Autonomous Driving Validation and Verification Using Digital Twins

Heiko Pikner[1], Mohsen Malayjerdi[1], Mauro Bellone[2], Barış Cem Baykara[1] and Raivo Sell[1]
[1]*Department of Mechanical and Industrial Engineering, Tallinn University of Technology, Tallinn, 19086 Estonia*
[2]*FinEst Smart City Centre of Excellence, Tallinn University of Technology, Tallinn, 19086 Estonia*

Keywords: Autonomous Vehicles, Validation and Verification, Modeling and Simulation, Artificial Intelligence.

Abstract: With the introduction of autonomous vehicles, there is an increasing requirement for reliable methods to validate and verify artificial intelligence components that are part of safety-critical systems. Validation and verification (V&V) in real-world physical environments is costly and unsafe. Thus, the focus is moving towards using simulation environments to perform the bulk of the V&V task through virtualization. However, the viability and usefulness of simulation is very dependent on its predictive value. This predictive value is a function of the modeling capabilities of the simulator and the ability to replicate real-world environments. This process is commonly known as building the digital twin. Digital twin construction is non-trivial because it inherently involves abstracting particular aspects from the multi-dimensional real world to build a virtual model that can have useful predictive properties in the context of the model-of-computation of the simulator. With a focus on the V&V task, this paper will review methodologies available today for the digital twinning process and its connection to the validation and verification process with an assessment of strengths/weaknesses and opportunities for future research. Furthermore, a case study involving our automated driving platforms will be discussed to show the current capabilities of digital twins connected to their physical counterparts and their operating environment.

1 INTRODUCTION

The Autonomous Vehicle (AV) industry aims to ensure system safety before mass deployment. Real-world testing would take decades to accumulate over tens of billion accident-free miles, which alone is not a reliable safety indicator (Kalra and Paddock, 2016). Among all testing methods, high-detail simulations show better performance considering cost and time (Thorn et al., 2018) (Matute-Peaspan et al., 2020). Leveraging physics engines and digital twins of real-world environments can significantly reduce testing time and cost and try any upcoming potential feature in varying operational design domains (ODDs), such as weather conditions or traffic patterns. While AI-based AV controllers are effective in real-world conditions, they may disregard physical rules, resulting in atypical decisions. As a result, the significance and complexity of validation and verification V&V of autonomous driving functionalities increases.

Verification and validation (V&V) is defined in the ISO-IEC-IEEE 24765 (ISO-IEC-IEEE, 2017), as the

> *"process of determining whether the requirements for a system or component are complete and correct, the products of each development phase fulfill the requirements (...), and the final system or component complies with specified requirements".*

It is clear from the definition in the standard that the V&V process is aimed to verify specific predefined requirements, typically described in a technical specification. However, the ISO also says that while the process of verification ensures that *the system has been built right*, the validation addresses the question of whether *the right system has been built* for the specific task.

In autonomous driving, V&V of systems with both deterministic and stochastic components poses a challenge. Deterministic systems have predictable behavior with known inputs and outputs, such as vehicle hardware and electronics. In contrast, stochastic processes, like object detection, have probabilistic outputs. In consideration of these aspects, the V&V process has to be carried out at the elementary level, in which each component is validated individually, and at an integration level, in which the V&V process is carried out to all components working together.

From a V&V standpoint, validating a stochastic

Pikner, H., Malayjerdi, M., Bellone, M., Baykara, B. and Sell, R.
Autonomous Driving Validation and Verification Using Digital Twins.
DOI: 10.5220/0012546400003702

In *Proceedings of the 10th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2024)*, pages 204-211
ISBN: 978-989-758-703-0; ISSN: 2184-495X

process means verifying its entire probability distribution. Take dice rolling as an example; you would need to roll the dice thousands of times to ensure each face appears equally. However, for complex systems like AVs, there are countless scenarios, making it impractical to physically test all outcomes. This is where digital twinning technology shines, allowing the computation of thousands of scenarios to predict system behavior. The precision of the digital twin directly impacts V&V fidelity. This paper explores recent digital twinning techniques in AVs and their distinctions from our custom platform.

2 RELATED WORK

Any industrial product, including AVs, starts its embryonic life from a Computer Aided Design (CAD) model with the goal of representing the idea, and continues to the Computer Aided Engineering (CAE) process that aims to optimize and test initial functionalities. Such a product eventually goes to the production stage in which Computer Aided Manufacturing (CAM) comes into the game, optimizing the manufacturing process. The industrial world very often confuses such processes as the digital twining process that, instead, has a fundamental difference: it represents a product as built, operating in the real world, and receiving data from it. These three characteristics are intrinsic and fundamental to defining a digital twin resembling a real product in its operational environment. CAD models represent a model as it could be, whereas digital twins represent the model as it is.

Literature in the field often refers to digital twins as an asset that improves products along their life cycle (Löcklin et al., 2020). From this point of view, it is clear that CAD-CAM models and digital twins are very different objects, but CAD models are elements of digital twins.

The definition of digital twins was introduced by NASA 2012 (Shafto et al., 2012) with the necessity of modeling as accurately as possible flight conditions for astronauts in space or other environments, then shifted to other domains, including industrial engineering and robotics (Negri et al., 2017). NASA defines a digital twin (Shafto et al., 2012) as

> *"an integrated multiphysics, multiscale simulation of a vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its corresponding flying twin".*

While the initial NASA's definition includes all the main components of digital twins, it lacks generality and new functionalities. For this reason, the definition has been updated and generalized, referring to a digital replica of a physical system able to mirror all its static and dynamic characteristics (Talkhestani et al., 2018). However, it is really when digital twins start receiving data from their physical counterparts that becomes powerful, exploiting computational capabilities to predict failures and drive update strategies. One can also see the digital twin as the feedback loop of a physical system, receiving data and thus correcting possible unexpected outcomes. In this approach, also AVs and their testing environments can be connected to their digital twins in the simulated space. Nowadays, a commercial car has an expected lifespan of about 10-15 years. These vehicles, autonomous or not, already have many software functionalities that could be improved and updated over time, keeping the same hardware components. Digital twinning allows manufacturers to continuously simulate each vehicle's behavior and receive data from their physical counterparts to verify and validate products and components, detecting possible faults in advance and releasing a fix via software update.

AV simulations, for example, in CARLA (Dosovitskiy et al., 2017), and Autoware (Kato et al., 2018), mainly use the concept of the digital twin to validate and verify the safety and performance of those vehicles. Autoware is an open-source software project for autonomous driving, while CARLA focuses on game-engine based simulation and providing assets to build environments (urban details, road users, etc.).

AWSIM (Autoware Foundation, 2022) and CARLA are simulators that were built on top of these game engines with a specific focus on automated driving. On the other side of the ocean, Baidu is also driving the sector with the Apollo [1] open-source simulation and verification platform focusing on autonomous driving with several iterations of development. A testing case of this framework can be found in (Li et al., 2023).

An example of V&V platform, the PolyVerif[2] framework is very well detailed in (Razdan et al., 2023) and (Alnaser et al., 2019). Since the verification of physical objects is costly, not scalable, and has obvious safety concerns, this platform has been developed based on simulation methods. With any form of simulation, one must directly address the nature of model abstraction, and this is typically aligned with the operational abstraction of the Device Under Test (DUT), the AV stack in our case. Overlaid on the simulation framework is the design-of-experiment (DOE)

[1] Apollo, 2022: https://github.com/ApolloAuto/apollo

[2] The Source code repository of polyVerif is available online and maintained at https://github.com/MaheshM99/PolyVerif

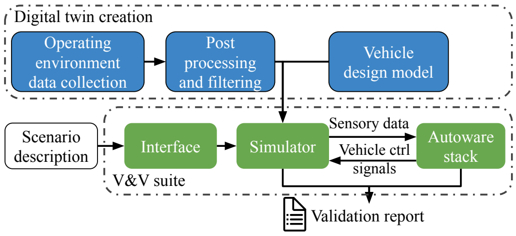

Figure 1: V&V suite workflow with digital twin, including environment and vehicles, as input to the V&V suite to provide a validation report.

unit consisting of a variety of scenarios (environment, dynamic actors) and some definition of correctness (pass/fail). The general workflow of a V&V platform is shown in Fig. 1. The framework defines an interface where the scenario definitions can be fed into the simulator. The digital twin, including a vehicle under the test and its operating environment, is a direct input to the simulator as an external loadable. It defines the environment domain and its properties, such as buildings, vegetation, road definitions, etc. The simulator runs alongside the Autoware stack to aggregate the scenario definitions within that digital twin environment, and based on the outcome, it produces validation reports. The scenario description includes the specific use case of the vehicle in the environment to be validated.

3 DOE VALIDATION FLOWS

For a serious V&V task, one must build a Design of Experiment (DOE) infrastructure that is programmatic in nature. Key elements of the DOE flow mimic the process for any large, sophisticated software project with elements. In summary, five concrete methods are provided to validate various parts of the AV stack. These flows provide researchers with an initial understanding of the framework and encourage them to build derivatives that extend the paradigm in interesting directions.

In terms of modeling abstraction, the Autoware AV stack (Kato et al., 2018) (or any AV stack) is operating in a conventional Newtonian physics universe. To be useful, any simulation environment must model key concepts such as momentum, graphic processing, sound dynamics, and more. These concepts can be modeled at various levels of fidelity with a tradeoff between accuracy and simulation performance (Malayjerdi et al., 2023b). At a component level, the internal useful abstractions of the major pieces of the Autoware AV stack are detection, control, localization, mission planning, and low-level control. Each of these modules is detailed below.

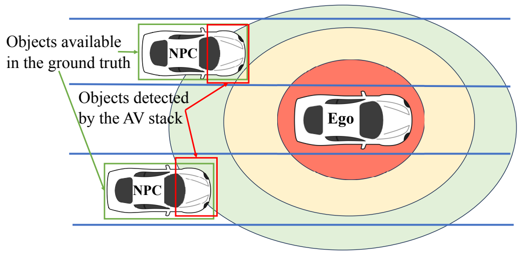

Figure 2: Detection validation example. The Ground truth of the detectable vehicle is indicated using green boxes while the detection is marked using red boxes.

Detection Validation. The V&V framework constructs detection validation by introducing stubs in the simulator to capture errors between the ground truth data and the Autoware stack detection log. This data logging is done on a per-frame basis, and the complete dataset is recorded in separate files for each of the test cases executed. Further, the framework automatically generates a figure of merit for the AV detection module performance. While generating results for object detection, below details can be considered (but not limited to):

- Frame-by-frame validation
- Report on objects detected by AV stack success and failure per object per frame.
- Distance-based accuracy report generation, as lesser distant objects are important for control. E.g. Detection success/failure rate in the range 0-10 meters, 10-20 meters, etc.

Figure 2 explains about object comparison. Green boxes are shown for objects captured by ground truth, while Red boxes are shown for objects detected by AV stack. Threshold-based rules are designed to compare the objects. It is expected to provide specific indicators of detectable vehicles in different ranges for safety and danger areas.

Control Validation. In Control Validation, the framework checks the impact of detection on the AV stacks control mechanism. This validation enables safety testing of controls like automatic braking mechanisms by computing response time and braking distance parameters. The objects ground truth is captured from the simulator while perception results are captured from the AV stack with CANBUS data, to know the control instructions from the AV stack to the CARLA simulator. V&V algorithms are written to compare data and validate the AV stacks algorithms' efficiency and accuracy. Computed Information is as below:

- Time-To-Collision $TTC_i = \frac{\Delta x}{v_{rel}}$

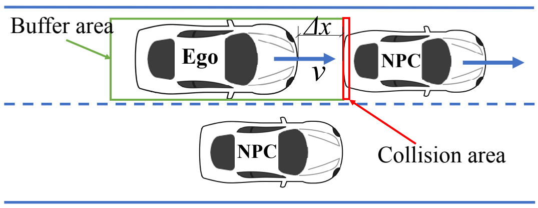

Figure 3: Time to Collisions Calculations and Collision Scenario.

- Simulator Response time on obstacle detection
- AV-stack Response time on obstacle detection
- Delay in response due to perception/detection

Figure 3 shows this concept in further detail, showing an ego-vehicle driving on the lane with other vehicles (NPCs), the time to collision is calculated using the simplest possible kinematic model using the relative speed between two vehicles.

Sufficient response time for AVs helps to return to a safer position without an imminent collision and by engaging the required braking force. Delay in response may cause collision and failure of AV systems.Computed parameters help in knowing the role of perception, their role in control initiation, and systems success/failure.

Current implementation rules consider highways and front/rear collisions from NPCs. Also, future plans are to consider all types of road infrastructures/junctions and static/pedestrian collisions from all directions.

Localization Validation. Vehicle localization failure leads to collision or accidents. Every AV stack has many inbuilt algorithms to ensure the accurate positioning of the vehicle. These algorithms use multiple sensors e.g. GPS/IMU for absolute position computation and other sensors like LIDAR/Camera/RADAR for relative position computation.

Under this validation, the V&V framework validates AV stacks localization algorithms and tests the capabilities of these algorithms in the case of GPS signal loss for a short period of time. This validation also helps in testing the localization mechanism by introducing different levels of noise into GPS/IMU sensor readings. The GPS and IMU noise can be modeled as per user requirements, and modified data can be published from the simulator to the AV stack to verify the behavior of the AV. The current validation method performs one-to-one mapping from the expected location vs. the actual location. Per frame, the vehicle position deviation value is computed and captured in the validation report. Later parameters like min/max/mean deviations are calculated from the same report.

In the validation procedure is also possible to modify the simulator to embed a mechanism to add noise in GPS/IMU data and provided the APIs to the end user. Through Python APIs, parameters can be passed to the simulator. The API internally models the noise and introduces the modified data in the simulation.

Mission Planning Validation. Each AV mission requires the capture of information from every possible sensor and the use of algorithms to move the vehicle safely to its destination based on that information. The success of the planned mission depends on the accuracy of these algorithms and the detection/perception of captured data by the sensors. Mission planning validation considers the start and goal position for the AV to navigate. Once these are set the AV generates a global trajectory based on the current location and the given destination. As shown in Fig. 4, the proposed platform validates that the trajectory is safely followed till the goal position. The validation report provides information on the trajectory following errors, collisions that have occurred, and whether the AV has reached its destination.

Low-Level Control Validation. Low-level control systems involve electronic control modules (ECUs), data networks, and mechanical actuators. In modern vehicles, there may be over 80 ECUs in some cases, therefore, validating a low-level control system requires substantial labor and effort.

Classic solutions involve recording vehicle data bus traffic for post-processing or playback. Often, data packages in networks include checksum and other security elements. Manipulating pre-stored logs and altering specific signals is only possible by recalculating the checksum for each modified data package. These packages also contain counters, so simply deleting them would result in corrupted counter values.

Building a network of physical controllers can address the package generation challenge, but creating and validating such a network is labor-intensive. Additionally, testing vehicle subsystems in this simplified manner may yield undesirable results.

The next objective is to create a simulated low-level control system model inside a digital twin. One such tool is MATLAB and Simulink software. Simulink software allows the generation of a sim-

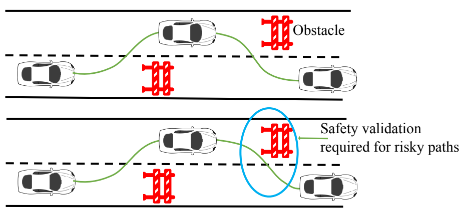

Figure 4: Trajectory validation example.

ulated low-level architecture for vehicles, including ECUs, and data buses as shown in Fig. 5. The autonomous software in ROS can generate navigation signals based on the virtual sensor data provided by the simulator. All navigation signals pass through the simulated low-level control system model and enter as actuation commands into the simulator. So, for example, the consequence of turning off the steering system model would be that the control signal from the ROS computer would no longer turn the simulated vehicle wheels.

The gateway module facilitates the connection between physical and simulated data flow. This setup enables testing stand-alone ECUs or vehicle subsystems in a hardware-in-loop (HIL) environment when a vehicle self-drives inside a simulation and simultaneously generates all the traffic on the data network.

Such a test system facilitates easy and rapid validation for developing control modules and simulating system operation. Different designed situations and disturbances allow for performing various tests. It also provides testing scenarios that would be too hazardous to conduct in real traffic scenarios. Stability and durability can be evaluated by running tests for an extended period. Furthermore, the parameters of an actual vehicle can be compared against the model, and any discrepancies between the vehicle and the digital twin in response to the same input might indicate a possible fault.

4 CASE STUDY: TESTING AN Av-shuttle

To decrease the entry barrier for researcher engagement, we provide a fully characterized AV-focused case study as a part of the V&V platform. We provide test cases by implementing an autonomous shuttle, iseAuto, in the simulated and real-world environment with the interesting premise that improvements

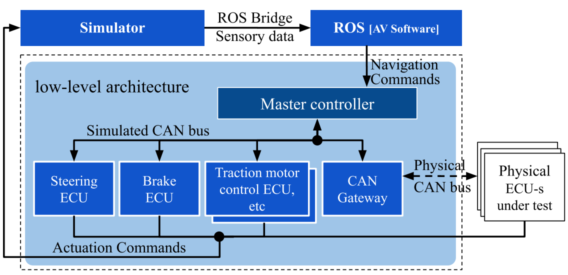

Figure 5: Low-level control system HIL simulation experimental structure. All of the vehicle's controllers are simulated, and while the simulation is running, traffic is generated on a simulated data network that can be used to test and develop physical controllers.

in Autoware or V&V can be tested in cooperation with other research groups. The iseAuto is an autonomous shuttle of Tallinn University of Technologys (TalTech) AV research group operating on the campus for experimental and study purposes. The AV shuttle and its related operating environment are connected to its digital twin, enabling running all developments first in a simulation. The simulation environments, interfaces, and concepts are described in detail in (Sell et al., 2022), and (Malayjerdi et al., 2023a).

4.1 Digital Twin of the iseAuto Shuttle

The initial design model of the iseAuto shuttle was used and constantly updated to deploy its digital twin, which is used as a DUT in any desired environment designed for testing and validation. The DUT digital twin contains the same sensor configuration as the real device and the 3D graphical model. The virtual environment also represents similar features to the actual test area; features such as urban details and vegetation are simulated within the environment. LGSVL (Rong et al., 2020) is deployed in the proposed platform as a vehicle simulator powered by the Unity game engine. This enables the creation of any desired virtual environment and the target vehicle to provide more flexibility and compliance in performing various tests. The simulator also benefits from a Python API toolkit to create different test scenarios based on pre-built features. It is also possible to import scenarios from a different platform (Malayjerdi et al., 2023a), e.g. SUMO (Behrisch et al., 2011).

To create a more complex test plan, multiple events can be included in one scenario. After running a simulation, the simulator provides virtual sensor inputs to the control algorithms provided by Autoware.ai. The raw data is received by the perception algorithms and then processed by various units. Finally, the software decides on the required actuation command and sends it back to the simulator environment. This communication is handled through a ROS bridge. Based on each study objective, various safety and performance KPIs are defined and the corresponding data is recorded during the runs. We then analyze and observe these criteria to find the vulnerabilities and corner cases where the DUT violated the metrics (Malayjerdi et al., 2023a; Roberts et al., 2023).

The data collection used in iseAuto is an end-to-end general-purpose AV data collection framework featuring algorithms for sensor calibration, information fusion, and data space to collect hours of robotics-related application that can generate data-driven models (Gu et al., 2023). The novelty of

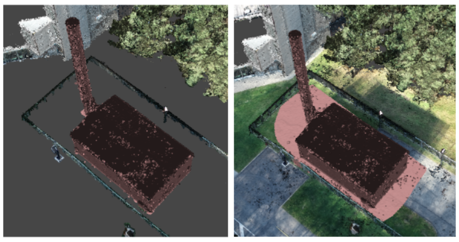

Figure 6: Comparison between point selection in segmented point-cloud and non-segmented point-cloud.

this dataset collection framework is that it covers the aspects from sensor hardware to the developed dataset that can be easily accessed and used for other AD-related research. The framework has backend data processing algorithms to fuse the camera, LiDAR, and radar sensing modalities together. Detailed hardware specifications and the procedures to build the data acquisition and processing systems can be found in (Gu et al., 2023). Data collection and update are crucial parts of the digital twin creation process, which involves several resource-demanding steps. However, it is worth mentioning that the digital map of an area can be reused in the digital twining process of several AVs or other types of robotic units as well.

The digital twin of the shuttle without its operational environment remains just a CAD model. To accurately represent the real environment in which the AV operates in a digital world (i.e., the workspace in which the AVs operate), aerial images of the environment must be collected. This can be done in various ways and with various sensors (LIDAR, RGB Camera, etc.).

In the case study proposed here, a drone with an RGB camera has been used in a grid flight path at a constant altitude to take sequential images of the environment. These images have been collected from three different angles to ensure the best possible coverage of the environments details. The images are georeferenced with a coordinate stamp by the drone acquisition system itself, the georeferencing process was supported by an RTK base station and ground markers to increase georeferencing accuracy. This makes it possible to photogrammetrically process them to obtain a point cloud of the environment. A small misalignment of the georeferenced images or unexpected glares on the camera's lens could degrade the point cloud's quality. Once the data has been collected, it goes through a photogrammetric alignment, point-cloud creation, and outlier removal process. This part is completely handled using commercially available software. This step makes it significantly easier to select and classify the point cloud and to clean it up from unwanted noise (see Fig. 6). The previously generated point clouds are then re-imported into Agisoft Metashape for classification and cleanup. It is also worth mentioning that after these processes are completed, one could easily generate buildings from this data directly in Metashape in any desired format.

4.2 TalTech iseAuto V&V

All of the steps required for the V&V process including the creation of the digital twin, scenario generation, and simulation are integrated into the simulation platform. As a primary step, an openDRIVE network map (xodr) of the target environment is needed. Figure 7 shows an example of a xodr map over the operating 3D virtual environment. In the next step, this map is used by the Scenic (Fremont et al., 2022) to generate distributed test cases all over the area. Scenic utilizes M-SDL, a human-readable, high-level scenario definition language, to describe scenarios. Several generated scenarios for a car parked in front of the AV are shown in Fig. 8. Scenic assists in the distribution of the target validation scenario over the entire operational area.

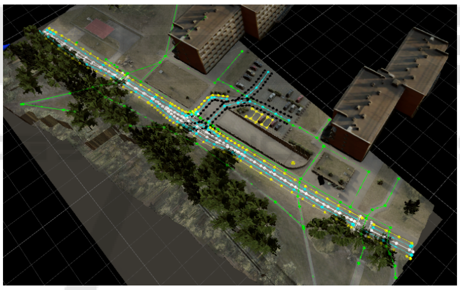

Figure 7: OpenDRIVE map over the 3D environment.

The generated scenarios are then simulated inside a high-fidelity simulator, which in this case is LGSVL. Fig. 9 displays 4 different passing scenarios generated by scenic and simulated in the simulator 3D environment.

5 DISCUSSION AND CONCLUSIONS

V&V of AV systems is a very difficult problem and there is a need to build research frameworks that can accelerate the state-of-art. However, a current limitation is that the current examples take into account AI components only in the detection module. Many research questions arise from the use of AI, for instance, AI fundamentally builds a model from data with ef-

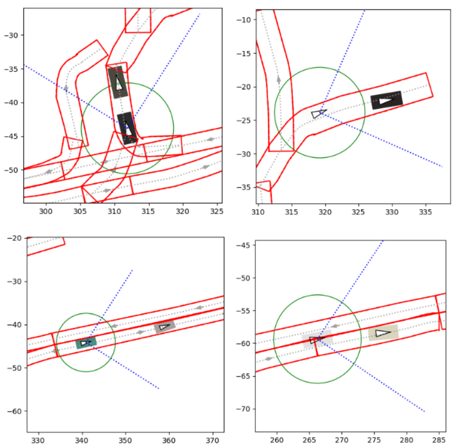

Figure 8: Scenic generates different scenarios over the xodr imported map.

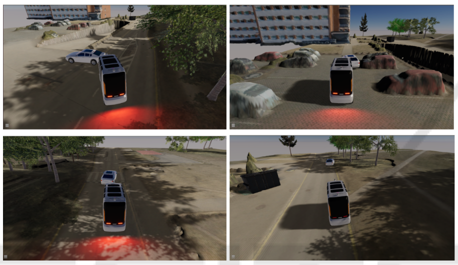

Figure 9: Scenarios generated by the scenic inside the LGSVL.

fectively an opaque lookup function for inference.

This means data in the "algorithm" does not have a deterministic outcome in the operational domain as even a slight variation might generate unexpected outcomes. How can one validate the data projected through training for conformance to the appropriate Operational Design Domain (ODD) state space and its behavioral transformations? For AI, how does one capture "expectation" functions to determine correctness when there is a lack of a system design modeling methodology? Many AI applications use AI to "discover" the highest level system transformation. The answers to the above questions lead to the computational convergence questions.

An intuition would be to build a formalization of ODD state spaces and create a method for examining the data sets under that constraint. In the AI area, the only well-established method is cross-validation, involving the swap between several train-validation sub-datasets to confirm the model performances within a specific variance threshold. While cross-validation provides a measure of the knowledge abstraction capabilities of AI modules, it does not ensure that the final model is built in compliance with any well-established standard in the area.

Research Problem 1. For AI training/inference, is there a more robust theory of convergence?

Current convergence criteria are based on loss-functions minimization and regularization methods. This means that the training stops when the minimization of the loss function does not improve anymore over time, and the best model is chosen over the best loss function value or using any early stopping criteria that measure the accuracy of the validation data. These criteria seem weak from a general knowledge abstraction point of view as validation and training datasets might be slightly different and the mathematical assurance of convergence exists only asymptotically (for the dataset size that goes to infinity).

Research Problem 2. For AI V&V, is there any theory of convergence?

The questions might seem similar at first glance, but they consider two different aspects, the training procedure of the model, and the validation procedure as the model is integrated into a product (e.g. a vehicle). Typically, V&V is exponential in terms of scenarios to consider, it is possible to use a number of techniques that employ abstraction to manage complexity but most of these techniques do not work with AI inference or work only on a limited subset of cases.

For AV in particular, further open research questions include:

- **Newtonian Physics.** Autonomy exists in the physical world. The physical world is governed by physics (Maxwell, Newton). This should be a great aid in helping set a governing framework for validation. How might one use the properties from physics to build a validation governor around AI-based autonomy systems?
- **Component Validation.** Each of the major steps in the AV pipeline (Detection, Perception, Location Services, Path Planning, etc) has its challenges. Can one build robust component-level validation for each of these?
- **Abstraction.** Complex problems are solved by the use of abstraction. Is it possible to leverage component validation such that deeper scenario validation can be done at a higher level of abstraction? If so, what are the abstractions of concern?

The field of AV and AV V&V is rich with open research problems. However, it is very difficult to make progress without a very large level of infrastructure. A cooperative open-source model is critical for progress, and the proposed platform is designed to help researchers quickly experiment with state-of-the-art ideas in this direction.

In conclusion, this paper underscores the pivotal role of digital twins in addressing validation and verification challenges associated with the principal components in AVs. Through a comprehensive review of current methodologies, this study elucidates the nuanced connection between the digital twining process and the imperative task of ensuring safety-critical systems reliability. The assessment of strengths, weaknesses, and opportunities for future research reveals the intricacies involved in constructing digital twins with high predictive value.

ACKNOWLEDGEMENTS

This work has been supported by the European Union through the H2020 project Finest Twins (grant No. 856602).

REFERENCES

Alnaser, A. J., Akbas, M. I., Sargolzaei, A., and Razdan, R. (2019). Autonomous vehicles scenario testing framework and model of computation. *SAE International Journal of Connected and Automated Vehicles*, 2(4).

Autoware Foundation (2022). TIER IV AWSIM. https://github.com/tier4/AWSIM.

Behrisch, M., Bieker, L., Erdmann, J., and Krajzewicz, D. (2011). Sumo–simulation of urban mobility: an overview. In *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation*. ThinkMind.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., and Koltun, V. (2017). Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR.

Fremont, D. J., Kim, E., Dreossi, T., Ghosh, S., Yue, X., Sangiovanni-Vincentelli, A. L., and Seshia, S. A. (2022). Scenic: A language for scenario specification and data generation. *Machine Learning*, pages 1–45.

Gu, J., Lind, A., Chhetri, T. R., Bellone, M., and Sell, R. (2023). End-to-end multimodal sensor dataset collection framework for autonomous vehicles.

ISO-IEC-IEEE (2017). Systems and software engineering -vocabulary. IEEE.

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

Kato, S., Tokunaga, S., Maruyama, Y., Maeda, S., Hirabayashi, M., Kitsukawa, Y., Monrroy, A., Ando, T., Fujii, Y., and Azumi, T. (2018). Autoware on board: Enabling autonomous vehicles with embedded systems. In *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*, pages 287–296. IEEE.

Li, H., Makkapati, V. P., Wan, L., Tomasch, E., Hoschopf, H., and Eichberger, A. (2023). Validation of automated driving function based on the apollo platform: A milestone for simulation with vehicle-in-the-loop testbed. *Vehicles*, 5(2):718–731.

Löcklin, A., Müller, M., Jung, T., Jazdi, N., White, D., and Weyrich, M. (2020). Digital twin for verification and validation of industrial automation systems–a survey. In *2020 25th IEEE international conference on emerging technologies and factory automation (ETFA)*, volume 1, pages 851–858. IEEE.

Malayjerdi, M., Goss, Q. A., Akbaş, M. İ., Sell, R., and Bellone, M. (2023a). A two-layered approach for the validation of an operational autonomous shuttle. *IEEE Access*.

Malayjerdi, M., Kaljavesi, G., Diermeyer, F., and Sell, R. (2023b). Scenario-based validation for autonomous vehicles with different fidelity levels. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6.

Matute-Peaspan, J. A., Zubizarreta-Pico, A., and Diaz-Briceno, S. E. (2020). A vehicle simulation model and automated driving features validation for low-speed high automation applications. *IEEE Transactions on Intelligent Transportation Systems*, 22(12):7772–7781.

Negri, E., Fumagalli, L., and Macchi, M. (2017). A review of the roles of digital twin in cps-based production systems. *Procedia manufacturing*, 11:939–948.

Razdan, R., Akba, M. ., Sell, R., Bellone, M., Menase, M., and Malayjerdi, M. (2023). Polyverif: An open-source environment for autonomous vehicle validation and verification research acceleration. *IEEE Access*, 11:28343–28354.

Roberts, A., Malayjerdi, M., Bellone, M., Maennel, O., and Malayjerdi, E. (2023). Analysing adversarial threats to rule-based local-planning algorithms for autonomous driving. *Network and Distributed System Security (NDSS) Symposium*.

Rong, G., Shin, B. H., Tabatabaee, H., Lu, Q., Lemke, S., Možeiko, M., Boise, E., Uhm, G., Gerow, M., Mehta, S., et al. (2020). LGSVL simulator: A high fidelity simulator for autonomous driving. In *23rd International conference on intelligent transportation systems (ITSC)*, pages 1–6. IEEE.

Sell, R., Malayjerdi, E., Malayjerdi, M., and Baykara, B. C. (2022). Safety toolkit for automated vehicle shuttle-practical implementation of digital twin. In *2022 International Conference on Connected Vehicle and Expo (ICCVE)*, pages 1–6. IEEE.

Shafto, M., Conroy, M., Doyle, R., Glaessgen, E., Kemp, C., LeMoigne, J., and Wang, L. (2012). Modeling, simulation, information technology & processing roadmap. *National Aeronautics and Space Administration*, 32(2012):1–38.

Talkhestani, B. A., Jazdi, N., Schlögl, W., and Weyrich, M. (2018). A concept in synchronization of virtual production system with real factory based on anchor-point method. *Procedia Cirp*, 67:13–17.

Thorn, E., Kimmel, S. C., Chaka, M., Hamilton, B. A., et al. (2018). A framework for automated driving system testable cases and scenarios. Technical report, United States. Department of Transportation. National Highway Traffic Safety .