Polaris GEM e2 Vehicle

Hang Cui

Feb. 10, 2022

Contents

Hardware of Polaris GEM e2 Vehicle

Velodyne VLP-16 LiDAR (Top LiDAR)

Mako G319C Camera (Front Camera)

Delphi ESR 2.5 Radar (Front Radar)

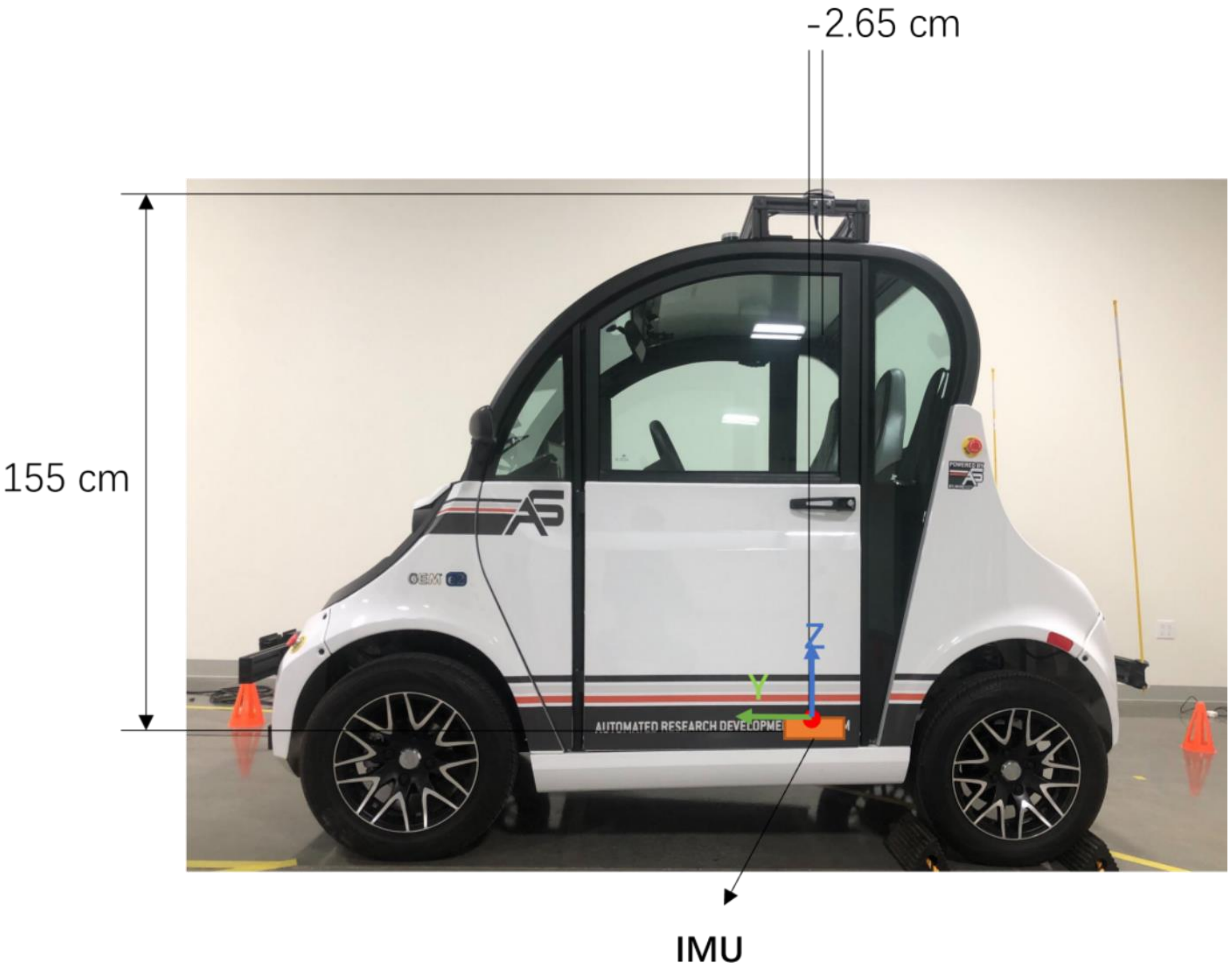
ProPak6 & SPAN-IGM-S1 (GNSS/INS)

PACMod (DBW Kit)

AStuff Spectra 2 (**2x** RTX 2080)

PACMod

Hardware of Polaris GEM e2 Vehicle



Software of Polaris GEM e2 Vehicle

System:

Ubuntu 20.04 with ROS Noetic (Python 3) and Gazebo 11

NVIDIA Driver Version: 460.80

Qt 5.12.11

CUDA 11.0.3

cuDNN 8.1.1

TensorRT 8.0.0.3

OpenCV 4.4.0

pytorch 1.7.1

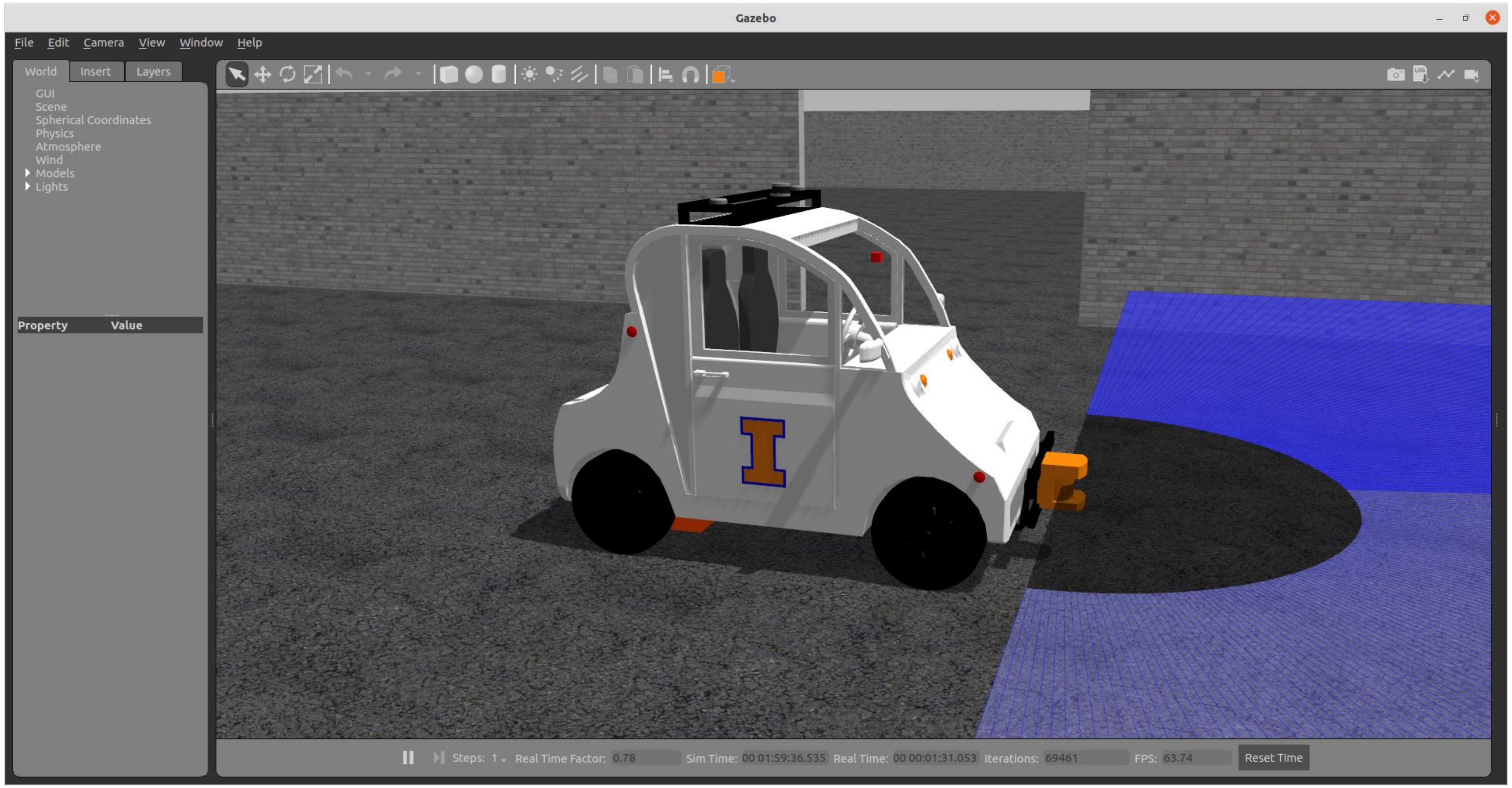
tensorflow 2.5

PCL 1.11.1

Simulator of Polaris GEM e2 Vehicle

The GEM simulator was originally built for ROS Melodic / Gazebo 9 (Ubuntu 18.04)

It has been **improved** and merged into **ROS Noetic / Gazebo 11 (Ubuntu 20.04)**



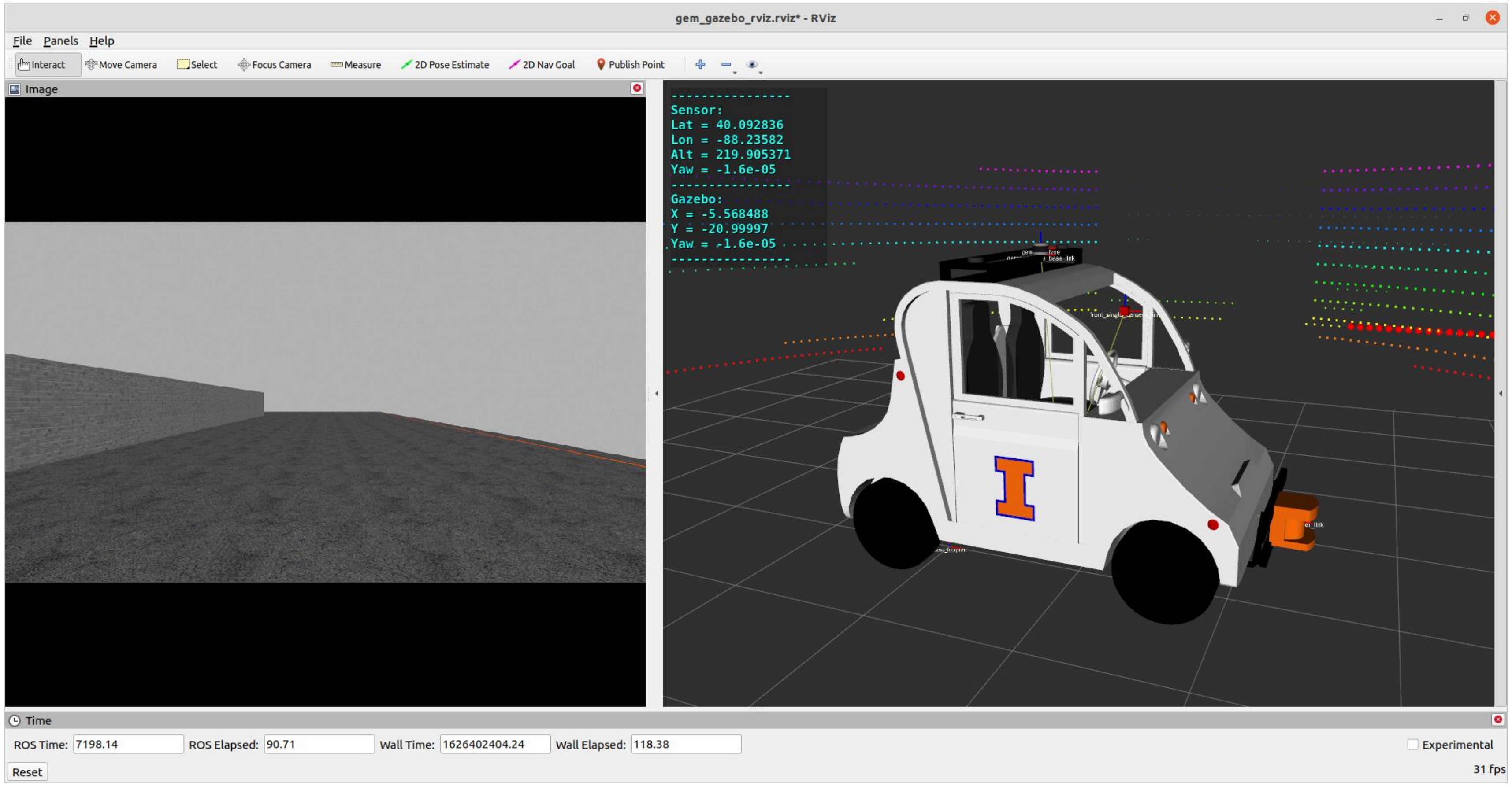
polaris_gem_simulator

ackermann_msgs	3 items
gem_simulator	2 items
hector_gazebo	6 items
jsk_rviz	6 items
sicktoolbox	10 items
sicktoolbox_wrapper	6 items
velodyne_simulator	7 items
waypoint_logger	3 items
gamepad.png	311.8 kB
gem_dimension.png	500.5 kB
GNSS_INS.png	5.6 MB
Polaris GEM e2 wtih Simulator.pdf	4.4 MB
readme.txt	1.4 kB
update_log.txt	341 bytes

Simulator of Polaris GEM e2 Vehicle

The GEM simulator was originally built for ROS Melodic / Gazebo 9 (Ubuntu 18.04)

It has been **improved** and merged into **ROS Noetic / Gazebo 11 (Ubuntu 20.04)**



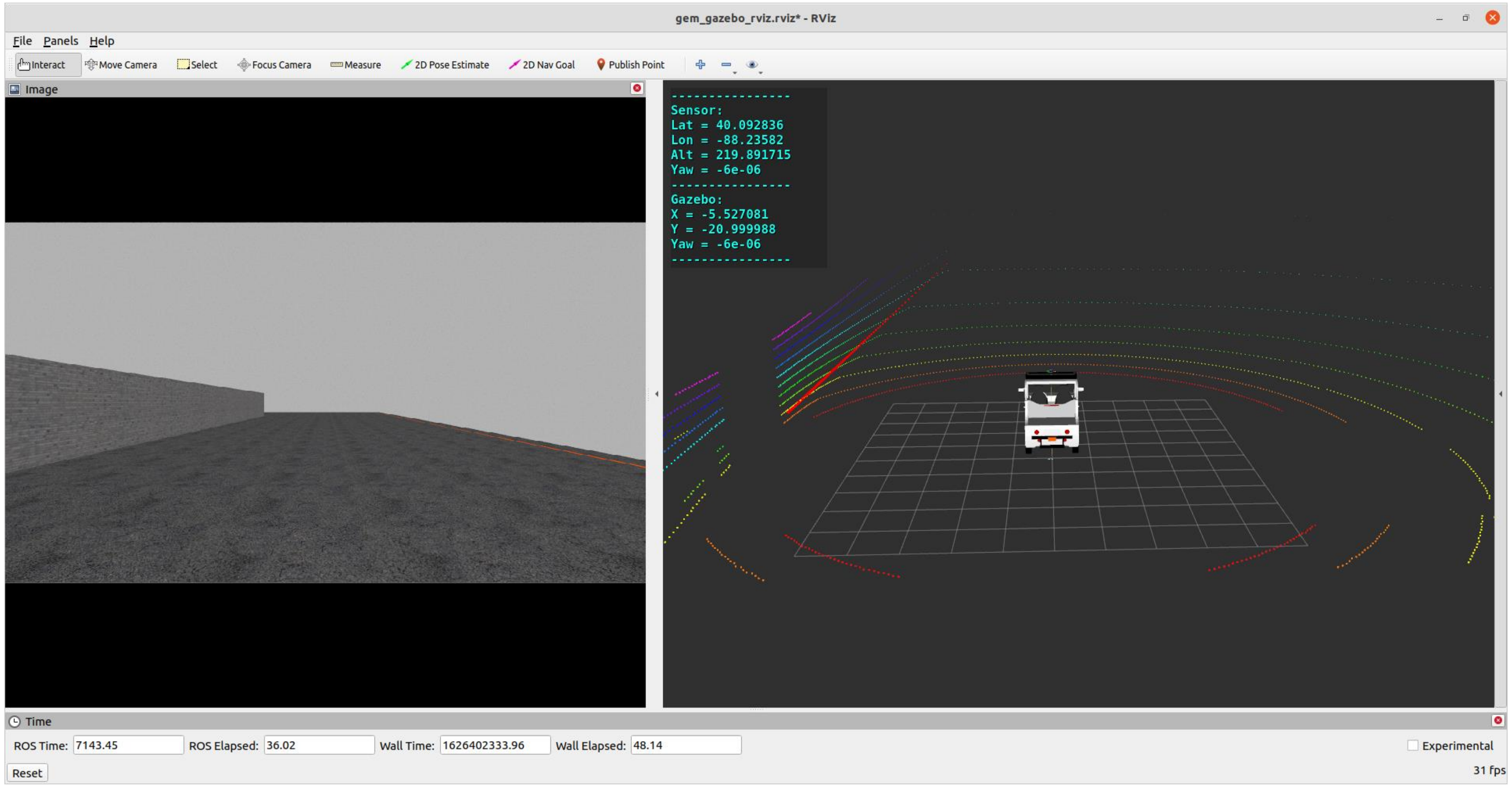
polaris_gem_simulator

ackermann_msgs	3 items
gem_simulator	2 items
hector_gazebo	6 items
jsk_rviz	6 items
sicktoolbox	10 items
sicktoolbox_wrapper	6 items
velodyne_simulator	7 items
waypoint_logger	3 items
gamepad.png	311.8 kB
gem_dimension.png	500.5 kB
GNSS_INS.png	5.6 MB
Polaris GEM e2 wtih Simulator.pdf	4.4 MB
readme.txt	1.4 kB
update_log.txt	341 bytes

Simulator of Polaris GEM e2 Vehicle

The GEM simulator was originally built for ROS Melodic / Gazebo 9 (Ubuntu 18.04)

It has been **improved** and merged into **ROS Noetic / Gazebo 11 (Ubuntu 20.04)**



polaris_gem_simulator

ackermann_msgs	3 items
gem_simulator	2 items
hector_gazebo	6 items
jsk_rviz	6 items
sicktoolbox	10 items
sicktoolbox_wrapper	6 items
velodyne_simulator	7 items
waypoint_logger	3 items
gamepad.png	311.8 kB
gem_dimension.png	500.5 kB
GNSS_INS.png	5.6 MB
Polaris GEM e2 wtih Simulator.pdf	4.4 MB
readme.txt	1.4 kB
update_log.txt	341 bytes

Simulator of Polaris GEM e2 Vehicle

Updates & New Features:

(1) Internel collision bugs were fixed.

(2) Model's inertia, friction and damping were tuned. (**may still need more tunings**)

(3) A frontal 2D laser was added.

(4) Optinally spawn LiDAR or laser sensor by using ROS launch commands

(5) Model's domain has been unified, such as /gem/imu/..., /gem/gps/..., /gem/...

(6) Model's base_link was redefined at the center of the rear axle.

(7) All sensor frames were calibrated/measured with repect to the base_link.

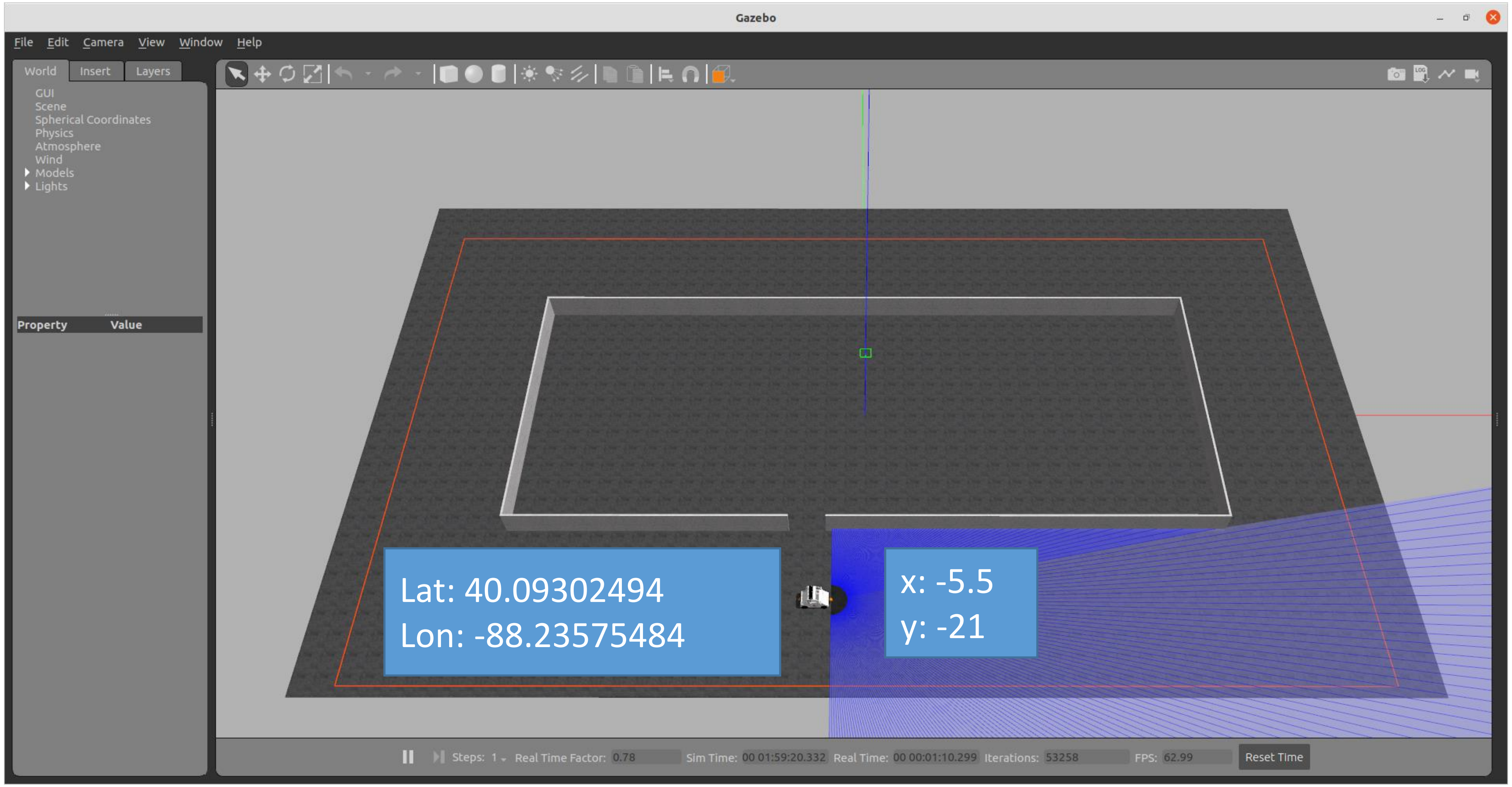
(8) Optianlly spawn multiple GEMs into the environmnent (**working on it**)

(9) A **polaris_gem_drivers_sim** is under development (*gem_gps_tracker_sim, gem_vision_tracker_sim, gem_teleop, gem_pure_pursuit_sim, etc*) to work with this simulator, namely **polaris_gem_simulator**

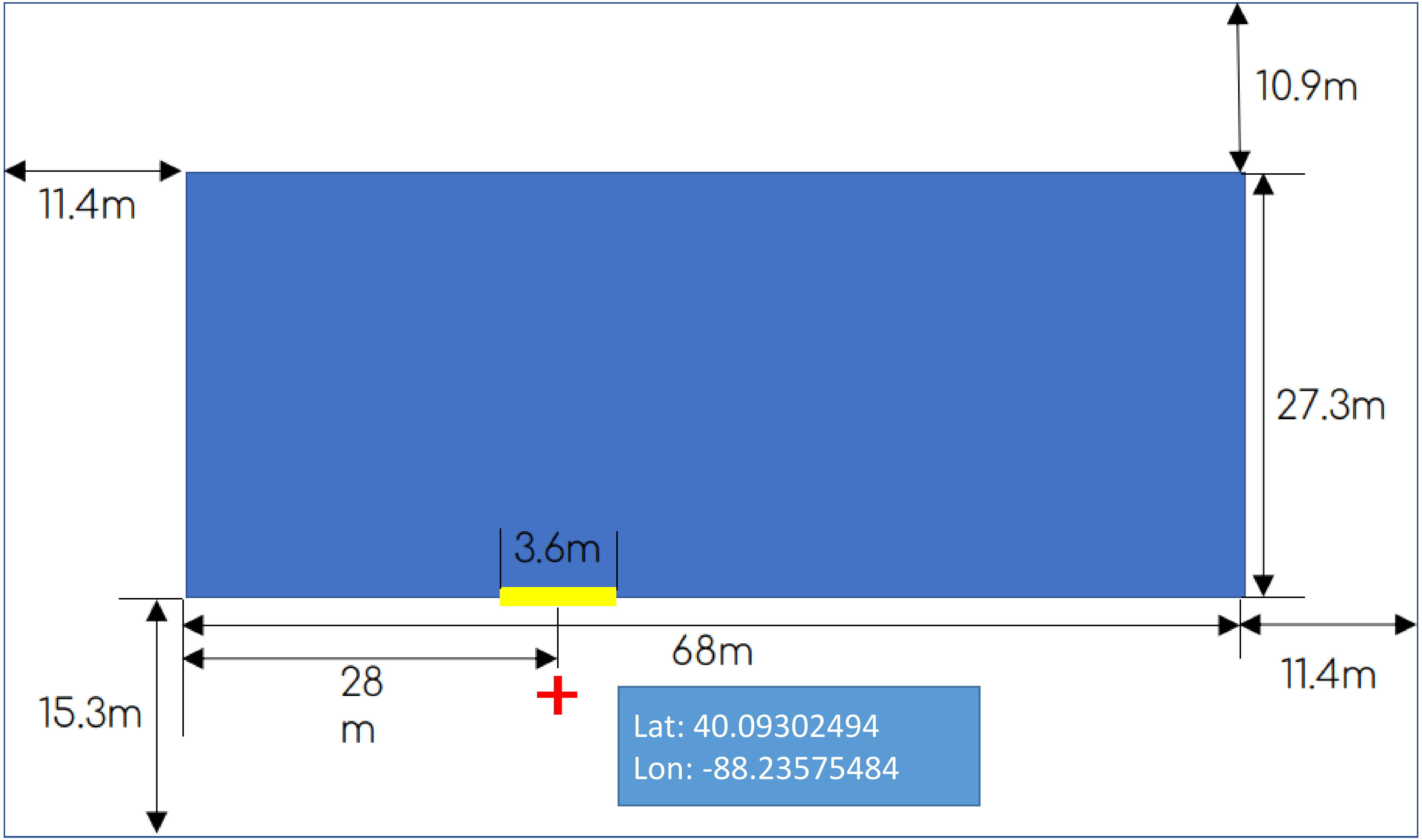
(10) Coming more

Simulator of Polaris GEM e2 Vehicle

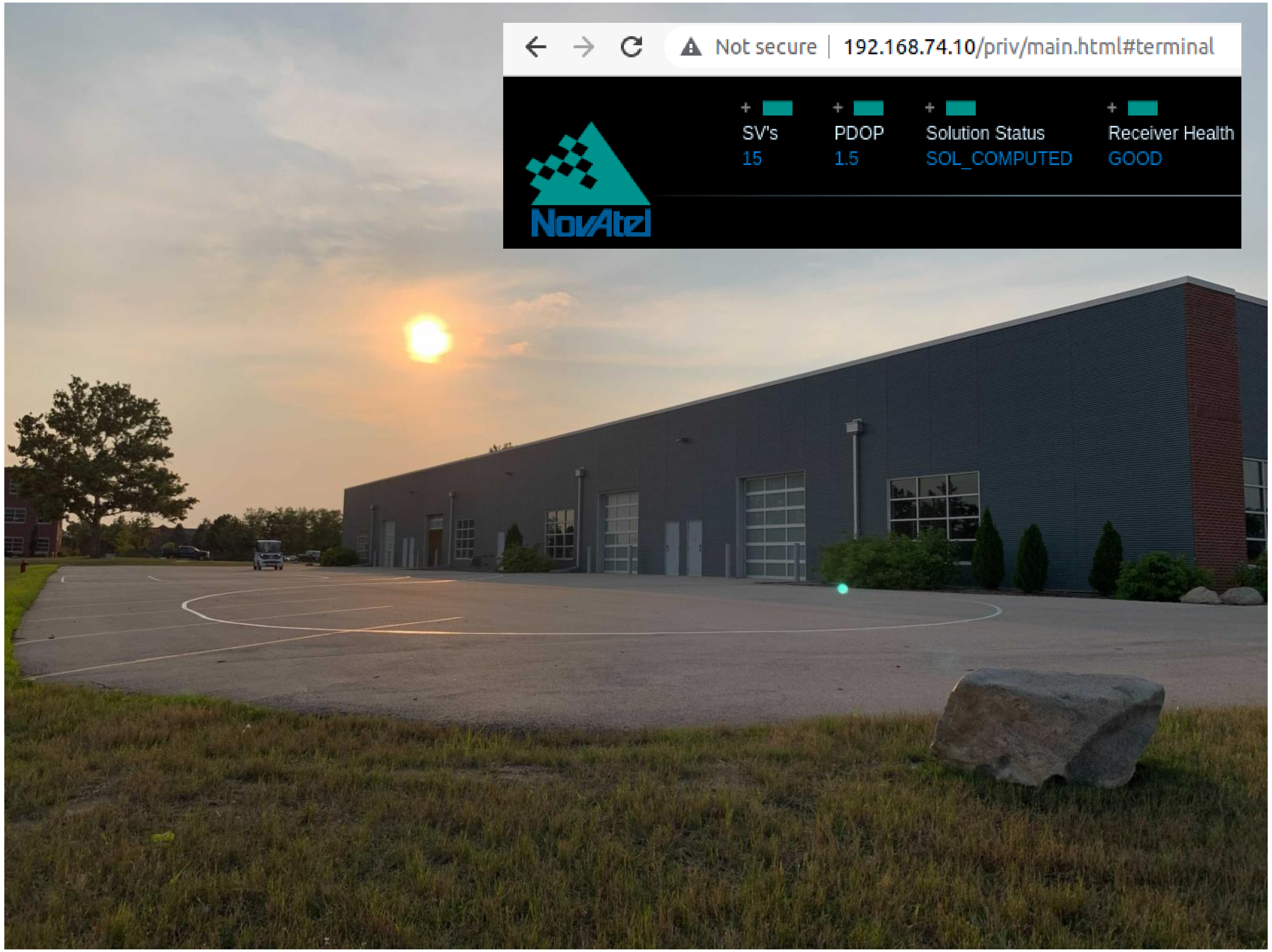
The high bay enviroment in real size **with matched GPS coordinates.**



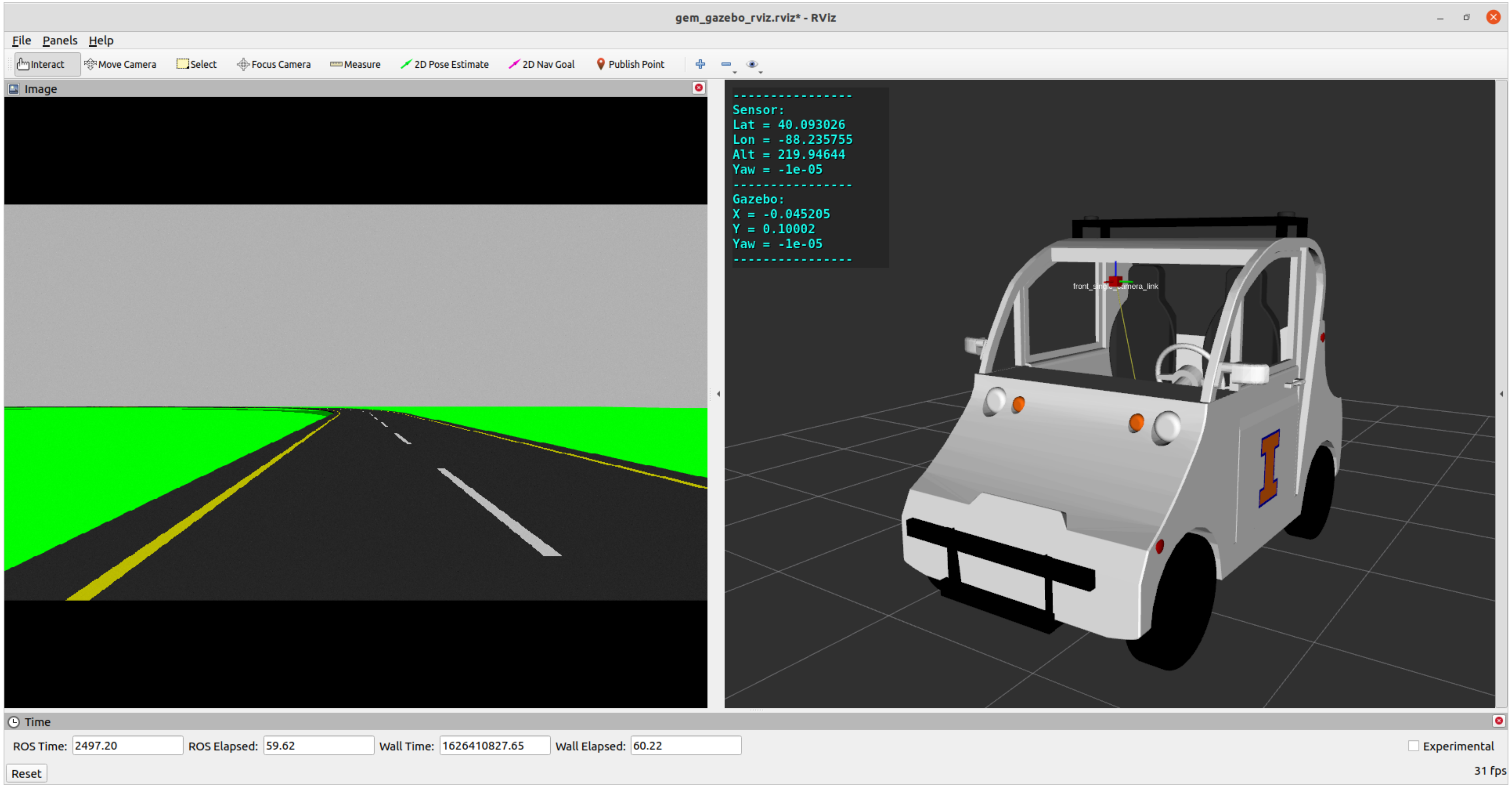
Simulator of Polaris GEM e2 Vehicle



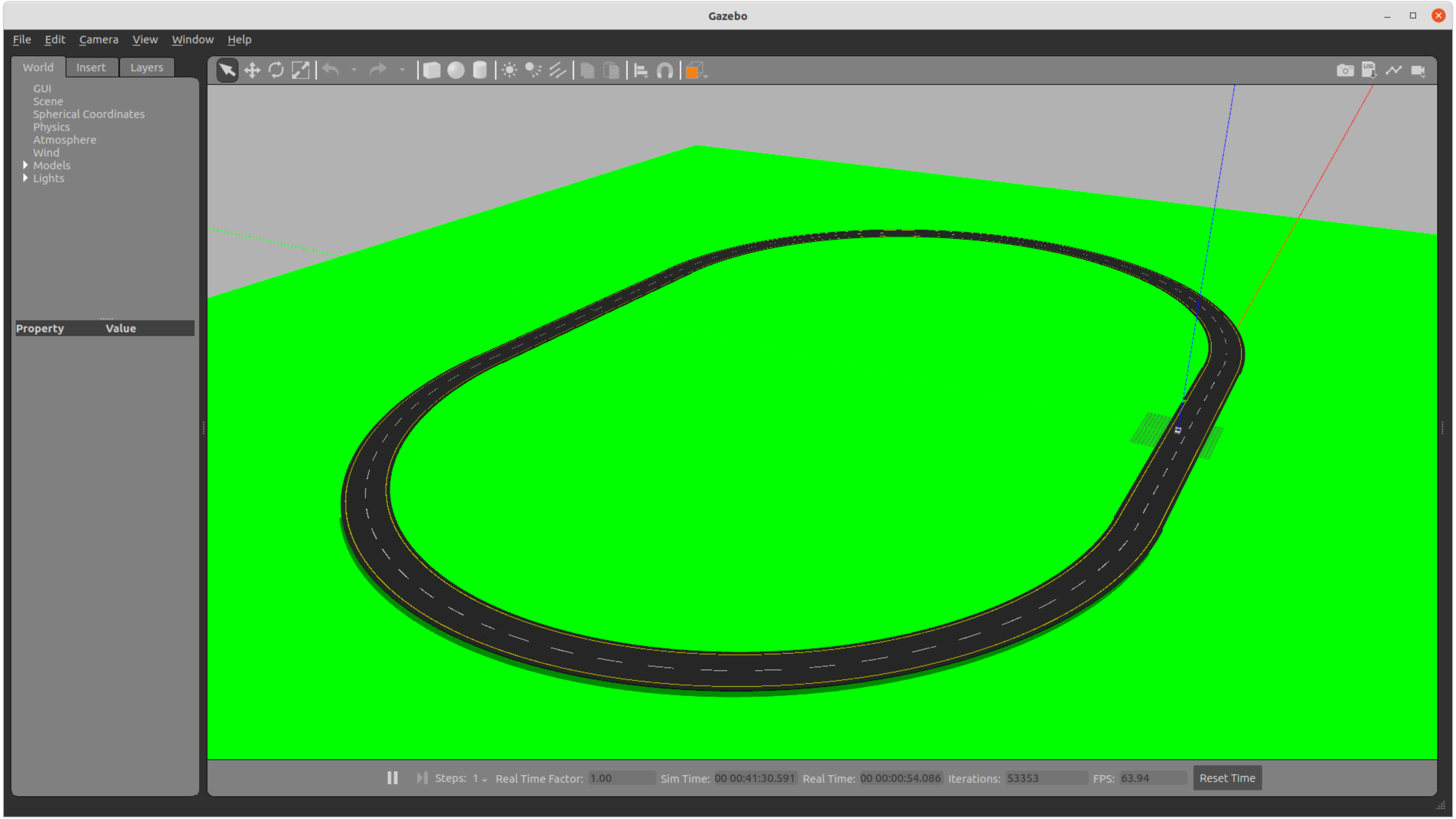
Simulator of Polaris GEM e2 Vehicle



Simulator of Polaris GEM e2 Vehicle



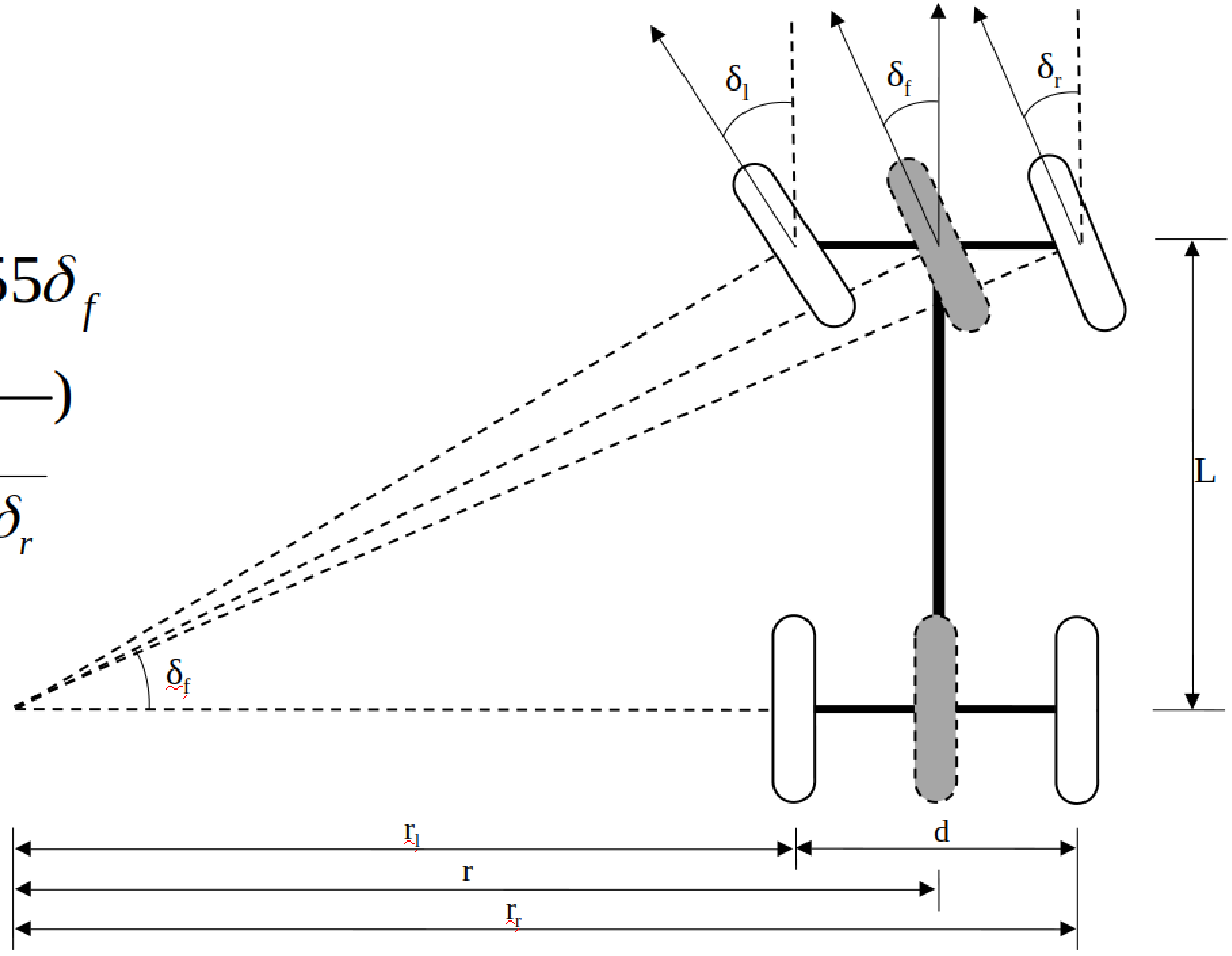
Simulator of Polaris GEM e2 Vehicle



Steering Wheel to Front Wheel Calibration

$$\delta_s = -0.1084\delta_f^2 + 21.755\delta_f$$

$$\delta_f = \arctan\left(\frac{2}{\frac{1}{\tan\delta_l} + \frac{1}{\tan\delta_r}}\right)$$

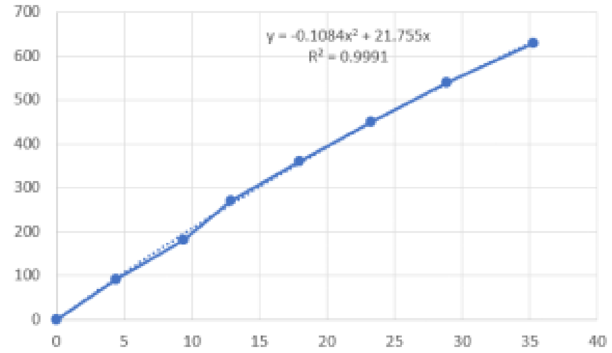


Steering Wheel to Front Wheel Mapping

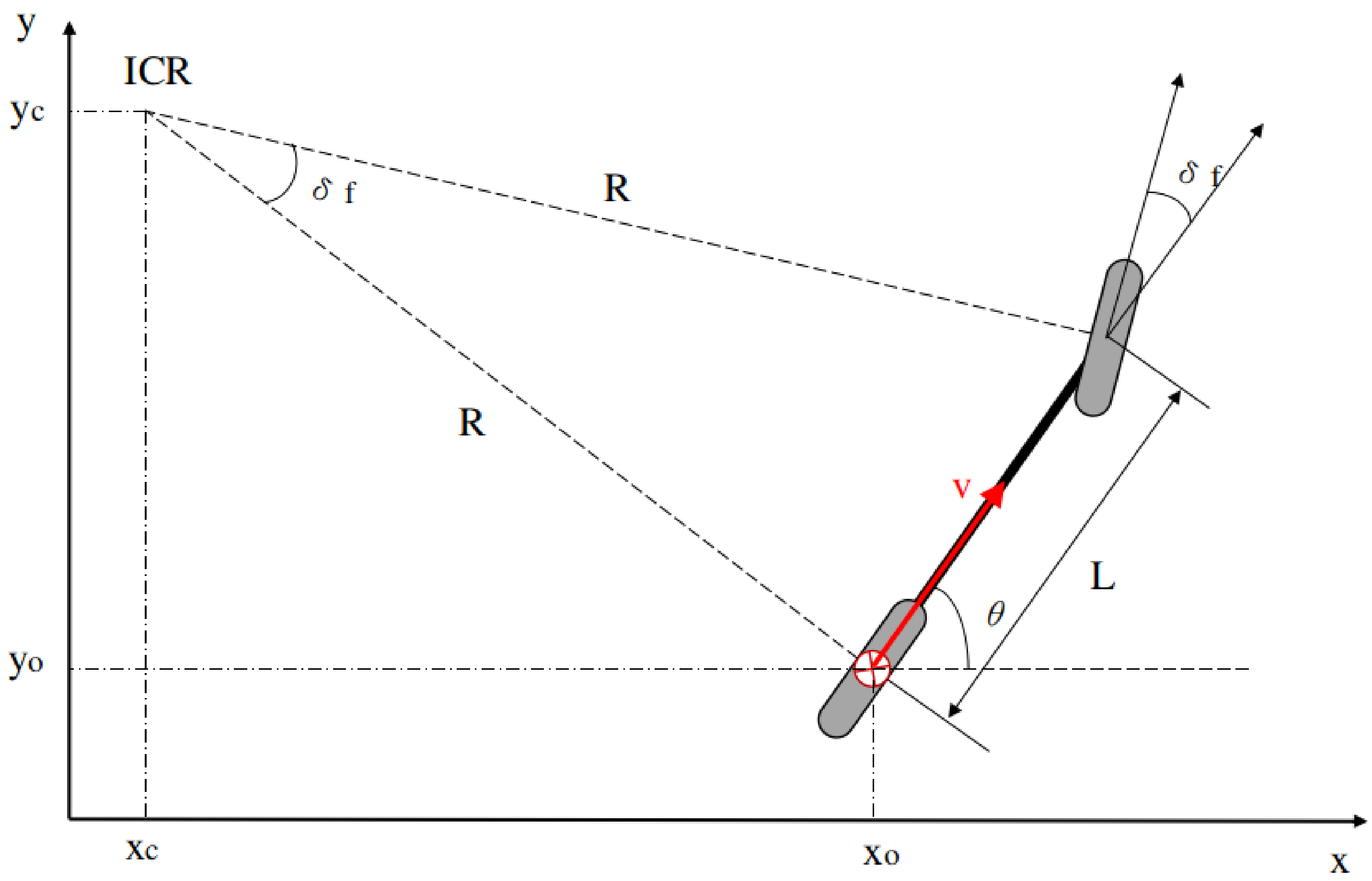
Steering Wheel to Front Wheel Calibration

Left Wheel	Right Wheel	Middle Angle	Steering Angle
0	0	0	0
4.6	4.2	4.4	90
10.5	8.5	9.4	180
14	12	12.9	270
18.8	17.3	18	360
24.2	22.4	23.3	450
31.7	26.5	28.9	540
38	33	35.3	630

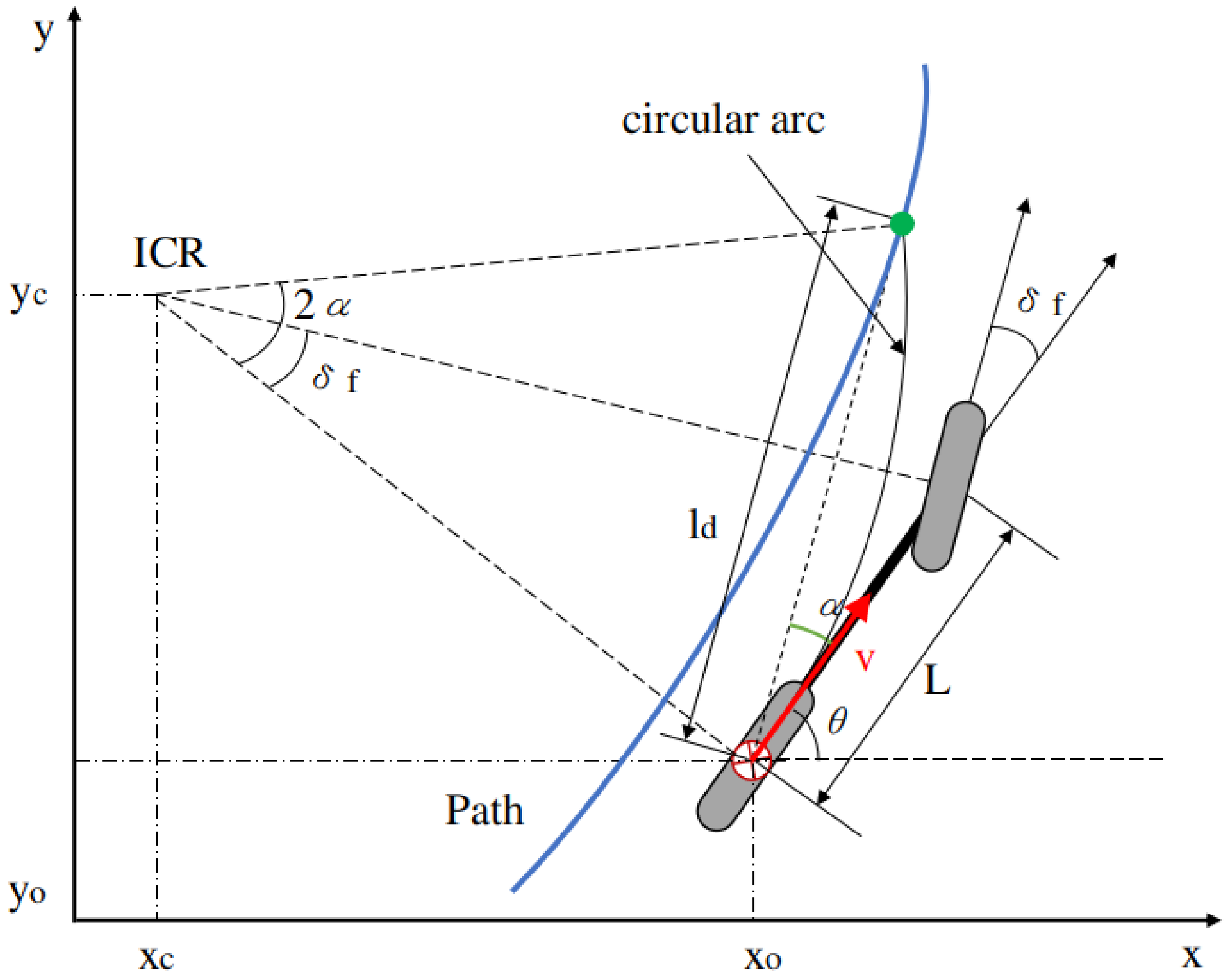
Steering wheel was controlled by program to ratotat at a particular angles, namely 0, 90, 180, 270, 360, ..., etc.



Lane Detection and Tracking - Bicycle Model



Lane Detection and Tracking - Pure Pursuit Controller



$$\dot{x} = v\cos\theta$$

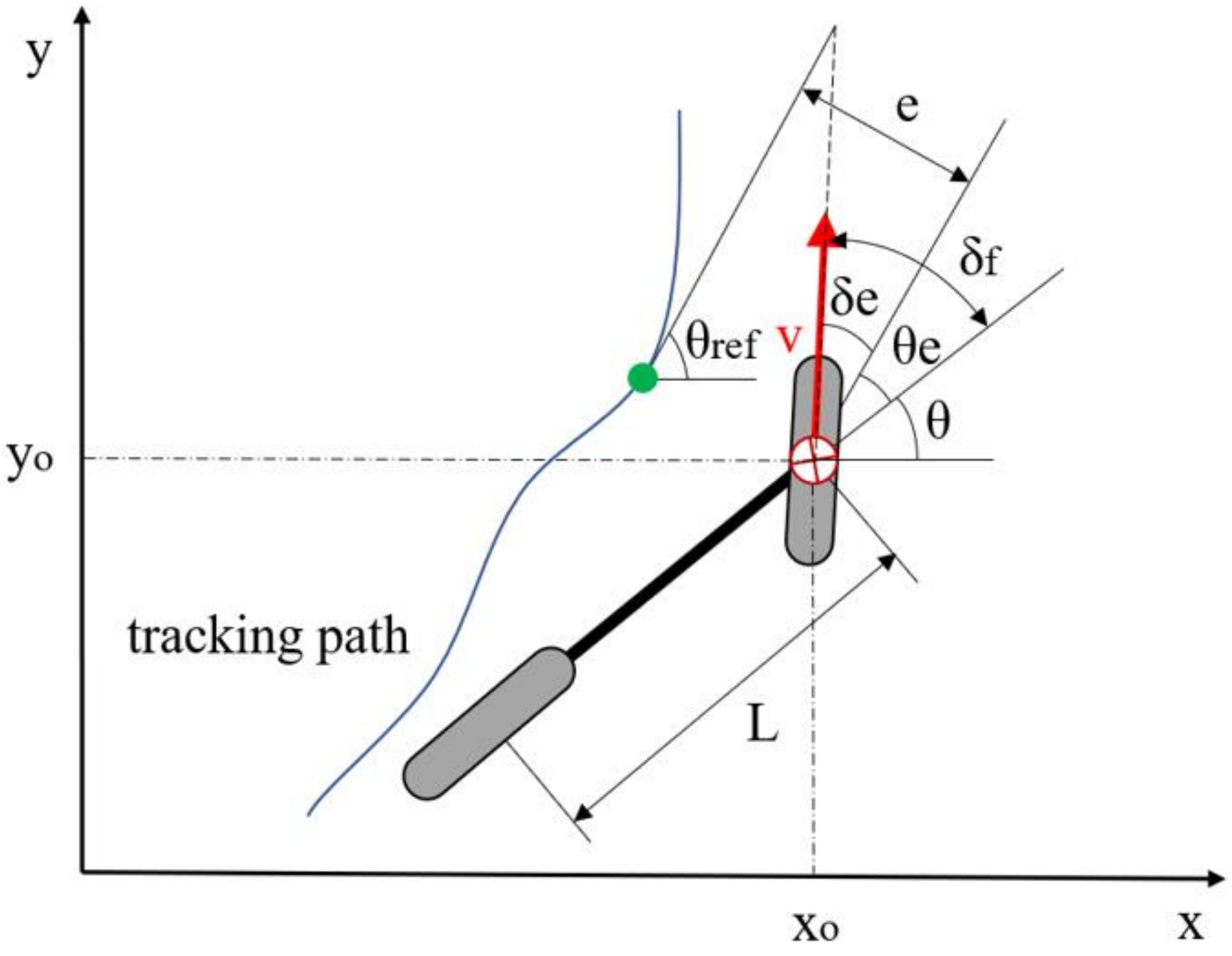
$$\dot{y} = v\sin\theta$$

$$\dot{\theta} = \frac{v}{L}\tan\delta_f$$

$$\dot{v} = a$$

$$\delta_f = \tan^{-1}\left(\frac{2L\sin\alpha}{l_d}\right)$$

Lane Detection and Tracking - Stanley Controller



$$\delta_e = \tan^{-1}\left(\frac{ke}{v}\right)$$

$$\theta_{ref} = \theta_{ref} - \theta$$

$$\delta_f = \theta_e + \delta_e$$

$$e = [\sin\theta_{ref}(x - x_{ref}) - \cos\theta_{ref}(y - y_{ref})]$$

L: wheelbase

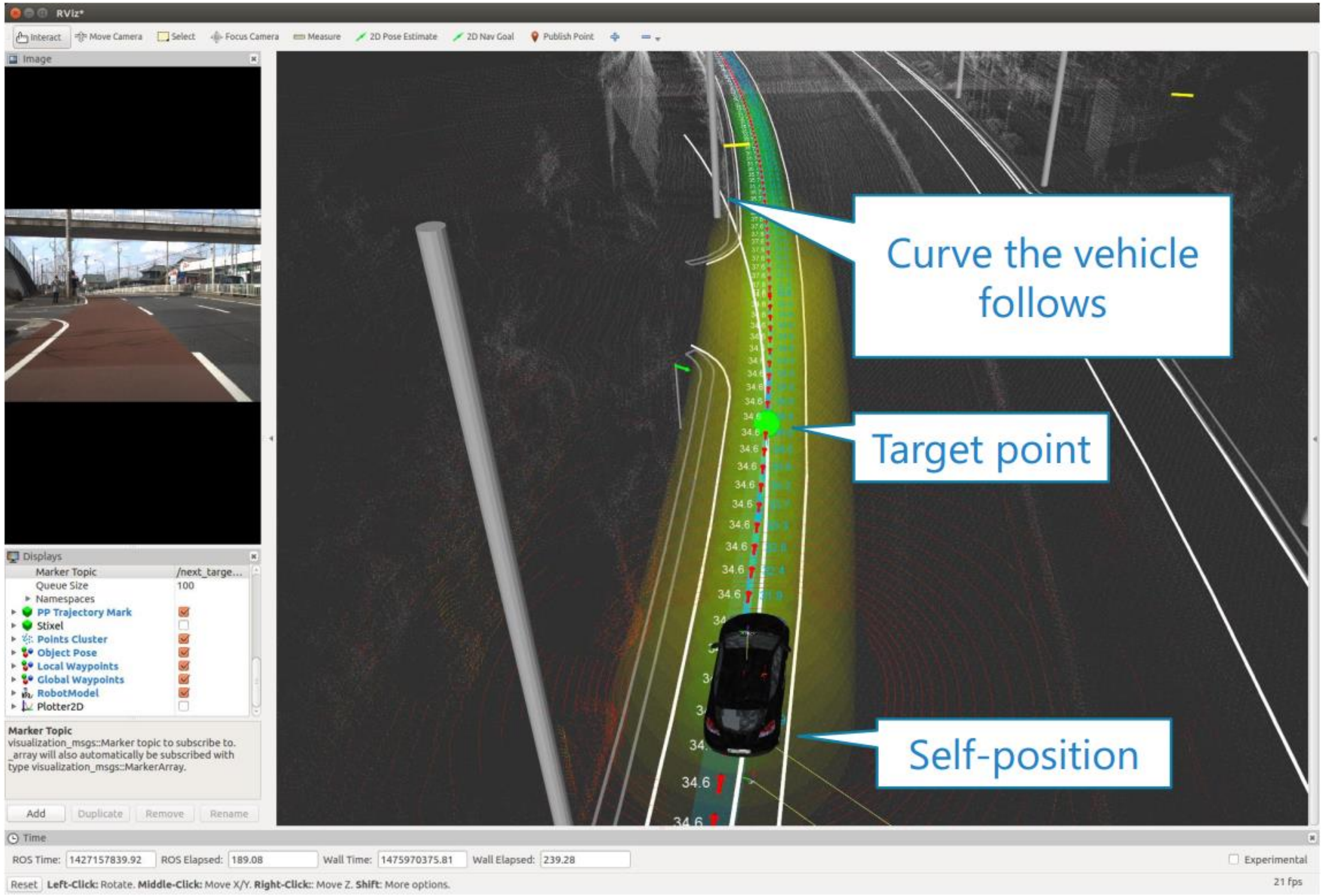
e: cross track error

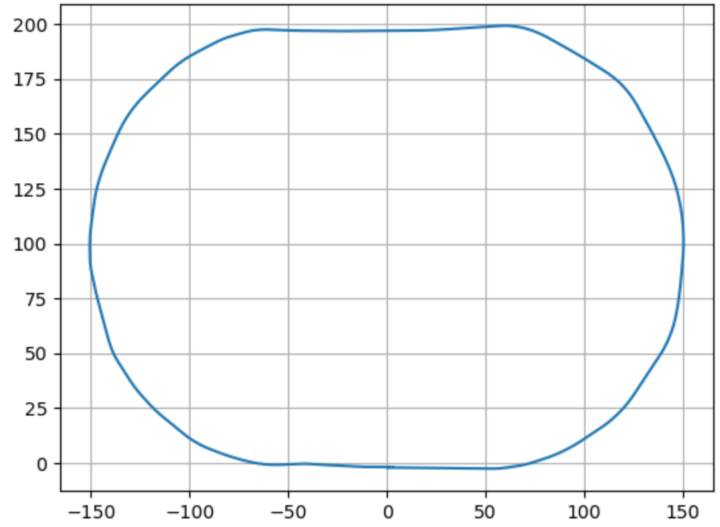
θ_e: heading error

δ_e: cross track steering

Lane Tracking with Pure Pursuit Controller

Autoware.AI





Reference

AStuff drivers: https://github.com/astuff

Polaris GEM e2 ROS Noetic simulator: https://github.com/hangcui1201/POLARIS_GEM_e2

Polaris GEM e2 user manual: https://github.com/hangcui1201/POLARIS_GEM_e2

Thanks!