

Polyverif Setup And Installation guide

Below you can find instructions on how to setup the self-driving demo of AWSIM simulation controlled by Autoware. The instruction assumes using the Ubuntu OS.

Required PC Specs:

Please make sure that your machine meets the following requirements in order to run the simulation correctly:

OS	Ubuntu 22.04 jammy 64-bit
CPU	11th Gen Intel® Core™ i7-11700K @ 3.60GHz × 16
GPU	NVIDIA GeForce RTX 4070
Nvidia Driver (Ubuntu 22)	535.161.07

Localhost settings:

The simulation is based on the appropriate network setting, which allows for trouble-free communication of the AWSIM simulation with the Autoware software. To apply required localhost settings please add the following lines to ~/.bashrc file

```
if [ ! -e /tmp/cycloneDDS_configured ]; then sudo sysctl -w
net.core.rmem_max=2147483647 sudo ip link set lo multicast on touch
/tmp/cycloneDDS_configured
fi
```

and these lines to ~/.profile or in either of files: ~/.bash_profile or ~/.bash_login:

```
export ROS_LOCALHOST_ONLY=1
export RMW_IMPLEMENTATION=rmw_cyclonedds_cpp
```

Note: A system restart is required for these changes to work.

1.Download and Run AWSIM Demo Binary

a. Install Vulkan Graphics Library

```
$ sudo apt update  
$ sudo apt install libvulkan1
```

b. In order to configure and run the PolyVerif framework Run following cmd's

```
$ mkdir $HOME/adehome/ && git clone --branch PolyVerif-AWSIM --single-branch https://github.com/MaheshM99/PolyVerif.git $HOME/adehome/  
$ wget --directory-prefix="$HOME/adehome" --content-disposition 'https://drive.usercontent.google.com/download?id=17p4Zne7auBrYpiz1TOMvHVQv3gagt-7E&confirm=t&uuid=a8b3a20c-7bf4-4f3e-9a1d-e1713d611152' && unzip "$HOME/adehome/AWSIM_v1.2.0.zip" -d "$HOME/adehome/"
```

c. Launch AWSIM

```
$ ./adehome/AWSIM_v1.2.0/sim.x86_64
```



2. Launching Autoware

In order to configure and run the Autoware software with the AWSIM demo, please:

1.Download map files (pcd, osm) in home and unzip them.

https://github.com/tier4/AWSIM/releases/download/v1.1.0/nishishinjuku_autoware_map.zip

2.Clone [Autoware](https://github.com/autowarefoundation/autoware.git) in home directory and move to the directory.

```
$ git clone https://github.com/autowarefoundation/autoware.git  
$ cd autoware
```

3.Switch branch to awsim-stable

```
$ git checkout awsim-stable
```

4.Configure the environment

```
$ ./setup-dev-env.sh
```

5.Create the src directory and clone external dependent repositories into it

```
$ mkdir src  
$ vcs import src < autoware.repos
```

6.Install dependent ROS packages

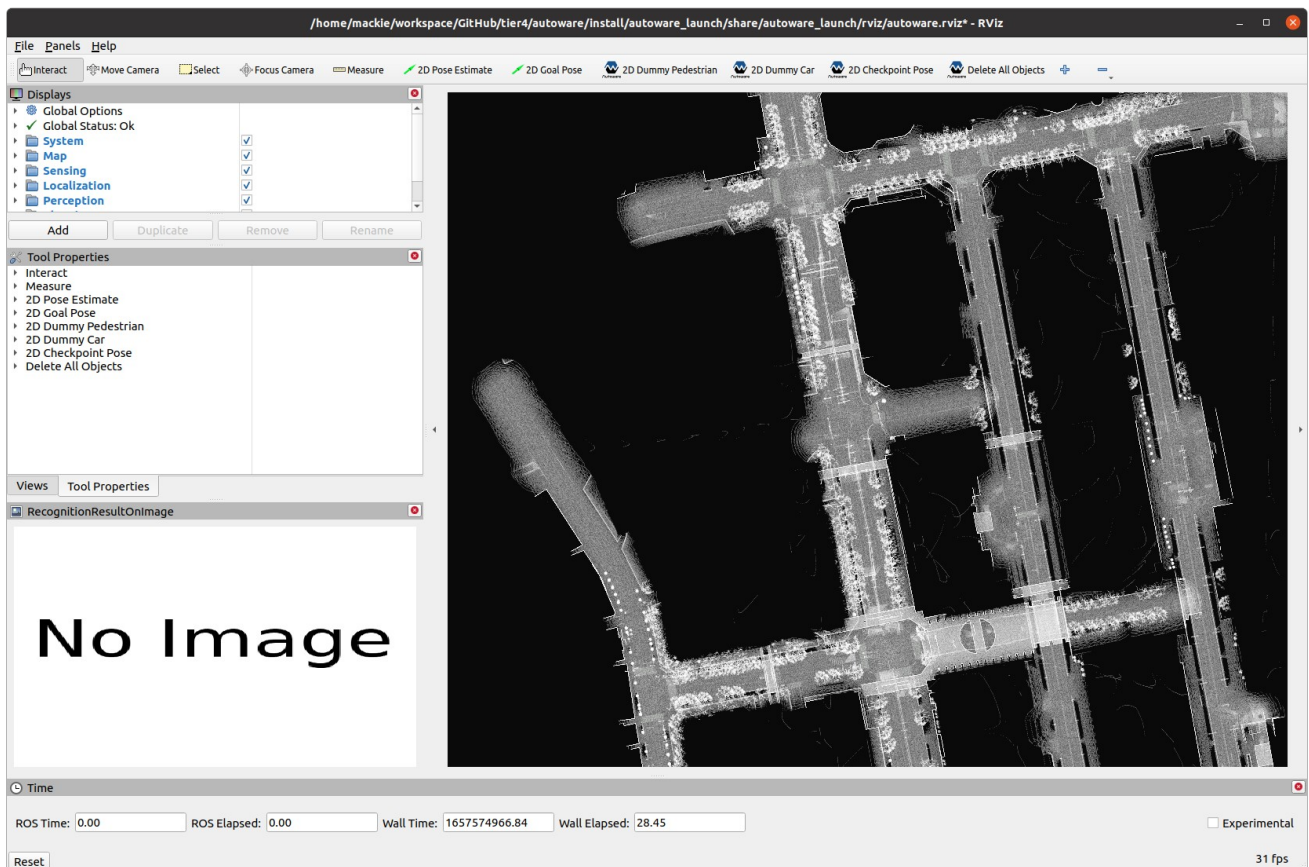
```
$ source /opt/ros/humble/setup.bash  
$ rosdep update  
$ rosdep install -y --from-paths src --ignore-src --rosdistro $ROS_DISTRO
```

7. Build the workspace

```
$ colcon build --symlink-install --cmake-args -DCMAKE_BUILD_TYPE=Release  
-DCMAKE_CXX_FLAGS="-w"
```

8. Launch Autoware.

```
$ source install/setup.bash  
$ ros2 launch autoware_launch e2e_simulator.launch.xml  
vehicle_model:=sample_vehicle sensor_model:=awsim_sensor_kit  
map_path:=<your mapfile location>
```

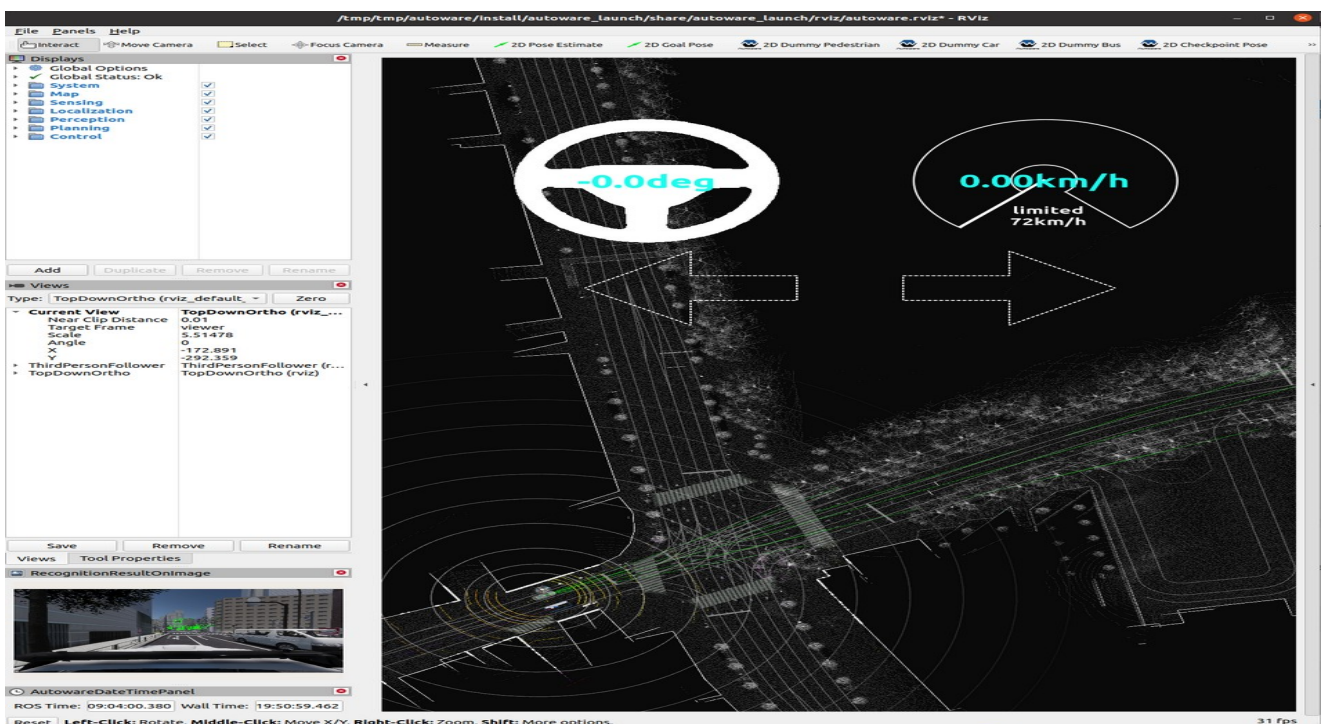


3 Let's run the self-Driving simulation

1. Launch AWSIM and Autoware according to the steps described earlier in this document.



2. The Autoware will automatically set its pose estimation as presented below.

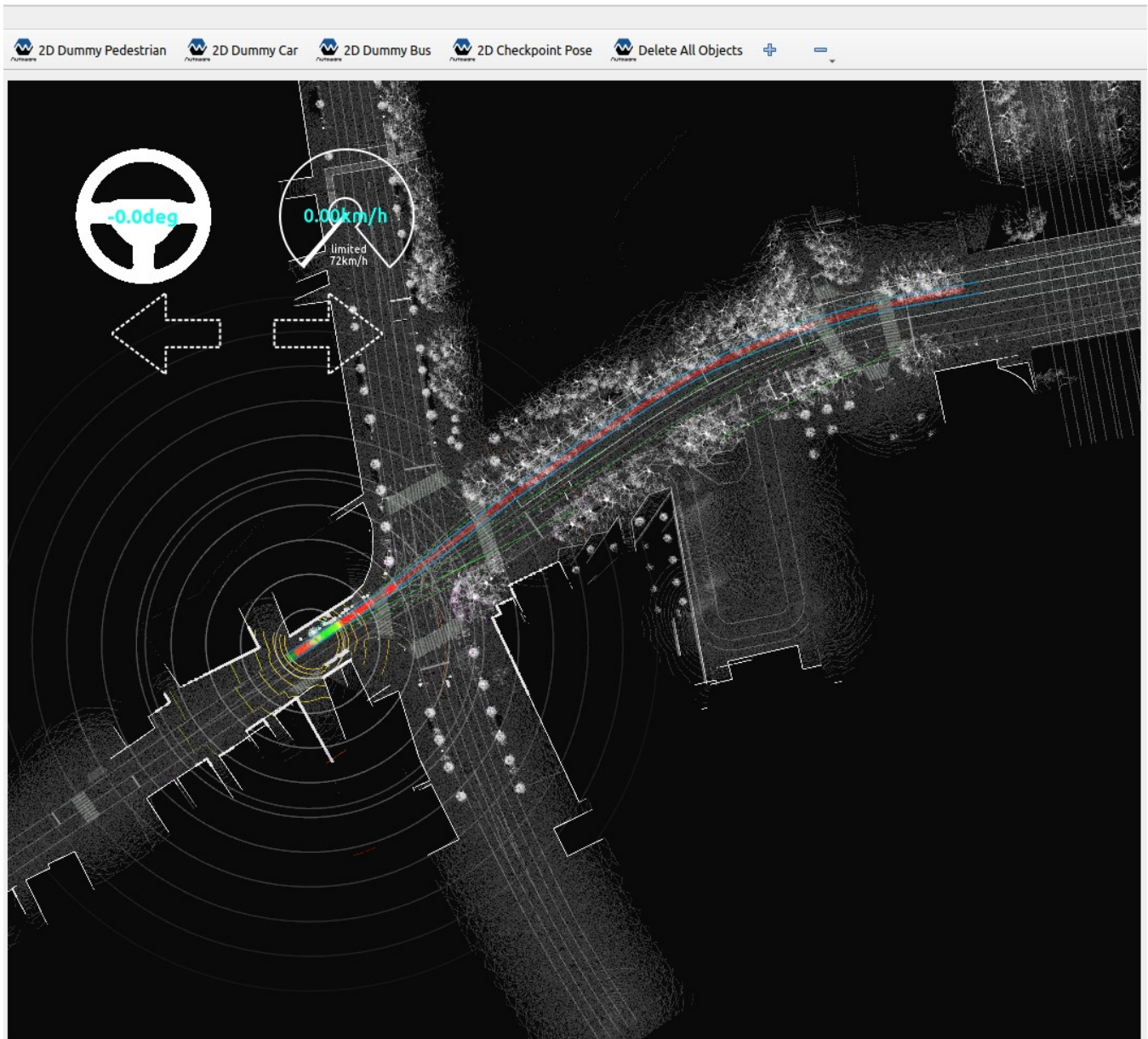


3. Set the navigation goal for the vehicle.

The screenshot shows the Autware simulation interface. The top toolbar has the '2D Goal Pose' tool selected, highlighted with a red box. The main window displays a top-down view of a vehicle on a road. The vehicle's current state is shown in the top-left corner: steering at -0.0deg and speed at 0.00km/h. A red arrow points to the vehicle's current position, labeled 'Here'. The interface includes several panels on the left:

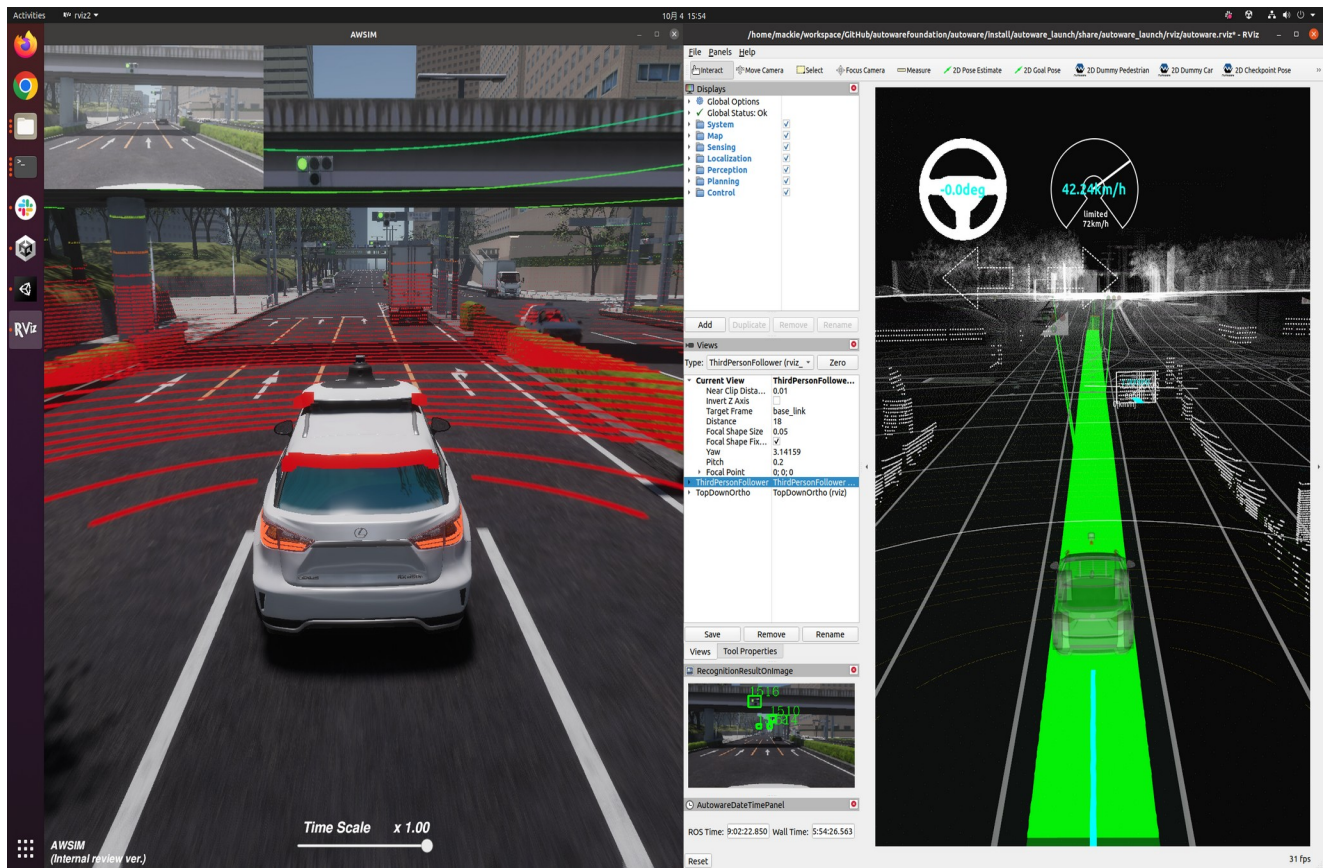
- Displays:** Global Options, Global Status: Ok, System, Map, Sensing, Localization, Perception.
- Views:** Type: TopDownOrtho (rviz_default_plugins), Current View: TopDownOrtho (rviz_default_plugins), Near Clip Distance: 0.01, Target Frame: viewer, Scale: 3.28768.
- AutwareStatePanel:** OperationMode (STOP, AUTO, STOP, LOCAL, REMOTE), AutwareControl (Enable, Enable, Disable).
- Routing:** UNSET, Clear Route, INITIALIZED, STOPPED, Accept Start, NONE, NONE.
- Localization:** Send Velocity Limit: 0 [km/h], Set Emergency.
- RecognitionResultOnImage:** RecognitionResultOnImage.
- AutwareDateTimePanel:** ROS Time: 1970-01-01 05:31:41.077, Wall Time: 2024-05-02 11:45:26.665.

The generated path can be seen.



4. Enable self-driving.

```
$ cd autoware
$ source install/setup.bash
$ ros2 topic pub /autoware/engage autoware_auto_vehicle_msgs/msg/Engage
'{engage: True}' -1
```



The self-driving simulation demo has been successfully launched !!!!!