ATOS: An Open-Source Platform for Testing AI-Based Automated Vehicle Systems in Integrated Simulated and Physical Scenarios

Victor Jarlow*, Lukas Wikander*, Samuel Thorén*, Robert Brenick*, Xianbiao Hu†, Timo Kero*
*AstaZero AB
{firstname.lastname}@ri.se
† Penn State University
xbhu@psu.edu

***Abstract*—The increasing complexity of automated and AI-based vehicle systems necessitates novel testing methodologies to ensure safety, reliability, and performance in diverse and dynamic scenarios. This paper introduces AV Test Operating System (ATOS), an open-source platform that seamlessly integrates simulated and physical testing environments for Advanced Driver-Assistance Systems and Automated Driving systems. ATOS supports comprehensive scenario evaluations under varied conditions such as weather, traffic, and connectivity environments by automating and orchestrating tests involving multiple simultaneous virtual and physical objects. Key contributions of this work include: 1) The introduction and design of ATOS, an open-source platform for automating and orchestrating complex vehicle testing scenarios. 2) A method for evaluating the repeatability of automated test orchestration systems. 3) Data analysis of two scenarios where ATOS was used in real-life AV testing. Results demonstrate ATOS's effectiveness in executing repeatable and reliable tests across diverse configurations, highlighting its utility for research institutions, vehicle manufacturers, and testing facilities. Using ATOS in repeated runs, the time delay when triggering dynamic events was within 1 millisecond with virtual objects and 7 milliseconds when using physical objects. The positional variation between runs using virtual and real-life objects amounted to 6 and 20 cm, respectively.**

I. Introduction

Automated Driving (AD) and Advanced Driver-Assistance Systems (ADAS) systems promise increased safety, sustainability, accessibility, and efficiency throughout the transportation system. These AI-based, safety-critical systems include sensors and software that must be tested and validated over millions of miles [1]. To handle the vast testing needs, methodologies must include extensive software simulations, frequent physical tests of critical scenarios in safe and controlled environments such as proving grounds, and ongoing monitoring during real-world use [2]. Here, tools for correlating proving ground testing and simulations play a key role [3].

Moreover, initiatives such as EuroNCAP, USNCAP, and ChinaNCAP incentivize vehicle safety by providing a rating system and appurtenant certification criteria. EuroNCAP's roadmap for 2030 points to system robustness as a critical enabler for increasing road safety, achieved by varying scenario parameters and several simultaneous test objects [4].

Additionally, experimentation and testing are required to investigate the effect of traffic safety of recent and emerging concepts such as driver monitoring systems [5], autonomous vehicle intent signaling [6], risk acceptance criteria [7], remote vehicle operation [8], collaborative vehicle platooning [9, 10, 11, 12], worksite flagman detection [13] and more.

Given the importance and broad scope of testing, validation, and certification of safety functions in automated vehicles, further research is needed into safe, efficient, and accurate testing methods for AD and ADAS system experimentation, testing, and validation, considering the interaction between physical testing and virtual or simulated testing.

As a reaction to today's closed-source commercial test systems based on proprietary protocols [14, 15], we designed and evaluated a novel integrated AV-testing platform for simulated and physical test automation of AI-based vehicle systems: ATOS [16], available as open-source software. ATOS aims to provide researchers and testers in the automotive industry with a safe, efficient, interoperable, and repeatable way to automate vehicle testing. Standards are a cornerstone of the design, giving a highly interoperable system that enables advanced research and testing of future use cases, such as dynamic connectivity control and weather simulation, beyond the capabilities of existing systems. While we touch on ATOS's safety, efficiency, and accuracy aspects, this work focuses on the aspect of *repeatability*, which is important in AD and ADAS system validation. To this end, we designed and conducted simulated and real-life experiments in different scenarios to understand an AV-testing platform's repeatability. We then analyzed the experimental data, concluding that ATOS maintains high repeatability across various configurations. We found that ATOS contributes with as little as a millisecond of latency during event processing in dynamic scenario execution.

To summarize, the following contributions were made:

- The introduction of ATOS, an open-source platform for automating and orchestrating complex traffic scenarios.

We thank Sweden's Innovation Agency, Vinnova, for funding Grant Nos. 2021-05042, 2022-01648, 2023-01704, and 2023-03297 making the work described here possible.

- A method for evaluating the repeatability of automated test orchestration systems.
- An analysis of data from two scenarios where ATOS was used in real-life AV testing.

The subsequent sections are structured as follows: Section II describes the requirements for automated vehicle testing and the current state of the art. The proposed system design is given in Section III. Section IV outlines the experimental method, Section V the results, and a discussion thereof. Lastly, our conclusions and identified future work follow in Section VI.

II. Background

To the best of our knowledge, full-scale automated vehicle testing systems exist external to the research literature as proprietary commercial systems [14, 15]. Therefore, we seek guidance from the interviews conducted by [17] together with vehicle manufacturers, suppliers, and researchers. They produced a set of requirements for automated vehicle testing complexity, data collection, and automation. In the discussions, standardized data formats in testing were deemed highly important, a sentiment underscored by a recent survey study involving engineers at Mercedes-Benz [18]. What follows is an adapted selection of the identified requirements:

R1 *Data, logs, and measurements need to be collected during driving and analyzed to determine the outcome of a test. To be able to exchange data, a certain standardized data format is required.*

R2 *Simulation environments must be connected to real environments through standardized data formats to allow seamless scenario execution and transition between the two.*

R3 *The test equipment should support communication between test objects to allow for synchronization, enabling future complex test scenarios.*

R4 *It should be possible to do testing of varying conditions such as different weather (e.g., snow, rain, sunshine, fog), times of the day, types of vehicles under test, certification targets, and equipment.*

R5 *Automated vehicles under test should never leave their respective driving lane or operational area.*

R6 *Automatic repetitive testing (i.e., running the same test several times with small changes) needs to be supported.*

With these requirements as a guide, the development of ATOS resulted in certain design choices, which we will discuss in further detail in Section III.

The first two requirements relate to standardization. In this context, the Association for Standardization of Automation and Measuring Systems (ASAM) is a non-profit organization that promotes standardization in automotive development and testing toolchains, improving interoperability between simulation tools [19]. ASAM's OpenSimulationInterface (OSI) ensures compatibility between simulation tools and automated driving software by providing standardized sensor integration and data exchange interfaces. Similarly, Open Data Services (ODS) offers a standard interface for storing, accessing, and analyzing test data generated during vehicle testing.

Standards like OpenDRIVE and OpenSCENARIO establish a common language for describing road networks and dynamic traffic scenarios. Although initially created for simulation, these standards apply to physical testing. The popular esmini package [20] features an OpenSCENARIO simulator, useful for playing such files. Recent studies, such as [21], demonstrate the use of these standards to enhance simulation realism for automated driving systems. Furthermore, [22] highlight the need for a standardized approach to sharing simulation results and digital twin data within the automotive test chain.

ISO/TS 22133:2023 [23] is another relevant standard that describes functional requirements, specifications, and a communication protocol for test object monitoring and control for active safety and automated/autonomous vehicle testing. This standard relates to the third requirement and defines a uniform communication interface between test control centers and equipment, allowing the coordination of complex test scenarios involving specialized equipment from different manufacturers.

Regarding the fourth requirement, conditions such as weather are emulated at proving grounds and research facilities such as Japan Automotive Research Insititute (JARI), Virginia Tech Transportation Institute (VTTI), Messring, and Mercedez Benz Immendingen Test Center have created methods that allow testing of various weather-related phenomena such as rain, fog, snow, as well as different lighting conditions [24, 25, 26, 27]. Such weather-emulating machinery could be integrated into automated testing to provide richer scenarios, e.g., trigger water flow at specific moments in a test scenario.

Requirement five underlines that fully automated AV testing requires a high degree of safety. The recent geofencing method described by [28] ensures that vehicles are contained to their operational area. Such a method is useful in automated scenario-based AV testing, providing additional safety by preventing hazardous situations.

Aside from safety, repeatability is an important aspect of testing, according to requirement six. It allows testers to verify their systems in isolation, minimizing the effect of errors in measurement equipment and tools. Guide to the Expression of Uncertainty in Measurement (GUM) [29] describes repeatability as the closeness of the agreement between the results of successive measurements of the same measurand carried out under the same conditions of measurement, which we explore further in Sections IV and V, but not before describing our proposed test system.

III. AV Test Operating System

We now present the open-source system for automated execution of AV testing, ATOS, available on GitHub [16]. It is part of a larger verification and validation process (see Figure 1, which includes planning in the form of site mapping and critical scenario selection [30, 31], and analysis of results. During the execution of a test scenario, ATOS continuously monitors and communicates with a set of *test objects*, which can be either virtual objects that can represent real-life (e.g., simulated cars, pedestrians, sensors, and more) or physical objects and phenomena in real life, such as cars, trucks, drones (aerial and ground-based), robot platforms, quality of connectivity, and weather phenomena, among others.

By basing the architecture of ATOS on Robot Operating System 2 (ROS2) [32], users and systems can interface via standardized ROS2 messages, in line with requirement *R1*. To manage the complexity of automated scenario test execution, ATOS is built as a collection of modules designed for specific tasks. To satisfy requirement *R2*, ATOS interfaces

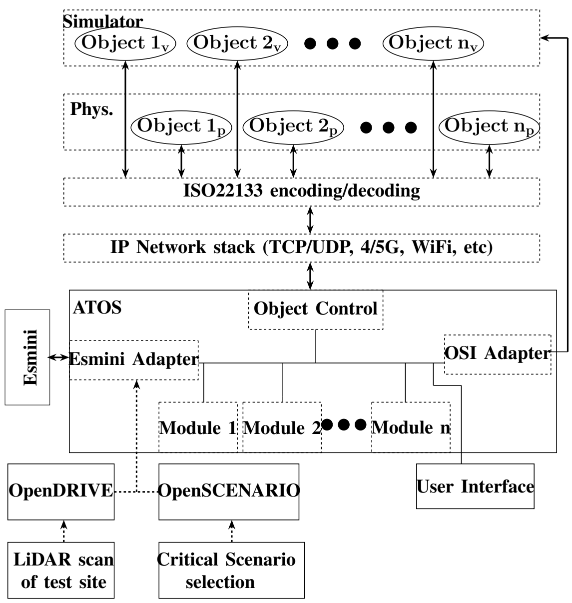

Fig. 1: Overview of the communication between modules within ATOS and externally.

with esmini [20], an openSCENARIO simulator. Through integration with its C-API, ATOS can trigger actions dynamically during runtime and support OpenSCENARIO and OpenDRIVE. When executing scenarios with physical and simulated objects, physics-based vehicle simulators require information regarding the physical objects. ATOS provides this information via the OSI Adapter module. At the same time, test equipment and objects use the communication protocol described in the ISO/TS 22133:2023 standard [23] through the Object Control module, in line with requirement *R3*. This means that test equipment, e.g., for emulating different weather phenomena or different types of vehicles or equipment, is possible in accordance with requirement *R4*. ISO/TS 22133:2023 describes functionalities such as test scenario execution, static and dynamic trajectory following, triggers and actions, and remote control driving, implemented to a high degree in ATOS. Furthermore, ATOS supports safety features such as emergency stop, object arm/disarm, geofencing (related to requirement *R5*), and millisecond time synchronization.

Test objects also adhere to [23] and communicate with ATOS using the following messages:

- **Trajectory Object Message (TRAJ)** for sending trajectories to objects with set points in time, position, heading, acceleration, and velocity.
- **Object Settings Message (OSEM)** for configuring test objects (e.g. test scenario origin coordinates, transfer rate for messages).
- **Object State Change Request Message (OSTM)** for requesting a state change of the test objects. Test objects can be in one of the states: init, armed, disarmed, remote_controlled, and running.
- **Start Message (STRT)** specifically for placing an object in the RUNNING state, causing it to start following its trajectory. The control center can only send this message when the test object is armed.
- **Heartbeat Message (HEAB)** is continuously sent with variable frequency from the control center to the test objects with a tuneable frequency. it reports the status of the control center and is used to signal, e.g., emergency abort to the test object.
- **Monitor Message (MONR)** is continuously sent with variable frequency from each test object to the control center to report the current state, pose, and status.

Executing a scenario scenario run with ATOS entails the following preparatory steps:

- Preparing an OpenSCENARIO and OpenDRIVE file describing the test location and scenario.
- Configuring settings of each test object (IP address, bounding-box dimensions, etc).
- Configuring ATOS modules to be used in the test (e.g., scenario name, test objects used in the test, send-rate of MONR and HEAB messages, etc.).

Once properly configured, what follows is a summary of the steps that occur during interaction with ATOS:

1) ATOS receives an initialization request, causing the configured scenario to be loaded into esmini, to create trajectories for ATOS to distribute to the test objects later.
2) ATOS receives a connection request, causing ATOS to establish an IP connection to each active configured test object using the ISO/TS 22133:2023 communication protocol.
3) ATOS receives an arm request, causing ATOS to send an OSTM message, transitioning each test object to the arm state. When performing testing with physical objects, no humans should be present in the test area when the test objects are armed.
4) ATOS receives a start request, causing ATOS to send an STRT message to each test object, placing them in the running state.
5) ATOS may abort the test at any time while running. This can occur because a) ATOS receives an abort request, b) ATOS loses network heartbeat connection to an object, and c) a test object reports an emergency (e.g., due to component failure). This causes ATOS to send an abort request to each test object.

These steps can be repeated sequentially for automated testing in line with requirement *R6*. Each test in the sequence generates a log compiled by ATOS from received MONR messages. These logs provide insights into the test and the repetability of ATOS, as discussed in the next section.

IV. Experimental Setup

This section outlines the choices in experimental design and analysis of the repeatability of ATOS. It covers the selected traffic scenarios, test objects, generated datasets, the types of errors relevant to the context, and the evaluation metrics.

A. Scenarios

Two traffic scenarios were selected (see Figure 2), typical of AD and ADAS testing. Scenario 1 is the EuroNCAP ADAS scenario called Car-to-Bicyclist Longitudinal Adult (CBLA) [33], containing two actors. Scenario 2 is a scenario designed to challenge AD vehicles and contains 6 actors. What follows is a detailed description of each scenario.

Scenario 1, as seen in Figure 2, is a CBLA EuroNCAP scenario [33] for automatic emergency braking (AEB) evaluation. A *car (ego)* and a *bicycle* follow parallel, overlapping trajectories in the scenario. The scenario starts with the bicycle accelerating until it reaches a longitudinal velocity of $4.16\,\mathrm{m/s}$, at which point the car accelerates to $13.88\,\mathrm{m/s}$ on a collision course. The purpose of the scenario is to evaluate the ability of the car's AEB system to prevent collision by braking before colliding with the bicycle.

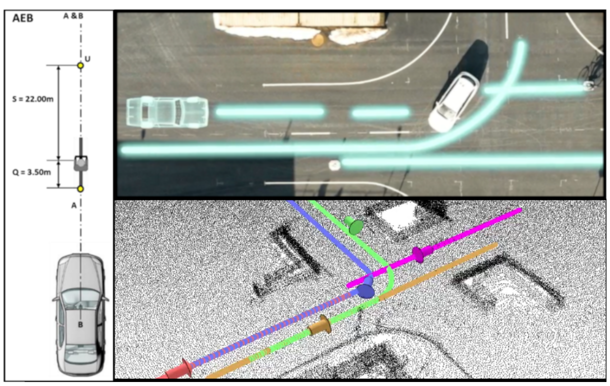

Fig. 2: Left: Scenario 1, CBLA from [33]. Right: Scenario 2, head-on collision, top: drone-photographed, bottom: view within the graphical user interface of ATOS.

Scenario 2, as seen in Figure 2, describes a head-on collision (HoC) situation between a reckless *speeder car* performing an overtake maneuver and a *car (ego)* in a busy 4-way intersection monitored by two video-surveillance *UAV drones*. The ego car enters the intersection, turning westward. At the same time, the reckless speeder overtakes a *road user* as it enters the intersection. As such, the speeder is in the wrong lane and on a direct collision course with the ego vehicle. At the same time, a *bicyclist* enters the intersection, trailing close behind the ego vehicle. This creates a contradictory situation for the AD system of the ego car, prompting the question: *How should it act to satisfy the opposing safety needs of the involved individuals, i.e., passengers in the ego car, the oncoming vehicle, and the bicyclist?*. The purpose of the scenario is to evaluate the ability of the car AD system to minimize risk in a complex situation with multiple road users.

B. Test Objects

Several different test objects, both virtual and physical, were used in the evaluation, detailed in Table I.

Fig. 3: Left to right: A car, a Humanetics UFO platform with a cyclist dummy, and AstaZero's Flexible Target Carrier with a yellow marker at FLXZone, AstaZero proving ground.

Virtual Objects of two types were used: objects with *idealized behavior* by interpolating time and position and objects from *physics-informed simulation*. Objects with idealized behavior use the *(IsoObject)* reference implementation of an ISO/TS 22133:2023 client [34]. The physics-informed simulation was set up in the CARLA simulator [35], including a virtual car. Figure 1 depicts the information flows between ATOS and Simulators, where ATOS sends and receives test object information via ISO/TS 22133:2023 and IP traffic while also feeding information about the physical test into the simulation via the OSIAdapter module.

Scenario	Conf.	Object	Role	Data link
1. CBLA	1.1	O_1: Ford Mondeo	Ego Car	Ethernet
		O_2: IsoObject	Bicycle	Loopback
	1.2	O_1-O_2: IsoObject	As above	Loopback
2. HoC	2.1	O_1: Ford Mondeo	Ego Car	Private 4G
		O_2: UFO	Bicyclist	WiFi
		O_3: FTC	Road user	Private 4G
		O_4: CARLA Car	Speeder Car	Ethernet
		O_5: DJI Drone	Filming UAV	Private 4G
		O_6: DJI Drone	Filming UAV	Private 4G
	2.2	O_1-O_6: IsoObject	As above	Loopback

TABLE I: The experimental configurations used to generate the evaluation data.

AstaZero's research vehicle, a Ford Mondeo Hybrid 2019, shown in Figure 3. Its characteristics are described in [28]. The other ground-based physical objects seen in Figure 3 are two target carriers, a Humanetics Ultra-Flat Overrunable Robot Platform (UFOnano) and a Flexible Target Carrier (FTC) developed by AstaZero. The filming UAVs are DJI Mavic 3 Pro drones. Although controlled by ATOS, they are not part of the traffic scenario and therefore excluded from the evaluation of Section V

C. Experiments

Each experiment consists of a number of experimental runs, wherein ATOS executes a scenario from start to finish with a specific configuration of objects, as described in Table I. Due to budget limitations and equipment availability reasons, Configuration 1.2 contains 15 executions, and Configuration 2.2 contains 7 executions. Moreover, these runs were executed with a MONR-rate of $100\,\mathrm{Hz}$. After initial analysis of these configurations, data from 40 additional executions, 20 of which were collected using a MONR-rate of $100\,\mathrm{Hz}$, and 20 of which had a MONR-rate of $1000\,\mathrm{Hz}$ for the fully virtual configurations 1.2 and 2.2. In each experiment, ATOS was run on a computer with 32 GB RAM and an Intel i7-1260P@$4.70\,\mathrm{GHz}$ CPU with 16 threads.

D. Intrinsic and extrinsic errors

When analyzing and interpreting the generated data, it is important to discern between errors intrinsic to ATOS and those sprung from external sources. Figure 4 illustrates the different types of errors that can cause spatiotemporal deviations between repeated runs of a scenario.

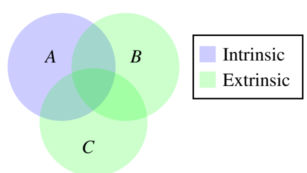

Fig. 4: A Venn diagram of error types intrinsic and extrinsic to ATOS: *A* represents ATOS processing time, *B* network latency, and *C* object trajectory deviation. Their intersection highlights potential error sources in automated ATOS testing.

Hypothetically, the chosen scenario configuration should affect the repeatability. For example, a virtual object should follow its trajectory perfectly. In contrast, physical objects may deviate from their trajectory for various reasons, such as path-tracking algorithm performance, positioning sensor accuracy, road geometry, etc. Similarly, the data link employed will affect scenario timings, such as start triggers and other dynamic events involving multiple objects. However, the errors that are intrinsic to ATOS consist of the processing and messaging times between internal modules. These are affected by, e.g., the number of concurrent objects, the configured rate of messages, and so on.

E. Metric

We use the repeatability coefficient (rc) to quantify the repeatability of ATOS. The repeatability coefficient is the value under which the difference between any two repeat measurements of the same measurand acquired under repeatability conditions (aka identical conditions) should fall with 95% probability, given that measurement differences are normally distributed [36]. It is defined as:

$$rc = 1.96 \times \sqrt{2\sigma_w^2} \tag{1}$$

where σ_w is the within-subject standard deviation, describing the standard deviation between measurements of each run at a specific time.

V. Results

We now present the results of our experiments, investigating *temporal* and *spatiotemporal* variability between repeated experimental runs. We measure temporal repeatability by examining the delay between a trigger being satisfied and its action occurring (focusing on variability between runs due to error types *A* and *B* from Figure 4). We also investigate the spatiotemporal repeatability of relative object distances between runs. Here, the inter-run variability due to error types *A*, *B*, and *C* in Figure 4 is measured. To this background, we now delve into a presentation and analysis of our results.

A. Temporal variability

Figures 5 and 6 show the time delay between a trigger and its action taking effect. Specifically, the time between a vehicle reaching a certain position in the scenario and another object being transitioned to the running state by ATOS via a *positional start trigger*.

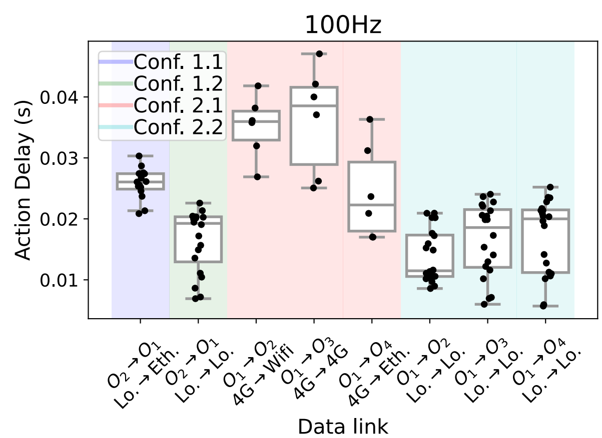

Fig. 5: Box plot of the time delay between trigger satisfaction and action. The x-axis categories describe triggering and acting objects, and the used communication media

Starting with Figure 5, containing the action delay when the MONR frequency is $100\,\mathrm{Hz}$. It is immediately noticeable that the box of each box plot of Scenario 2 is larger than in Scenario 1, indicating higher variability. Both fully virtual configurations have lower action delay times, as is expected. Moreover, the box plots visualize data generated using different data links between ATOS and the objects.

Starting with Configuration 1.1 in Figure 5, the mean and median action delay times are $25\,\mathrm{ms}$, as can be seen in Table II, and each measurement is quite closely clustered around those values. The difference in extremal points is $10\,\mathrm{ms}$, and the standard deviation is $2\,\mathrm{ms}$. Shifting to Configuration 1.2, both the median and mean action delay times are close to $10\,\mathrm{ms}$. The standard deviation is $2\,\mathrm{ms}$, and the range between extremal values is $12\,\mathrm{ms}$. The low delay times are expected since both objects have a loopback interface data link to ATOS. The rc value of both configurations in Scenario 1 is $7\,\mathrm{ms}$, in line with expectations due to the stability of loopback and Ethernet interfaces.

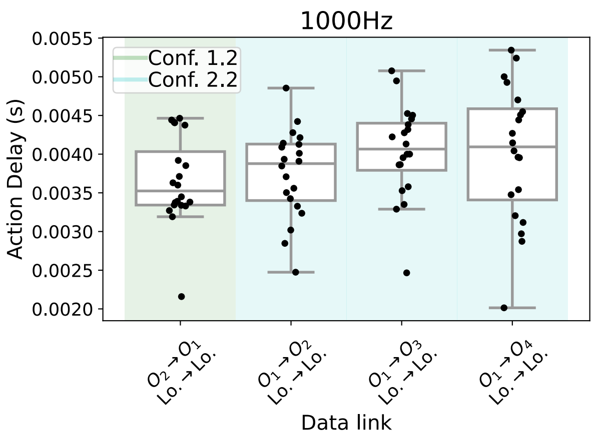

Fig. 6: Box plots of the time delay between trigger satisfaction and action. The x-axis categories describe triggering and acting objects, and the used communication media

MONR freq.	Scenario	Conf.	Runs	Objects	Mean (s)	σ (s)	Min (s)	Max (s)	rc (s)
100hz	1	1.1	15	O_2, O_1	0.0257	0.0024	0.0208	0.0303	0.0069
		1.2	20	O_2, O_1	0.0104	0.0027	0.0046	0.0171	0.0078
	2	2.1	7	O_1, O_2	0.0351	0.0047	0.0268	0.0418	0.0143
				O_1, O_3	0.0362	0.0080	0.0250	0.0470	0.0245
				O_1, O_4	0.0243	0.0071	0.0169	0.0363	0.0218
		2.2	20	O_1, O_2	0.0165	0.0058	0.0059	0.0240	0.0166
				O_1, O_3	0.0170	0.0060	0.0056	0.0252	0.0171
				O_1, O_4	0.0138	0.0042	0.0085	0.0209	0.0119
1000hz	1	1.2	20	O_2, O_1	0.0036	0.0005	0.0021	0.0044	0.0015
	2	2.2		O_1, O_2	0.0037	0.0005	0.0024	0.0048	0.0016
				O_1, O_3	0.0040	0.0005	0.0024	0.0050	0.0016
				O_1, O_4	0.0041	0.0008	0.0020	0.0053	0.0024

TABLE II: Descriptive statistics of the measured action delays. One sample was collected per run.

Comparing the two cases of Scenario 2 in Figure 5, the fully virtual configuration has an overall lower action delay time. The values of Configuration 2.1 do not seem to follow any specific trend, whereas the action delay times of Configuration 2.2 seem to be clustered at either 10 ms or 20 ms. The means of Configuration 2.1 range from 24 ms to 36 ms, and Configuration 2.2's means range from 14 ms to 17 ms. In that same vein, rc's range from 14 ms to 24 ms for Configuration 2.1 and from 12 ms to 17 ms, indicating generally lower delays with less variability, and thus higher repeatability.

Figure 6 visualizes Configuration 1.2 and 2.2. In contrast to Figure 5, MONR messages are sent every millisecond, giving a more fine-grained time step size to more precisely investigate the action delay times. According to Table II, all statistics are lower, indicating an increase in fidelity when ATOS is configured to use higher MONR frequencies. For Configuration 1.2, using a MONR frequency of 1000 Hz compared to 100 Hz results in a mean value that is 68 ms lower across runs, which is significant. Similarly, the rc is significantly lowered by 63 ms. For Configuration 2.2, there is a similar trend, where, e.g., the rc when O_1 triggers O_3 is lowered to 2 ms down from 17 ms. This points to the benefit of using high MONR frequencies whenever possible to ensure high performance in dynamic event triggering.

B. Spatiotemporal variability

Figure 7 shows the relative distances between object O_1 and object O_2 in configurations 1.1 and 1.2. It is clear that Configuration 1.2 has a lower within-subject standard deviation. Moreover, there is an immediately noticeable difference in the shape of the curve of object distances. This is due to the fact that no AEB function is implemented in Configuration 1.2, causing Object O_1 to drive straight through Object O_2 (seen between seconds 23 and 24).

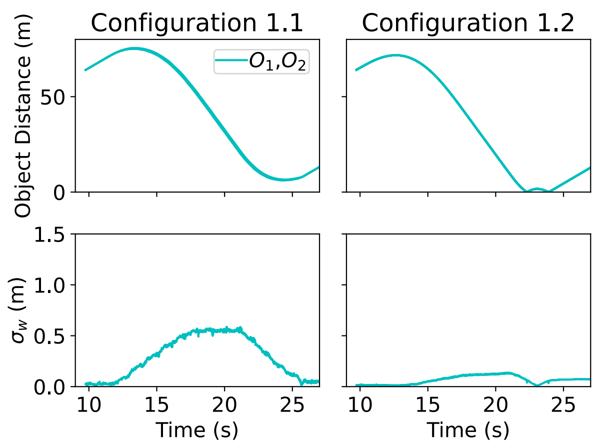

Fig. 7: Scenario 1 (CBLA). Top: Relative object distances, bottom: within-subject standard deviation

Focusing on Configuration 1.1 in Figure 7, the within-subject standard deviation ramps up until the midpoint of the scenario, after which it subsides closer to the end. Examining the scenario in greater detail, it is clear that the physical object O_1 is stationary between 10 and 12.5 seconds. At around 12.5 seconds into the scenario, the virtual object O_2 triggers the start of O_1. As O_1 starts to move, inter-run variability is introduced. Towards the end of the scenario, Object O_1 stops due to its AEB function activating, causing a lower within-subject standard deviation. In this case, rc is 1.0499 m with maximum and minimum within-subject standard deviation of 0.5874 and 0.0106, according to table III.

Looking at the fully virtual Configuration 1.2 in 7, the within-subject standard deviation is clearly lower than for Configuration 1.1, with an rc value of 0.2003, maximum and minimum within-subject standard deviations of 0.1378 and 0. The difference in repeatability could be attributed to the absence of physical objects, pointing to extrinsic errors of type *B* and *C*.

Figure 8 show the distances between Object O_1 and the other ground-borne objects (O_2,O_3, and O_4). Each plotted color in the upper plots represents an experimental run. Similar to Figure 7, there is a clear difference between the fully simulated runs and those involving physical objects, both in object distances over time due to simulation fidelity and in within-subject standard deviation.

Concentrating on Configuration 2.1, there is some within-subject standard deviation between runs, similar to Configura-

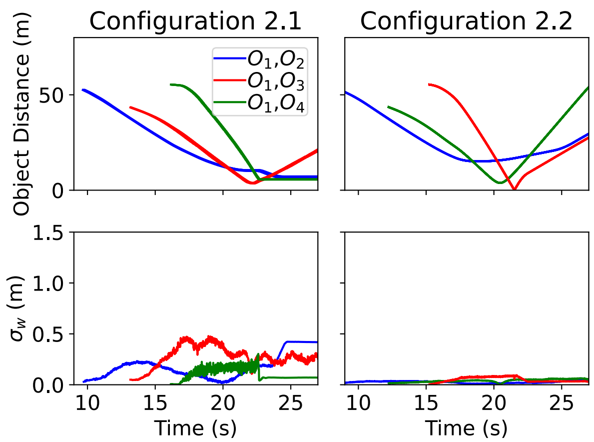

Fig. 8: Scenario 2 (HoC). Top: Relative object distances, bottom: within-subject standard deviation

Scenario	Conf.	Runs	Samples	Objects	Maxσ_w(m)	Minσ_w(m)	rc (m)
1	1.1	15	1898	O_2, O_1	0.5874	0.0106	1.0499
	1.2	20	2381	O_2, O_1	0.1378	0	0.2003
2	2.1	7	1857	O_1, O_2	0.4227	0.0160	0.7221
			1521	O_1, O_3	1.4565	0.0451	1.2504
			1226	O_1, O_4	0.3033	0.0022	0.3514
	2.2	20	2602	O_1, O_2	0.0451	0	0.0669
			2246	O_1, O_3	0.0649	0	0.1003
			1950	O_1, O_4	0.0944	0	0.1232

TABLE III: Descriptive statistics of the relative object distances. All results were obtained with a MONR frequency of 100 Hz.

tion 1.1. When two observed objects stop, the within-subject standard deviation stabilizes to a constant value over time, seen at 22.5 seconds and 23 seconds into the scenario for the distance between O_1 and O_2 and the distance between O_1 and O_4 respectively. The data with O_1 and O_3 has the highest maximum and minimum within-subject standard deviation with 1.4565 and 0.0451 m, respectively. The rc is also highest at 1.2504. The lowest rc of the configuration is between O_1 and O_4. This is expected since the data links between ATOS and O_1 and O_3 are cellular both ways, and both objects are physical, while in the case of O_1 and O_4, O_4 is a virtual object with an Ethernet cable connection to ATOS.

Configuration 2.2 in Figure 8 contains the fully virtual runs of Scenario 2. Compared to Configuration 2.1, it is clear that the within-subject standard deviation is lower between each object. The rc for each object combination is between 0.0669 and 0.1232, and the maximum and minimum within-subject standard deviation are between 9 and 4 cm and 0 cm, respectively, which is low compared to Configuration 1.1.

VI. Conclusion

We have introduced ATOS, an interoperable AV testing platform designed to meet the growing complexity of automated and AI-based vehicle systems, following the automated testing requirements outlined in Section II. The evaluation focused on ATOS's temporal and spatiotemporal reliability across various scenarios and configurations. Temporal repeatability was assessed by measuring the delay between trigger satisfaction and action execution, while spatiotemporal repeatability was evaluated through inter-object distance measurements across multiple runs.

Results showed higher MONR message frequencies (e.g., 1000 Hz) reduced temporal variability and delays. Configurations with more reliable data links also exhibited lower delays, smaller standard deviations, and improved repeatability coefficients. Spatiotemporal variability was influenced by object types (physical vs. virtual) and communication link stability. Virtual configurations achieved better repeatability due to the absence of physical-world noise and more stable data links. In contrast, physical objects introduced greater variability, especially in wireless data link setups, as observed in Scenario 2, Configuration 2.1. Overall, the findings highlight the critical impact of system configuration, communication link stability, and MONR frequency on automated testing systems' temporal and spatiotemporal reliability.

Future work should examine which configurations lead to highly repeatable results, e.g., investigating data links (e.g., 5G and 6G), path-tracking algorithms, and positioning system solutions that maximize repeatability and comply with EuroNCAP tolerance requirements. Beyond repeatability, ATOS efficiency, safety, and accuracy are possible targets for future research. Here, ATOS could be compared on cost-efficiency and capabilities to other systems and methods. Finally, support for the ASAM ODS standard can further enhance ATOS's interoperability and data compatibility in future development.

References

[1] N. Kalra and S. Paddock M. "Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?" In: *Transportation Research Part A: Policy and Practice* 94 (2016), pp. 182–193.

[2] *New Assessment/Test Method for Automated Driving (NATM) Guidelines for Validating Automated Driving System (ADS)*. URL: https://2cm.es/WVie (visited on 12/26/2024).

[3] Z Fei, M. Andersson, and A. Tingberg. *Correlation of Software-in-the-Loop Simulation with Physical Testing for Autonomous Driving*. 2024. arXiv: 2406.03040.

[4] *Euro NCAP Vision 2030: A Safer Future for Mobility*. URL: https://2cm.es/WViv (visited on 12/26/2024).

[5] *EuroNCAP's Current and Future In-cabin Monitoring Systems Assessment*. 27th ESV Conference Proceedings. 2023.

[6] A. Habibovic et al. "Communicating Intent of Automated Vehicles to Pedestrians". In: *Frontiers in Psychology* 9 (Aug. 2018), p. 1336.

[7] F. Sandblom et al. "Choosing Risk Acceptance Criteria for Safe Automated Driving". In: *CARS@EDCC2024 Workshop - Critical Automotive applications: Robustness & Safety*. Apr. 2024.

[8] O. Amador, M. Aramrattana, and A. Vinel. "A Survey on Remote Operation of Road Vehicles". In: *IEEE Access* 10 (2022), pp. 130135–130154.

[9] G. Xie et al. "Sustainable Distributed Adaptive Platoon in Multi-Agent Mobile-Edge Computing Networks for Lane Reduction Scenario". In: *IEEE Transactions on Intelligent Transportation Systems* 25.11 (2024), pp. 15673–15686.

[10] C. Earnhardt et al. "Cooperative Exchange-Based Platooning Using Predicted Fuel-Optimal Operation of Heavy-Duty Vehicles". In: *IEEE Transactions on Intelligent Transportation Systems* 23.10 (2022), pp. 17312–17324.

[11] E. Pelletier and S. Brennan. "Defining the Operational Cost of En-Route Platoon Formation Scenarios". In: *2021 IEEE Vehicle Power and Propulsion Conference (VPPC)*. 2021, pp. 1–7.

[12] M. Hjälmdahl, S. Krupenia, and B. Thorslund. "Driver behaviour and driver experience of partial and fully automated truck platooning: a simulator study". In: *European Transport Research Review* 9.1, 8 (2017).

[13] M. He, W. Shi, and R. Rajkumar. "Enhancing Flagman Gesture Recognition with Virtual Augmentation". In: *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*. 2022, pp. 2926–2933.

[14] *AB Dynamics Ground Traffic Control*. URL: https://l8.nu/tJN3 (visited on 12/26/2024).

[15] *iMAR iARGUS-CMD*. URL: https://l8.nu/tJN1 (visited on 12/26/2024).

[16] *ATOS*. https://github.com/RI-SE/ATOS. 2025.

[17] A. Knauss et al. "Proving Ground Support for Automation of Testing of Active Safety Systems and Automated Vehicles". In: *Proceedings of 2017 4th International Symposium on Future Active Safety Technology towards Zero-Traffic-Accidents (FAST-zero) :* Society of Automotive Engineers of Japan, 2017.

[18] N. Katzorke et al. "Fields of interest and demands for a digital proving ground twin". en. In: *Transportation Research Interdisciplinary Perspectives* 18 (Mar. 2023), p. 100782.

[19] *ASAM Standards*. URL: https://www.asam.net/standards/ (visited on 12/26/2024).

[20] *esmini*. https://github.com/esmini/esmini. 2025.

[21] P. Baker et al. "Exploring Realism in Virtual Testing: Towards a Scalable Platform Using Open-Source Solutions for Automated Driving Systems". In: *2024 IEEE Intelligent Vehicles Symposium (IV)*. 2024, pp. 2517–2522.

[22] B. Cabrero-Daniel, Ahmed Y. Abdelkarim, and A. Broberg. "Digital Twins for Early Verification and Validation of Autonomous Driving Features: Open-source Tools and Standard Formats". In: *2024 IEEE Intelligent Vehicles Symposium (IV)*. 2024, pp. 2477–2482.

[23] *Road vehicles — Test object monitoring and control for active safety and automated/autonomous vehicle testing — Functional requirements, specifications and communication protocol*. Standard. Geneva, CH: International Organization for Standardization, Mar. 2023.

[24] Y. Zhang et al. "Perception and sensing for autonomous vehicles under adverse weather conditions: A survey". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 196 (2023), pp. 146–177.

[25] I. Doric and R. Werner. "Wassersprühsystem zur Erzeugung eines Sprühnebels für die Erprobung von Fahrerassistenzsystemen und Prüfkörper mit einem Wassersprühsystem". European pat. 102019131642. MESSRING Active Safety GmbH. 2022.

[26] N. Katzorke et al. "Regensimulationsvorrichtung". European pat. 102021004659. Mercedes-Benz Group AG.

[27] P. Eriksson et al. "Repeatable Visibility Degradation using Water Spray for AD and ADAS Testing". In: *2024 IEEE International Automated Vehicle Validation Conference (IAVVC)*. IEEE. 2024, pp. 1–6.

[28] S. Thorén et al. "Model Predictive Geofencing for Vehicle Containment". In: *Proceedings of the 2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. 2024.

[29] Internationale Organisation für Normung and Bureau international des poids et mesures, eds. *Guide to the expression of uncertainty in measurement*. eng. 1. ed., corr. and reprinted. Genève: International Organization for Standardization, 1995.

[30] Jacobo Antona-Makoshi et al. *Leveraging the Automated Mobility Partnership (AMP) to Support the Evaluation of Safety of the Intended Functionality (SOTIF) in Automated Driving Systems*. Tech. rep. 2025–01–8674. SAE Technical Paper.

[31] Gibran Ali et al. "Integrated Scenario-Based Analysis: A Data-Driven Approach to Support Automated Driving Systems Development and Safety Evaluation". In: *2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2671–2678.

[32] S. Macenski et al. "Robot Operating System 2: Design, architecture, and uses in the wild". In: *Science Robotics* 7.66 (May 2022).

[33] *Test Protocol – AEB VRU systems*. URL: https://2cm.es/ZNXU (visited on 12/26/2024).

[34] *IsoObject*. https://github.com/RI-SE/isoObject. 2025.

[35] A. Dosovitskiy et al. *CARLA: An Open Urban Driving Simulator*. 2017. arXiv: 1711.03938.

[36] J Martin Bland and Douglas G Altman. "Measuring agreement in method comparison studies". In: *Statistical Methods in Medical Research* 8.2 (1999), pp. 135–160.