# Polaris GEM e2 Vehicle

Hang Cui

Sep. 6, 2022

# Contents

**Hardware** & **Software** of Polaris GEM e2 Vehicle

**Simulator** of Polaris GEM e2 Vehicle

**Demos** of Polaris GEM e2 Vehicle in Simulator or Real World

- Sensor/Actuator Calibration for Control
- GNSS-based Waypoints Following w/wo RTK using Pure Pursuit or Stanley Controller

# Hardware of Polaris GEM e2 Vehicle

Velodyne VLP-16 LiDAR (Top LiDAR)

ZED2 Stereo Camera (Front Camera)

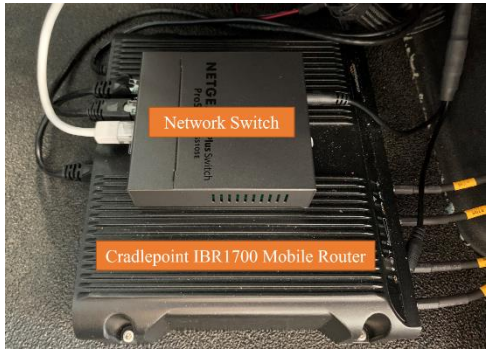
Delphi ESR 2.5 Radar (Front Radar)

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

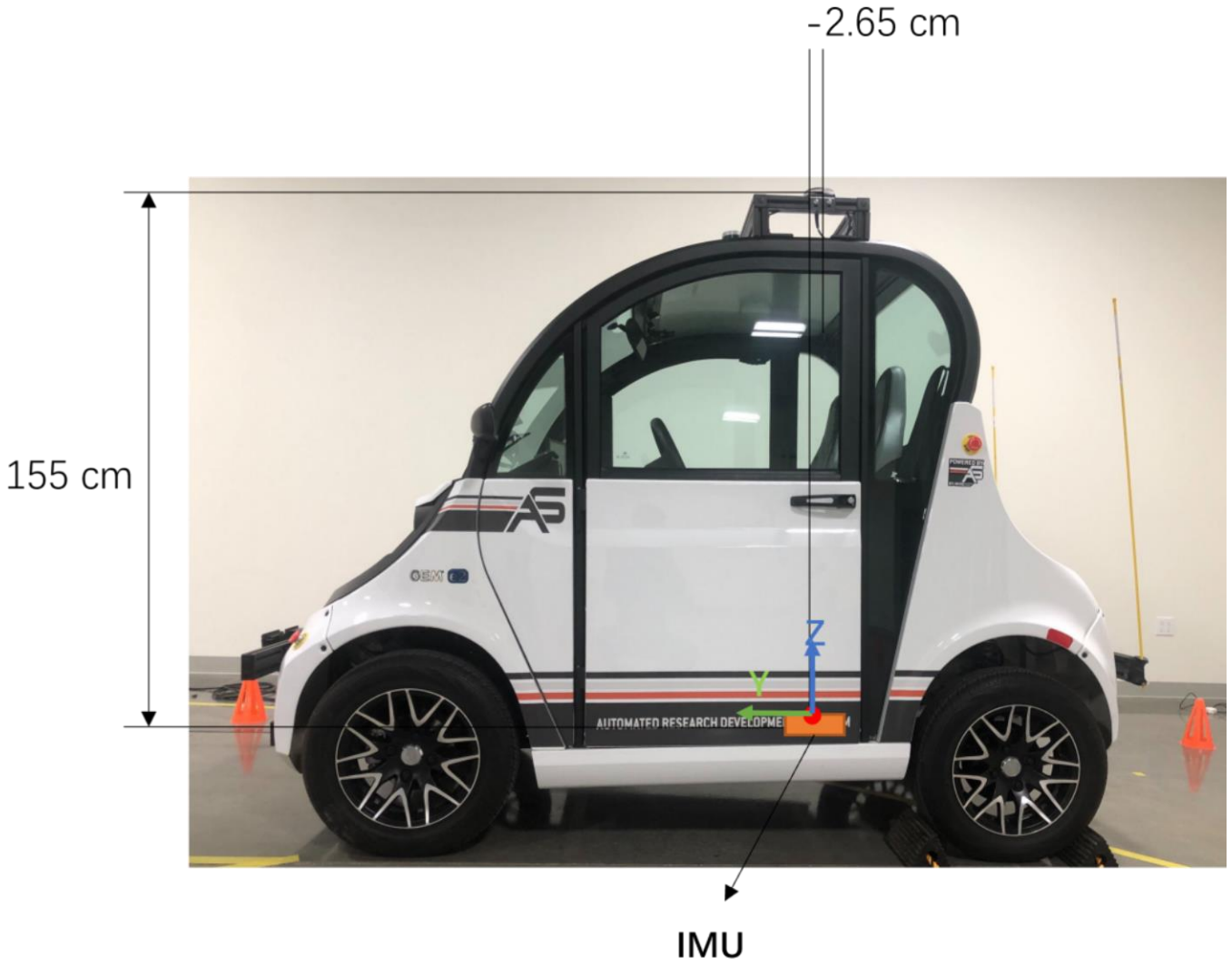
PACMod (DBW Kit)

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



# Hardware of Polaris GEM e2 Vehicle

# Software of Polaris GEM e2 Vehicle

**System:**

**Ubuntu 20.04 with ROS Noetic and Gazebo 11**

NVIDIA Driver Version: 450

CUDA 11.0.3

OpenCV 4.6.0

Pytorch 1.7.1

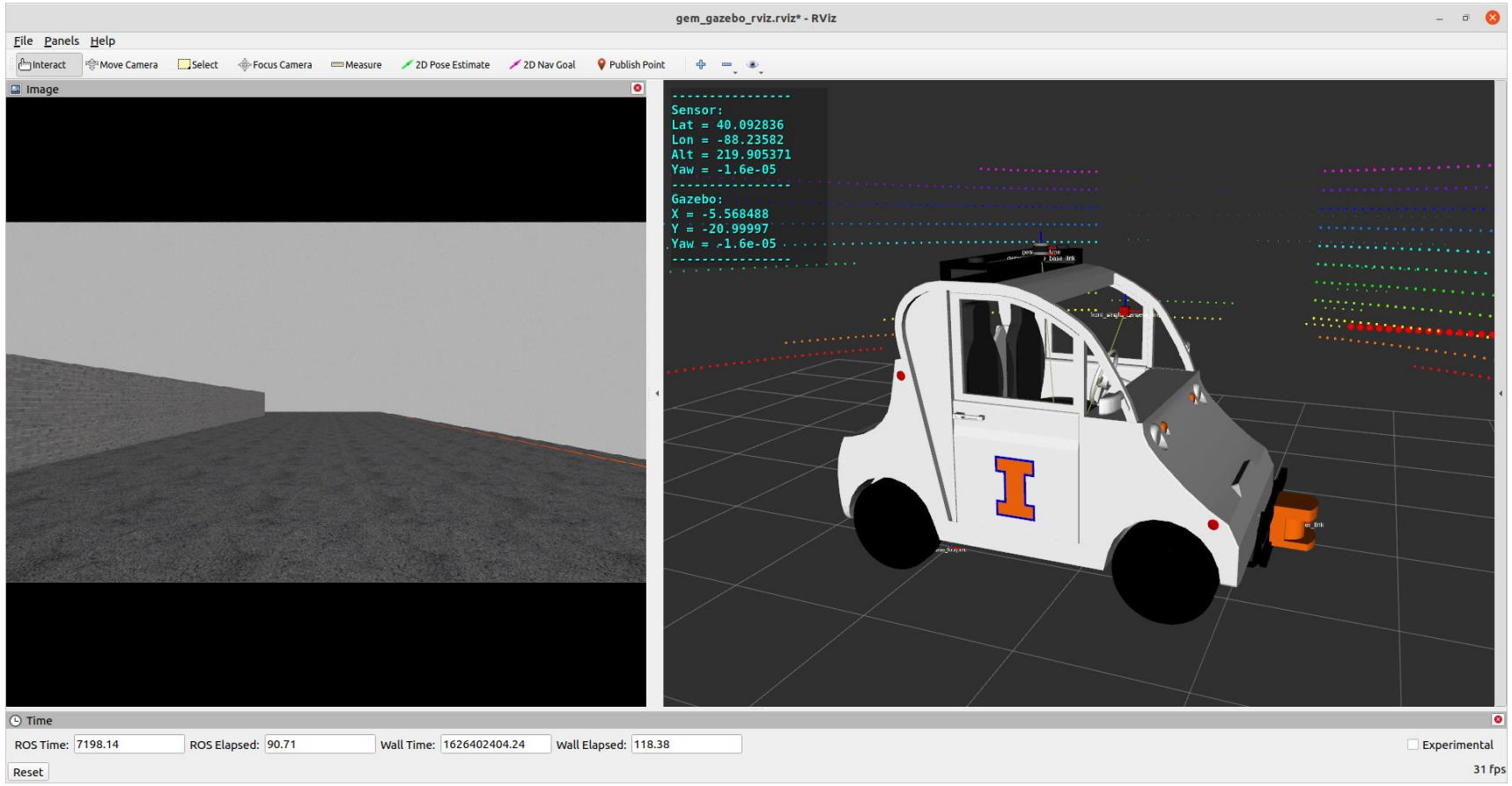
More to add …

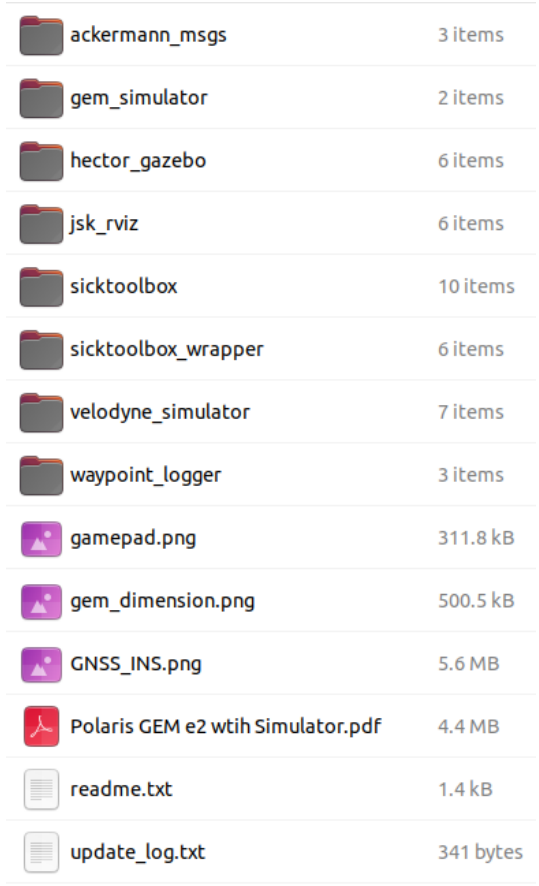
# Simulator of Polaris GEM e2 Vehicle

**System & Software:** Ubuntu 20.04, ROS Noetic and Gazebo 11

**Environment:** Highbay

**Github**: https://github.com/hangcui1201/POLARIS_GEM_e2_Simulator



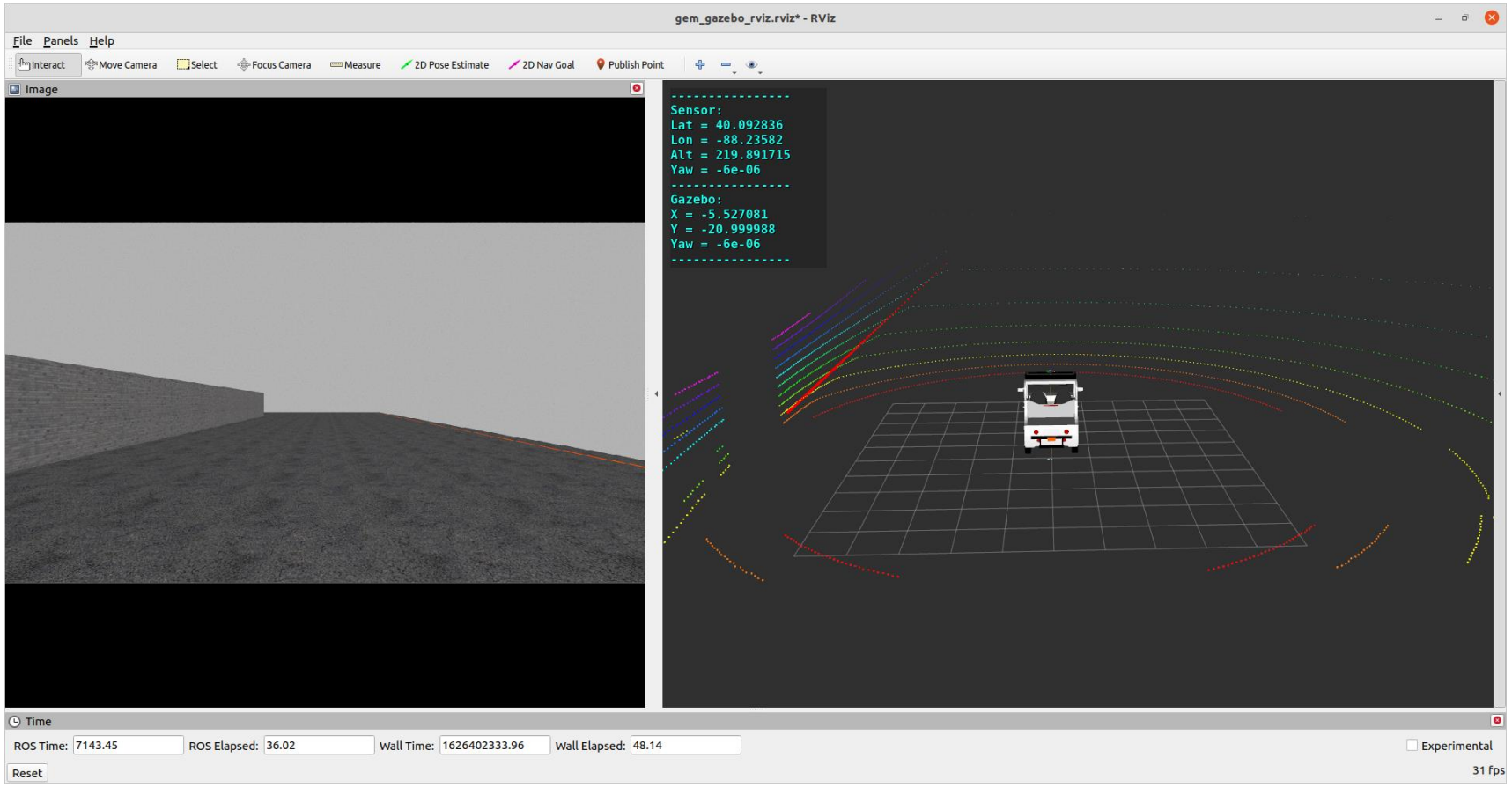



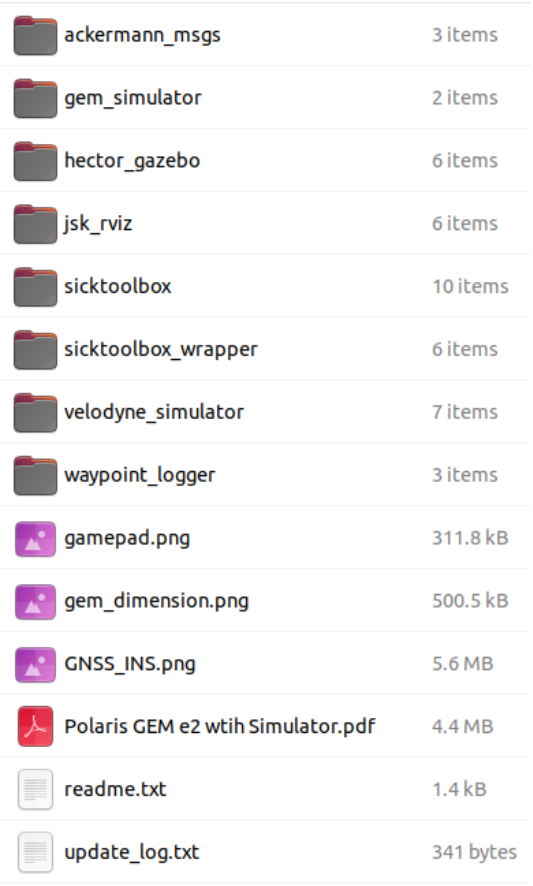
# Simulator of Polaris GEM e2 Vehicle

**System & Software:** Ubuntu 20.04, ROS Noetic and Gazebo 11

**Environment:** Highbay

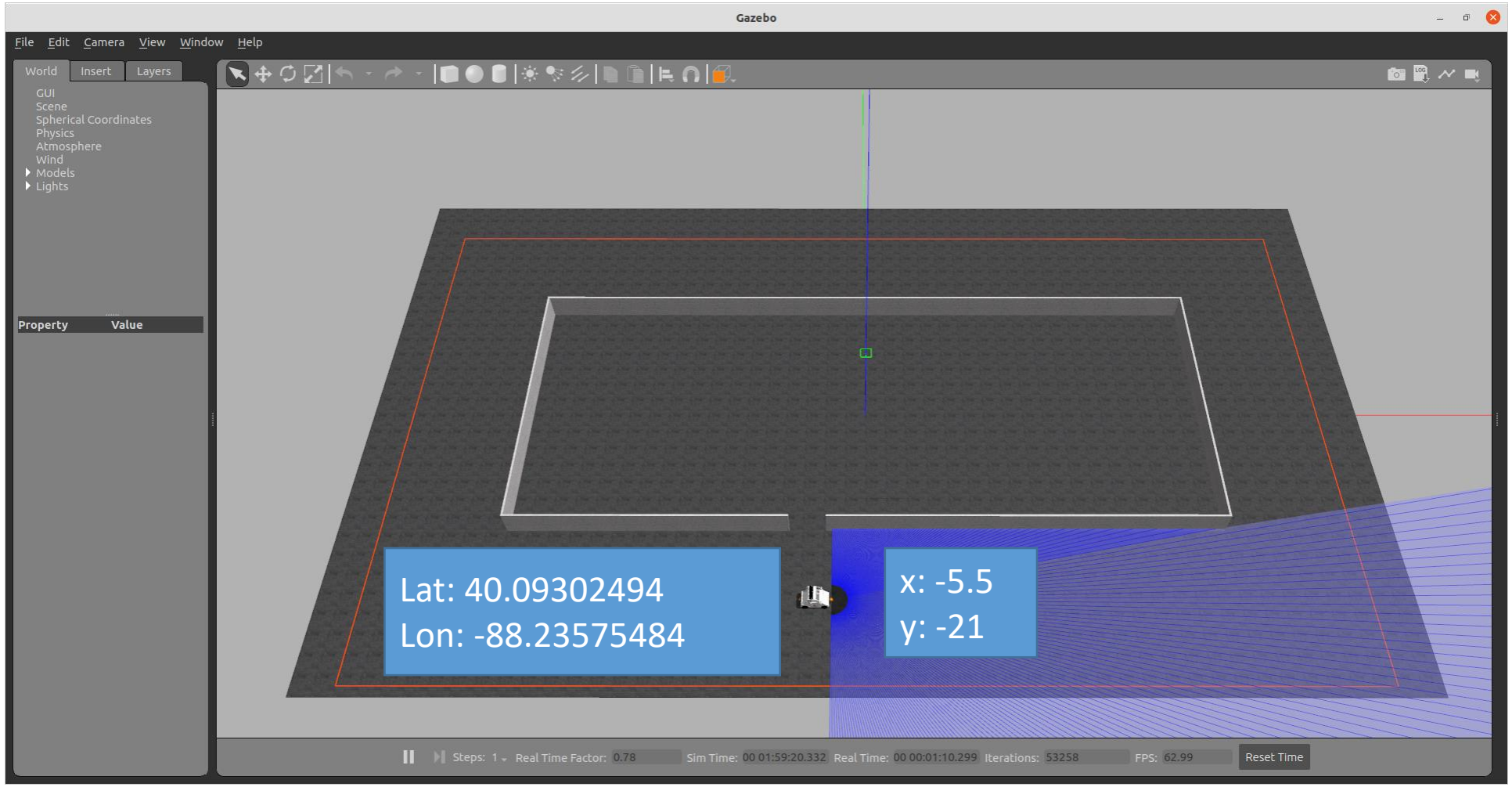
**Github**: https://github.com/hangcui1201/POLARIS_GEM_e2_Simulator



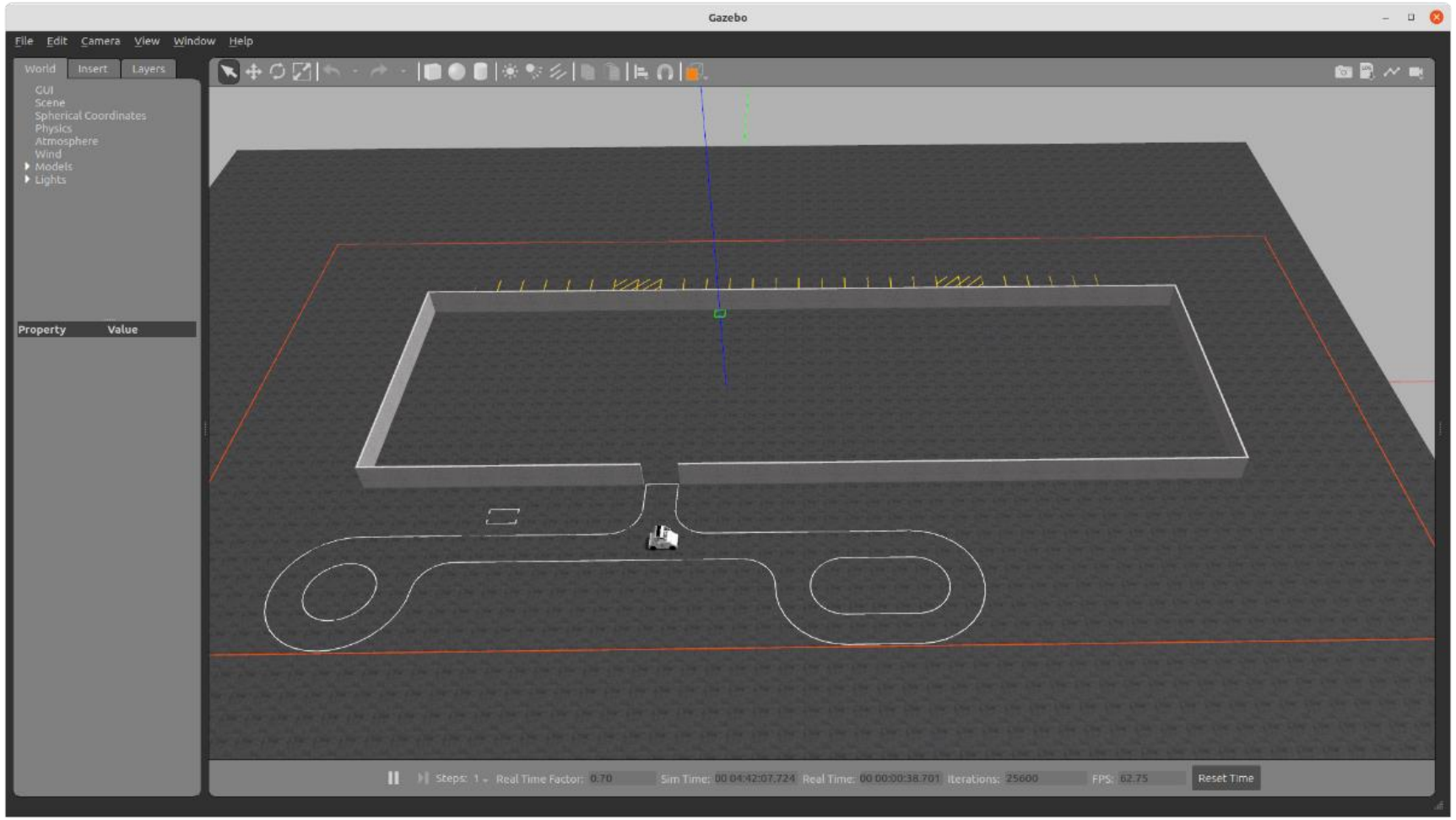
# Simulator of Polaris GEM e2 Vehicle

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

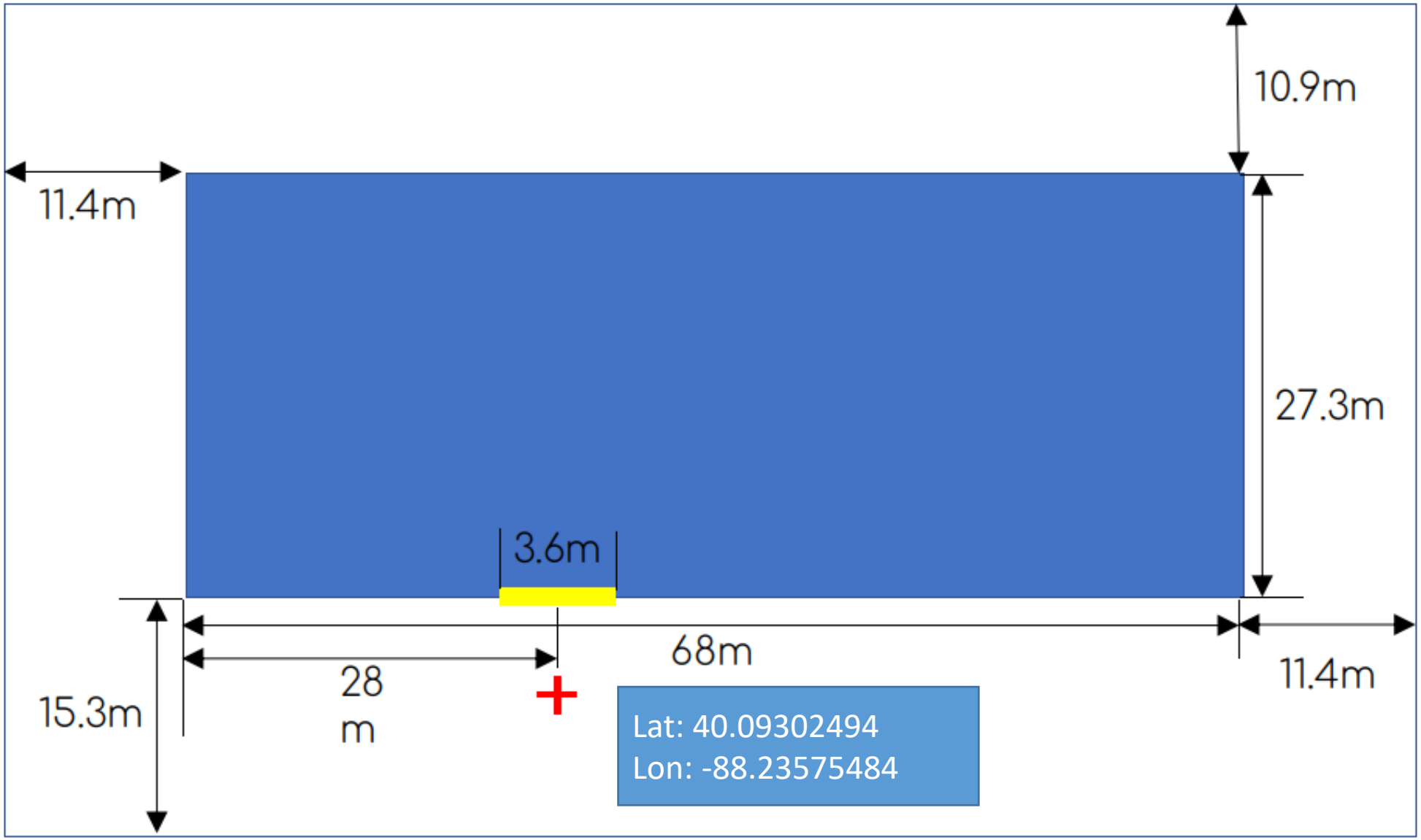


# Simulator of Polaris GEM e2 Vehicle

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

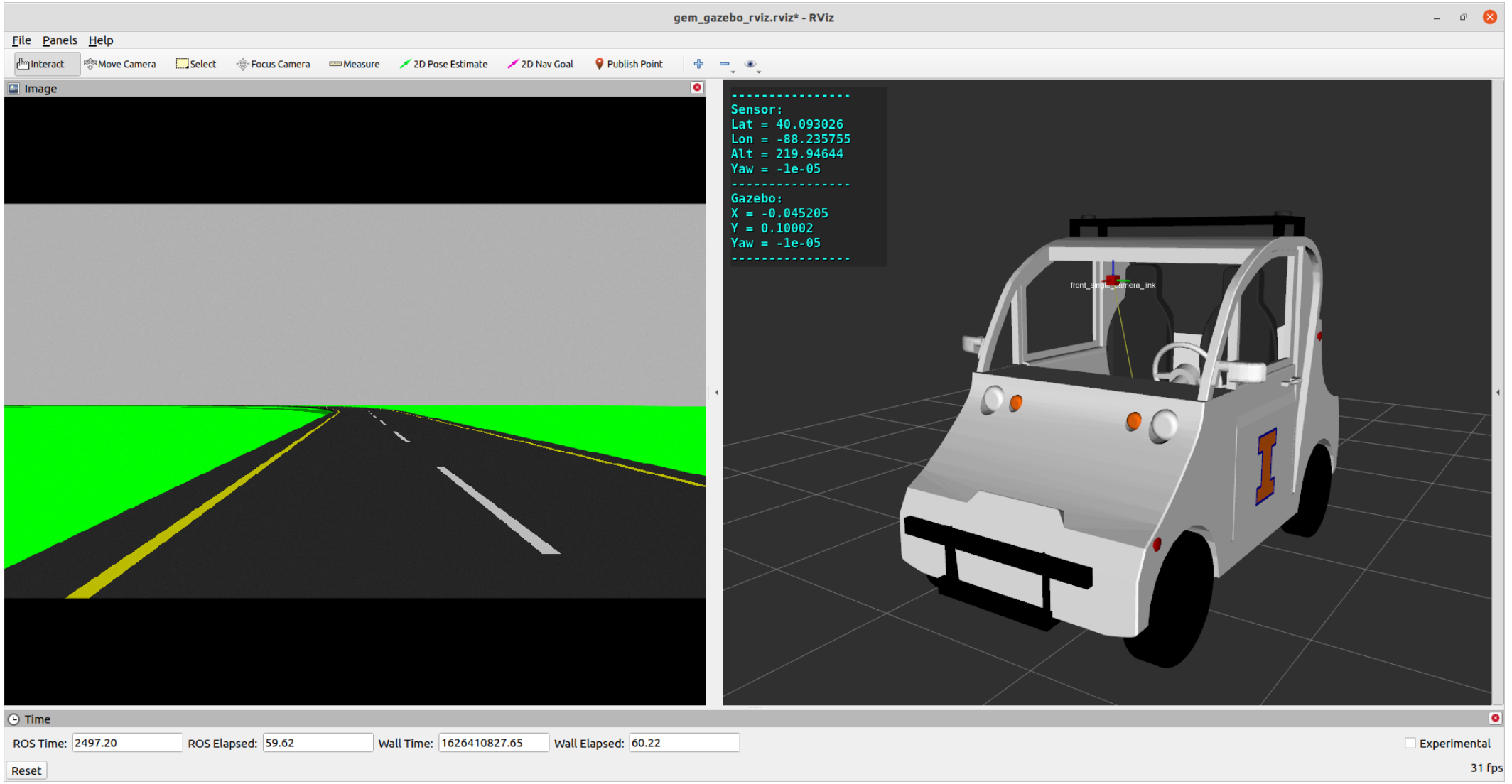


# Simulator of Polaris GEM e2 Vehicle

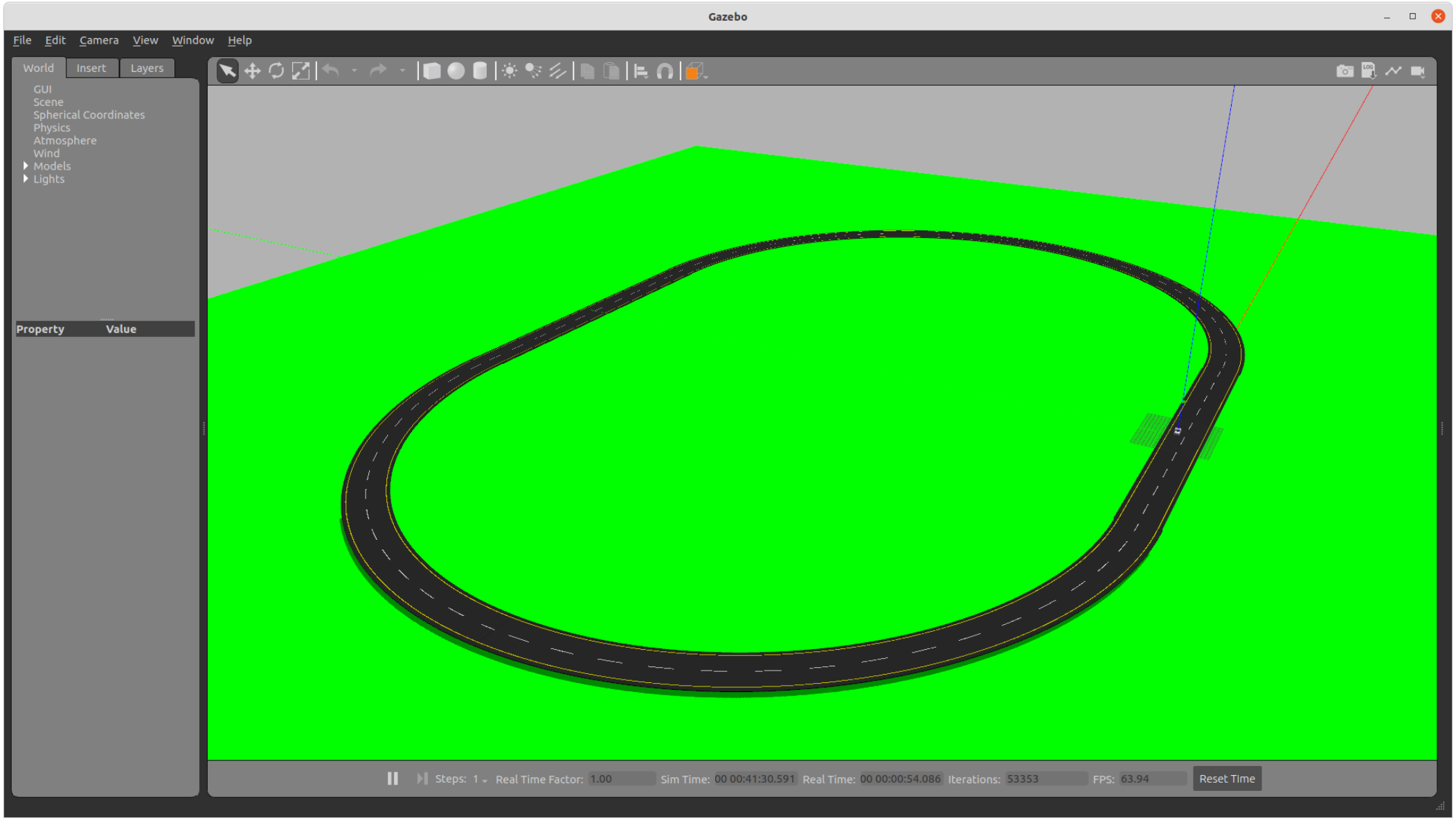


# Simulator of Polaris GEM e2 Vehicle

# Simulator of Polaris GEM e2 Vehicle



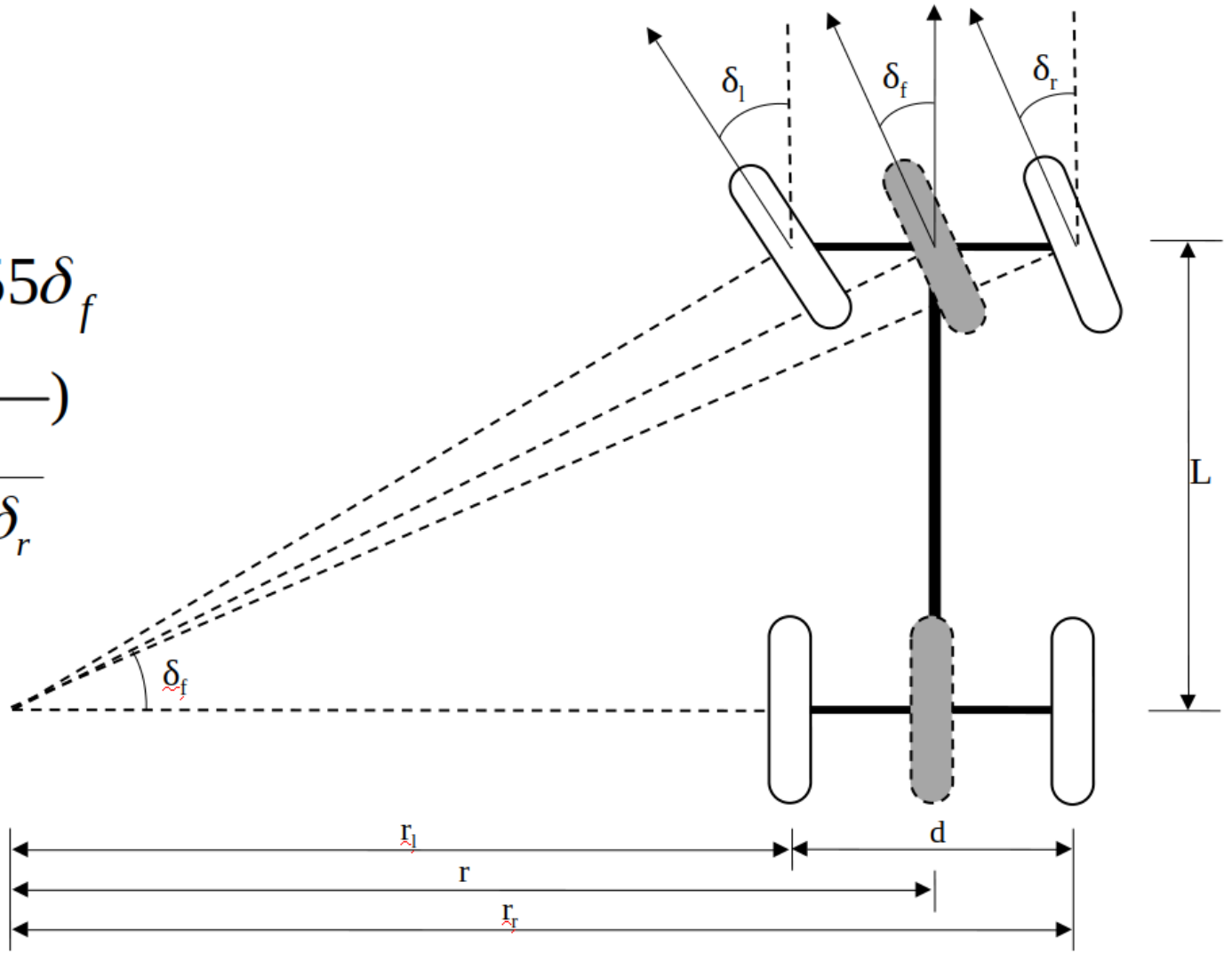
# Simulator of Polaris GEM e2 Vehicle



# Steering Wheel to Front Wheel Calibration

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

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

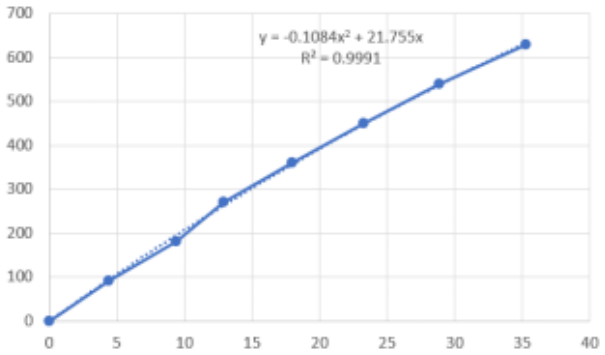


# Steering Wheel to Front Wheel Mapping

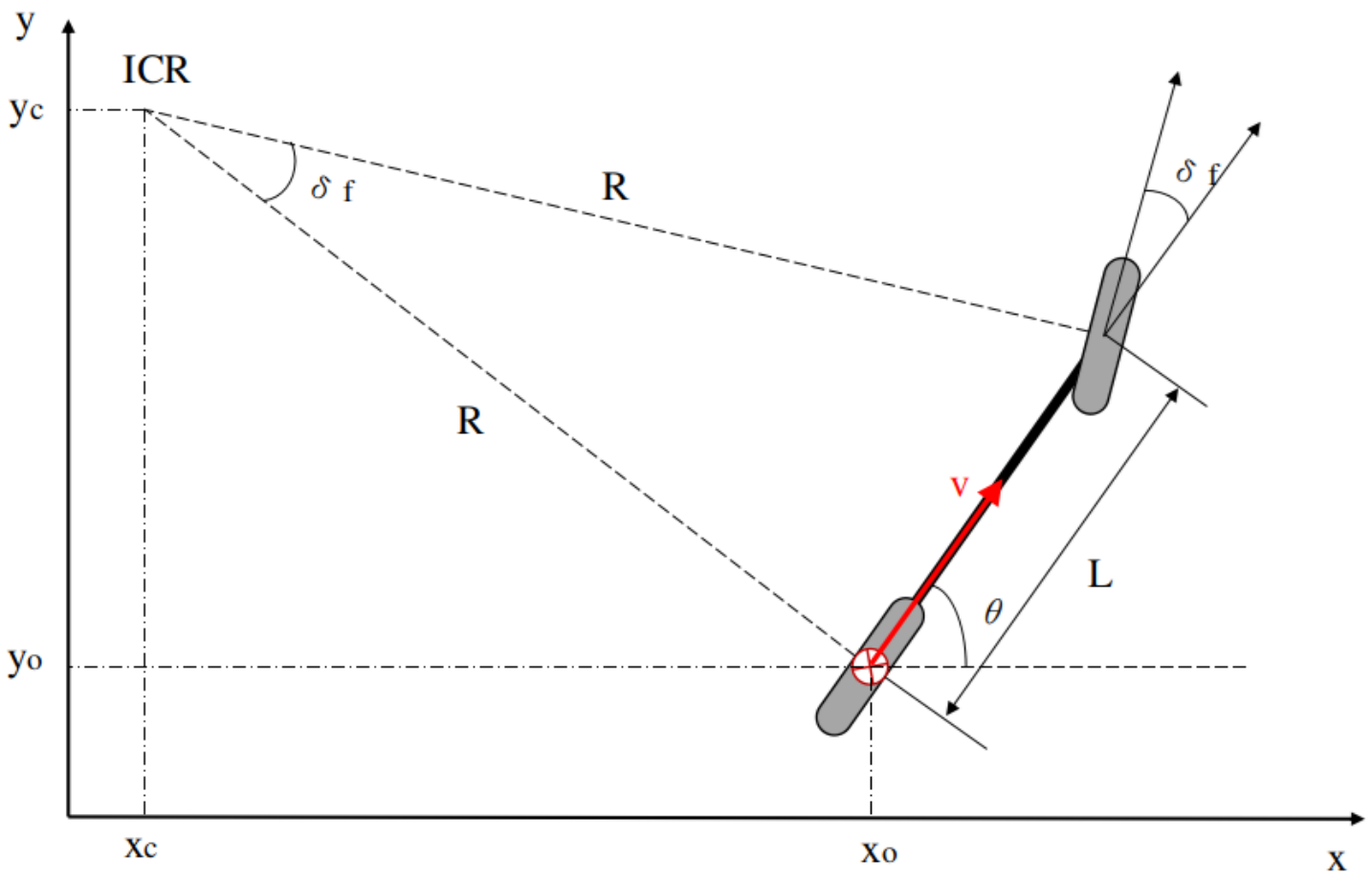

# Steering Wheel to Front Wheel Calibration

| 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 |

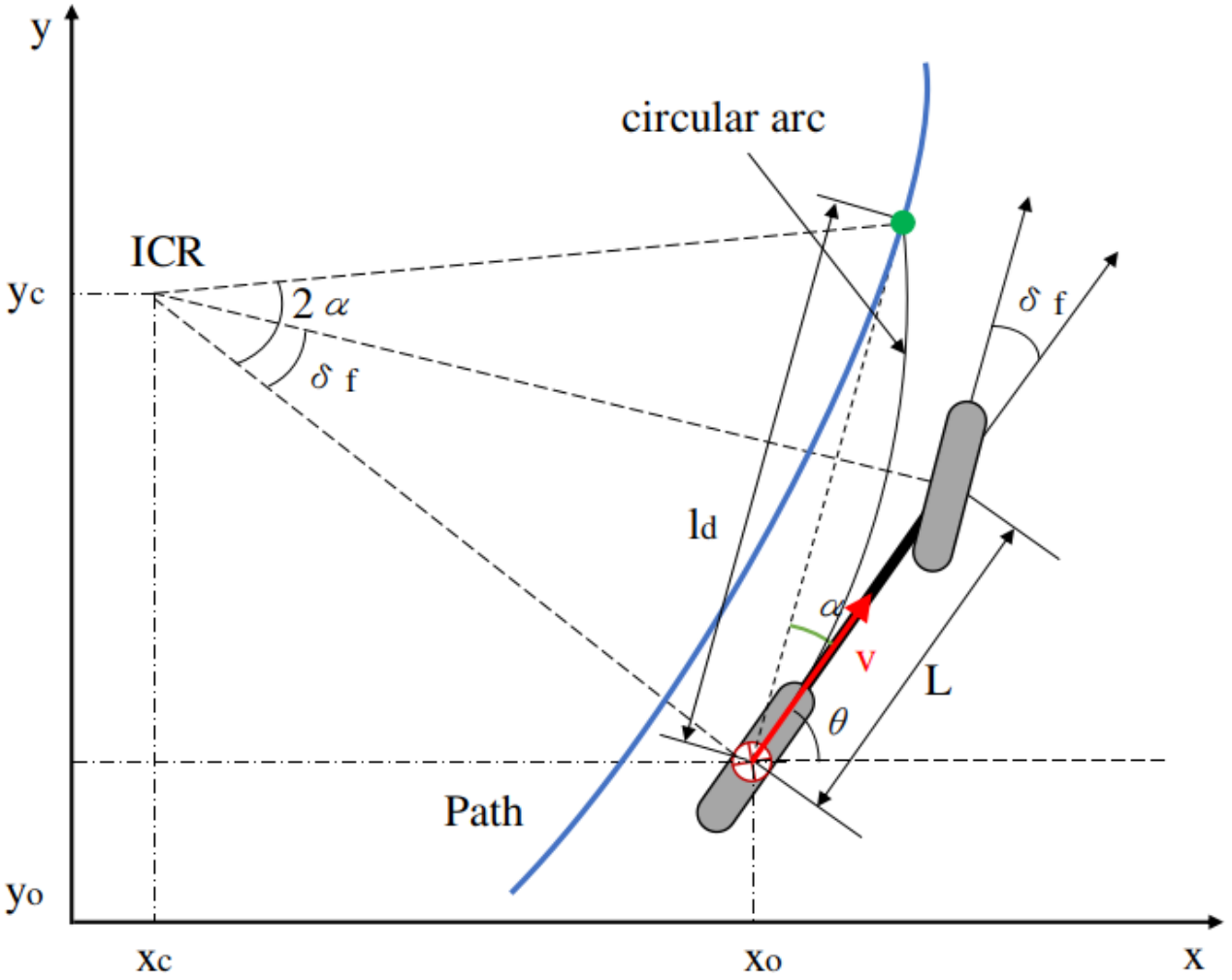
Steering wheel was controlled by program to ratotat at a particular angles, namely 0, 90, 180, 270, 360, ..., etc.



# Lane Detection and Tracking - Bicycle Model



# Lane Detection and Tracking - Pure Pursuit Controller




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

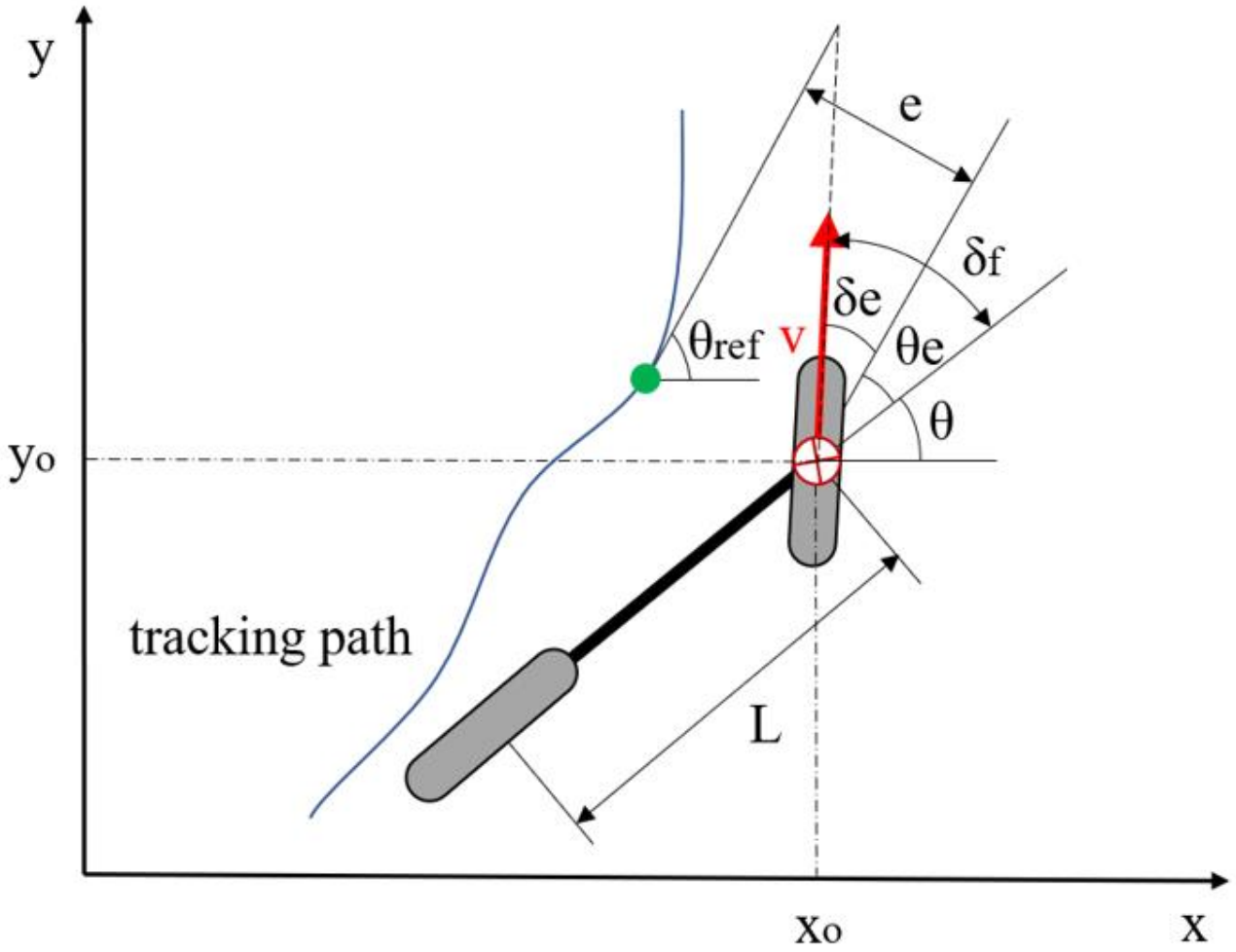
$$\dot{y} = v\sin\theta$$

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

$$\dot{v} = a$$

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

# 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

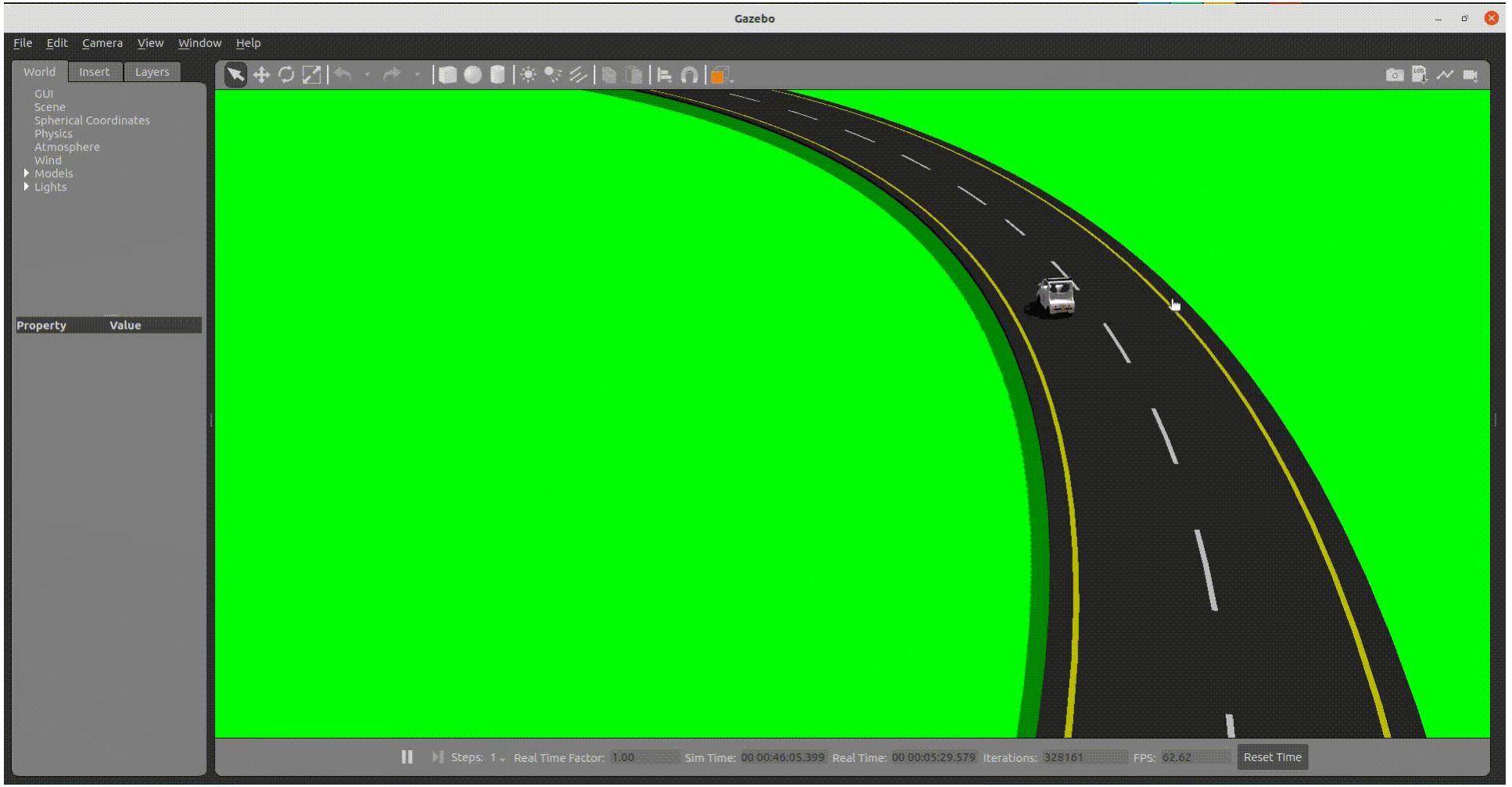
$e$: cross track error

$\theta_e$: heading error

$\delta_e$: cross track steering

# Demo of Pure Pursuit Controller in Simulator

**Github**: https://github.com/hangcui1201/POLARIS_GEM_e2_Simulator

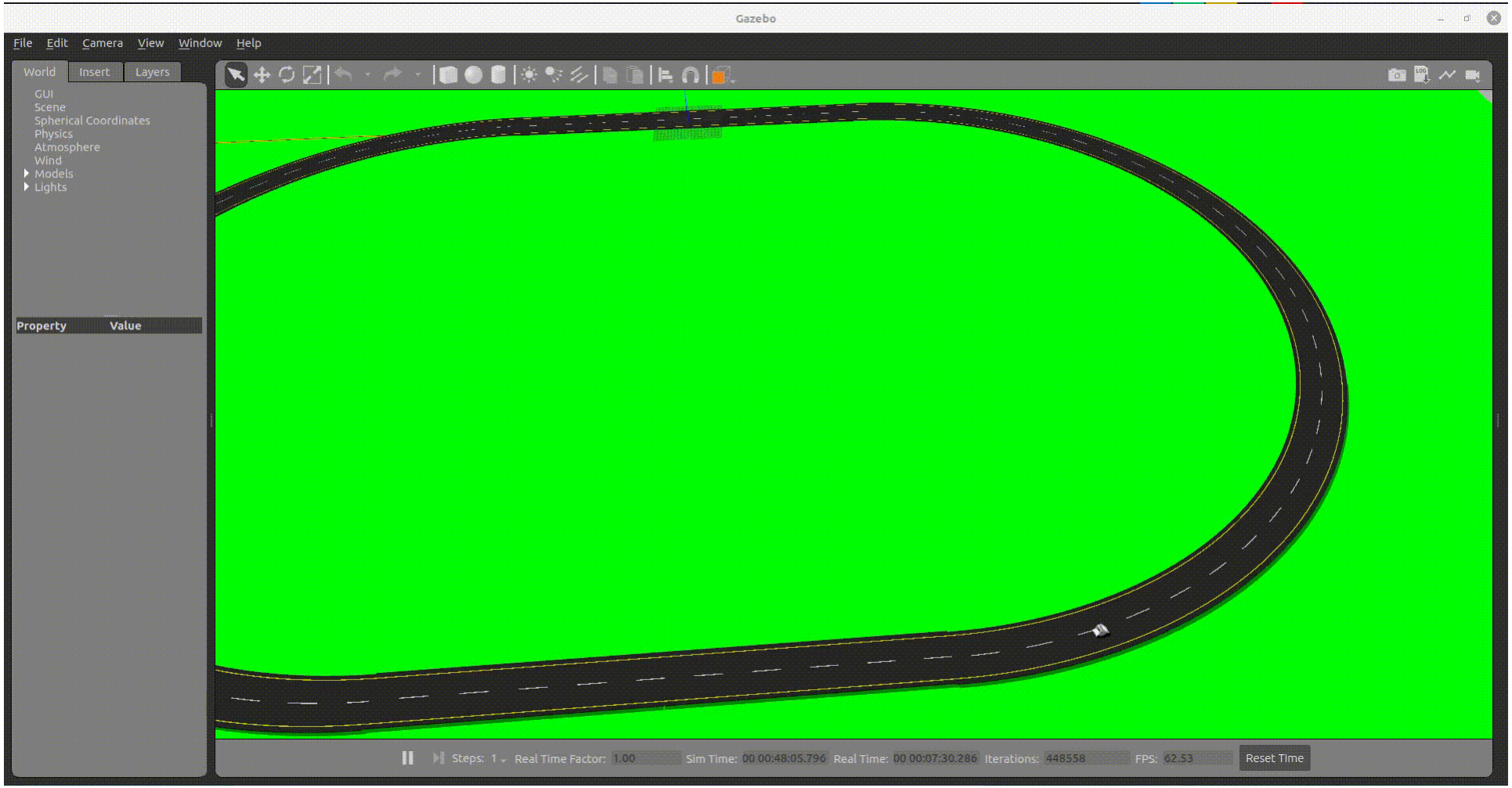


# Demo of Pure Pursuit Controller in Real World

**Github**: https://github.com/hangcui1201/POLARIS_GEM_e2_Real

# Demo of Stanley Controller in Simulator

**Github**: https://github.com/hangcui1201/POLARIS_GEM_e2_Simulator



# Demo of Stanley Controller in Real World

**Github**: https://github.com/hangcui1201/POLARIS_GEM_e2_Real

# Others

Lab Safety Certificate: https://old.drs.illinois.edu/Training/TrainingChecklist

Safety Procedures and Documents:
https://wiki.illinois.edu/wiki/display/GAG/a%29+Safety+roles+and+procedures

Thanks!