

[illegible]

## Document Control

### Distribution List

The following list of people will receive a copy of this document every time a new version of this document becomes available:

Teaching assistants:

Clay Pate

Customer(s):

M. I. Akbas

AVVC

Project team members:

Isabella Acosta

Serena Conticello

Hannah Ramsden

William Reimer

Davian Rosario-Ortiz

### Change Summary

The following table details changes made between versions of this document:

Version	Date	Modifier	Description
0.1	12-5-24	Serena Conticello	Initialization of document and base information
0.2	12-5-24	All team members	Adding information and finalizing submission.

# Table of Contents

Document Control .....	1
Distribution List.....	2
Change Summary .....	2
1. Introduction .....	3
1.1. Purpose.....	4
1.2. Scope .....	4
1.3. System Overview .....	4
1.4. Testing Approach Overview .....	4
1.5. Testing Entrance Criteria .....	4
1.6. Document Overview .....	4
1.7. References.....	4
2. Testing Approach .....	4
2.1. Testing Types .....	4
2.1.1. Usability Testing .....	4
2.1.2. Functionality Testing .....	4
2.2. Testing Suspension Criteria and Resumption Requirements .....	4
2.2.1. Suspension Criteria.....	4
2.2.2. Resumption Requirements.....	4
2.3. Testing Environment.....	4
2.4. Testing Assumptions.....	4
The assumptions that can be made are that to download PolyVerif must be connected to Wi-Fi, ensuring a successful install. It must also be connected to Wi-Fi to run the simulation. The user has a list of test cases that are pre-determined, as well as adding their test simulations (the goal of our project). These assumptions will ensure that our project runs smoothly and produces accurate test cases. ....	4
2.5. Testing Risks and Contingencies .....	4
2.6. Test Plan .....	4
3. Test Schedule .....	4
4. Traceability Matrix and Defect Tracking.....	4
4.1. Traceability Matrix .....	4
4.2. Defect Severity Definitions.....	4
5. Test Cases .....	4
5.1. Test Case <N>.....	4



# 1. Introduction

## 1.1. Purpose

The purpose of this System Test Plan (STP) is to define the testing strategy for evaluating the PolyVerif simulation platform used in autonomous vehicle testing. This document outlines the testing approach, responsibilities, schedule, and success criteria to ensure the system meets functional, technical, and performance requirements.

## 1.2. Scope

PolyVerif provides a flexible simulation environment for autonomous vehicle validation and verification, supporting a wide range of scenarios with customizable parameters such as traffic conditions and environmental factors. This test plan focuses on assessing the system's ability to simulate real-world conditions accurately, identifying potential issues early in the development process to improve safety and reliability.

## 1.3. System Overview

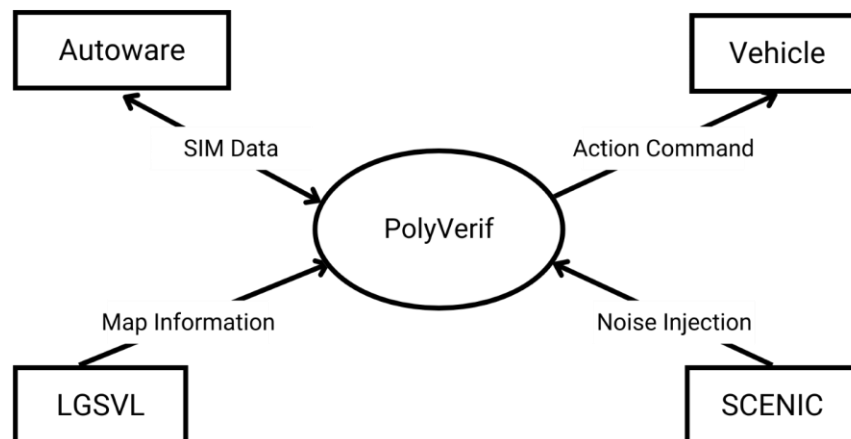


Figure 1. System context diagram

## 1.4. Testing Approach Overview

Our testing approach is currently still in the process of being developed since PolyVerif is not working. We will properly implement these test scenarios as well as the test case as soon as Polyverif is fully functional.

### 1.5. Testing Entrance Criteria

Before testing, it must be ensured that PolyVerif is installed successfully and all systems required to run PolyVerif are integrated correctly. There must be test cases within the interface (already pre-determined within the interface) and ensure that the testing environment for these test cases is set up properly to handle all the parameters needed. We must have the correct data test preparation and documentation needed to run these tests. Once all of these requirements have been met, we can then begin the testing process for our project.

### 1.6. Document Overview

This document goes over the execution of test cases and the overall testing of our project.

### 1.7. References

- [1] R. Razdan, Mustafa İlhan Akbaş, R. Sell, M. Bellone, Mahesh Menase, and Mohsen Malayjerdi, "PolyVerif: An Open-Source Environment for Autonomous Vehicle Validation and Verification Research Acceleration," *IEEE Access*, vol. 11, pp. 28343–28354, Jan. 2023, doi: <https://doi.org/10.1109/access.2023.3258681>.
- [2] M. I. Akbaş, M. Menase, S. Verma, and R. Razdan, "PolyFlows: Modular Test Framework Design for Autonomous Vehicles," *2024 IEEE International Conference on Mobility, Operations, Services and Technologies (MOST)*, pp. 50–59, May 2024, doi: <https://doi.org/10.1109/most60774.2024.00014>.
- [3] M. I. Akbas, "Testing and Validation Framework for Autonomous Aerial Vehicles," *Journal of Aviation/Aerospace Education & Research*, Jan. 2021, doi: <https://doi.org/10.15394/jaaer.2021.1849>.
- [4] Q. Goss, W. C. Pate, and Mustafa İlhan Akbaş, "An Integrated Scenario-Based Testing and Explanation Framework for Autonomous Vehicles," May 2024, doi: <https://doi.org/10.1109/most60774.2024.00015>.
- [5] D. J. Fremont *et al.*, "Scenic: a language for scenario specification and data generation," *Machine Learning*, vol. 112, Feb. 2022, doi: <https://doi.org/10.1007/s10994-021-06120-5>.
- [6] "Eclipse SUMO - Simulation of Urban MObility," *Eclipse SUMO - Simulation of Urban MObility*. <https://eclipse.dev/sumo/>

- [7] Geeks, G. for. (2024, June 19). *Verification and validation in software engineering*. GeeksforGeeks. <https://www.geeksforgeeks.org/software-engineering-verification-and-validation/>
- [8] TWI. (2024). *What is an autonomous vehicle?* <https://www.twi-global.com/technical-knowledge/faqs/what-is-an-autonomous-vehicle>
- [9] *Autonomous Vehicle Verification Consortium*. (n.d.). <https://www.avvc.net/>

## 2. Testing Approach

### 2.1. Testing Types

#### 2.1.1. Usability Testing

The PolyVerif Suite is designed for an intuitive experience for the user to navigate PolyVerif effectively and collect data from their simulations. Thus, the following tests will occur to ensure usability

- Validation Suite
  - Validation Selection: The user can select one of four different validation options. Detection Validation, Control Validation, Localization Validation, and Mission Planning Validation. No other options are available at this time; a validation option must be selected to enable the next functionalities as described below.
  - Scene Selection: Several tests will be run when a Validation Suite option is selected. The only functional option that can be selected are the default scenarios installed with PolyVerif and the ERAU campus. Additionally, Stop Scenario, AVP, and Stop ADE should be able to be selected, but at this stage in the test these buttons when clicked should do nothing. Additionally at this stage, the Show Report Button can be clicked, and a Report will be generated, but the generated variables will be N/A
  - Script Selection: The Select Scripts pane will be accessible to the user when the Scene Selection pane has met a successful condition (by selecting a scenario). When accessible, the user can click on the *Test Cases* option. If successful, a sub-option is presented to the user (number of sub-options are subject to change based on the progress of test case development and implementation) When selected, the user is presented below with several specific test cases based on their previous selections. This selection also allows for the option to select several different buttons: *Start ADE*, *Select Scenario*, *Set Count*, *Run Scenic*, *Run Scenario*, *Initial Pos*, *Goal Pos*.
  - Starting Scenario: There are several options that can be selected before running the scenario. Selecting *Initial Pos* will include the starting point of the vehicle in the simulation report, and Selecting *Goal Pos* will include the Goal Position for the vehicle to reach, and if the Vehicle reached that position. Additionally, before starting the scenario, the *Start ADE* button must first be pressed, as this initializes the simulation software. Not doing so will cause the scenario to fail to run. When the desired options are selected, the user will select *Run Scenario* which will immediately cause the simulation to load and play.

#### 2.1.2. Functionality Testing

The functionality test for PolyVerif ensures that the system performs as expected by validating key features such as creating and executing simulation scenarios (e.g., lane changes, collisions), customizing simulation parameters (e.g., vehicle specifications, sensors,



environmental factors), and integrating external models (e.g., 3D maps, traffic data). It also tests sensor and perception accuracy, error handling, and logging of simulation results. Additionally, the test assesses system performance under high loads, ensuring stability when handling complex, large-scale simulations. Overall, the goal is to confirm that PolyVerif meets its functional requirements and provides reliable, accurate results for autonomous vehicle testing and validation.

## **2.2. Testing Suspension Criteria and Resumption Requirements**

### **2.2.1. Suspension Criteria**

The testing within this interface will be suspended if there is a critical incident within the software, if the test environment is subject to change, if the requirements need to be changed for a specific simulation, or if the results are inconclusive based on the test scenario. We would then troubleshoot these specifications and try to figure out how to fix/improve upon them before continuing with our testing. Our simulation is currently not in a working stable condition but if any of the above scenarios were to occur once it is stable, we would proceed accordingly with the troubleshooting and continue working through the issues.

### **2.2.2. Resumption Requirements**

Our testing will be executed once PolyVerif is in a stable and working condition to produce viable and accurate test cases.

## **2.3. Testing Environment**

Our testing environment is based within the PolyVerif interface. The PolyVerif interface tests the verification and validation of autonomous vehicles through multiple testing scenarios. These scenarios have custom environments, parameters, and traffic patterns needed to gain accurate and reliable testing results. Our test cases will be executed once the program is in a stable condition to produce test scenarios and test cases.

## **2.4. Testing Assumptions**

The assumptions that can be made are that to download PolyVerif must be connected to Wi-Fi, ensuring a successful install. It must also be connected to Wi-Fi to run the simulation. The user has a list of test cases that are pre-determined, as well as adding their test simulations (the goal of our project). These assumptions will ensure that our project runs smoothly and produces accurate test cases.

## **2.5. Testing Risks and Contingencies**

The CPU that is being used to run PolyVerif must have an 8-minimum core. The operating system is running Ubuntu 20.04 64-bit and Python must be running on version 3.8. The GPU must be running NVIDIA GTX 1080 (8 GB or higher). The computer that is being used to run the simulation is in the Micaplex in the MP 224 WiDe Lab. It is required to have ID badge access to enter the Micaplex as well as ID badge access to enter the MP 224 WiDe Lab. Login credentials are also needed to login into the computer that the simulation is being run on. The computer must be connected to the Wi-Fi network and have all the required programs and their versions (listed above) on it to run a successful simulation. These contingencies would cause our interface to not work correctly, causing us to not be able to produce any viable test cases.

## 2.6. Test Plan

These are theoretical test cases that we are planning to implement once PolyVerif is fully functional and is able to produce test cases.

**Table 1: Test Plan**

ID	Test	Status <sup>1</sup>	Date	Notes
001	<p>Running detection validation simulation with ERAU campus as environment:</p> <p>This test is designed to test the functionality of the 3D environment of our product. The user shall begin the test when PolyVerif Suite opens, and makes the following choices (specifics are subject to change)</p> <p>Detection Validation ERAUCampus Test Cases Test3DEnvironment</p> <p>*The environment selected is made simply to ensure that the ERAU campus map functions properly. No additional parameters are assigned to this scenario</p>	TBD	TBD	TBD
002	<p>Running control validation simulation with ERAU campus as environment and making a turn at an intersection as a scenario:</p> <p>This test is designed to test the functionality of the 3D environment of our product, and the functionality of the scenario creation and data gathering. The user shall begin the test when PolyVerif Suite opens, and makes the following choices (specifics are subject to change)</p> <p>Control Validation ERAUCampus Test Cases IntersectionTurn Run Scenario *Wait for simulation to end Stop Scenario Generate Report Check if the report reflects vehicle actions in the simulation</p>	TBD	TBD	TBD

<sup>1</sup> Unwritten, Incomplete, Pass, Fail

	<<tests are given in Section 5 and later>>			
--	--	--	--	--

### **3. Test Schedule**

PolyVerif is currently not in a working or stable condition, which is why we are unable to provide a testing schedule. Next semester we plan to get it functioning as soon as possible so we can begin creating test scenarios, performing tests, and generating test cases to provide results.

## **4. Traceability Matrix and Defect Tracking**

### **4.1. Traceability Matrix**

PolyVerif is currently not operational, so these processes are not applicable to our project. Once PolyVerif is functional, we will be able to implement these processes and make sure that they work within our project.

### **4.2. Defect Severity Definitions**

PolyVerif is currently not operational, so these processes are not applicable to our project. Once PolyVerif is functional, we will be able to implement these processes and make sure that they work within our project.

## **5. Test Cases**

This section is currently not applicable to our project since PolyVerif is not operational. Once it is fully functional, we will be able to integrate our test cases into our simulations and produce accurate results.

<< Describe each test as shown below. >>

### 5.1. Test Case <N>

**Objective:** << Define the objective of Test <N>. >>

**Notes:** <<This area provides general notes concerning the test procedure. Such notes might include comments on how to execute the test procedure, an estimate of the test duration, the requirements of the procedure tests, or a statement of resources needed for this test.>>

Test No.: << Unique test ID >>		Current Status: << Passed / Failed / Pending >>		
Test title: <<This line contains the long title of the test procedure. This title should come from the test plan.>>				
Testing approach: <<Define the strategy that is being used on this test.>>				
STEP	OPERATOR ACTION	PURPOSE	EXEPECTED RESULTS	COMMENTS
N	Describe the actions taken by the person executing the test procedure. Include the test suite, or the name of the test file (in this case, the contents of the file could be given in the appendix).	Describe the reason for the step.	Describe the expected response of the system being tested to the action specified under OPERATOR ACTION. This should be derived from the SRS and SDD.	
Concluding Remarks: << Filled in by the person who completed the test >>				
Testing Team: << List members of testing team and lead >>		Date Completed:		