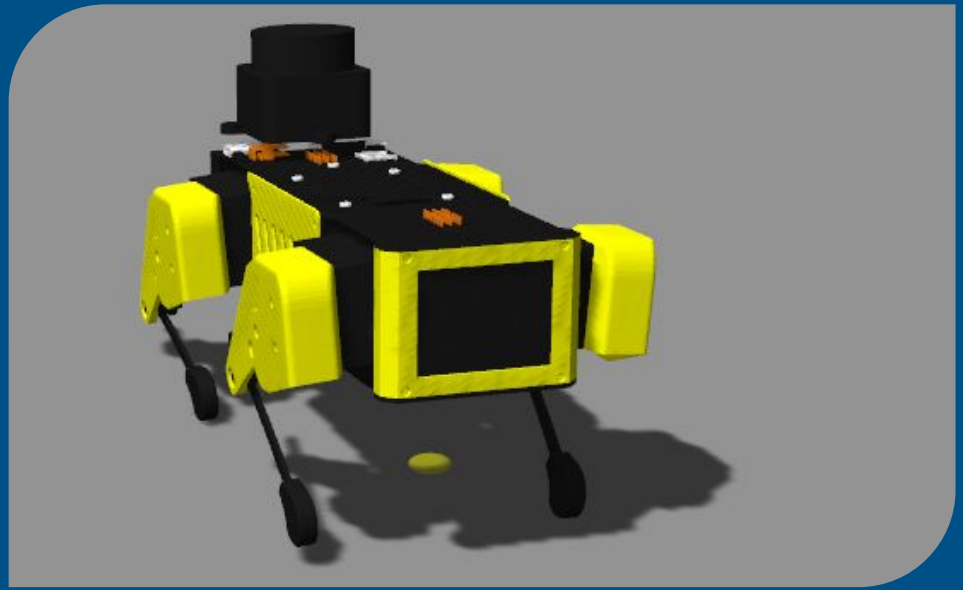
# QuadrupedSim

OOP Course Special Project : Group I

Vimarsh Shah (2022B5A71060G)
Nilesh Bhatia (2023A7PS0418G)
Sudhanshu Kulkarni (2023A7PS0414G)
Priyanshu Singawat (2023A7PS0417G)
Kanishk Rai (2023A7PS1026G)

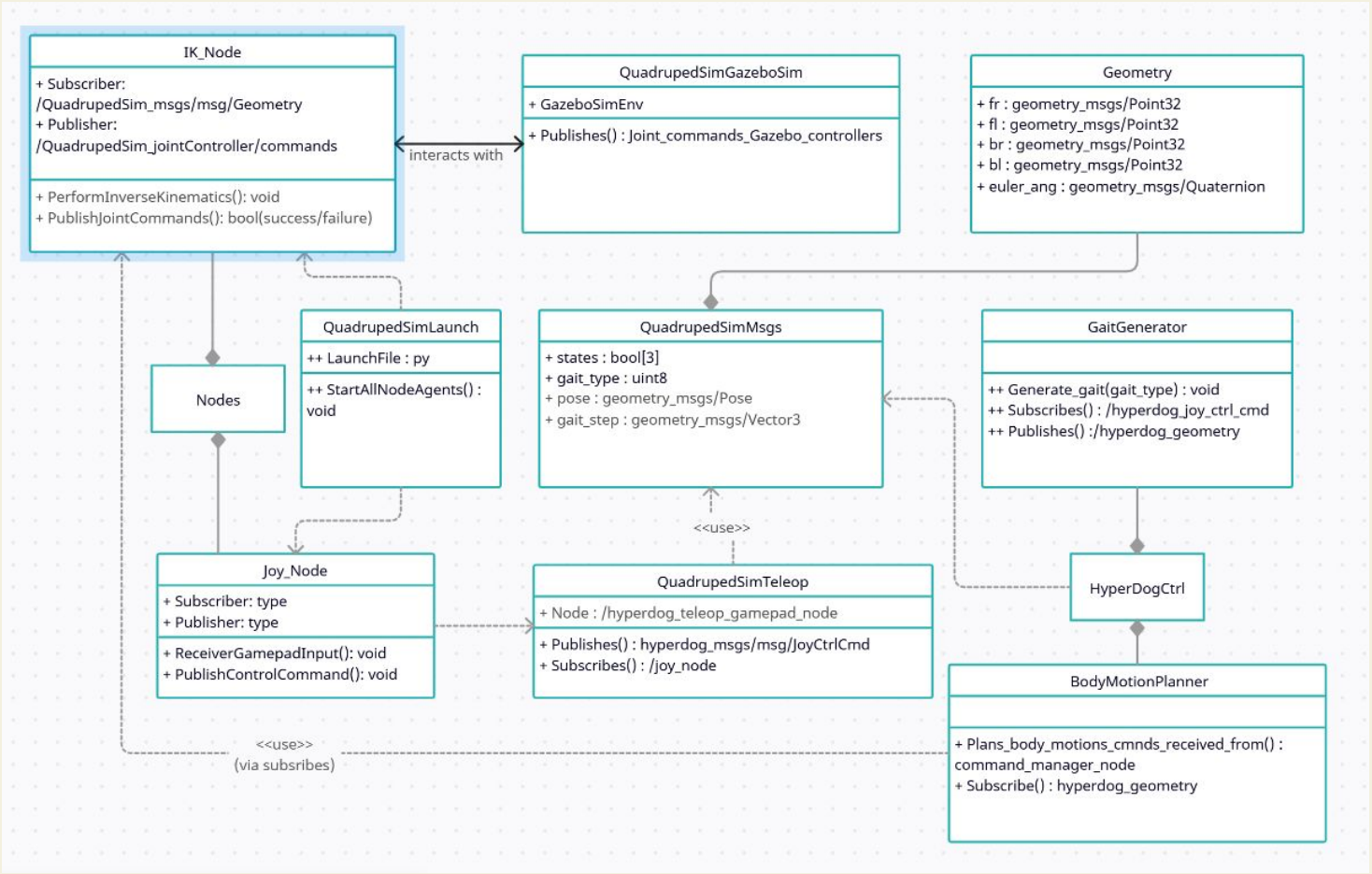


Electronics & Robotics Club

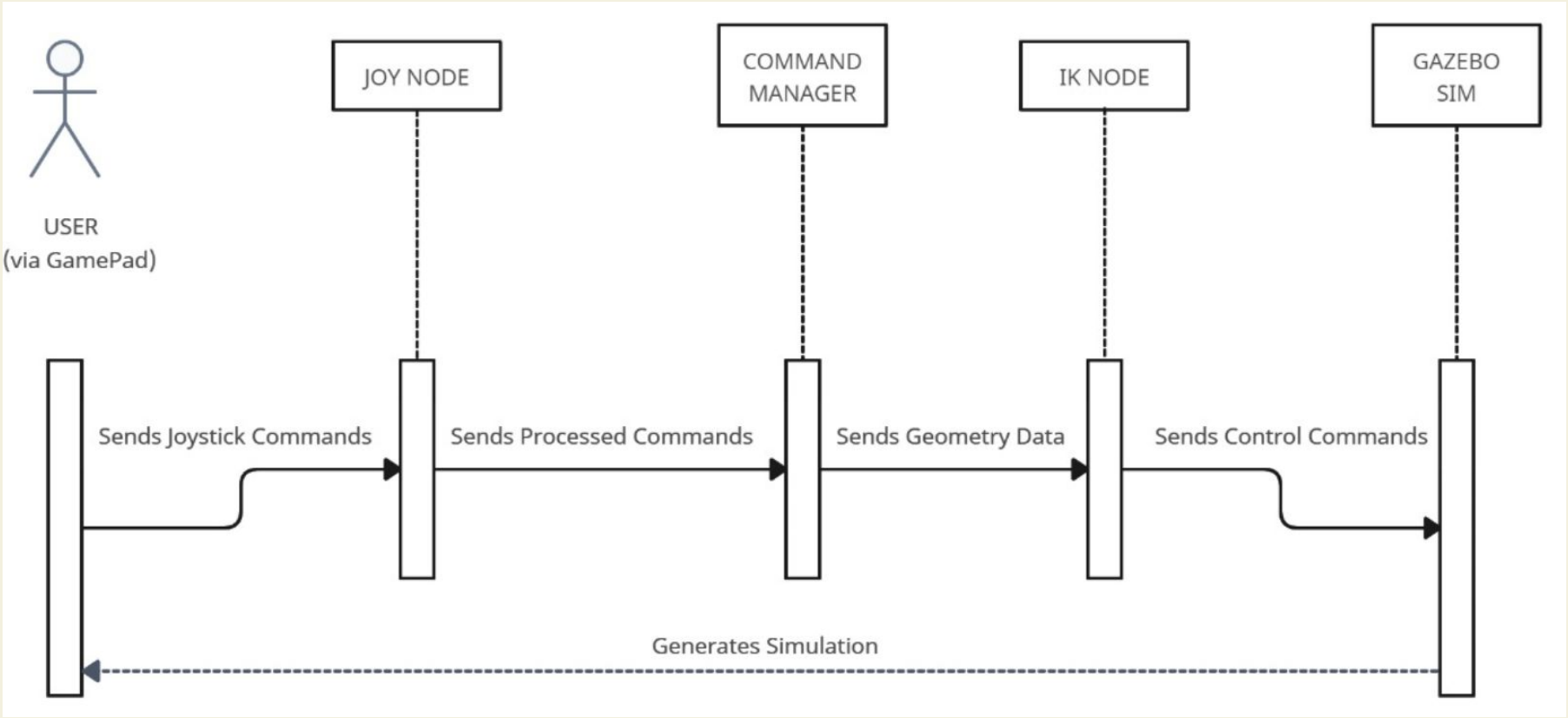
1. UML Diagrams
2. On ROS + Gazebo
3. Simulation Result
4. Reinforcement Learning
5. Future Work

1. **UML Diagrams**
2. On ROS + Gazebo
3. Simulation Result
4. Reinforcement Learning
5. Future Work

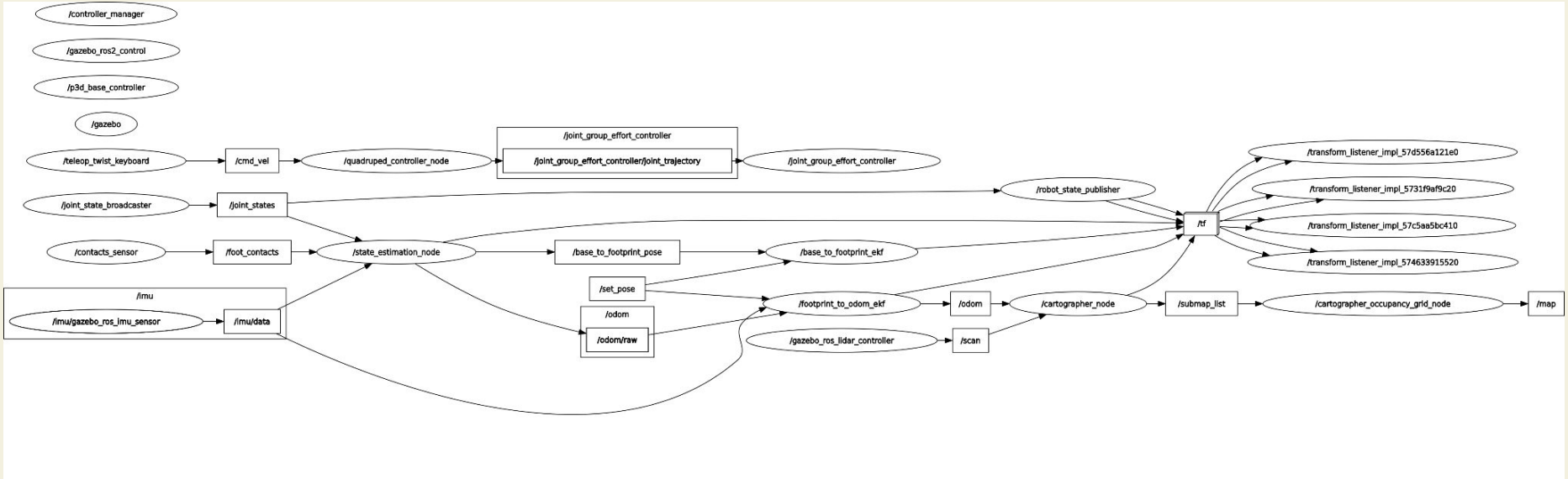
# CLASS DIAGRAM



# SEQUENCE DIAGRAM



RQT Graph


/controller_manager
/gazebo_ros2_control
/p3d_base_controller
/gazebo
/teleop_twist_keyboard
/cmd_vel
/quadruped_controller_node
/joint_group_effort_controller
/joint_group_effort_controller/joint_trajectory
/joint_group_effort_controller
/joint_state_broadcaster
/joint_states
/robot_state_publisher
/contacts_sensor
/foot_contacts
/state_estimation_node
/base_to_footprint_pose
/base_to_footprint_ekf
/imu
/imu/gazebo_ros_imu_sensor
/imu/data
/set_pose
/odom
/odom/raw
/footprint_to_odom_ekf
/odom
/cartographer_node
/submap_list
/cartographer_occupancy_grid_node
/map
/gazebo_ros_lidar_controller
/scan
/tf
/transform_listener_impl_57d556a121e0
/transform_listener_impl_5731f9af9c20
/transform_listener_impl_57c5aa5bc410
/transform_listener_impl_574633915520

# About the Quadruped

**1. Design and Actuation:**

- Developed a quadruped robot with 3 Degrees of Freedom (DOF) per leg, totaling 12 DOF.

- BLDC motors with FOC control were used to actuate the robot's links, enabling smooth and precise movement.

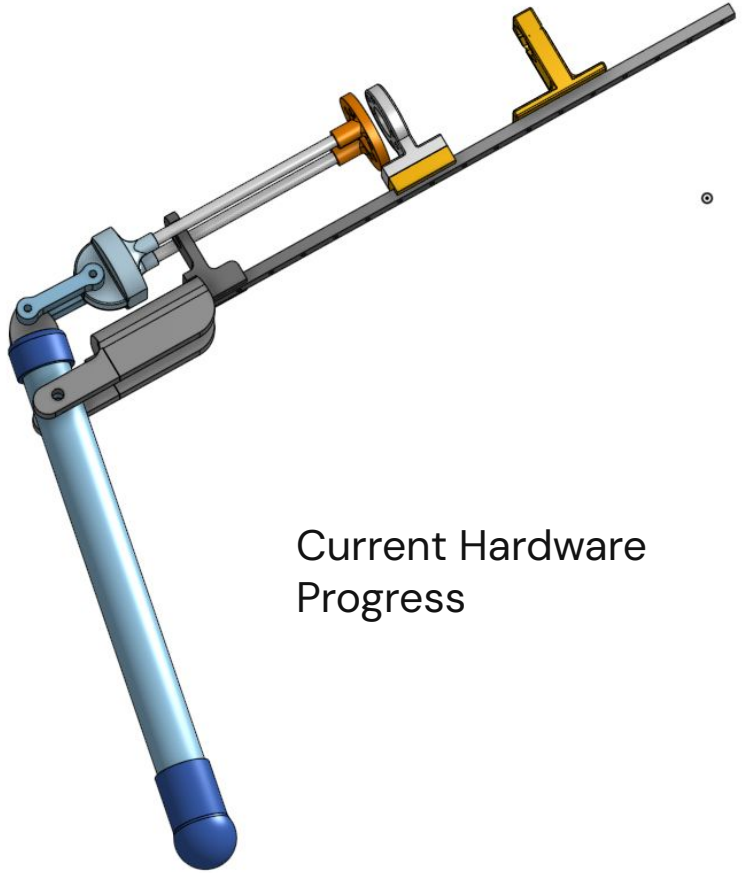
**2. Hardware Challenges:**

- The custom 3 DOF x 4 legs design faced complexity in hardware implementation, particularly in stability and actuation control.

- Shifted to Pupper and its derivative, **Mini Pupper**, developed by Stanford University's researchers, to progress the project.

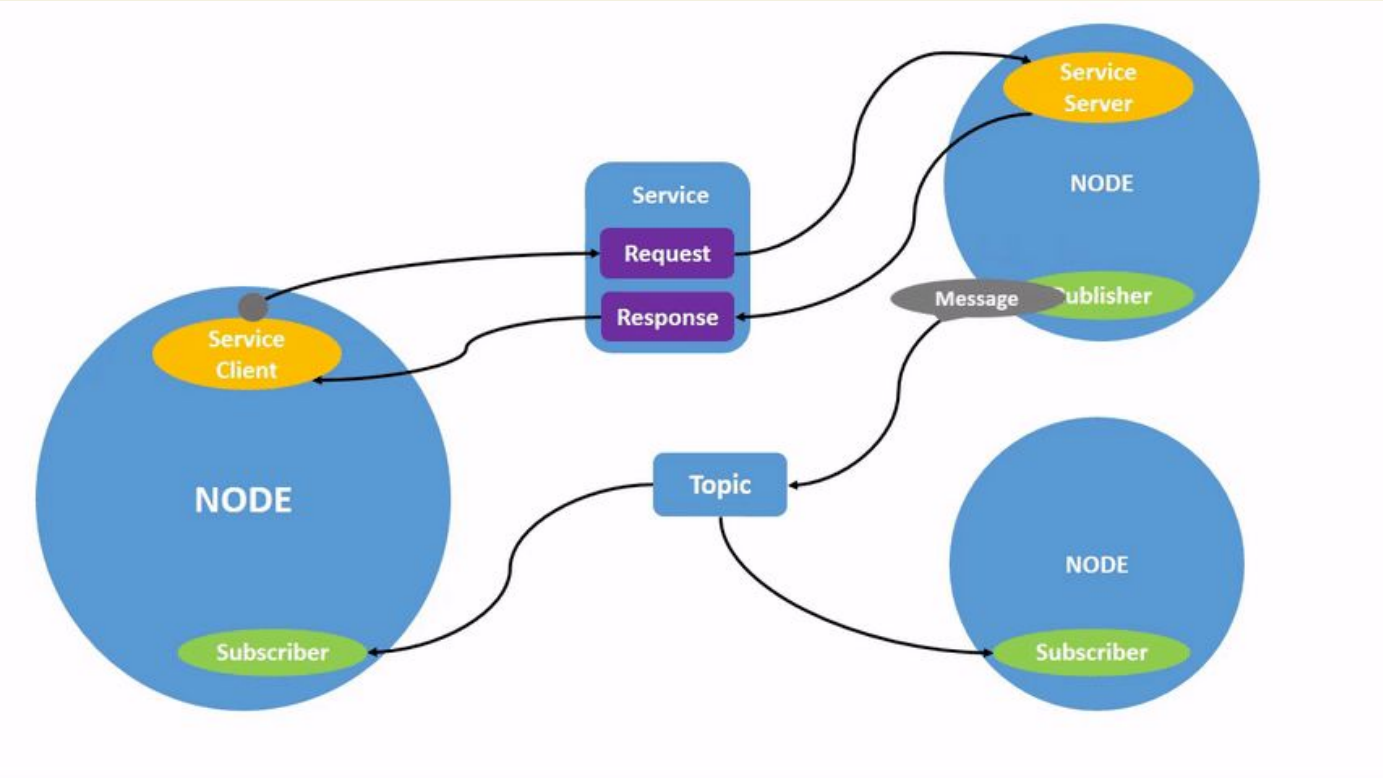
- Lightweight, affordable, and open-source design facilitated rapid prototyping and development of locomotion algorithms.

- Used this and developed custom environment with BITS Goa's Map in gazebo

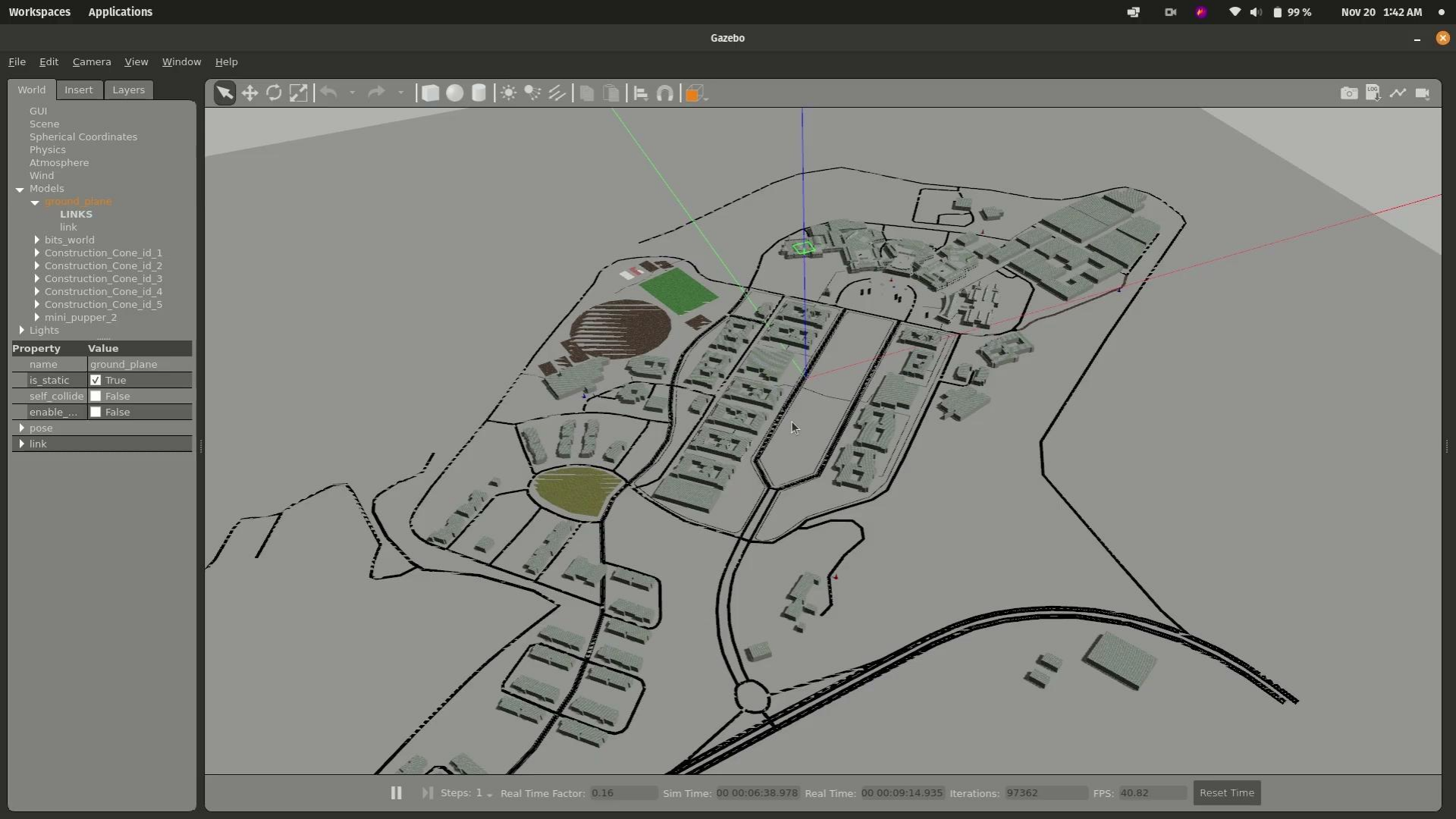


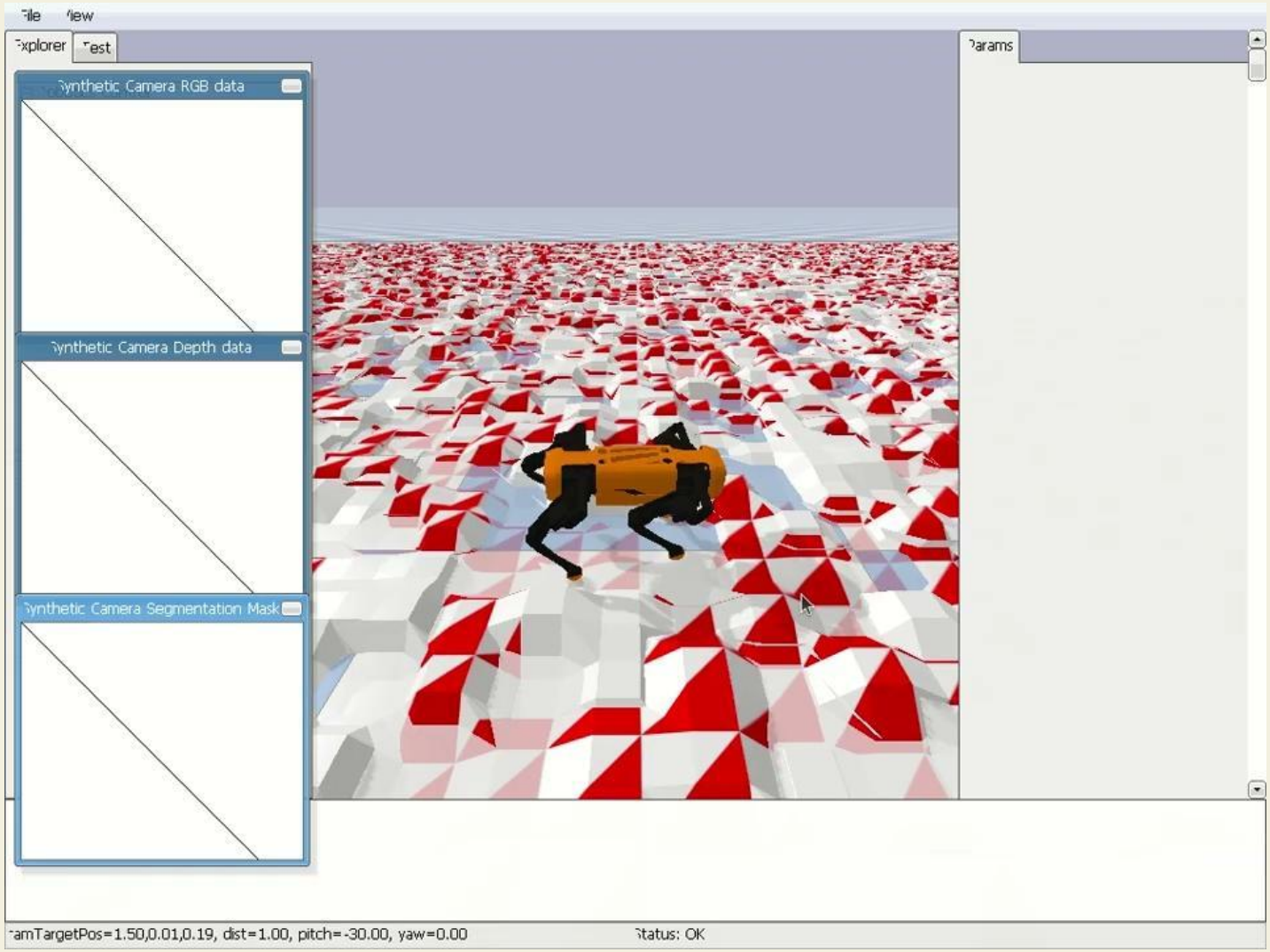
Current Hardware Progress

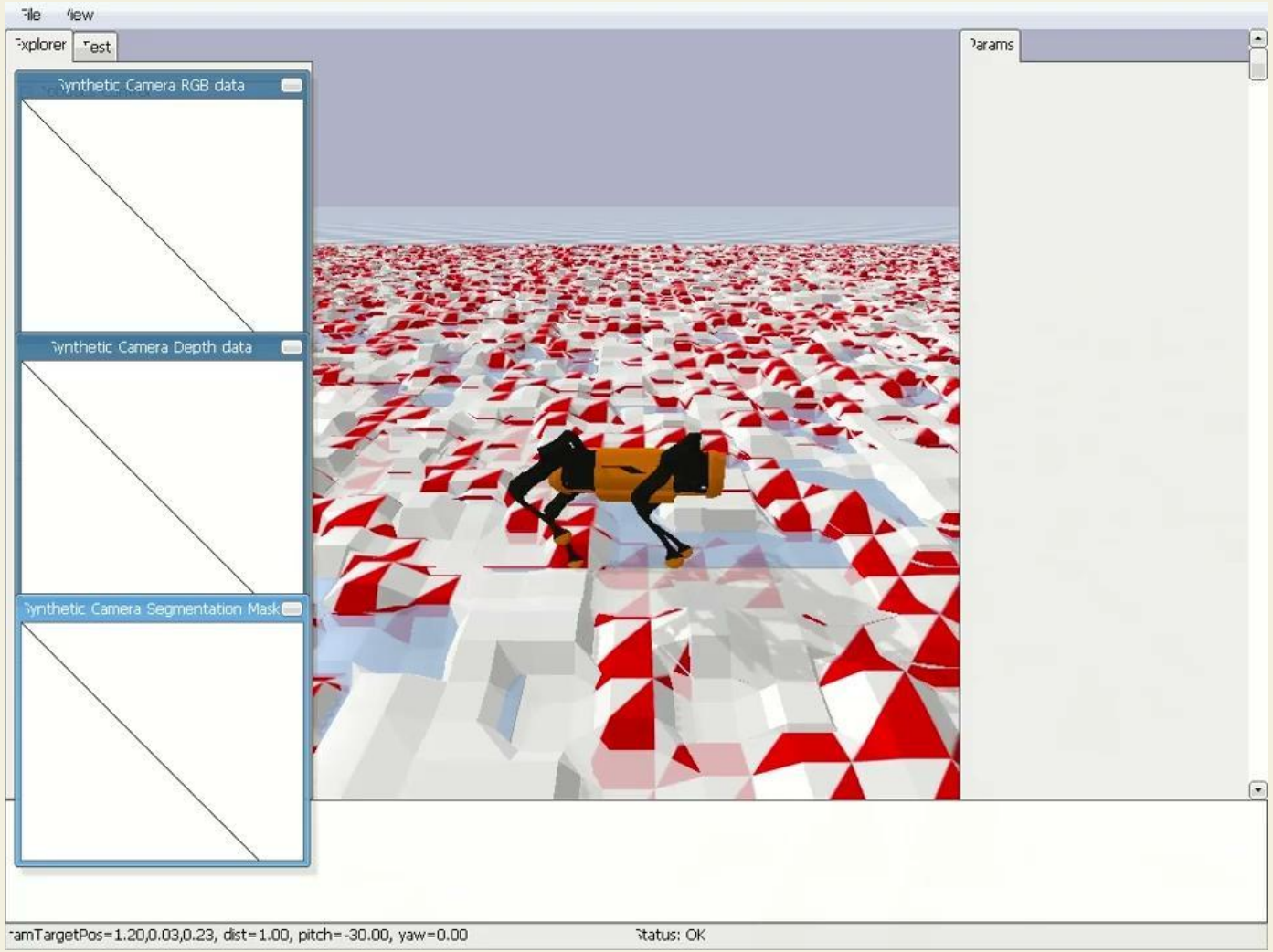
# About ROS (2) in brief



1. UML Diagrams
2. On ROS + Gazebo
3. **Simulation Result**
4. Reinforcement Learning
5. Future Work


Workspaces
Applications
99 %
Nov 20 1:42 AM
Gazebo
File
Edit
Camera
View
Window
Help
World
Insert
Layers
GUI
Scene
Spherical Coordinates
Physics
Atmosphere
Wind
Models
ground_plane
LINKS
link
bits_world
Construction_Cone_id_1
Construction_Cone_id_2
Construction_Cone_id_3
Construction_Cone_id_4
Construction_Cone_id_5
mini_pupper_2
Lights
Property
Value
name
ground_plane
is_static
True
self_collide
False
enable_...
False
pose
link
Steps: 1
Real Time Factor: 0.16
Sim Time: 00 00:06:38.978
Real Time: 00 00:09:14.935
Iterations: 97362
FPS: 40.82
Reset Time

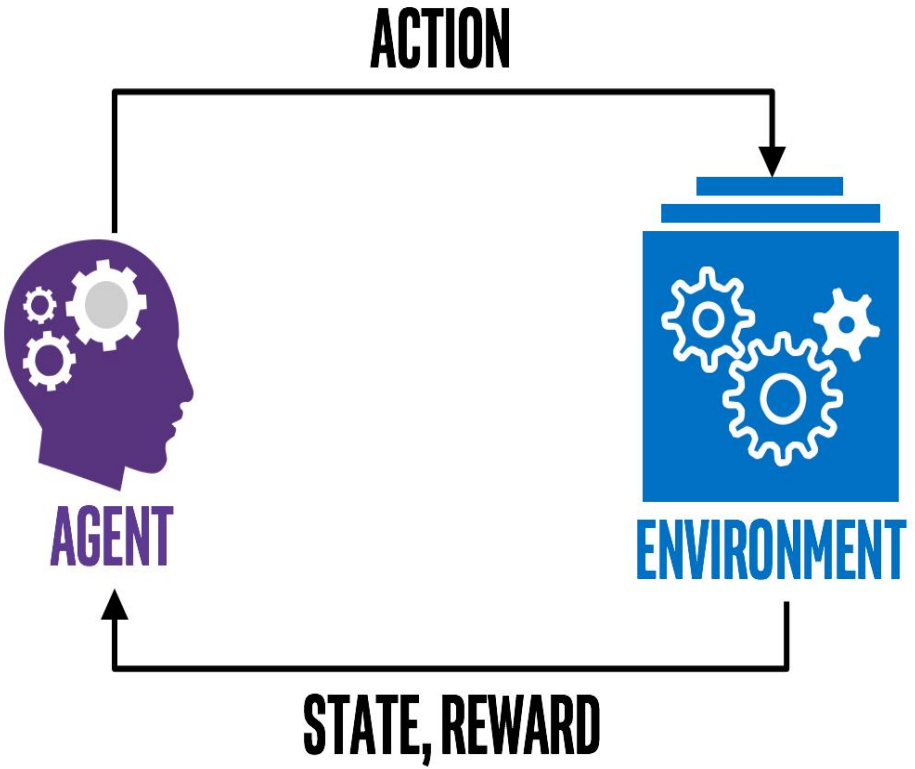

File View
Explorer Test
Synthetic Camera RGB data
Synthetic Camera Depth data
Synthetic Camera Segmentation Mask
Params
amTargetPos=1.50,0.01,0.19, dist=1.00, pitch=-30.00, yaw=0.00
Status: OK


Synthetic Camera RGB data
Synthetic Camera Depth data
Synthetic Camera Segmentation Mask
Params

1. UML Diagrams
2. On ROS + Gazebo
3. Simulation Result
4. **Reinforcement Learning**
5. Future Work

# Reinforcement Learning

Reinforcement Learning (RL) is a machine learning branch where agents learn to make decisions by interacting with their environment, in order to maximize rewards, relying on experience rather than predefined answers.



# RL in short

**Objective:**

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{T}\gamma^t r(s_t, a_t)\right]$$

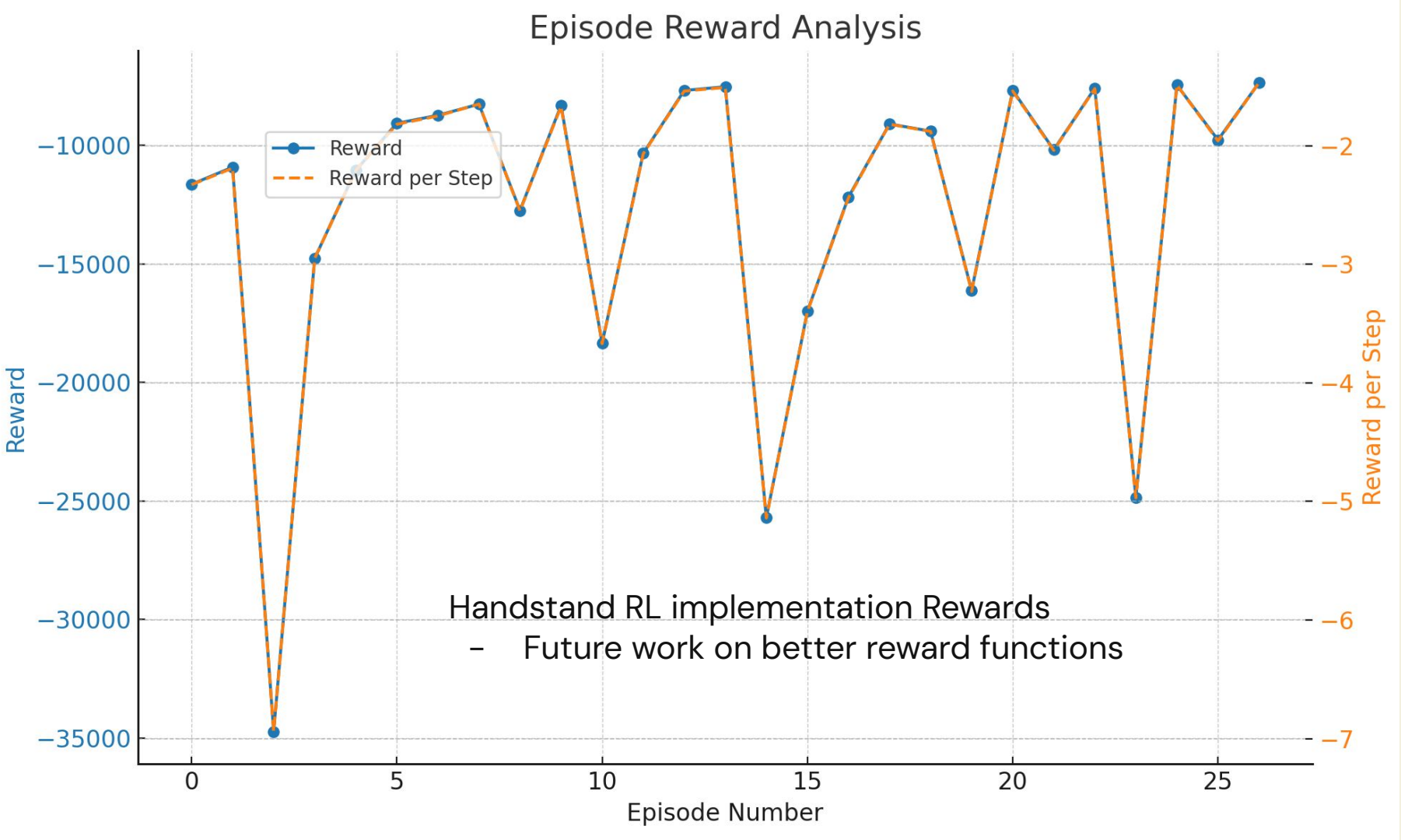
**Proximal Policy Update (PPO):**

$$L^{CLIP}(\theta) = \mathbb{E}_t\left[\min\left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\right)\right]$$

where: $r_t(\theta) = \dfrac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}\hat{A}_t = \text{Advantage Function}$

**Value Function Update:**

$$L^{VF}(\theta) = \mathbb{E}_t\left[\left(V_\theta(s_t) - R_t\right)^2\right]$$


Episode Reward Analysis
Reward
Reward per Step
−10000
−15000
−20000
−25000
−30000
−35000
−2
−3
−4
−5
−6
−7
Reward per Step
0
5
10
15
20
25
Episode Number
Handstand RL implementation Rewards
- Future work on better reward functions

1. UML Diagrams
2. On ROS + Gazebo
3. Simulation Result
4. Reinforcement Learning
5. **Future Work**

# Future Work

- Making the model more capable to navigate via jumps, gallop gait, other gait patterns, etc.
- Performance Optimization
- Better RL model performance.
- Implementing it on hardware (quadruped model).

Open Sourced:
https://github.com/vimarsh244/QuadrupedSim

# References


1. Mini Pupper ROS :

   MangDang Robotics Club, *Mini Pupper ROS*, [GitHub Repository].

2. Stanford Pupper:

   Stanford University, *Pupper: An Inexpensive & Open-source Quadruped Robot*.

3. HyperDog Paper:

   Mudalige et al., *HyperDog: An Open-Source Quadruped Robot Platform Based on ROS2 and micro-ROS*, arXiv:2209.09171 (2022),

4. ROS2 and Gazebo:

   *ROS 2 Humble* and *Gazebo Simulation Tools*, Open Robotics.

5. PyBullet:

   Erwin Coumans, *PyBullet Physics Simulation Framework*, [GitHub Repository]