duration simulation set. Associated with these safety violations are significant performance degradation. In scenarios with low noise levels (0.01), the maximum lateral

TABLE V: Summary of the Safety and Performance Evaluation - Overtake Scenario. No attack was carried out in the baseline experiment.

SAFETY							
Noise	NotF	SafetyV	V_{SiIn}	V_{Col}	V_{NDTLs}	V_{DTC}	V_{BrD}
baseline	0%	1%	0%	0%	0%	1%	0%
0.01	7%	18%	2%	3%	4%	14%	1%
0.05	16%	23%	8%	3%	11%	10%	2%
0.1	29%	40%	18%	2%	26%	14%	1%
0.2	33%	39%	23%	7%	28%	14%	2%

PERFORMANCE						
Noise	Dur	RlOut	$\overline{DTC}$	MxNDTSr	NDTer	S-NDTer
baseline	104.7s	8.2	0.4m	19.4	0.2m	0.1m
0.01	107.3s	7.8	0.2m	55.9	0.2m	0.2m
0.05	121.4s	8.9	0.2m	73.9	0.4m	0.5m
0.1	125.4s	10.0	0.2m	63.9	0.7m	0.9m
0.2	124.7s	10.2	0.2m	53.1	0.6m	0.8m

deviation is limited to around 0.2 meters. However, under maximum noise (0.2) and 20-meter duration conditions, the lateral deviation increases dramatically to 8.2 meters, showcasing the substantial impact of noise on the AV's ability to maintain its path. This severe lateral deviation illustrates the danger posed by noise-induced errors in the vehicle's steering and localization systems.

Moreover, the *RlOut* metric—which tracks the average number of local trajectory transitions during a mission—shows a significant increase under high-noise conditions. This indicates the motion planner's growing uncertainty and inability to maintain a stable trajectory. As the AV continuously switches between trajectories, it struggles to converge on an optimal path, leading to erratic driving behavior and further deviations. Another factor exacerbating these challenges is the increased mission duration under noise attacks. The AV, displaced from its efficient path due to trajectory deviations and localization errors, takes longer to complete the mission. In the 0.2 noise / 20-meter scenario, the mission duration extended by nearly 28 seconds compared to the no-attack baseline, reflecting the inefficiency introduced by the noise attacks.

B. Scenario 2: Overtake Maneuver

In this experiment, the attack length was fixed at 10 meters while varying the noise levels to assess their impact on the vehicle's performance and safety. In the no-attack scenario (see Table. IV), the AV successfully completed the overtaking maneuver with minimal disruptions. The mission failure rate (NotF) was 0%, and a 1% violation of distance to collision (V_{DTC}) was recorded, indicating that in one case, the vehicle exceeded the safe distance from nearby objects. Despite this, there were no sidewalk incursions (V_{SiIn}), collisions (V_{Col}), or localization loss (V_{NDTLs}). The vehicle maintained a safe average DTC of 0.4 meters. The mission duration was 104.7 seconds, with an NDT error of 0.2 and a standard deviation of 0.1.

In the 0.01 noise scenario, V_{NotF} increased to 7%, and by the 0.2 noise level, it reached 33%. Similarly, $V_{NDTLoss}$ was first observed at 0.01 noise (4%), growing to 28% in the 0.2 noise scenario. These results indicate that noise in the sensor input significantly disrupts the vehicle's ability to maintain accurate localization, directly impacting mission success.

In the no-attack scenario, V_{SiIn} and V_{Col} were recorded at 0%, reflecting ideal behavior where the AV stayed within its designated path and successfully avoided NPCs during overtaking. However, as noise levels increased, both metrics worsened. In the 0.01 noise scenario, V_{SiIn} rose to 2%, and V_{Col} to 3%, showing the system's diminished capacity to maintain lane discipline and avoid nearby vehicles. At the highest noise level (0.2), sidewalk incursions increased to 23%, while collisions reached 7%, a significant rise indicating the AV's inability to safely manage the overtaking maneuver under heavy noise interference. These results suggest that sensor noise not only disrupts the vehicle's path but also critically impacts its ability to avoid hazards that could lead to severe accidents involving both pedestrians and other vehicles.

The V_{DTC}, which reflects the rate at which the AV exceeded safe distances from nearby objects, increased from 1% in the no-attack case to 14% in the 0.2 noise scenario. This was accompanied by a rise in sharp braking events as the AV's control system struggled to compensate for the noisy input, leading to more frequent sudden stops. As the noise level increased, the RollOut metric showed greater instability. In the 0.2 noise case, the RollOut metric increased from 8.2 (in the no-attack scenario) to 10.2, indicating the planner's increasing uncertainty in maintaining a stable trajectory.

The mission duration increased as the noise level rose. In the 0.2 noise scenario, the AV took 124.7 seconds to complete the maneuver, an increase from 104.7 seconds in the no-attack scenario. Additionally, the NDT error and its standard deviation saw significant increases, with the NDTer rising from 0.2 to 0.6 and the S-NDTer increasing from 0.1 to 0.8, highlighting the degradation in localization performance under noisy conditions.

C. Scenario 3: Intersection

In the intersection scenario, the attack length remained unchanged at 10 m, while the noise levels varied to assess their impact on the AV's performance during this complex maneuver. In the baseline scenario, the AV successfully navigated the intersection without mission failure (0%) or significant safety violations, aside from a small 3% V_{DTC}. There were no recorded V_{SiIn} or V_{Col}, and the AV maintained an average DTC of 0.4 meters, with an NDTer of 0.1 and a minimal deviation from the reference path of 20.4 meters. The overall mission duration was 65.8 seconds, and the system performed with only 2.2 RollOut changes, indicating a stable and efficient planning process.

As noise levels increased, the NotF rate rose from 8% at 0.01 noise to 25% at 0.2 noise. Safety violations also saw a sharp increase, particularly in terms of V_{NDTLs}, which jumped from 7% at 0.01 noise to 22% at 0.2 noise. This degradation in localization directly impacted the AV's ability to make timely decisions and follow the intended trajectory, leading to more dangerous driving behavior.

While sidewalk incursions and collisions were rare in the baseline scenario, they became more frequent as noise levels rose. At 0.2 noise, 4% of the runs resulted in V_{SiIn}, and 4% in V_{Col} with non-player characters (NPCs) within the intersection. This behavior indicates a critical safety failure, where the AV not only lost control of its lane discipline but also failed to avoid NPCs and pedestrian zones.

TABLE VI: Summary of the Safety and Performance Evaluation - Intersection Scenario. No attack was carried out in the baseline experiment.

SAFETY							
Noise	NotF	SafetyV	V_{SiIn}	V_{Col}	V_{NDTLoss}	V_{DTC}	$\overline{DTC}$
baseline	0%	3%	0%	0%	0%	3%	0.4m
0.01	8%	15%	0%	1%	7%	10%	0.2m
0.05	19%	27%	2%	3%	16%	13%	0.2m
0.1	23%	32%	6%	3%	19%	16%	0.2m
0.2	25%	28%	4%	4%	22%	7%	0.1m

PERFORMANCE							
Noise	Dur	RlOut	MxNDTSr	NDTer	S-NDTer	Dev2Ref	S-Dev2Ref
baseline	65.8s	2.2	38.5	0.1m	0.1m	20.4m	8.2m
0.01	70.5s	3.1	39.5	0.2m	0.2m	39.1m	98.8m
0.05	72.9s	3.9	40.9	0.4m	0.4m	63.6m	170.4m
0.1	74.2s	4.5	37.5	0.5m	0.5m	69.6m	147.5m
0.2	74.5s	4.1	39.1	0.4m	0.5m	77.9m	154.9m

The cumulative deviation remained relatively low in the no-attack baseline scenario, indicating stable performance. However, under the influence of noise, this deviation increased significantly. For example, in the 0.2 noise scenario, the Dev2Ref reached 77.9 meters, with a high standard deviation of 154.9 meters, demonstrating the system's growing instability under attack. The high standard deviation reflects the inconsistency in the AV's ability to maintain a predictable trajectory, as deviations varied considerably at different points along the path. The increasing Dev2Ref values show that the AV struggled to recover from noise-induced errors, leading to significant drift from the planned path.

The results show that the roll-out metric increased as noise levels rose. In the 0.01 noise scenario, the roll-out increased to 3.1, and by 0.2 noise, it rose to 4.1, indicating the planning system's growing uncertainty in selecting and maintaining a stable path. The maximum NDT score also fluctuated, reaching a high of 40.9 in the 0.05 noise scenario, highlighting the deteriorating localization performance.

The NDT error and its standard deviation also increased with higher noise levels. At 0.2 noise, the NDT error rose to 0.4, with a standard deviation of 0.5, indicating significant localization drift. This localization instability contributed to unsafe driving behavior, as reflected in the increased V_{DTC} and collisions. The mission duration also increased with noise levels, from 65.8 seconds in the baseline scenario to 74.5 seconds at 0.2 noise. This duration increase indicates the AV's struggle to efficiently navigate the intersection under attack, as the planning algorithm and control systems were frequently forced to adjust to counteract the noise-induced deviations.

D. Comparison Between Safety Violations and Simulated Scenario

Figure 8 represents radar graphs that provide a clear visual representation of the impact of noise attacks on the AV across all different mission types: straight-line driving, overtaking, and intersection maneuvers, with varying attack lengths (10 meters and 20 meters) for the straight-line scenario. By comparing these radar graphs, we can discern how the attack influences the AV in different maneuvers and understand whether the vulnerability is related to the nature of each maneuver.

In the straight-line scenario (Fig. 8 (a) and (b)), the radar plots show a clear difference between the 10-meter and 20-meter attack lengths. With the 10-meter attack (Figure (a)), the V_{DTL} and V_{NDTLs} are relatively contained at noise levels below 0.1, but they spike at 0.2 noise, indicating that longer attack lengths exacerbate the vehicle's struggle to maintain its trajectory. By contrast, in the 20-meter attack scenario (Figure (b)), the impact of noise is more pronounced across all noise levels, with a higher percentage of NotF and significantly greater V_{DTL} and V_{NDTLs} values. This suggests that the longer attack duration amplifies the system's inability to recover from perturbations in the steering sensor, causing the AV to deviate further from the planned path.

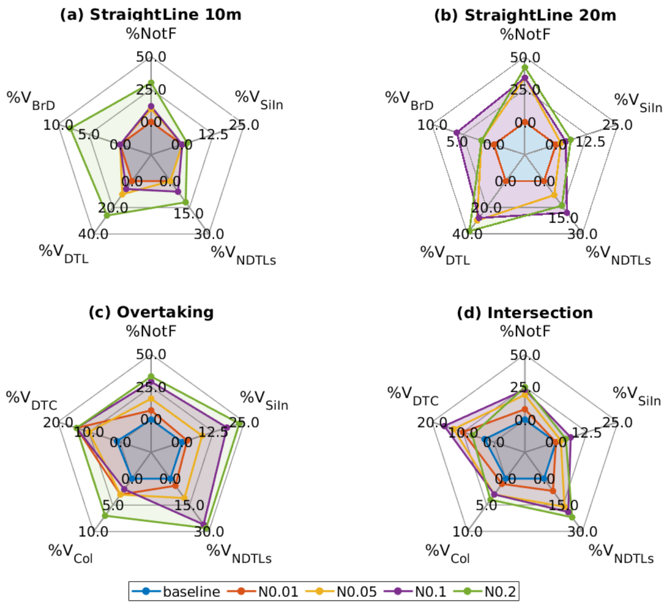

Fig. 8: Safety violation of simulated scenarios.

In the overtaking scenario (Fig. 8 (c)), the radar plot highlights that this maneuver is particularly vulnerable to V_{NDTLs} and V_{DTC} as noise levels increase. Even at 0.01 noise, the AV shows a marked increase in these safety violations, and by 0.2 noise, V_{NDTLs} and V_{DTC} reach critical levels. This indicates that overtaking is a more complex and challenging maneuver for the AV compared to straight-line driving, as it requires the vehicle to safely execute lane changes and avoid collisions with NPCs. The complexity of coordinating between localization, path planning, and collision avoidance makes the system more prone to safety violations when noise is introduced.

In the intersection scenario (Fig. 8 (d)), the radar plot demonstrates that this maneuver is less affected by V_{DTC} compared to the overtaking scenario, but the mission failure rate and localization loss are notably higher. Even at 0.01 noise, NotF jumps to 8%, and V_{NDTLs} reaches 7%, while at 0.2 noise, NotF reaches 25%, indicating a substantial failure rate. The intersection maneuver places a high demand on the AV's localization and planning systems, as it requires precise decision-making in a constrained environment with multiple potential collision points. The increase in safety violations with rising noise levels reflects the difficulty the AV faces in maintaining control during complex navigation tasks in intersections, where it must simultaneously monitor multiple

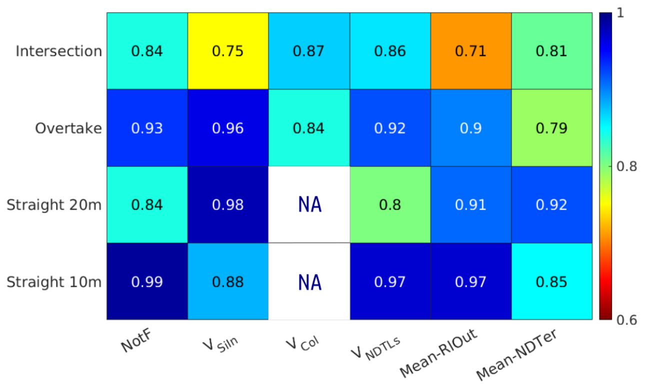

Fig. 9: Correlation coefficients between violation metrics (horizontal axis) and noise levels ([0, 0.01, 0.05, 0.1, 0.2]) for each scenario (vertical axis). The values indicate the strength of the relationship between the likelihood of each violation and changes in noise levels.

potential threats and adjust its trajectory.

The vulnerability of the AV to noise attacks appears closely tied to the nature of the maneuver. Straight-line driving is less demanding in terms of control and localization, and as a result, the AV is able to handle noise better—though longer attack durations (as in Fig. 8 (b)) significantly increase the risk of mission failure. In contrast, overtaking involves more dynamic path changes and collision avoidance, making it more susceptible to noise, as seen in the sharp rise in V_{DTC} and V_{NDTLs} even at low noise levels. Intersection maneuvers also present significant challenges, particularly due to the need for precise localization and decision-making at multiple points, resulting in higher mission failure rates and localization loss as noise levels increase. These findings suggest that the more complex the maneuver (i.e., those requiring more dynamic control and interaction with external factors like NPCs or intersection points), the more vulnerable the AV is to noise attacks.

E. Violation to noise correlation analysis

The correlation heatmap shown in Fig. 9 reveals significant insights into how different safety violations and performance metrics are affected by noise levels across various maneuvers and attack durations. Among all the maneuvers, straight-line driving (10m attack) demonstrates the highest correlation between noise levels and mission failure, with a coefficient of 0.99, indicating that shorter attack duration in straight-line driving are highly sensitive to noise. The overtake scenario follows this with a correlation of 0.93. Both the intersection and straight-line 20m scenarios show a correlation of 0.84 for mission failure, suggesting that longer attack duration and intersection maneuvers are somewhat less sensitive to noise, possibly due to the nature of the mission. Regarding sidewalk incursions, longer attack duration in the straight-line (20m) and overtake scenarios show the strongest correlations, at 0.98 and 0.96, respectively. In contrast, the intersection maneuver displays the weakest correlation for sidewalk incursions, reflecting the controlled, slower nature of this maneuver.

When examining localization loss, straight-line 10m and overtake show the highest correlations, 0.97 and 0.92, respectively, indicating that these scenarios are most affected by noise in terms of localization. The intersection scenario, though still sensitive to noise (0.86), shows a somewhat lower correlation, likely due to the AV's reduced speed and static behavior at stop points. Collision, on the other hand, shows similarly strong correlations in overtaking (0.84) and intersection (0.87) scenarios, but this metric is irrelevant in straight-line driving, as there are no NPCs involved in those maneuvers. The correlation for RollOut switches is also highest in straight-line 10m attacks (0.97), followed by straight-line 20m and overtake, while intersections have the lowest correlation (0.71) in this category. For NDTer, longer attack durations in straight-line scenarios show the highest correlation (0.92), while intersections and overtakes show lower values.

Overall, the straight-line (10m) and overtake scenarios exhibit the highest sensitivity to noise across several metrics, such as mission failure, sidewalk incursions, and localization loss. Intersection scenarios, in contrast, show consistently lower correlations, likely due to the nature of the maneuver, where the vehicle slows down or stops, reducing the dynamic impact of noise during attacks. This behavior at intersections explains the weaker overall correlation with noise, as the AV is generally at lower speeds and is less engaged in continuous movement compared to the overtake and straight-line scenarios. This highlights how the nature of each maneuver, particularly its dynamic or static characteristics, influences the vehicle's vulnerability to noise-induced safety violations and performance degradation.

VIII. Discussion

Throughout the paper, we demonstrated that AD software is sensitive to EMI attacks that can generate different levels of safety violations from low-priority violations, from which the vehicle can recover but resulting in suboptimal behaviour, to severe violations causing collisions or endangering other road users.

RQ1 How does a manipulation to the electromechanical component propagate through the AD software stack?

From our results, it emerges that an EMI attack at the steering sensor level often causes SiIn, DTL, or DTC violations, which are the most commonly visible in Fig. 8. To back-step this behaviour, to eventually debug such a complex AD software stack in a general purpose approach, developers will require an accurate analysis of each block in terms of data input-output relation. In our case, we carried out a back-step analysis at the ROS-topic level to identify the nodes that subscribe to specific messages. Here, we found out that the most probable user of steering sensor data, thus generating violations, is the mission and motion planning module, visible in Fig. 4, and composed of several sub-blocks including `op_trajectory_generator` and `op_waypoint_follower`, that represent the most probable components generating wrong decisions. While at the low level, PID controllers might be able to withstand noise to some extent, intelligent controllers have shown inherent vulnerability to this attack propagating from the low level up as raw sensor data to the master controller and up to the ROS topics.

RQ2 What dependencies exist between the AD control algorithm and low-level control?

High-level intelligent controllers trust digital data flowing over the in-vehicle network communication level. The interdependence of control algorithms resides in the feedback loop reading data from the low level while the AD acts in a hybrid deliberate/reactive robotic paradigm. In such a paradigm, well studied in robotics, an AD reacts quickly upon sensing without performing global-planning, which is typically a computationally demanding task running concurrently. SiIn, DTL, or DTC violations, which are the most commonly found in our analysis, are a typical result of the reactive behaviour of ADs. Similarly, the planner might generate unsafe trajectories in case of localization data corruption such as NDTLs violation or increase in NDTer margin. Eventually, the vehicle can recover from some violation when the global-planner generates a new waypoint, but this is not always guaranteed when some stochasticity is involved in the process.

RQ3 Where in the architecture of the autonomous vehicle can defensive mechanisms be placed to defend against control invariants?

Strategies to detect and mitigate low-level sensor data input manipulation focus on redundancy and multiple levels of data integrity checks. To investigate this question we step through each of the layers of the AV:

- **Low-Level PID Controller:** Integrity and plausibility checking of the PID can mitigate but not stop the injection of anomalous sensor input values. The PID has its own robustness, which is mathematically proved, the PID lacks the intelligence to interpret the meaning behind the input data. Therefore, attacks which manipulate the sensor input always have the possibility of traversing the PID. It is also possible to implement analog filters and hardware saturation, however, as mentioned, at this level, there is no means to discern attack behaviour which resembles regular signal/circuit specification and its operating characteristic.
- **Intermediate Layer:** At this level, it is possible to conduct inspection of the CAN data. The master controller has low-computational capacity. Therefore, implementation of mechanisms to interpret and provide intelligence of the CAN data is limited. Data saturation and filtering is possible at this level. However, filtering and saturation strategies would be challenged to defend against an adaptive sensor manipulation attack which searches for the filtering and saturation parameters and develop a 1-step or n-step attack which falls outside the range.
- **High-Level Control Layer:** A redundant, fall-back controller has a cost in terms of financial, compute and network resources, and cannot guarantee that an attack would also aim to manipulate the redundant controller. Furthermore, redundant controllers accessing the same sensor data might generate the same unexpected behaviour.

Our recommendations, for this particular use case, is to accurately model the sensor behaviour at the physical level considering the physical world world we live in. In this context, sensors, such as everything else, should obey Newton (for motion) and Maxwell equations (for electromagnetism). To detect sensor data anomaly our knowledge of the physical model of the sensor can be utilised to predict variances to this model. This would effectively detect a possible attack much earlier and thus prevent DTC & DTL violations occurring in the motion planning block. The validation of sensor data can run in a concurrent process throwing exceptions in case of unexpected levels of noise. The response action to an exception need to be modelled on the level of risk.

IX. Related Work

The closest work to our study is that of Berdich and Groza [21], which conducted multiple injection attacks (Fuzzing, Replay, DDoS) within the CAN Bus, targeted at diverse low-level sensors (Steering, Braking, Advanced Driver Assistance Systems (ADAS) ECU), within a cruise-control vehicle architecture. The experiment is only conducted in Simulink and the wider software stack including control properties and high-fidelity sensing of the vehicle are not included. The experimentation is conducted on the basis of establishing the feasibility of attack and model potential risk and safety consequences. Similarly, Pöllny et al. [6] developed an EMI attack using a helmholz coil which successfully manipulated a sensor popular used in ADAS. The study which focused at the low-level recommenced the possibility of plausibility check to mitigate steering angle attacks. Within, automotive software, studies of Garcia et al. [22] and Kim et al. [19] have discussed the problem of attacks on the low-level control with software developers, discovering issues with software implementation and development of defensive mechanisms, however, the scope of these studies did not include practical experimentation.

X. Conclusion

Our study analysed the robustness of a real-world AV to attacks on the low-level actuation using the example of an EMI attack on the steering angle sensor. We developed a hybrid low-level sensor and high-fidelity simulation environment, which we used to conduct approx. 1900 runs of diverse attack and driving scenarios. Our results demonstrated that our real-world AV is vulnerable to attacks on the low-level actuation. The effects of these attacks demonstrate that manipulated sensor input can propagate through to the higher-level control, and, in our case, impact modules for AD such as localisation and planning. Given the reactive nature of the AD to sensing, once a malicious input is within the system, the vehicle is to shown to react to this input with unsafe driving actions. Our recommendation to designers of AV software is to accurately model the sensor behaviour at a physical level and use this knowledge to predict variances in the model. This would provide the ability to detect a possible attack earlier and prevent collisions and localisation loss.

Acknowledgment

This work is supported by the Horizon Europe project CyberSecDome (Agreement No.: 101120779) and EU Horizon 2020 project MariCybERA (Agreement No.: 952360) and by the European Commission through the H2020 teaming project Finest Twins (Grant No. 856602).

References

[1] F. Zhang, H. A. D. E. Kodituwakku, J. W. Hines, and J. Coble, "Multilayer data-driven cyber-attack detection system for industrial control systems based on network, system, and process data," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4362–4369, 2019.

[2] C.-V. Briciu, I. Filip, and F. Heininger, "A new trend in automotive software: Autosar concept," in *2013 IEEE 8th International Symposium on Applied Computational Intelligence and Informatics (SACI)*, 2013, pp. 251–256.

[3] F. Munir, S. Azam, M. I. Hussain, A. M. Sheri, and M. Jeon, "Autonomous vehicle: The architecture aspect of self driving car," in *Proceedings of the 2018 International Conference on Sensors, Signal and Image Processing*, ser. SSIP '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 1–5. [Online]. Available: https://doi.org/10.1145/3290589.3290599

[4] N. United States National Transportation Safety Board, "Investigation of lion air flight 610 and ethiopian airlines flight 302," *Safety Recommendation Report NTSB ASR1901. PDF document*, 2019.

[5] E. S. I. B. ESIB, "Accident, loss of control with airbus a320-214 near tallinn airport on 28.02.2018." *Safety Investigations. Investigation report ESIB: A2802118 EECAIRS: EE0180. PDF document.*, 2019.

[6] O. Pöllny, F. Kargl, and A. Held, "Steering your car with electromagnetic fields," in *Proceedings of the 6th ACM Computer Science in Cars Symposium*, ser. CSCS '22. New York, NY, USA: Association for Computing Machinery, 2022. [Online]. Available: https://doi.org/10.1145/3568160.3570228

[7] Y. Tu, V. S. Tida, Z. Pan, and X. Hei, "Transduction shield: A low-complexity method to detect and correct the effects of emi injection attacks on sensors," in *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*, ser. ASIA CCS '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 901–915. [Online]. Available: https://doi.org/10.1145/3433210.3453097

[8] Y. Zhang and K. Rasmussen, "Detection of electromagnetic interference attacks on sensor systems," in *2020 IEEE Symposium on Security and Privacy (SP)*, 2020, pp. 203–216.

[9] Y. Tu, Z. Lin, I. Lee, and X. Hei, "Injected and delivered: Fabricating implicit control over actuation systems by spoofing inertial sensors," in *27th USENIX Security Symposium (USENIX Security 18)*. Baltimore, MD: USENIX Association, Aug. 2018, pp. 1545–1562. [Online]. Available: https://www.usenix.org/conference/usenixsecurity18/presentation/tu

[10] T. Trippel, O. Weisse, W. Xu, P. Honeyman, and K. Fu, "Walnut: Waging doubt on the integrity of mems accelerometers with acoustic injection attacks," in *2017 IEEE European Symposium on Security and Privacy (EuroS&P)*, 2017, pp. 3–18.

[11] P. Dash, M. Karimibiuki, and K. Pattabiraman, "Stealthy attacks against robotic vehicles protected by control-based intrusion detection techniques," *Digital Threats*, vol. 2, no. 1, jan 2021. [Online]. Available: https://doi.org/10.1145/3419474

[12] S. Jha, S. Banerjee, T. Tsai, S. S. Hari, M. B. Sullivan, Z. T. Kalbarczyk, S. W. Keckler, and R. K. Iyer, "Ml-based fault injection for autonomous vehicles: A case for bayesian fault injection," in *2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. Los Alamitos, CA, USA: IEEE Computer Society, jun 2019, pp. 112–124. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/DSN.2019.00025

[13] L. J. Moukahal, M. Zulkernine, and M. Soukup, "Boosting grey-box fuzzing for connected autonomous vehicle systems," in *2021 IEEE 21st International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, 2021, pp. 516–527.

[14] H. Choi, W.-C. Lee, Y. Aafer, F. Fei, Z. Tu, X. Zhang, D. Xu, and X. Deng, "Detecting attacks against robotic vehicles: A control invariant approach," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 801–816. [Online]. Available: https://doi.org/10.1145/3243734.3243752

[15] H. J. Jo and W. Choi, "A survey of attacks on controller area networks and corresponding countermeasures," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 6123–6141, 2022.

[16] A. Buscemi, I. Turcanu, G. Castignani, A. Panchenko, T. Engel, and K. G. Shin, "A survey on controller area network reverse engineering," *IEEE Communications Surveys and Tutorials*, vol. 25, no. 3, pp. 1445–1481, 2023.

[17] S. K. T. U. of Tokyo), "Autoware: Ros-based oss for urban self-driving mobility," in *ROSCon Vancouver 2017*. Open Robotics, September 2017. [Online]. Available: https://doi.org/10.36288/ROSCon2017-900813

[18] H. Darweesh, E. Takeuchi, and K. Takeda, "Openplanner 2.0: The portable open source planner for autonomous driving applications," in *2021 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)*, 2021, pp. 313–318.

[19] H. Kim, R. Bandyopadhyay, M. Ozmen, Z. Celik, A. Bianchi, Y. Kim, and D. Xu, "A systematic study of physical sensor attack hardness," in *2024 IEEE Symposium on Security and Privacy (SP)*. Los Alamitos, CA, USA: IEEE Computer Society, may 2024, pp. 146–146. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/SP54263.2024.00143

[20] G. Lou, Y. Deng, X. Zheng, M. Zhang, and T. Zhang, "Testing of autonomous driving systems: where are we and where should we go?" in *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ser. ESEC/FSE 2022. New York, NY, USA: Association for Computing Machinery, 2022, p. 31–43. [Online]. Available: https://doi.org/10.1145/3540250.3549111

[21] A. Berdich and B. Groza, "Cyberattacks on adaptive cruise controls and emergency braking systems: Adversary models, impact assessment, and countermeasures," *IEEE Transactions on Reliability*, vol. 73, no. 2, pp. 1216–1230, 2024.

[22] J. Garcia, Y. Feng, J. Shen, S. Almanee, Y. Xia, Chen, and Q. Alfred, "A comprehensive study of autonomous vehicle bugs," in *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, ser. ICSE '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 385–396. [Online]. Available: https://doi.org/10.1145/3377811.3380397