

# POLY-VERIFICATION USER GUIDE

This document contains the information on how to run the simulation through the Poly-Verification Suit. Once the installation and setup is completed you are ready to use the PolyVerif framework.

If you have not setup and installed the pre-requisites then for reference please follow **Setup\_And\_Installation** document.

This framework uses predefined metrics for the validation of the stacks which will be calculated after running the test cases. On the basis of metrics threshold the success/failure of the stack is validated. Any user can set their own values by modifying the config.ini file in **adehome/Poly\_Suite/config.ini**.

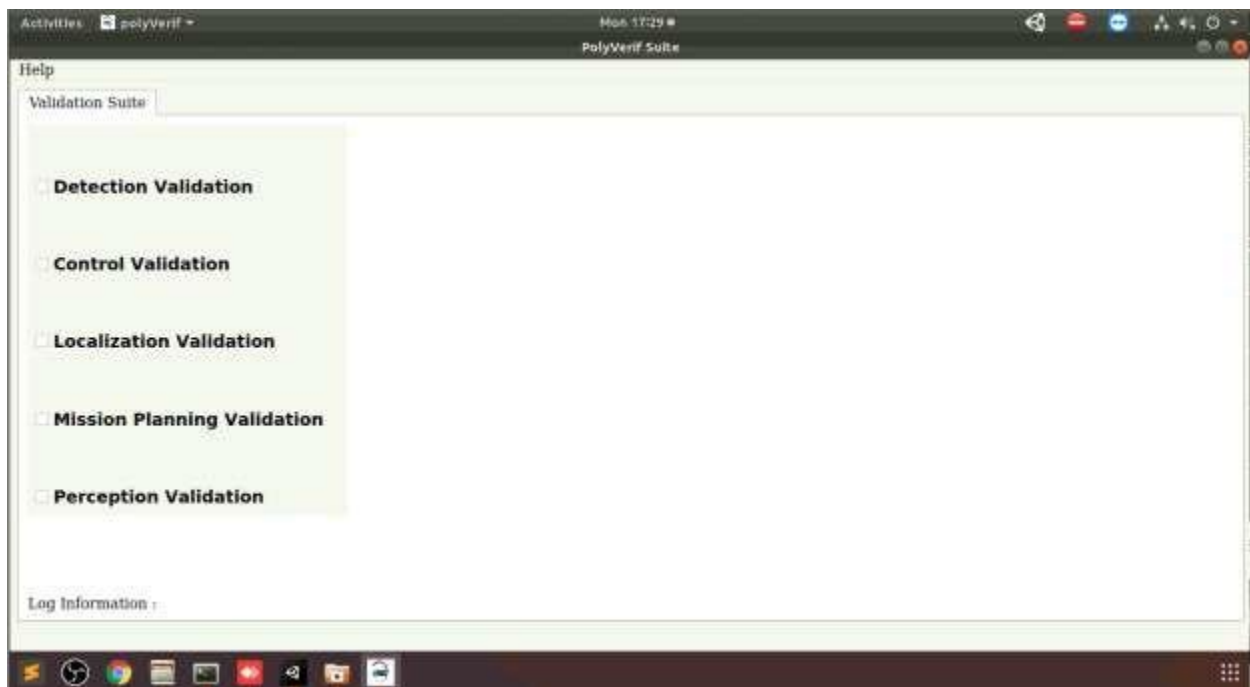
## **Below is the config.ini file parameters-**

```
[autonomous_stack_config]
  detection_max_threshold=40      #MAX value for detection
validation
  detection_min_threshold=30      #MIN value for detection
validation
  control_collion_count=0 #Collision count
  localize_max_threshold=5 #MAX value for localization
validation
  localize_min_threshold=1 #MAX value for localization
validation
  planner_goalpose_max_deviation=5 #MAX value of goal
position deviation
  planner_goalpose_min_deviation=2 #MIN value of goal
position deviation
```

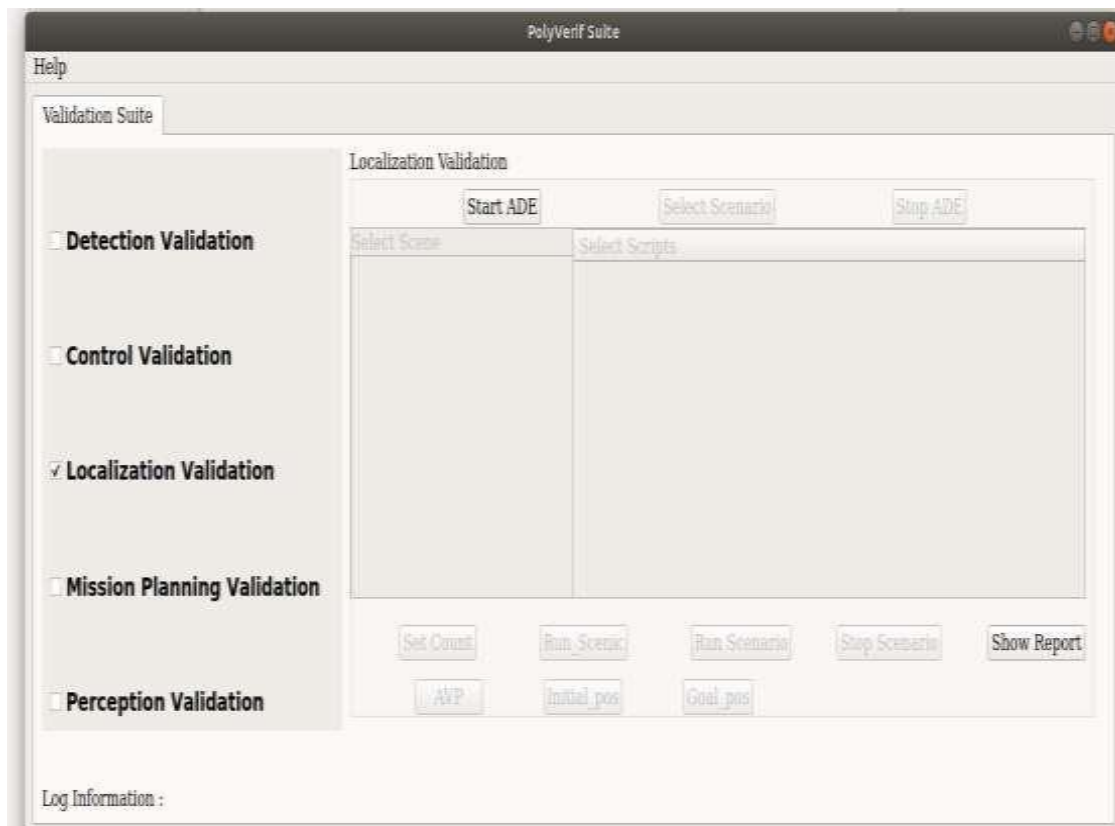
**Follow the steps provided below-**

- Go to the Poly\_Suite directory and run  
**\$ ./polyVerif**
- It will start the polyVerif framework
  - As of now four validations are working-
    - Detection Validation
    - Control Validation
    - Localization Validation
    - Mission Planning Validation

Click any one of them-

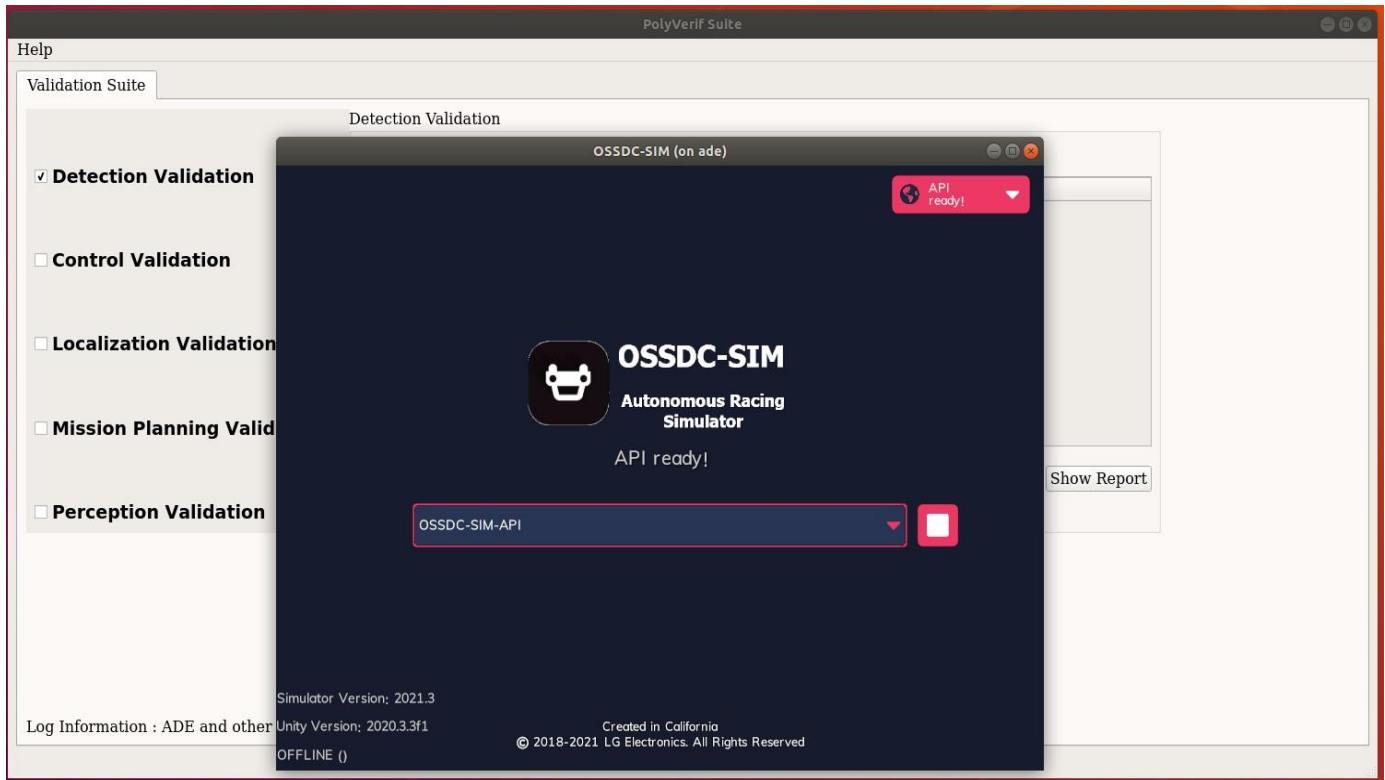


- Detection Validation is for detecting the vehicle from perception stack.
  - Click on the **Start ADE** button, it will start all the required modules.
    - AutowareAuto
    - Perception Stack
    - OSSDC simulator
    - Rviz
    - Ros2-lgsvl-bridge

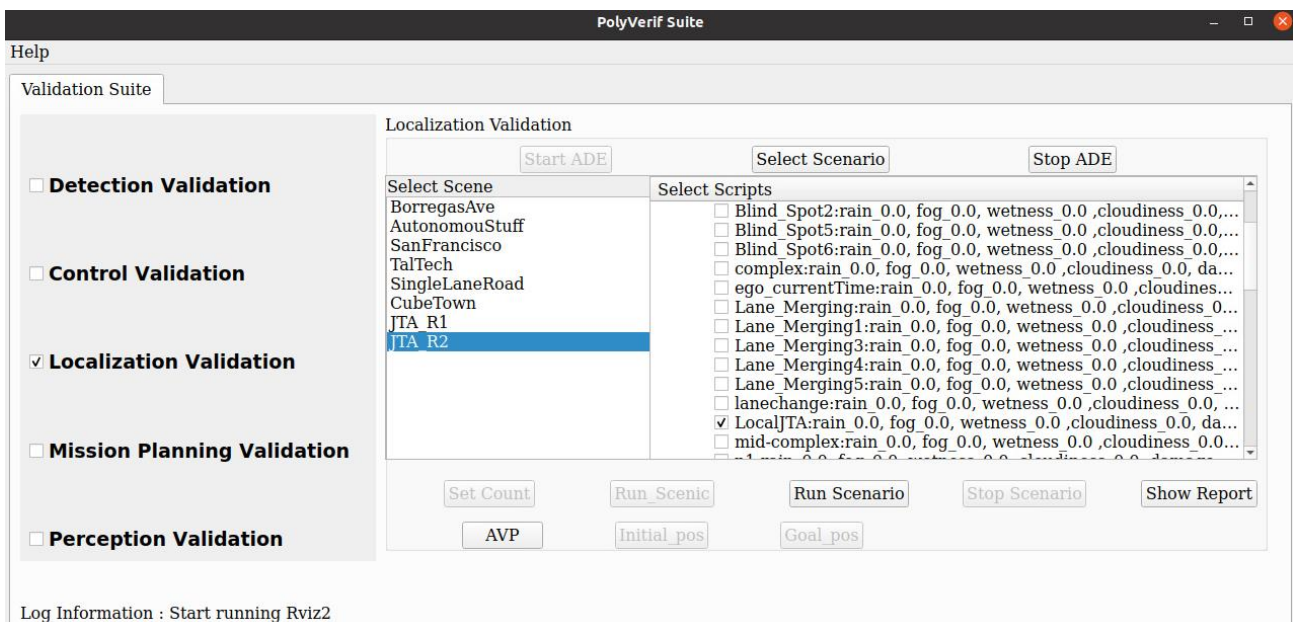


It will take some time to start the ADE docker and other components, please wait for a while.

- Now the simulator is in the **API\_Mode**



- Now click on the **Select Scenario** button
  - Select **JTA\_R2** map from the **Select Scene**
  - Select **LocalJTA** scenario from the **Select Scripts** list



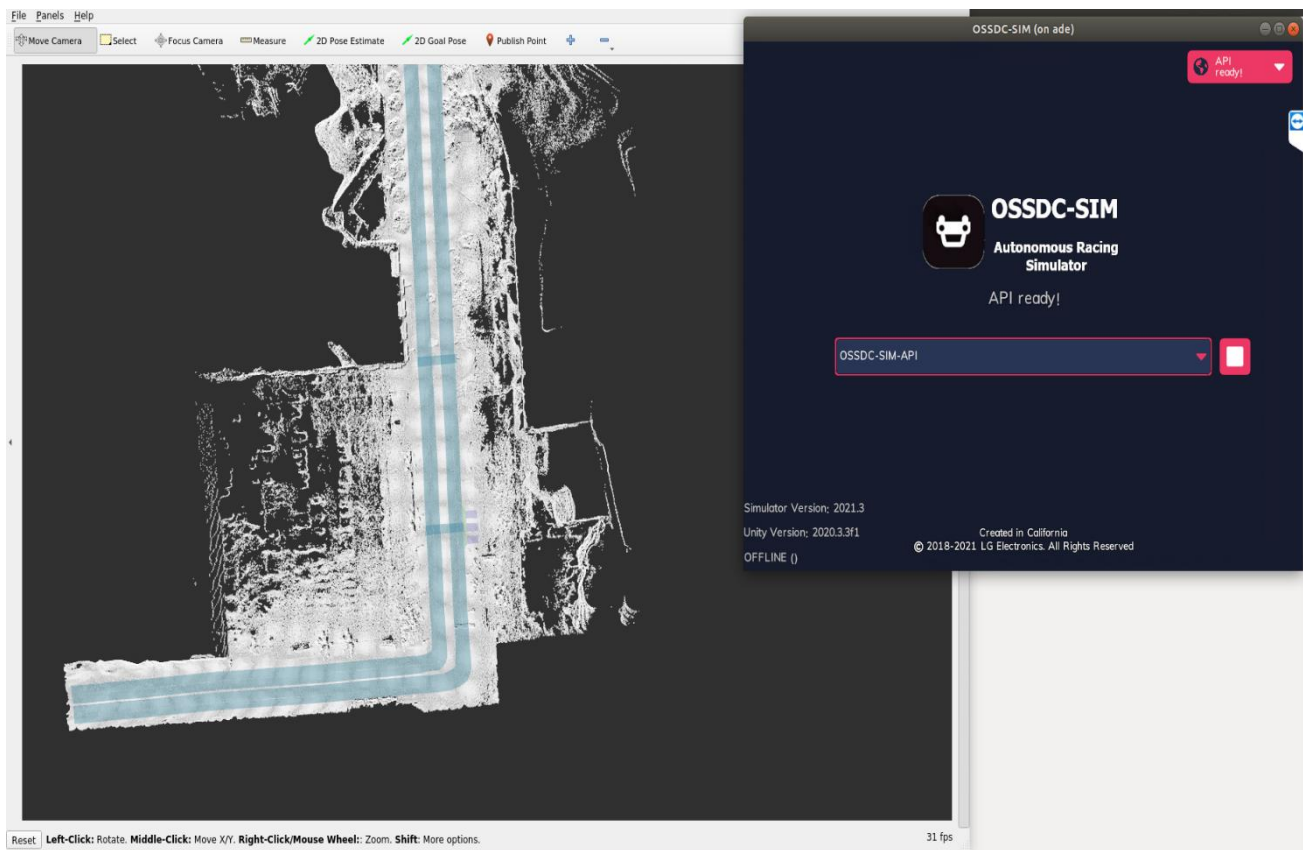
- Once you select script from the list it enables the respective button to run.
- Now click on the AVP button which will start the autoware\_auto\_avp demo.

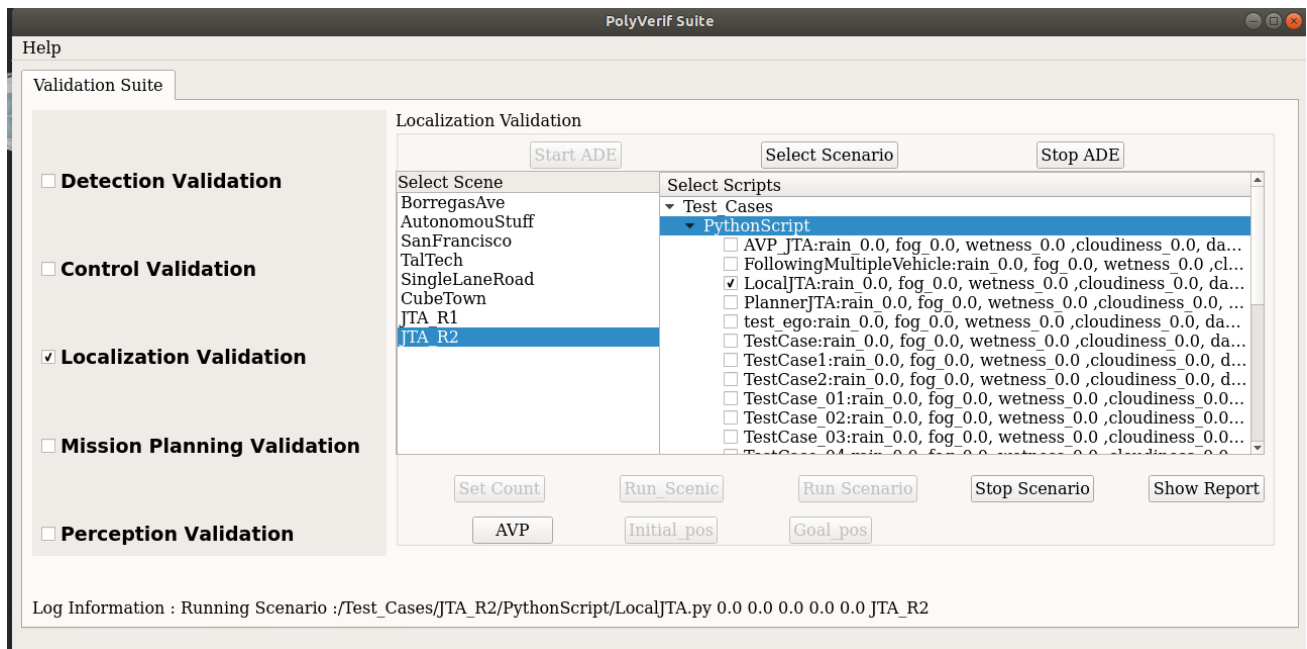
We have modified the demo to run with JTA\_R2 map. The link provided below is demo of AutowareAuto–

<https://autowarefoundation.gitlab.io/autoware.auto/AutowareAuto/avpdemo.html>

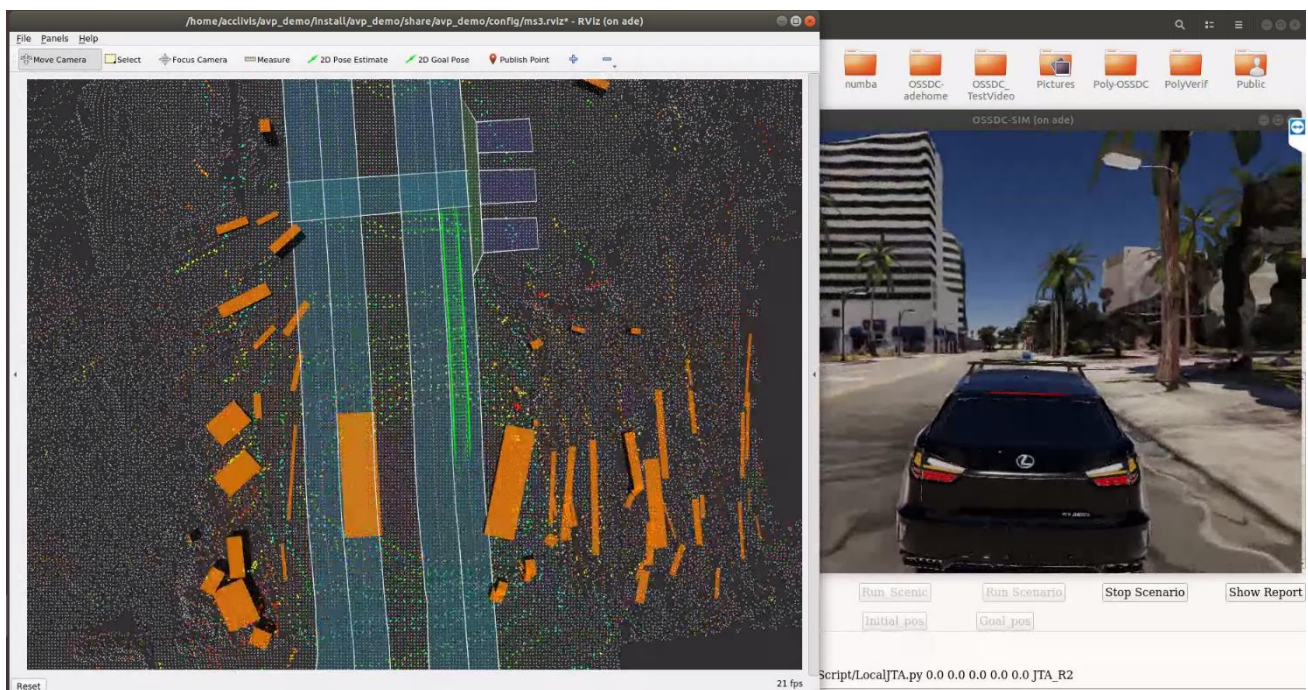
- Once the rviz starts then click on the **Run Scenario** button followed by the **Initial pos** and **Goal pos** button, it will start the simulation in OSSDC simulator which will be controlled by the AutowareAuto decisions.

**Note:** - Initial pos and goal pos values are static which is set specifically for the JTA\_R2 map. If you try these values in a different map then it will not be accurate.



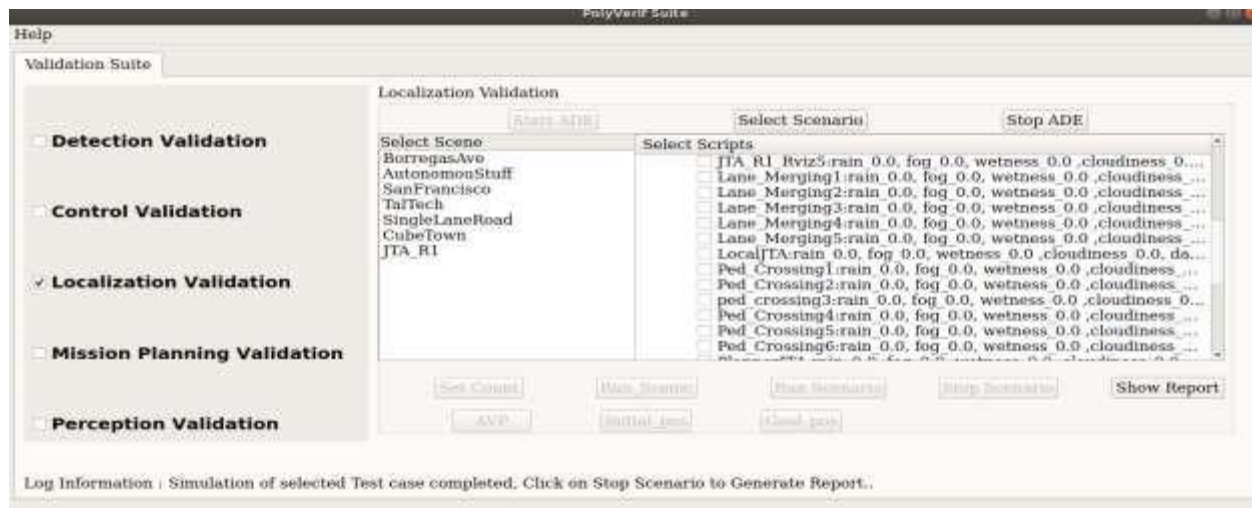


- Below is the screenshot of trajectory followed from the AutwareAuto.

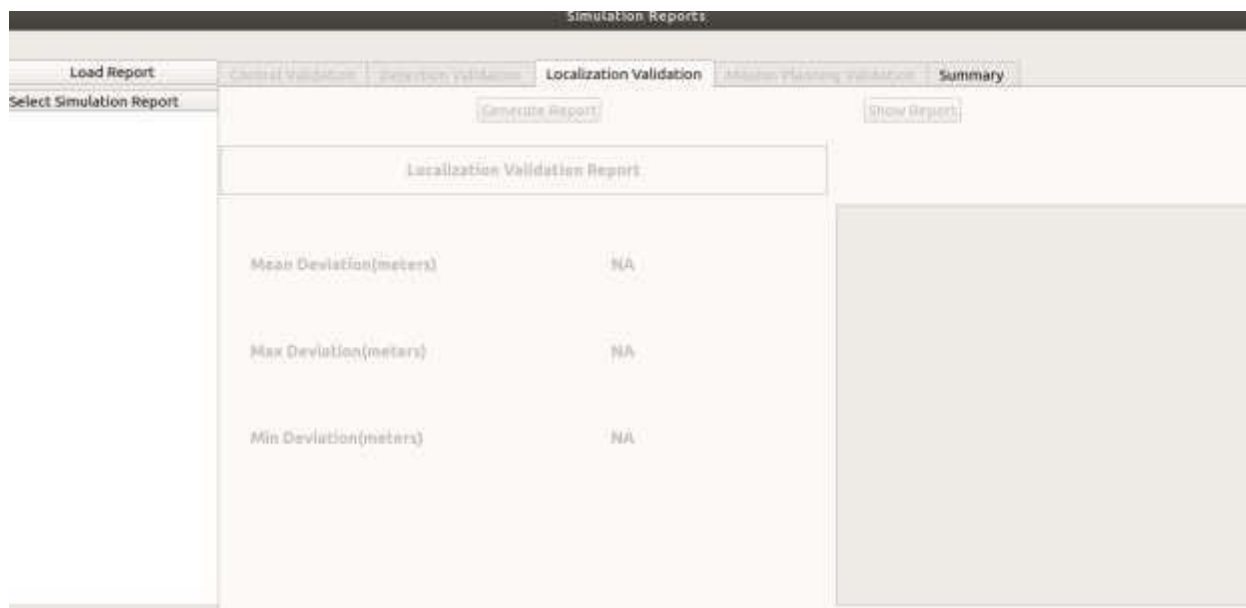




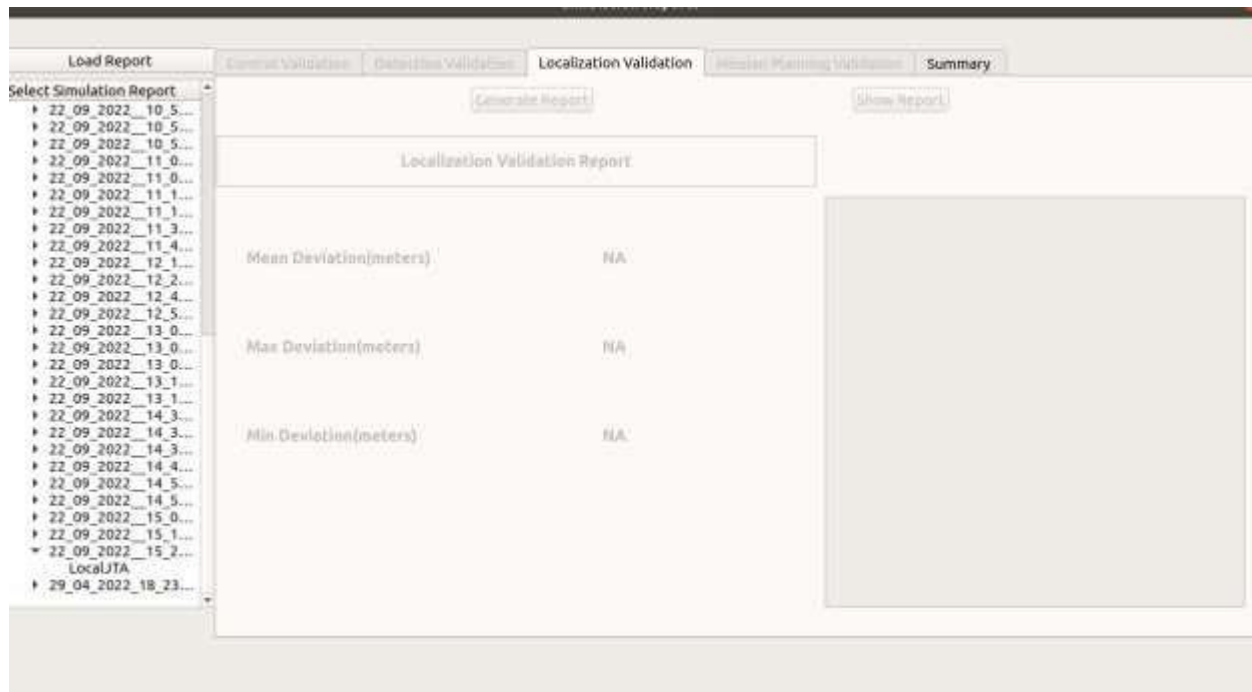
- Once the simulation is completed or if you want to stop the simulation. Click on the **Stop Scenario** button and click on the **Show Report** button. It will redirect to another form where you can see the respective simulation reports.



- This form contains reports of the respective simulation. Since we ran **Localization Validation** so it will generate and show the **Localization report**.
- Click on the **Load Report** button, it will list all the simulation reports with date and time.



- The last completed simulation report is set at the end of the list. Select and click on the **Generate Report** button. It will take some time to generate the report. Once report is generated. It will enable the **Show Report** button. By clicking on **Show Report** button you will see few respective parameters on the UI.





### Assumptions/Issues:

- AVP demo is tested with test cases (LocalJTA and PlannerJTA) for the JTA\_R2 map .
- You can modify Test Cases (Python and scenic script) as per your map environment.
- If there is only ego vehicle in the simulation then data will not compute.
- Sometimes rviz crashes but the perception stack is running in the back ground
- While running scenario using scenic, it hangs while connecting to Ros2 Bridge. So you need to forcefully terminate using Ctrl+C and start again.
- Offset for lanelet map should be set correctly in order to run AVP demo. Otherwise map will not load successfully

### References:

- OSSDC Simulator - <https://github.com/OSSDC/OSSDCSIM/releases/tag/OSSDC-SIM-v1.1>
- Scenic-[https://scenic-lang.readthedocs.io/en/latest/syntax\\_guide.html?highlight=facing#specifiers](https://scenic-lang.readthedocs.io/en/latest/syntax_guide.html?highlight=facing#specifiers)
- PythonAPI -<https://github.com/OSSDC/PythonAPI>
- AutowareAuto avp demo - <https://autowarefoundation.gitlab.io/autoware.auto/AutowareAuto/avpdemo.html>