Tribhuvan University
Institute of Engineering, Pulchowk Campus
Department of Mechanical and Aerospace Engineering
Pulchowk, Lalitpur

DEVELOPMENT AND TESTING OF AUTONOMOUS MOBILE ROBOT FOR MATERIAL HANDLING

Project Supervisor: Assistant Professor Dr. Surya Prasad Adhikari

Presented By:

Nirmal Prasad Panta (076BME025)

Pawan Shrestha (076BME027)

Prince Panta (076BME029)

Saki Basnet (076BME036)

Outline

- ❑ Introduction
- ❑ Problem Statement
- ❑ Scope
- ❑ Objectives
- ❑ Literature Review
- ❑ Methodology
- ❑ Output
- ❑ Work Completed
- ❑ Limitations
- ❑ Problems Faced
- ❑ Budget Analysis
- ❑ Work Schedule (Gantt Chart)
- ❑ Conclusion and Future Enhancement
- ❑ References

Introduction

- Agile corporations leveraging robots and autonomous technologies will thrive.[1]

- Automation has ability to tackle problems like labor shortage, safety risks.

- Flexible manufacturing systems can bring tremendous economic advantages, where AMR becomes of at most importance.

- AMRs transport raw materials, semi-finished products and products from inventory, job stations and assembly line to respective place.

Objectives

Main Objective

To develop an autonomous mobile robot for autonomous transportation
using
Simultaneous Localization and Mapping (SLAM) and Nav2 for mapping, localization, and navigation and computer vision (CV)
for Material handling

Specific Objectives

- To create an AMR prototype using conventional control for navigation

- To employ SLAM algorithm for mapping and Nav2 to enhance the AMR’s autonomous navigation, precise localization, and effective obstacle avoidance

- To integrate computer vision technology to augment the AMR's material handling capabilities, enabling it to accurately identify and handle payloads.

Literature review

Evolution of AGV to AMR	G. Ullrich, The History of Automated Guided Vehicle Systems, 2015
	W. Grzechca, Manufacturing in Flow Shop and Assembly Line Structure, 2016
	N. Zghair et.al , A one decade survey of autonomous mobile robot systems, 2021
	L. Lynch et al., Integration of autonomous intelligent vehicles into manufacturing environments: Challenges, 2020
ROS2	S. Macenski et.al , Robot Operating System 2: Design, architecture, and uses in the wild, 2022 (ROS 2 Documentation in https://docs.ros.org/)
SLAM	S. Macenski and I. Jambrecic, SLAM Toolbox: SLAM for the dynamic world, 2021
Nav2	S. Macenski, et.al, The Marathon 2: A Navigation System, 2020 (Nav2 Documentation in https://navigation.ros.org/)
OpenCV	J. Wubben et.al, Accurate Landing of Unmanned Aerial Vehicles Using Ground Pattern Recognition, 2019
	G. Bradski, The OpenCV Library, 2000.

Methodology

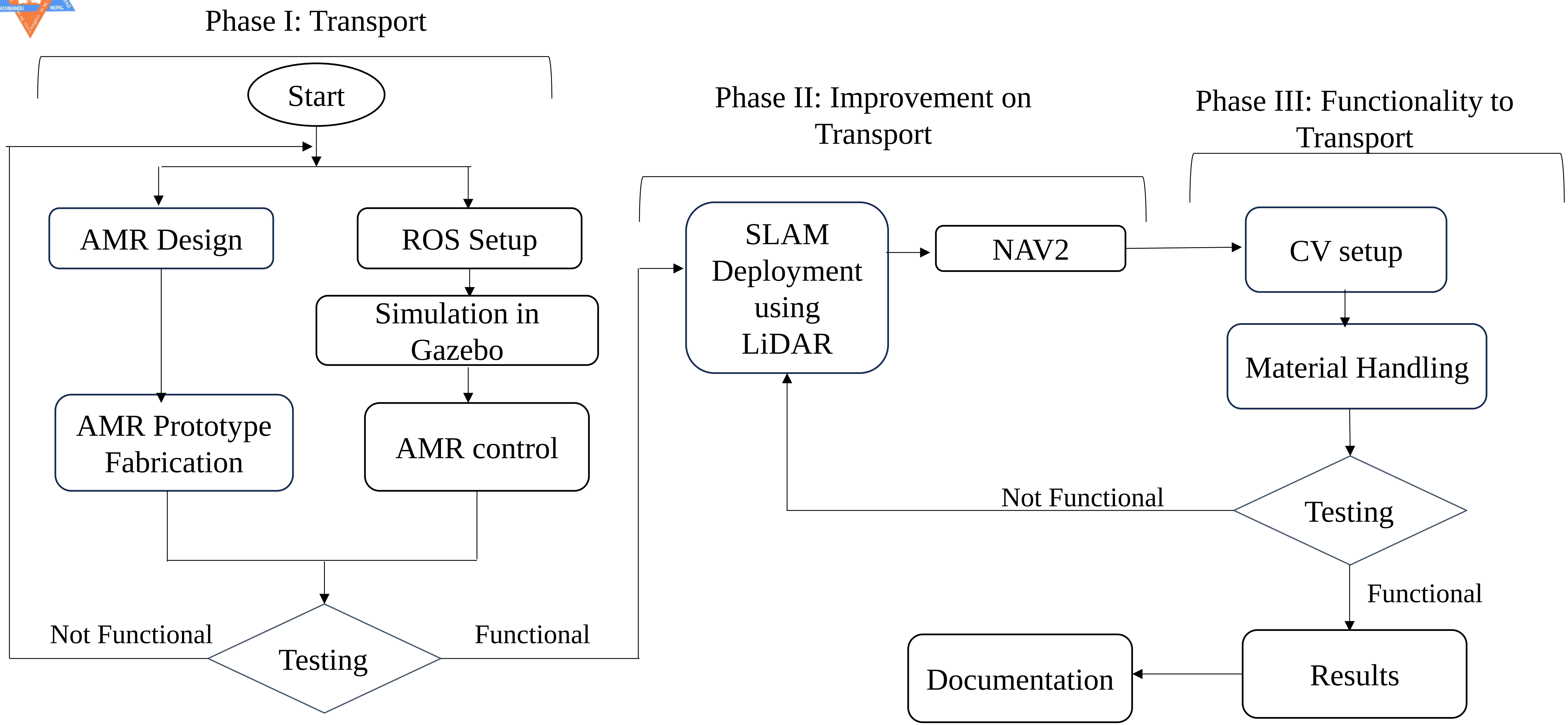


Figure 1: Methodology Flow Chart

AMR CAD Model

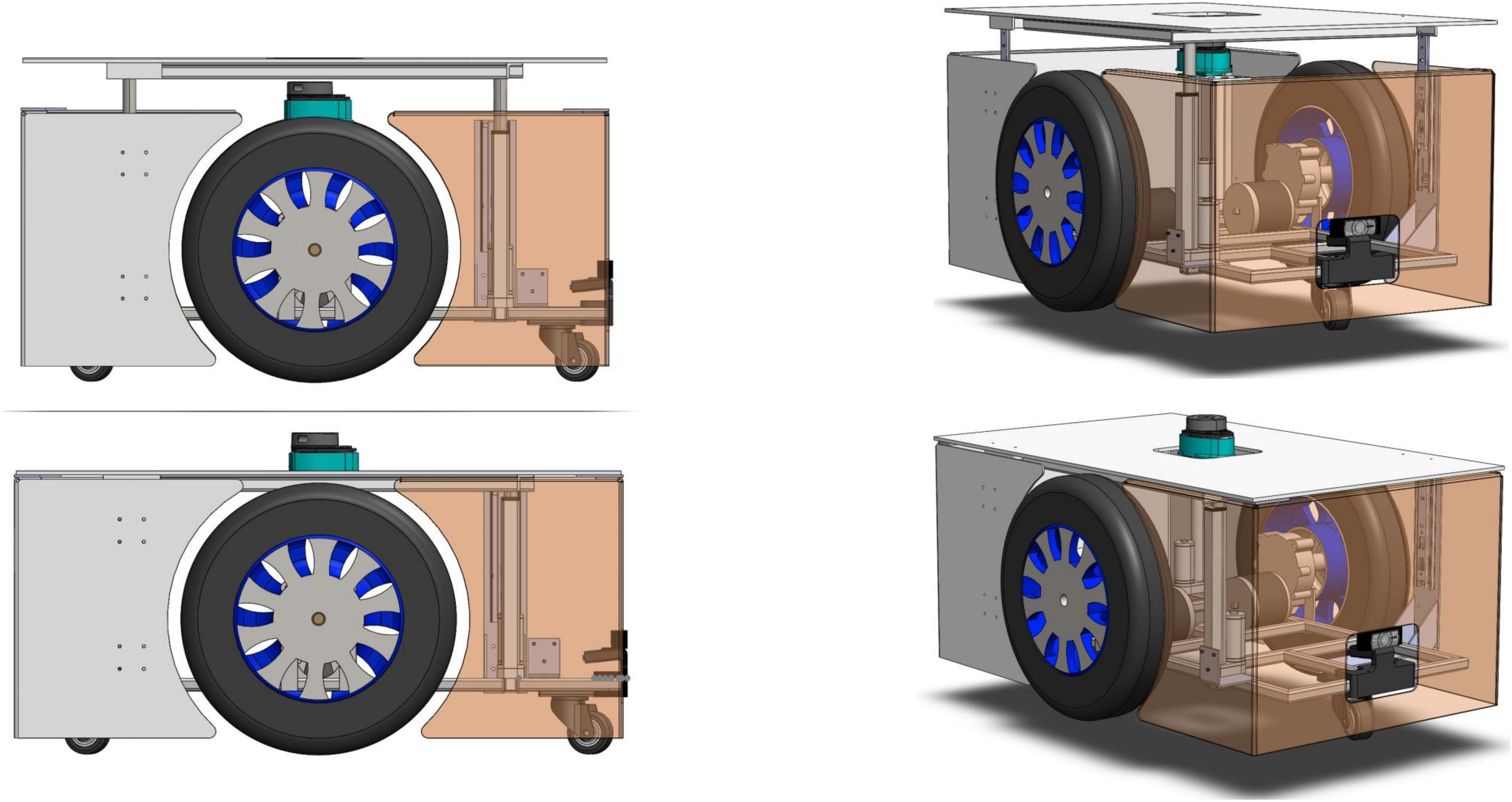
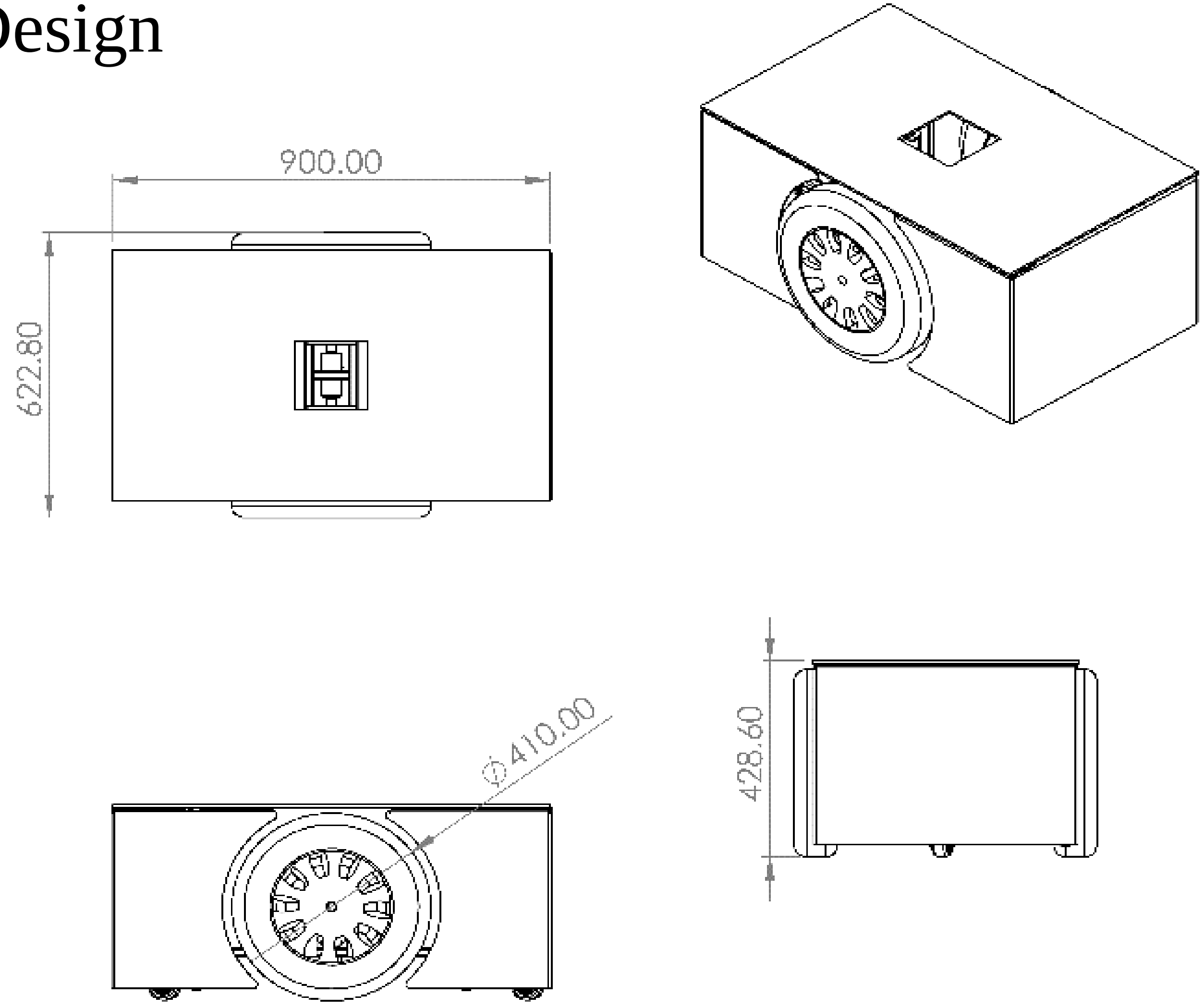


Figure 2: AMR prototype CAD model

AMR Design



Max payload capacity : 127.7kg
Motor torque : 125 kg.cm
Weight of robot : 30.8 kg

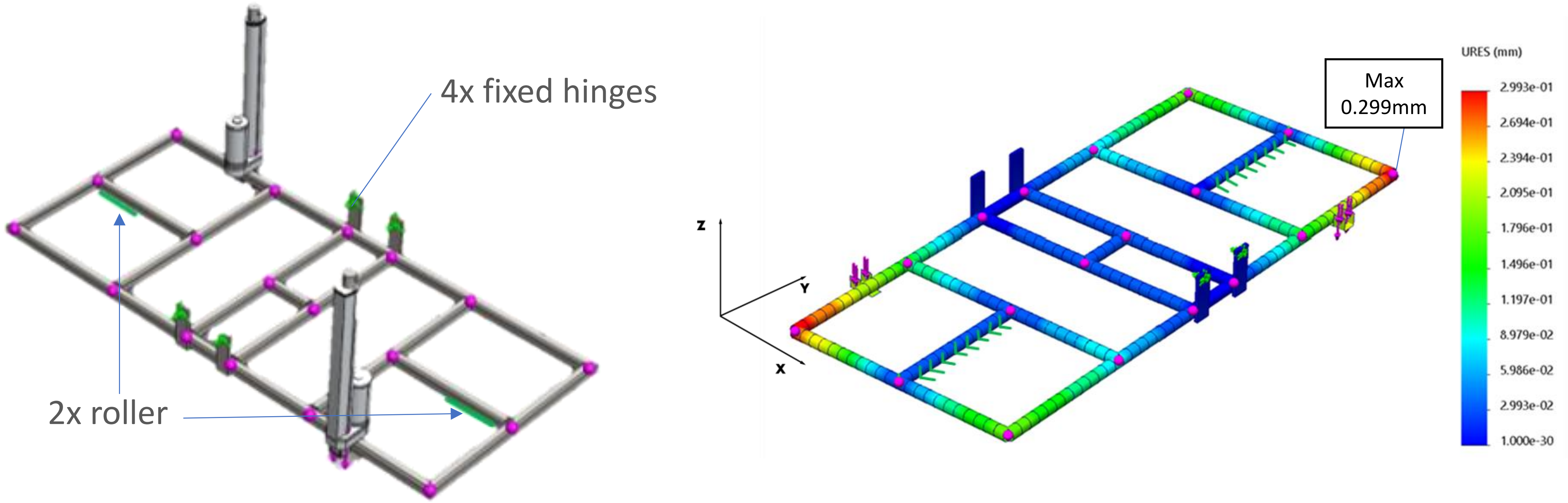
Figure 3 : AMR Dimensions

Table 1: Material Properties

Name:	**Plain Carbon Steel**
Model type:	**Linear Elastic Isotropic**
Yield strength:	**2.20594e+08 N/m^2**
Tensile strength:	**3.99826e+08 N/m^2**
Elastic modulus:	**2.1e+11 N/m^2**
Poisson's ratio:	**0.28**
Mass density:	**7,800 kg/m^3**
Shear modulus:	**7.9e+10 N/m^2**

- Static structural simulation results show the frame deflecting under load, with maximum displacement of 0.2993 millimeters
- The maximum von Mises stress of 115.4 MPa
- The minimum Factor of Safety is 1.9

AMR Static Study



Load in -Z from each actuator: 50kg

Figure 4: Model reference and the displacement plot

AMR Chassis Fabrication

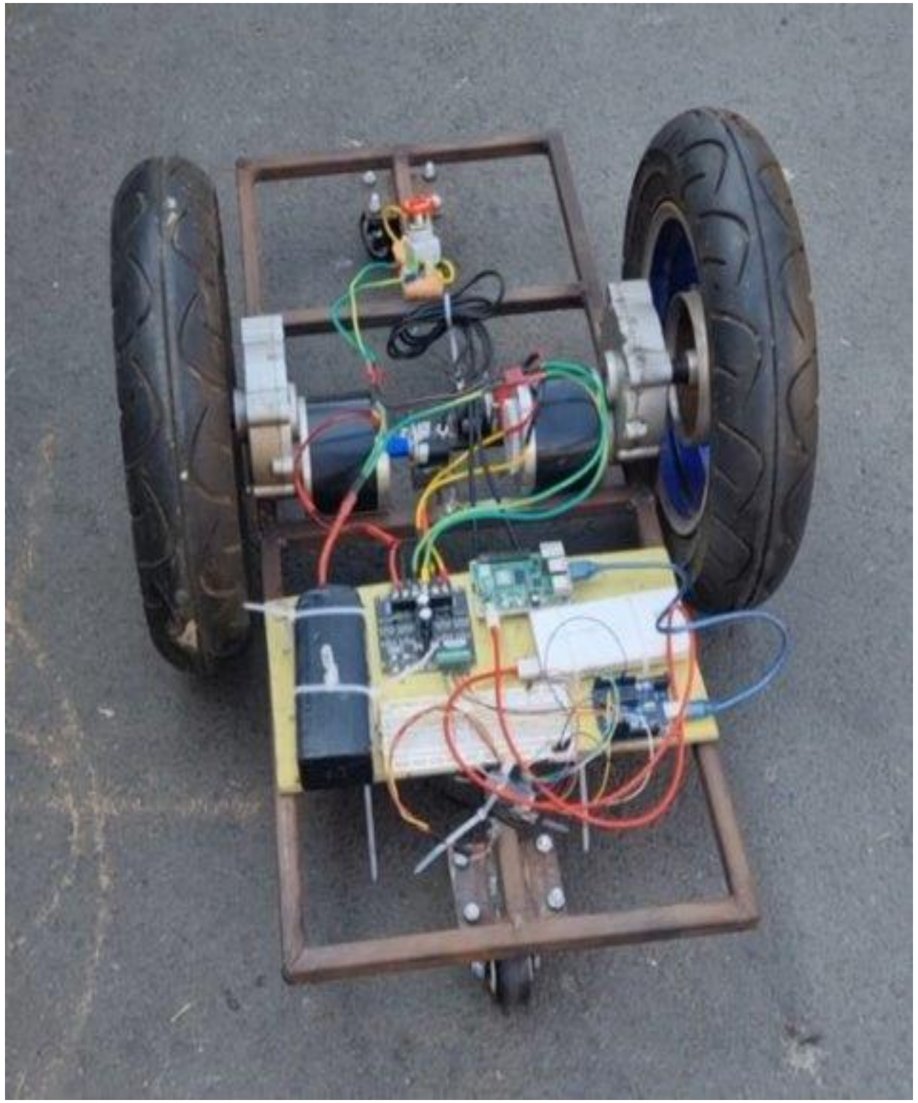


Figure 5: Chassis Fabrication

Simulation of AMR in ROS 2

- Package project_amr created and built with colcon
- Launch files for the simulation created
- Simplified robot model created with the help of urdf and xacro
- Gazebo reference and sensor also added in the urdf file for simulation in gazebo and RVIZ
- SLAM and Nav2 documentation utilized for mapping and navigation

Creation of Robot Model

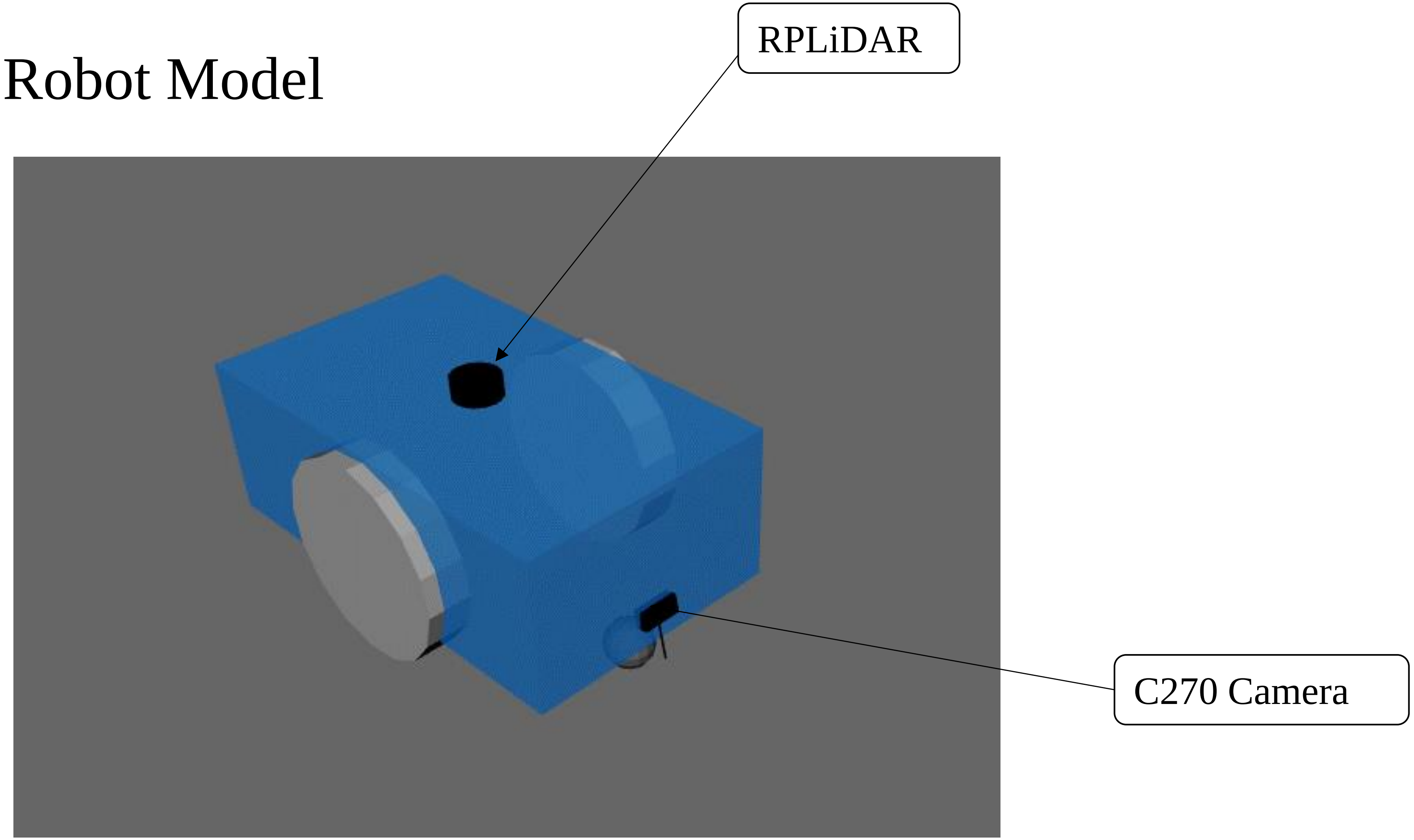


Figure 6 : Creation of simplified model for LIDAR and camera

Simulation of AMR in ROS 2

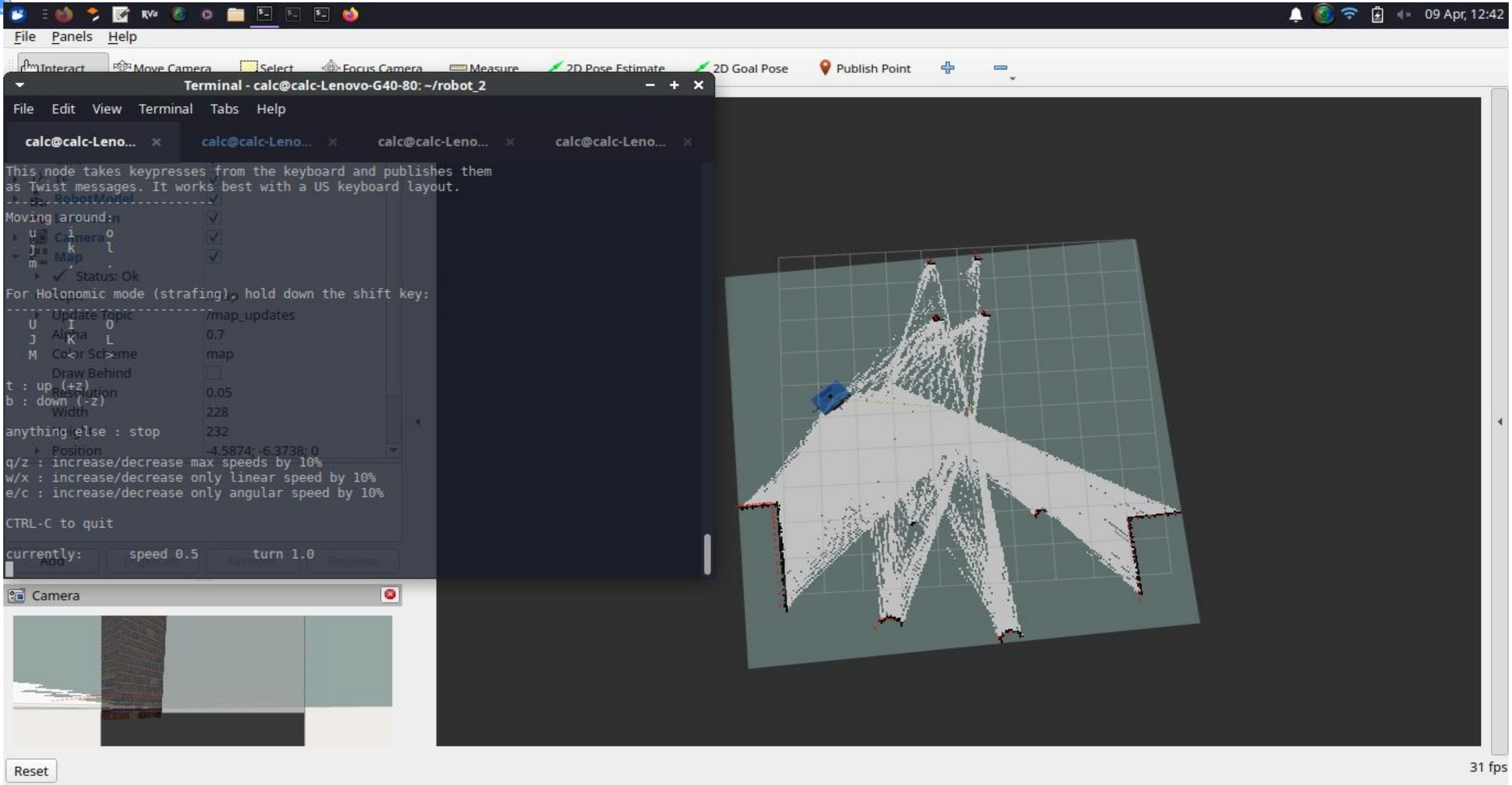
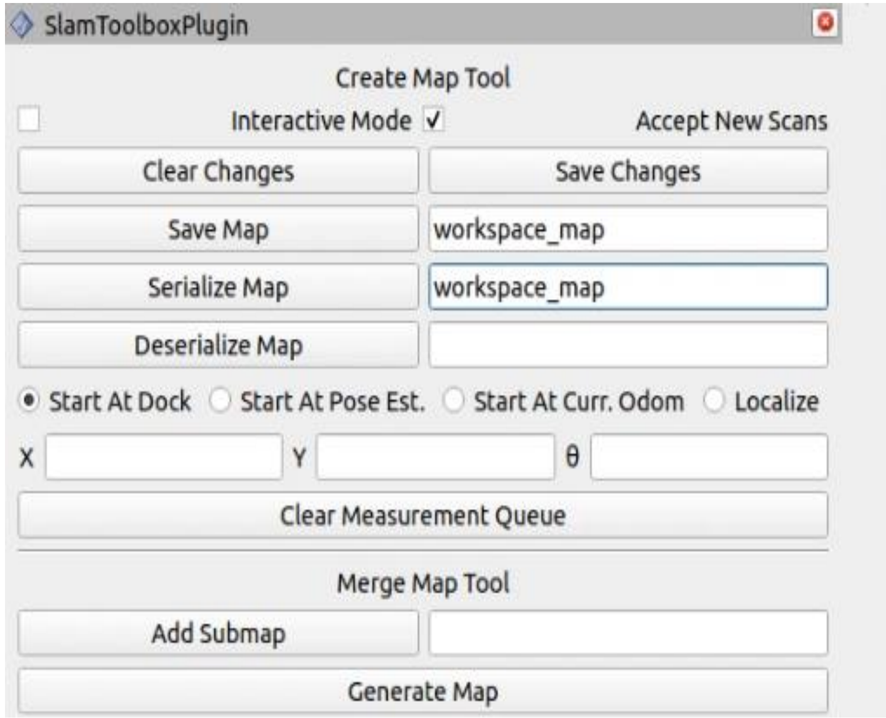


Figure 7: Creating map of virtual workspace using SLAM

SLAM



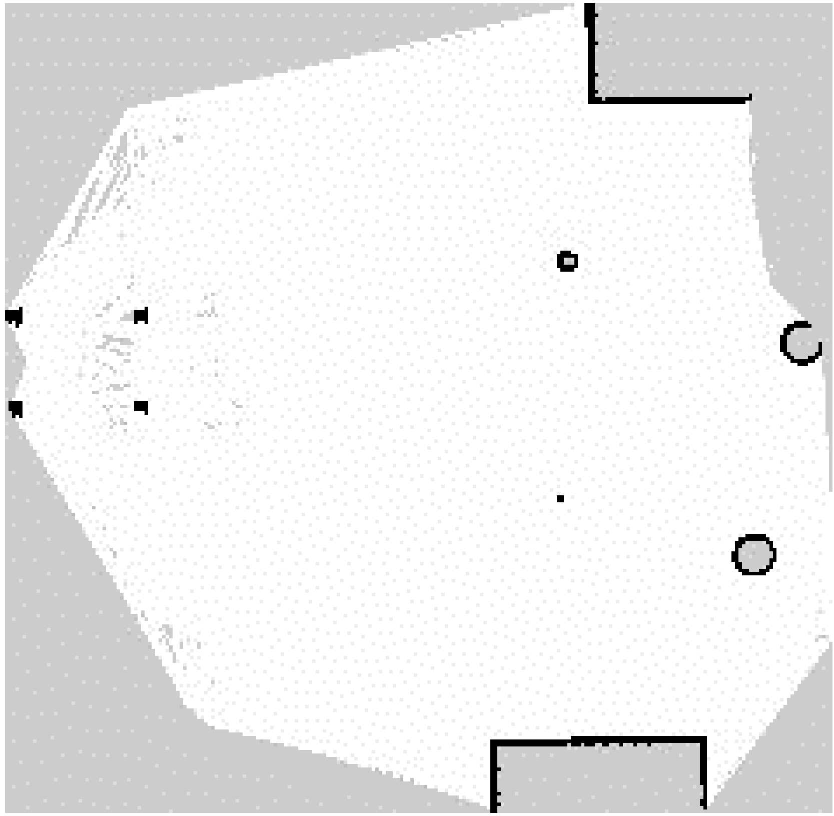


Figure 8 : Use of SlamToolboxPlugin to create workspace_map

Nav 2

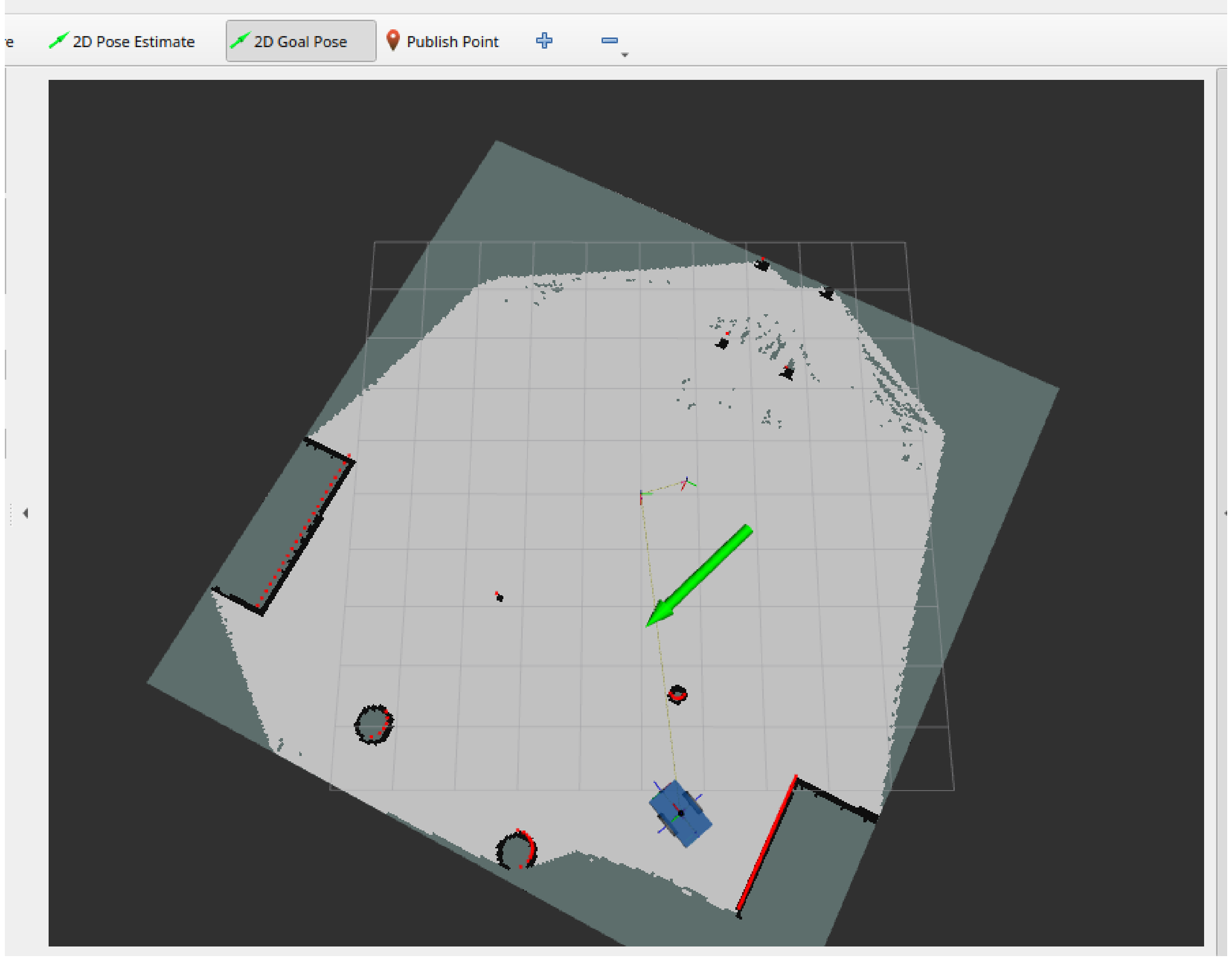


Figure 9 : Giving goal pose using Nav2

AMR Remote Control

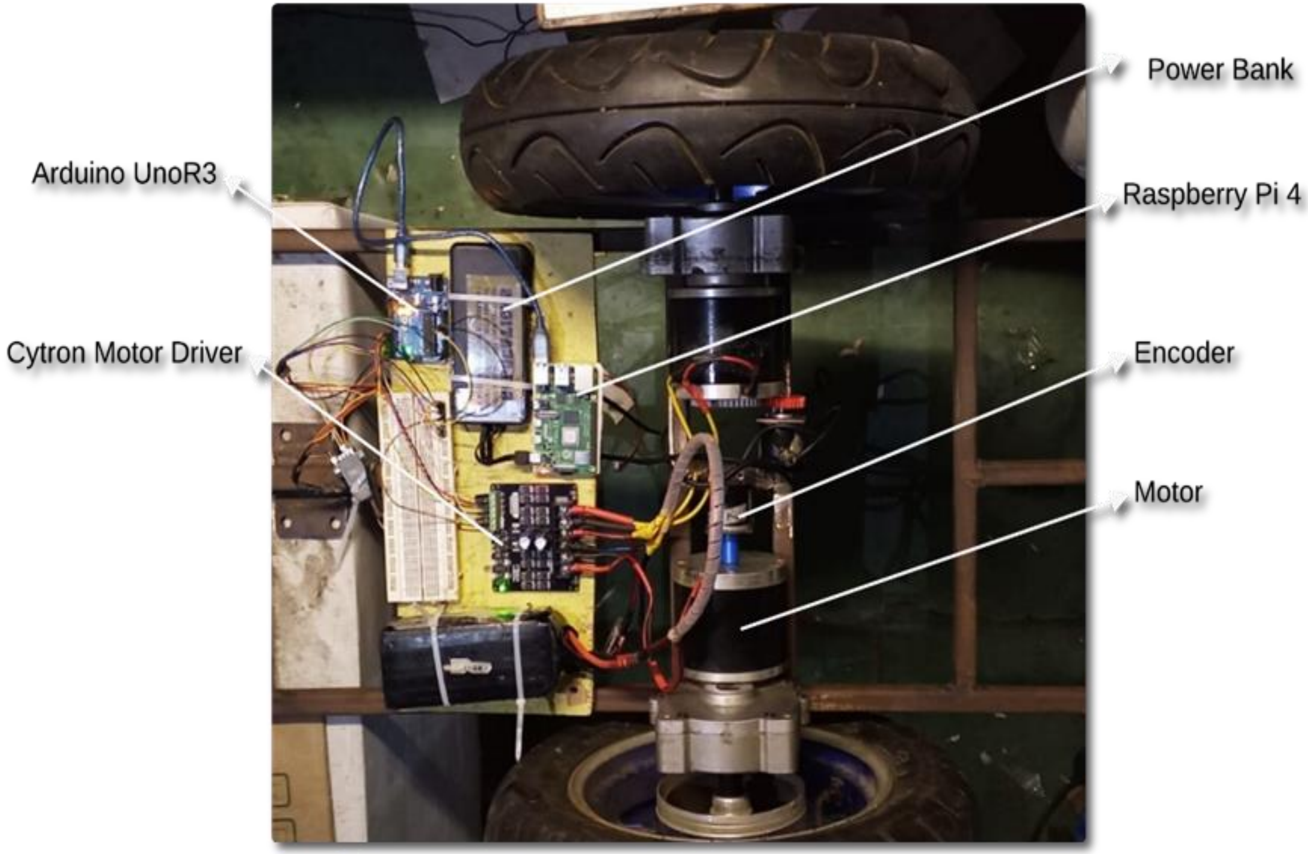


Figure 10 : Actual Control Circuit of AMR

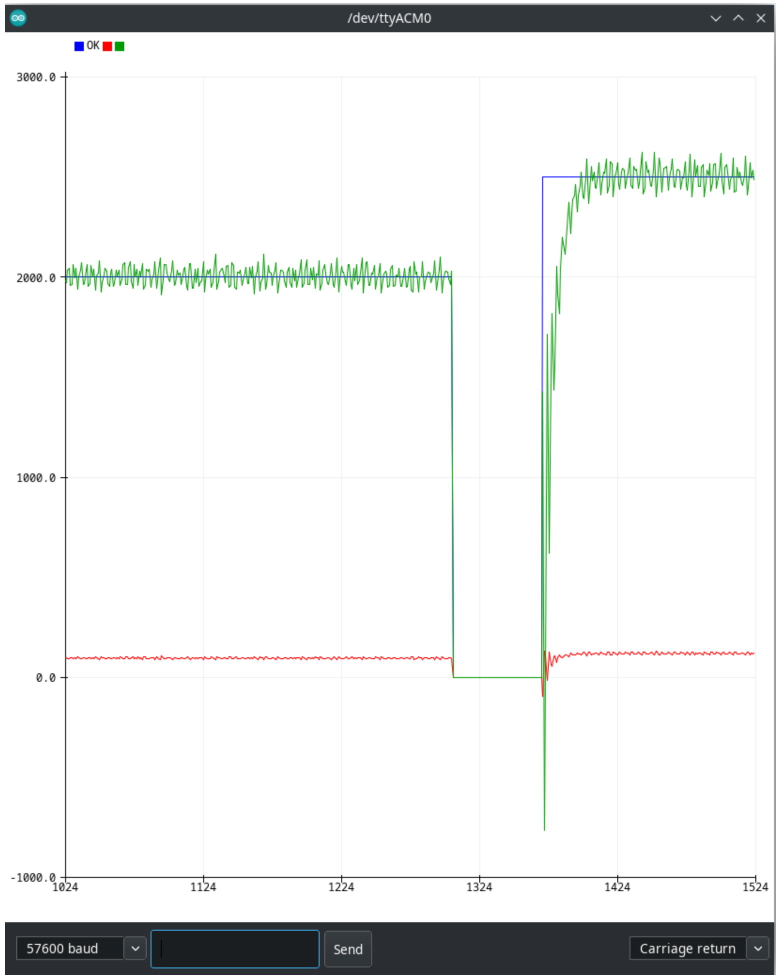


Figure 11 : Tuned PID response;
Kp=0.75, Ki=0.0001, Kd=4and Ko=50

Remote Controlled AMR Locomotion

- Two motors/actuators velocity controlled using **PWM** from **Cytron Motor Driver**

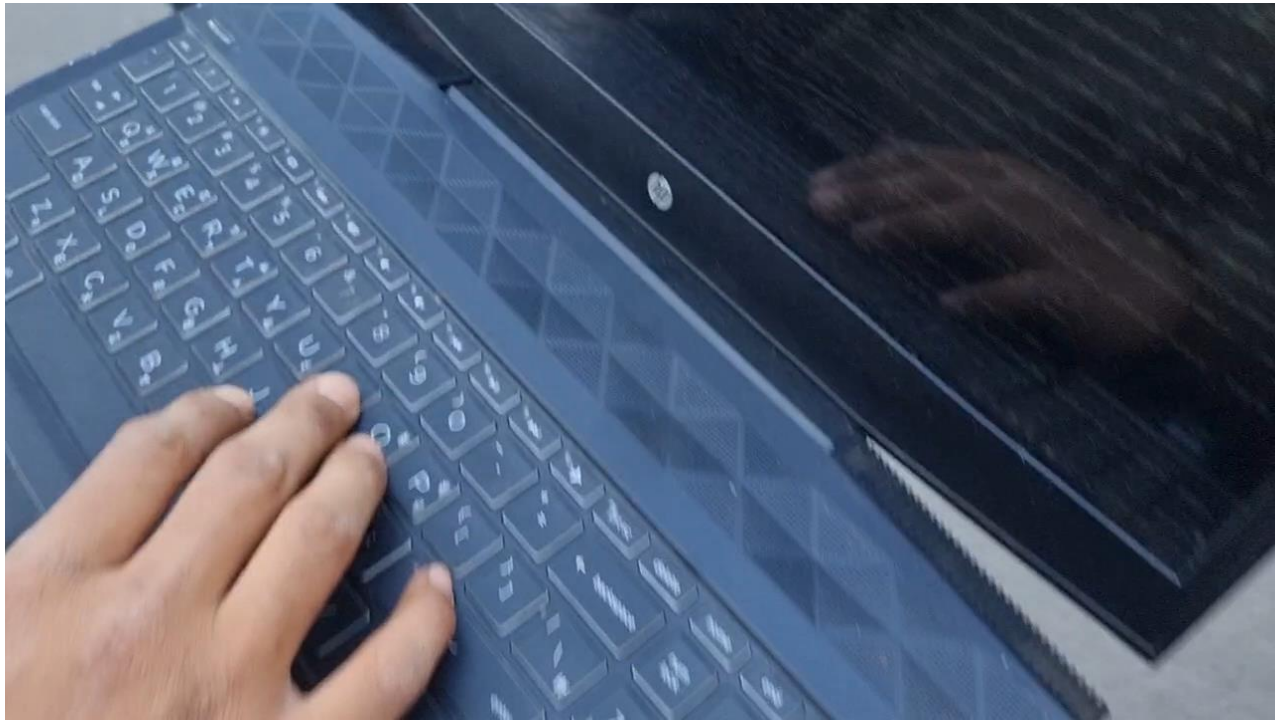
- Feedback from **Encoder** to enable **closed loop PID** control

- **Arduino** computes PID and uses **Encoder Counts per loop** as velocity variable at 30Hz.

- **Raspberry Pi** runs **ROS2 Control** that computes the required wheel velocity and corresponding Encoder Counts per loop and passes to Arduino using serial communication

- Server or another **PC** (for now) passes command to Raspberry PI about the task of the robot via **ssh**.

- **Command velocity** given from PC using either teleop_twist_**keyboard** or the **joystick** (joy and teleop_twist_joy node)



ROS2 Control

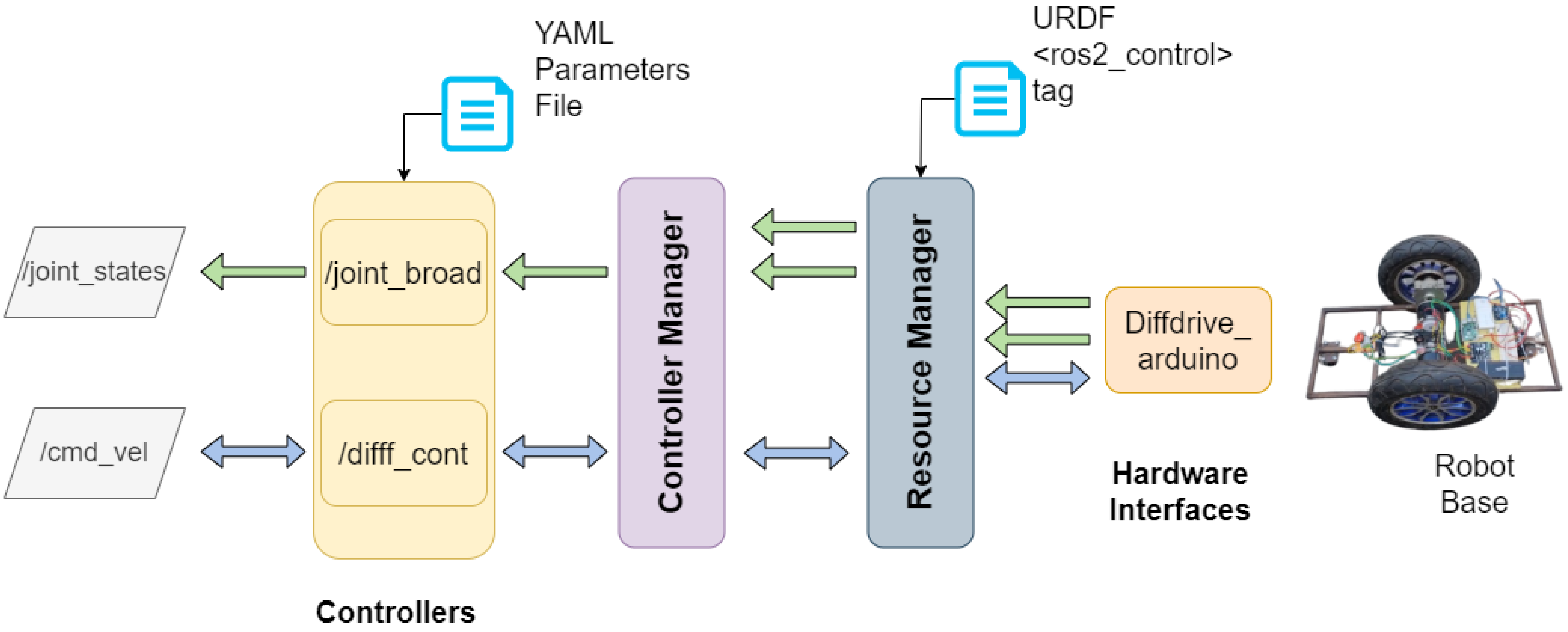


Figure 12: ros2 control architecture

ROS2 Control

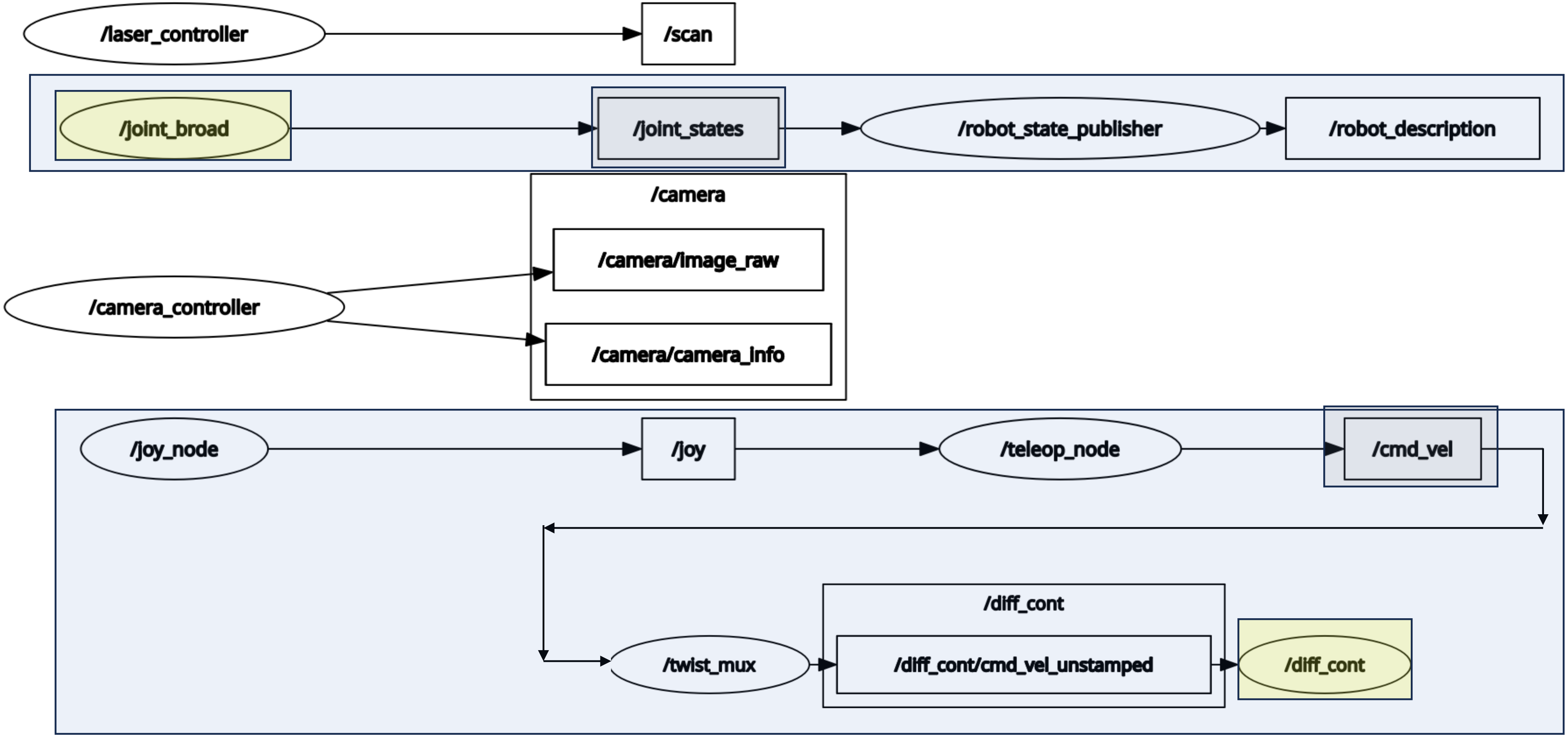


Figure 13: 8 Rqt_graph for ros2_control in simulation

Testing RPLIDAR

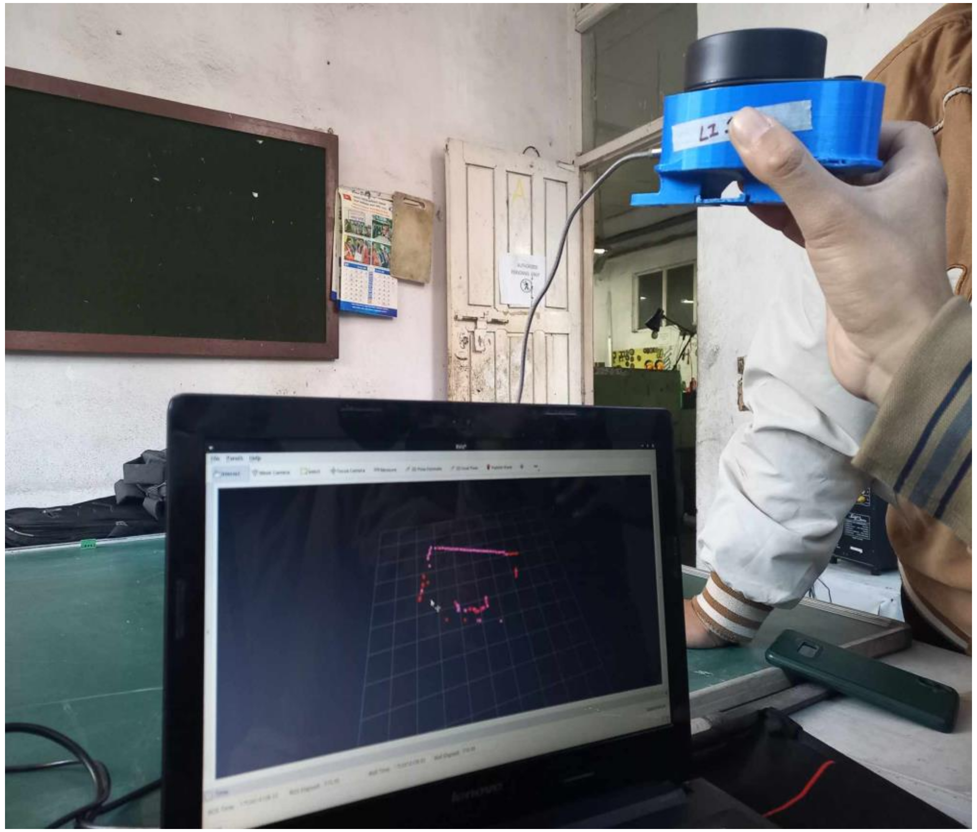
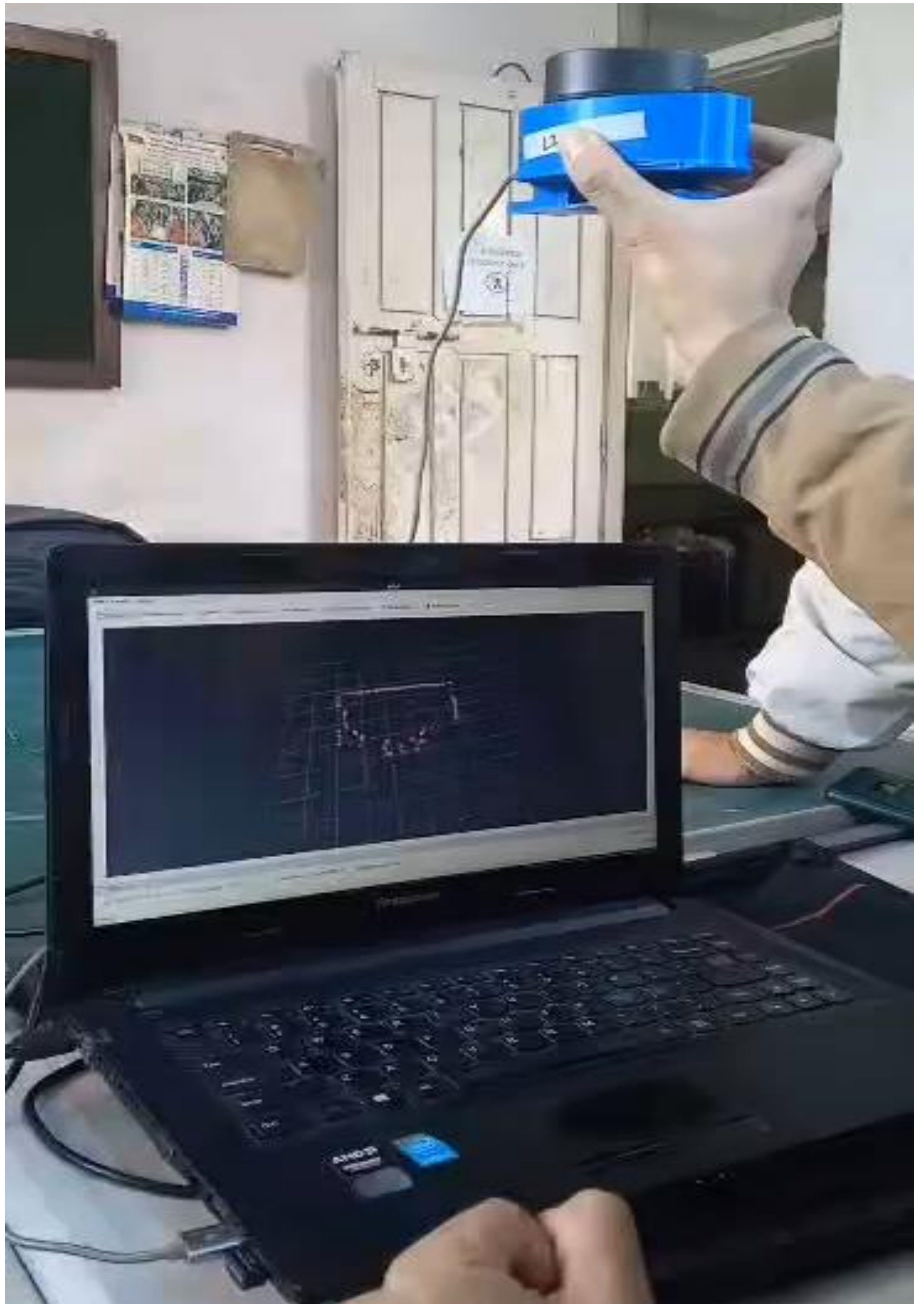


Figure 14 : Reading LIDAR’s data using RVIZ



AMR Autonomous Navigation

- RPLiDAR generates the point cloud of surrounding and passes it to Raspberry Pi
- Server or PC computes SLAM, Navigation and publishes the command velocity for the required path.
- ROS2 Control's Diff Cont controller computes the desired AMR wheel velocity
- The hardware interface of ROS2 control communicates the required counts per PID loop via serial port.
- Closed loop control same as in Phase I

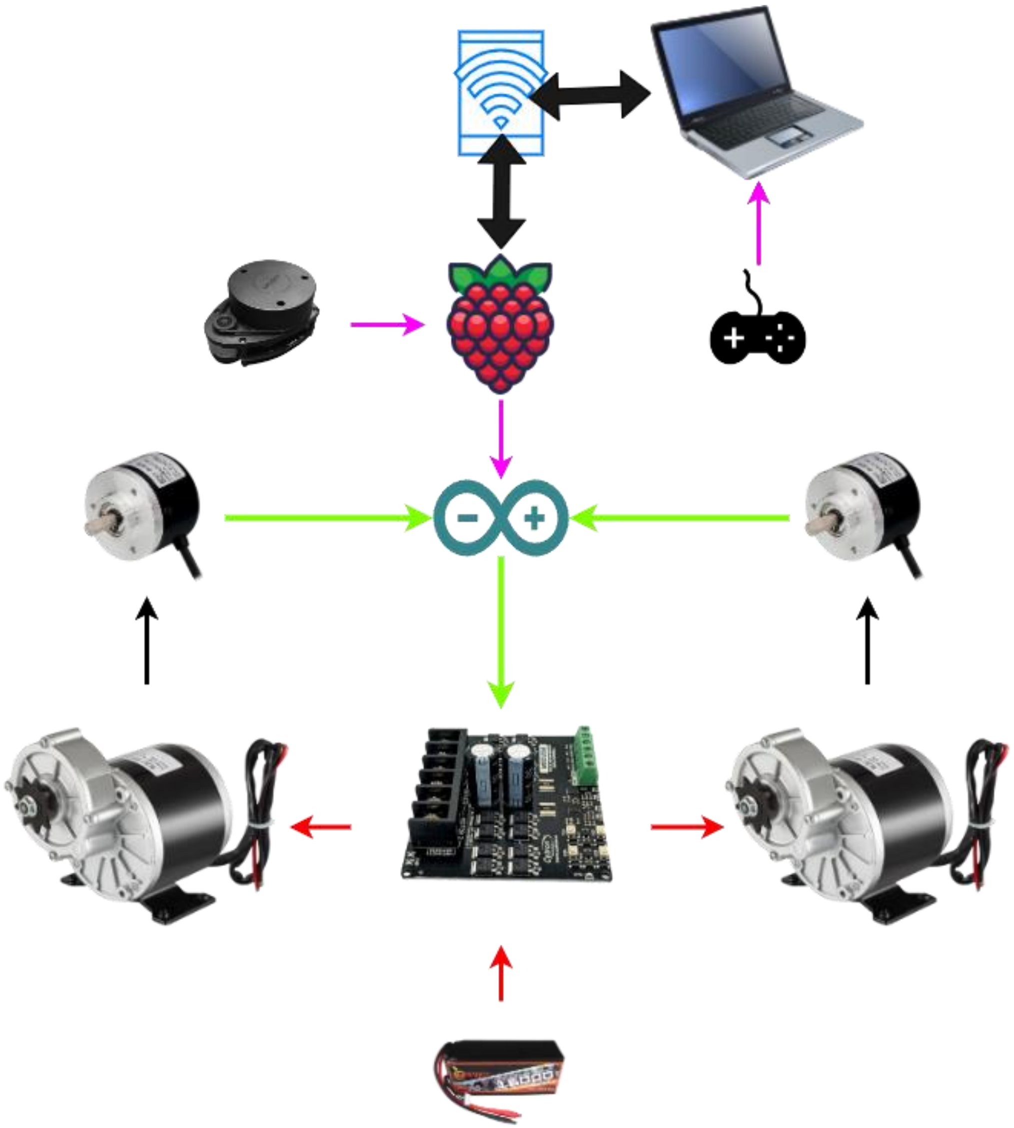
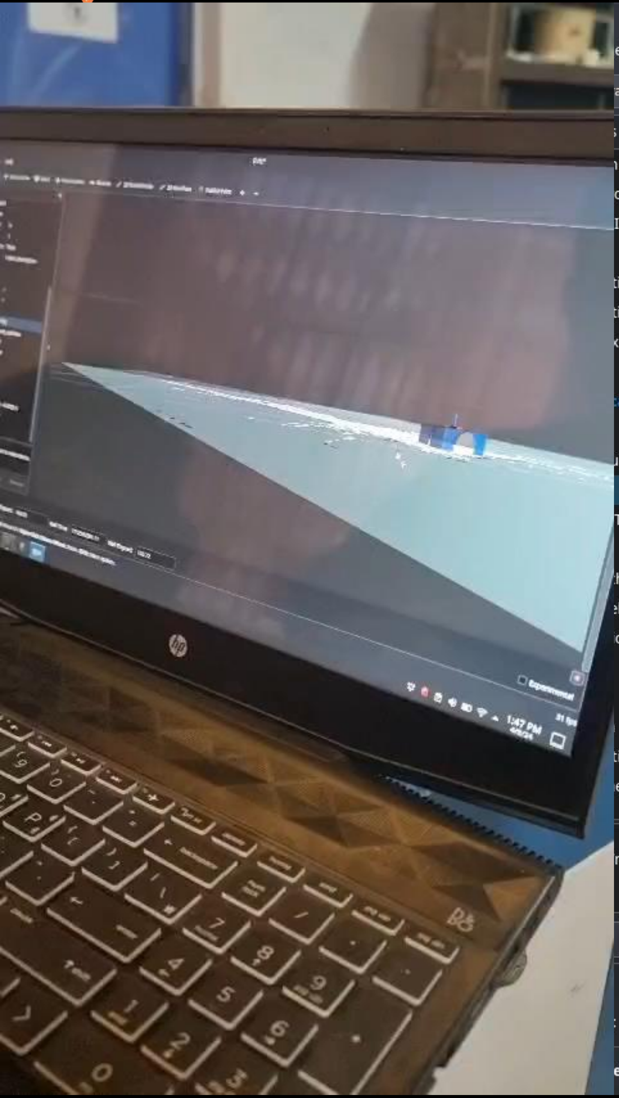
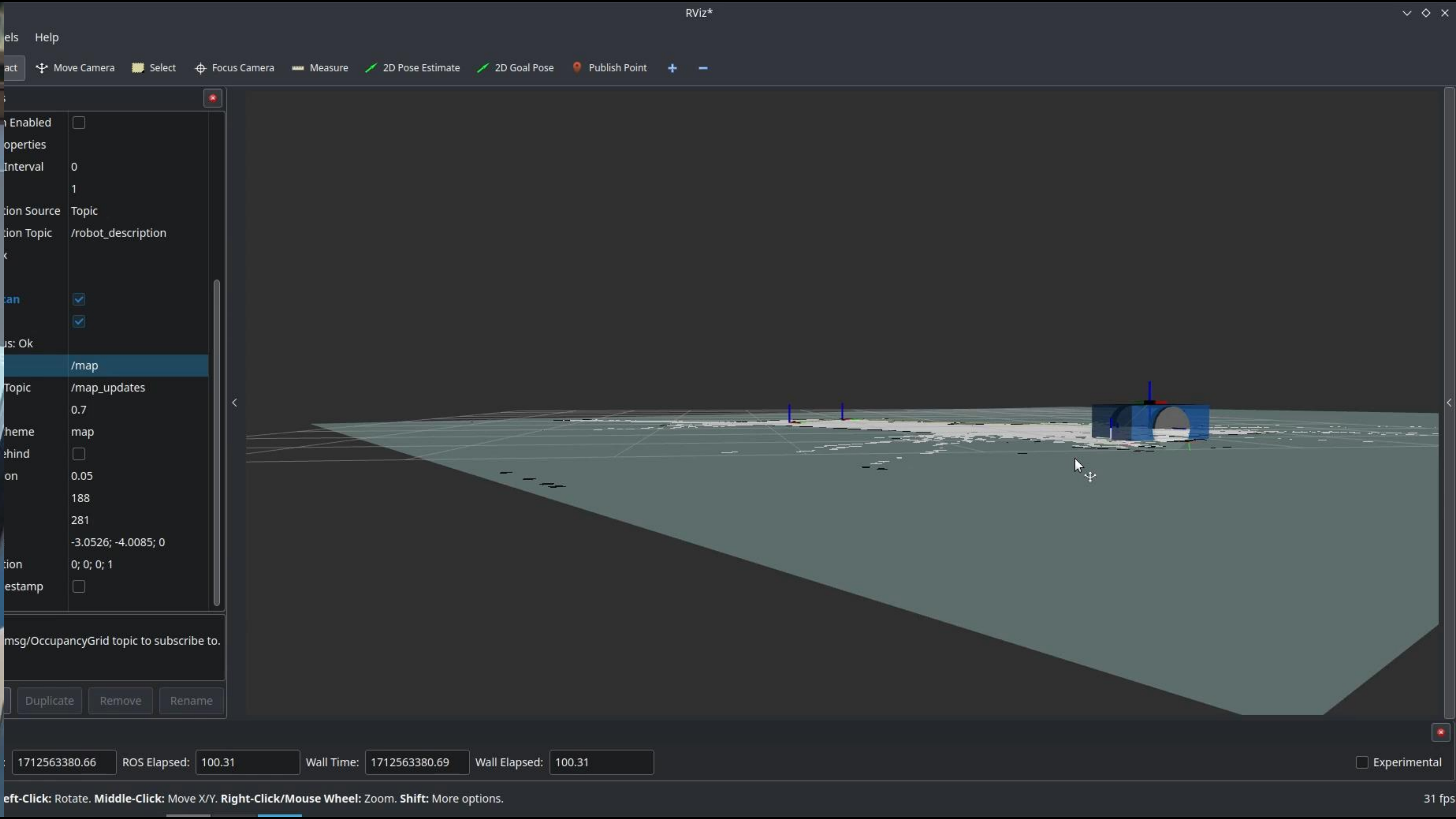


Figure 15 : Overall Navigation Control Pipeline

SLAM Deployment Using Lidar





Map of workspace in Robotics Club

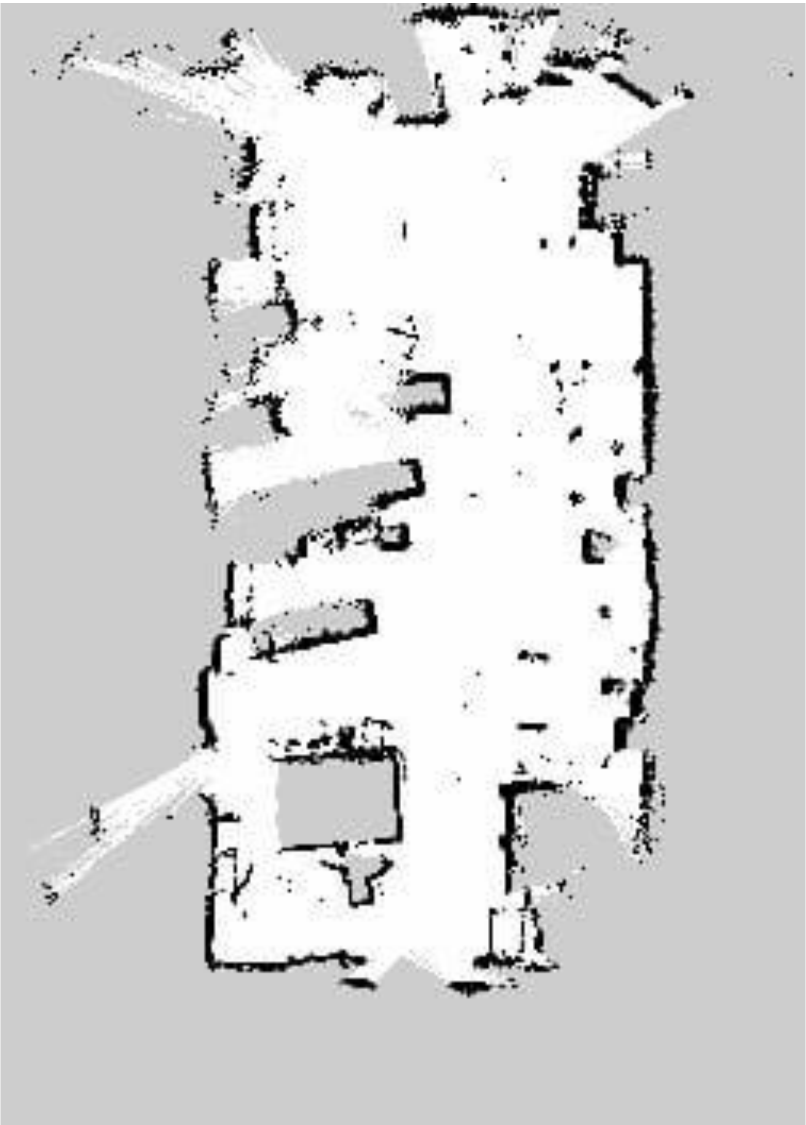
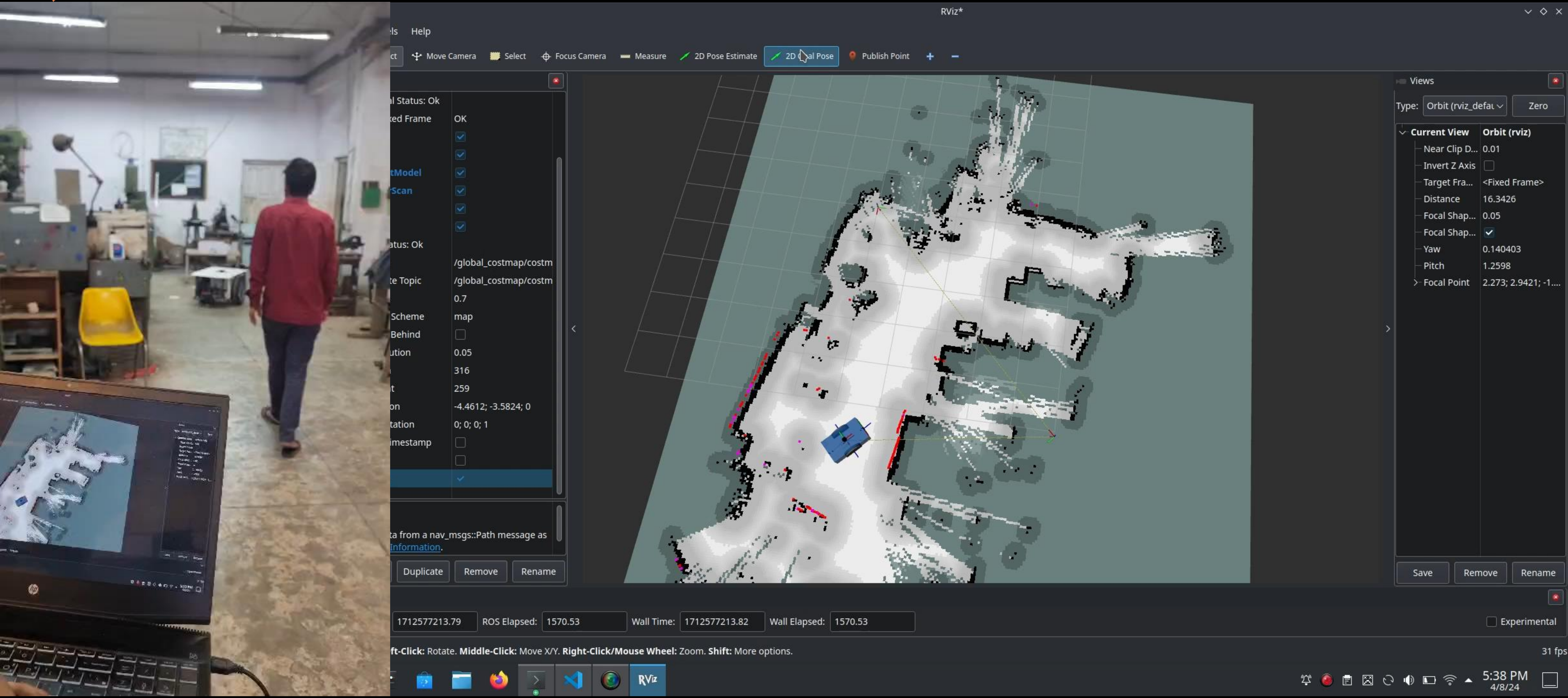


Figure 16: Created Map of Robotics Club workspace using SLAMToolbox Plugin

AMR Navigation using Nav2



Material Handling Design and Fabrication

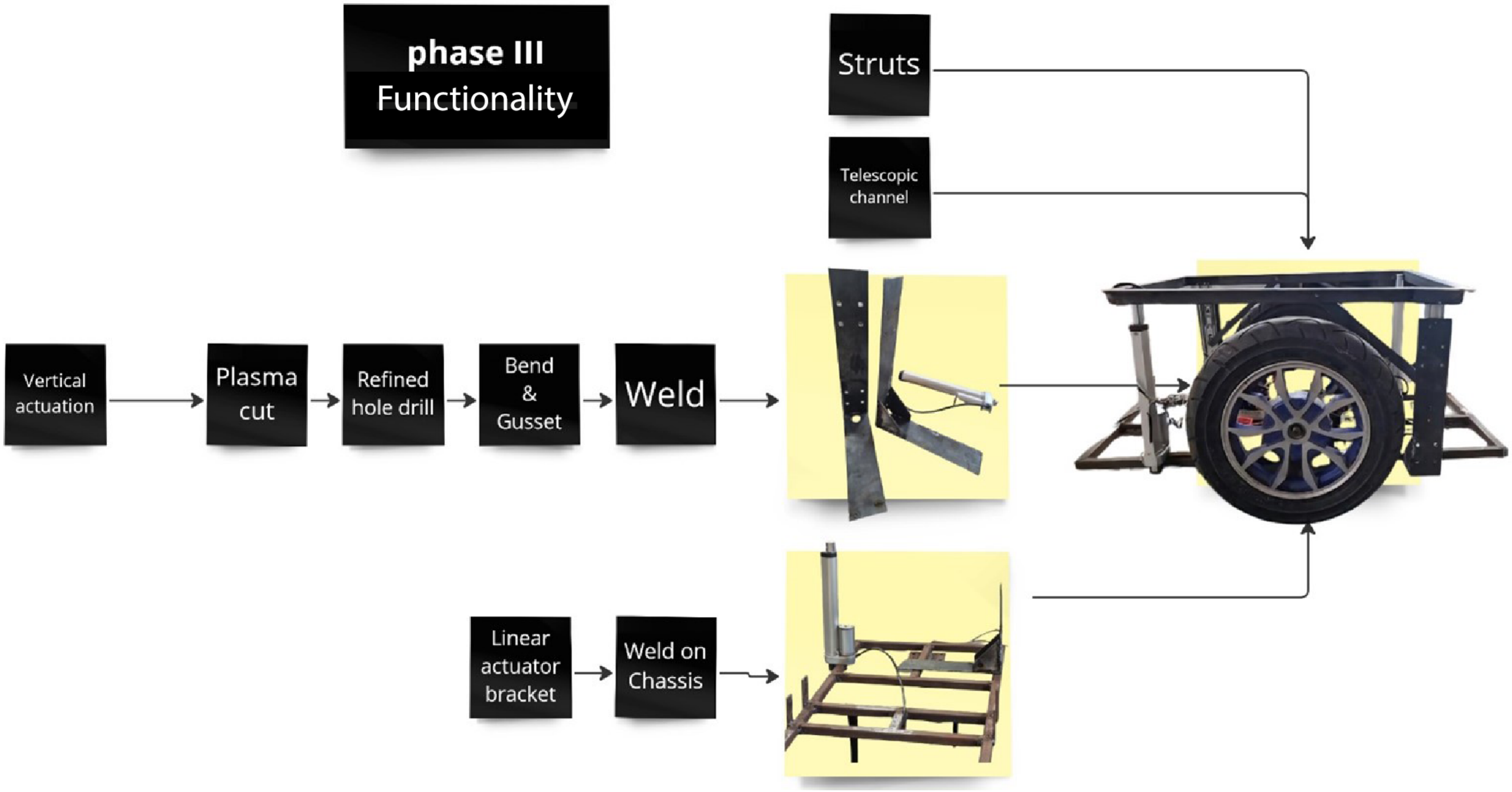
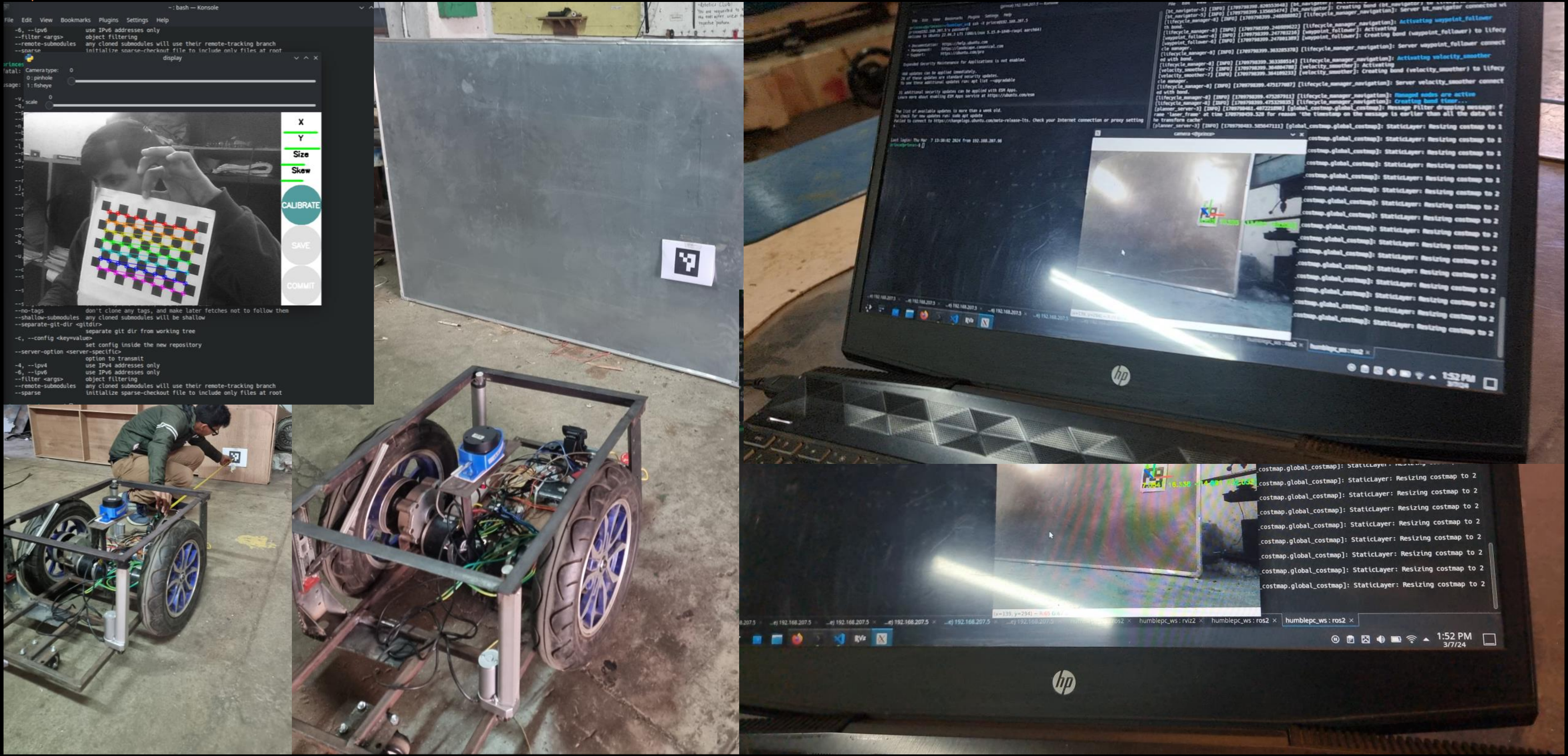


Figure 17 : Fabrication of Material Handling subsystem

CV Setup


~ : bash — Konsole
File Edit View Bookmarks Plugins Settings Help
-6, --ipv6 use IPv6 addresses only
--filter <args> object filtering
--remote-submodules any cloned submodules will use their remote-tracking branch
display
Camera type: 0
0 : pinhole
1 : fisheye
scale
X
Y
Size
Skew
CALIBRATE
SAVE
COMMIT
--no-tags don't clone any tags, and make later fetches not to follow them
--shallow-submodules any cloned submodules will be shallow
--separate-git-dir <gitdir>
separate git dir from working tree
-c, --config <key=value>
set config inside the new repository
--server-option <server-specific>
option to transmit
-4, --ipv4 use IPv4 addresses only
-6, --ipv6 use IPv6 addresses only
--filter <args> object filtering
--remote-submodules any cloned submodules will use their remote-tracking branch
--sparse initialize sparse-checkout file to include only files at root
costmap.global_costmap]: StaticLayer: Resizing costmap to 2
humblepc_ws : rviz2
humblepc_ws : ros2
1:52 PM
3/7/24
hp

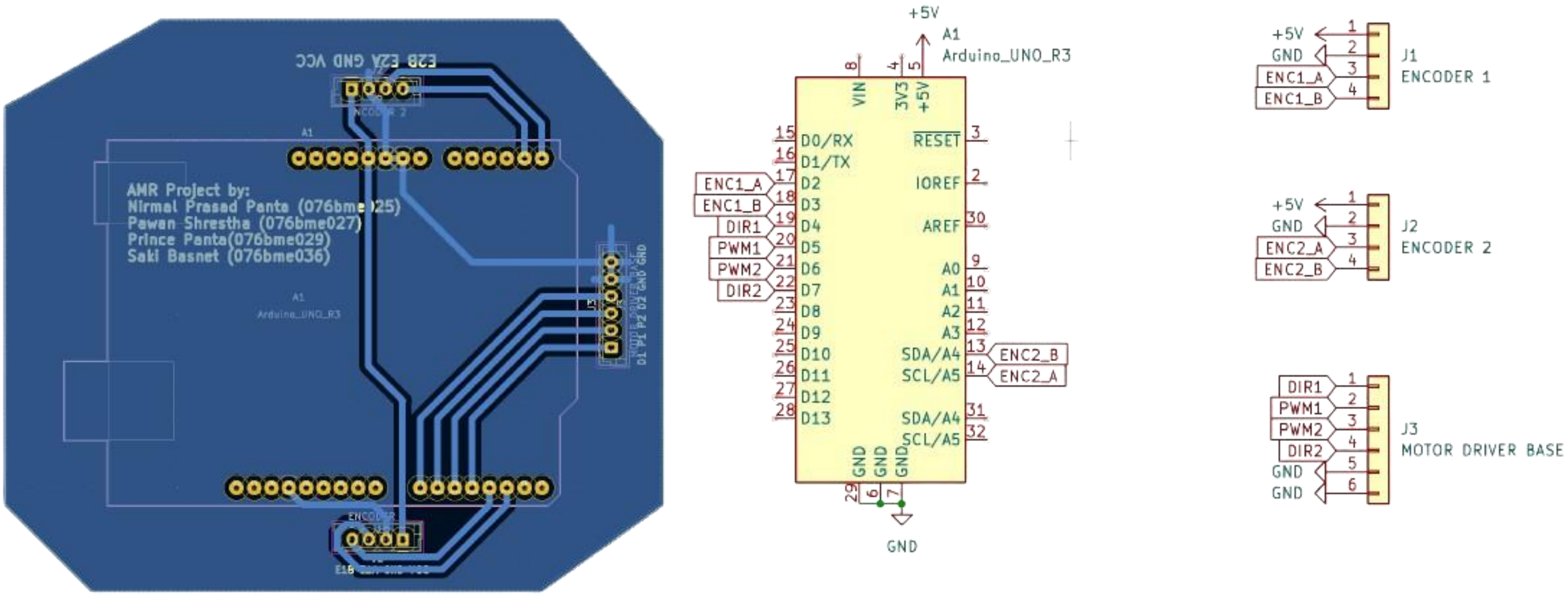


Figure 18: Circuit design in KiCAD

Circuit Design and Fabrication

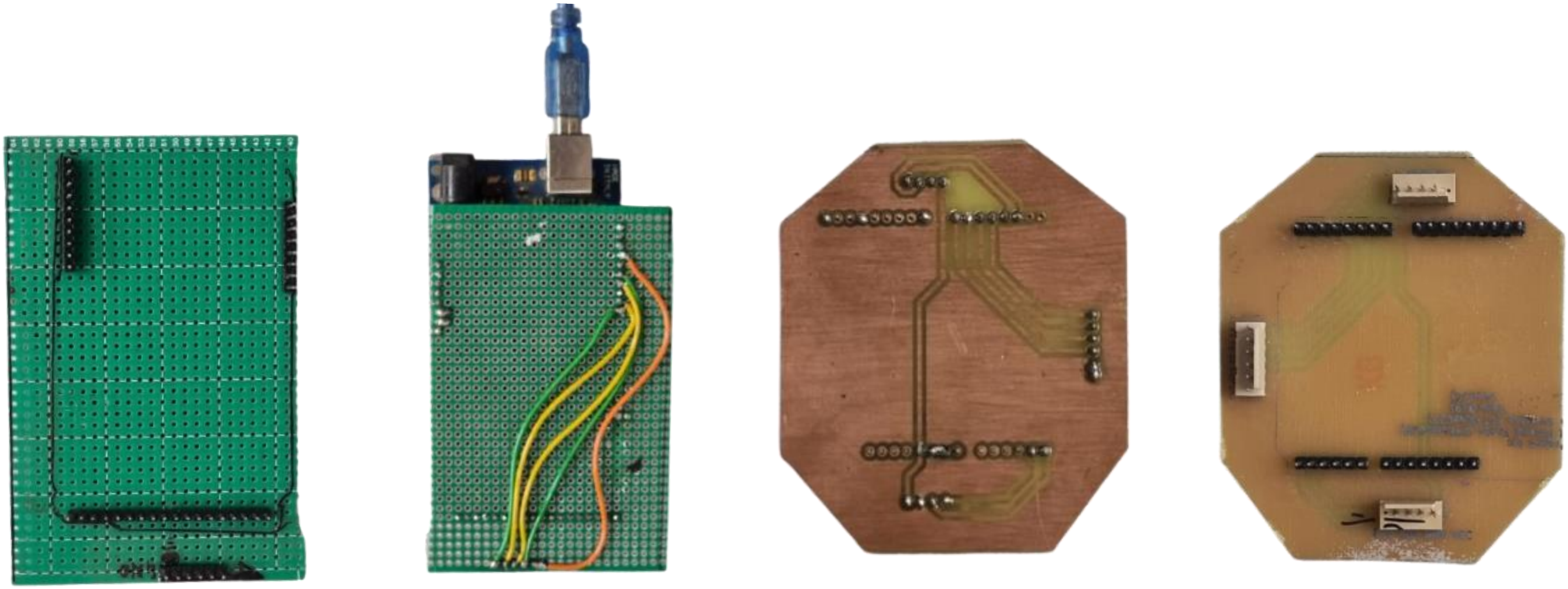
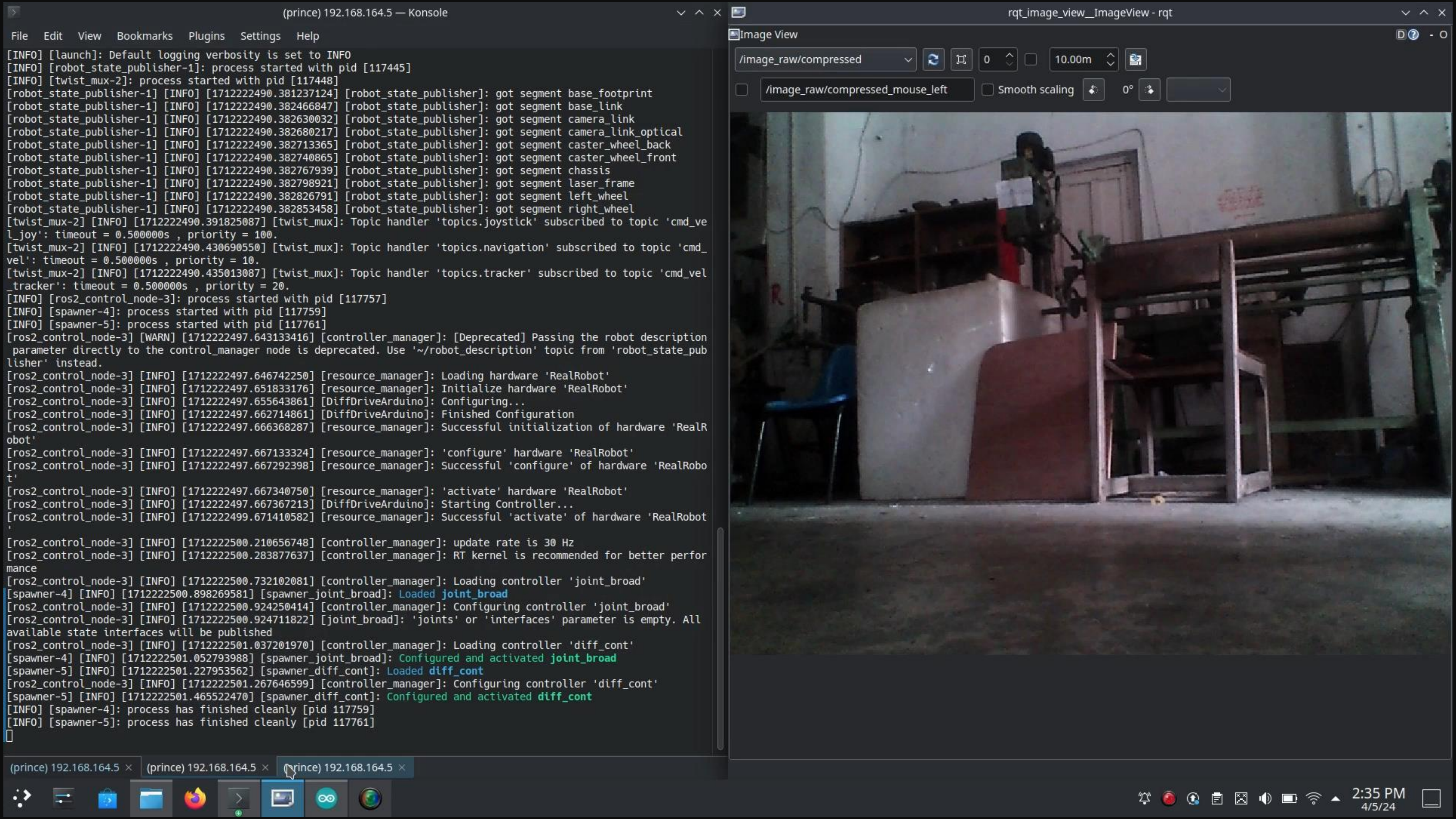


Figure19: Matrix board and fabricated PCB

Teleop Material Handling



Teleop Material Handling

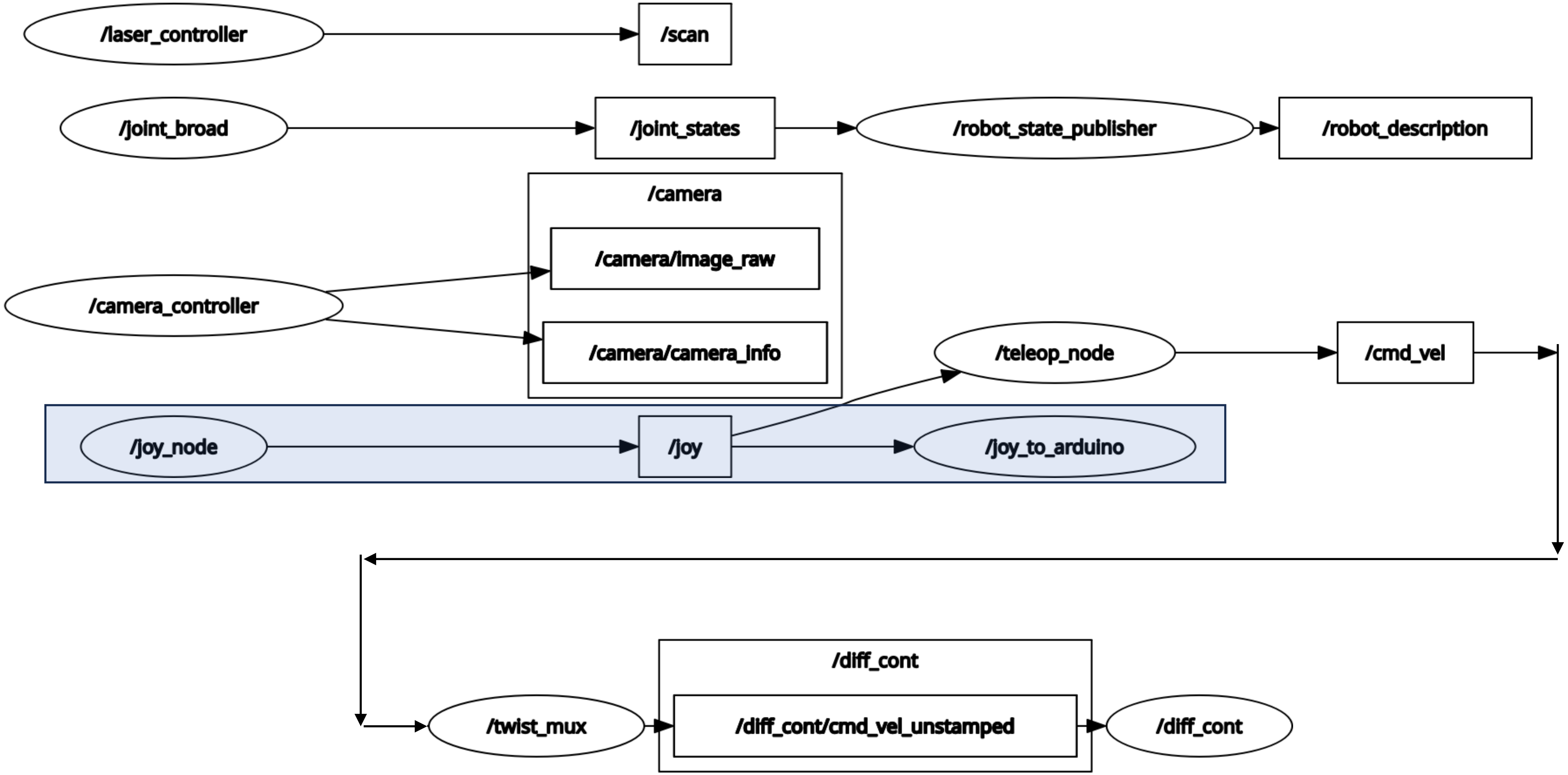
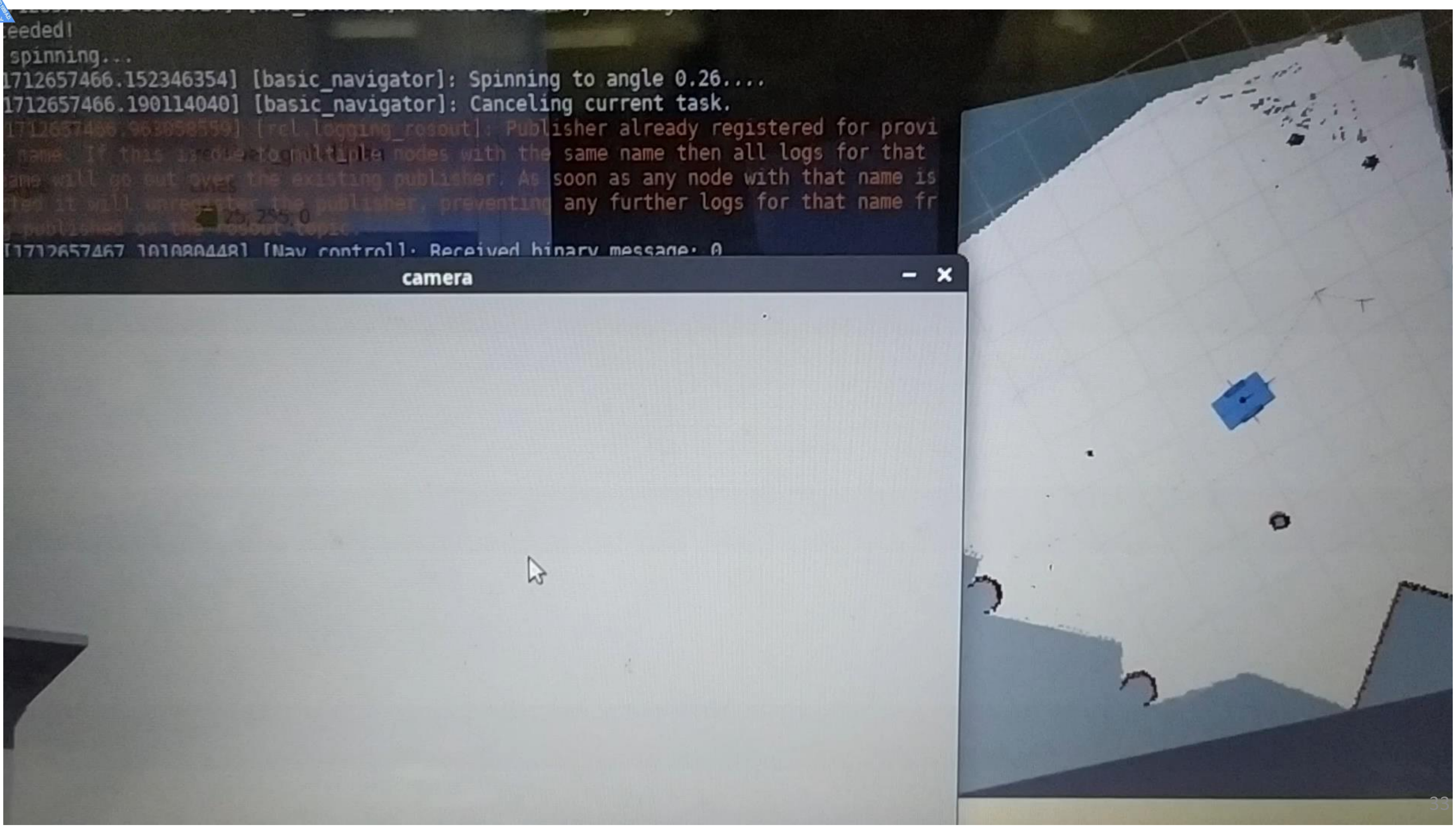


Figure 20: RQT graph during Teleop Material Handling

Autonomous Material Handling Simulation



AMR Control Architecture

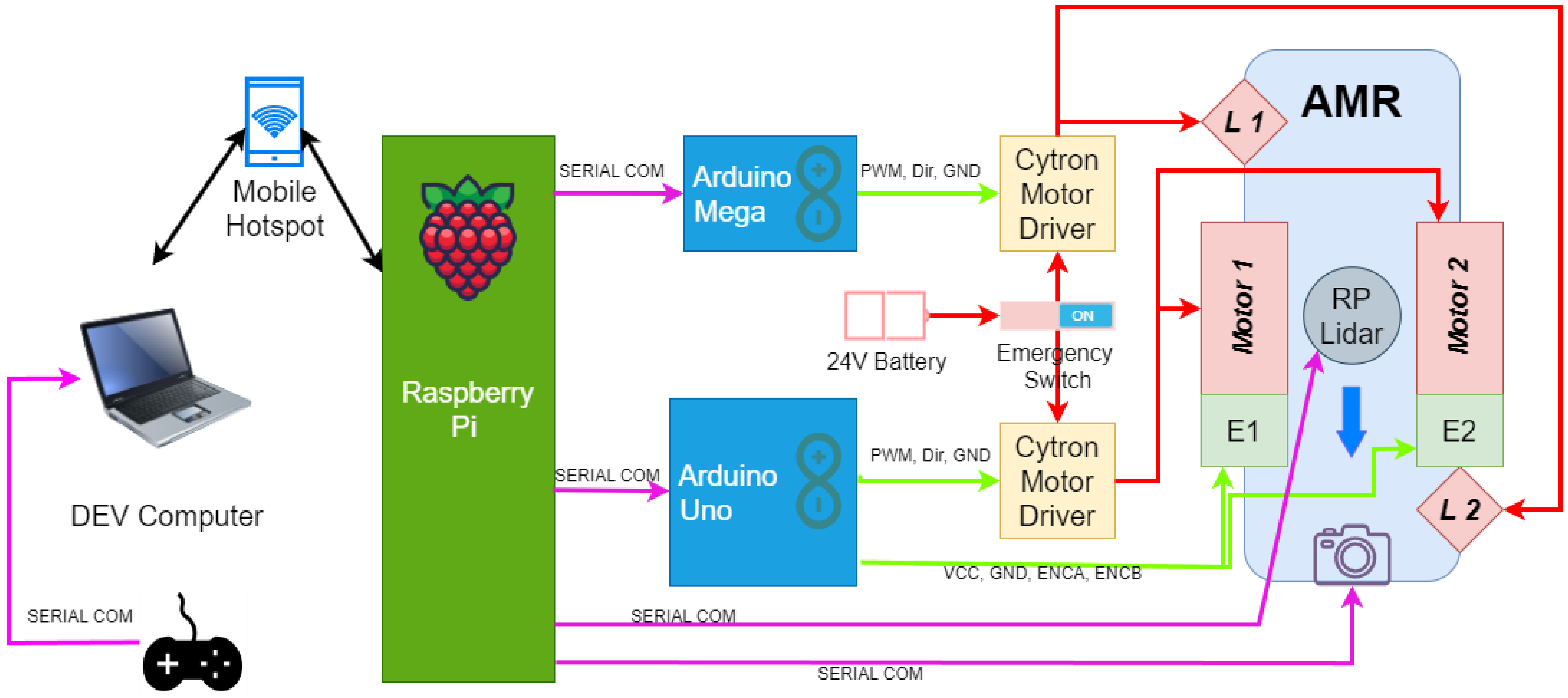
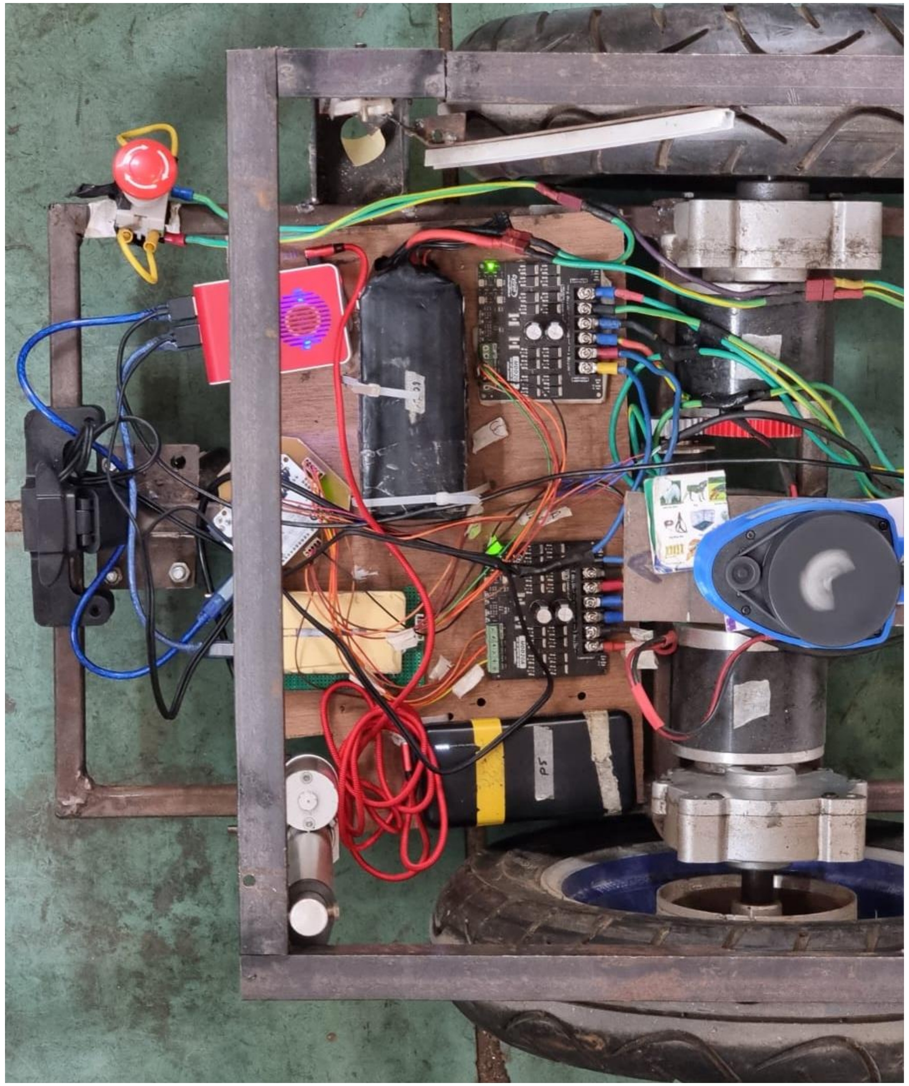


Figure 21: Final AMR Control Architecture



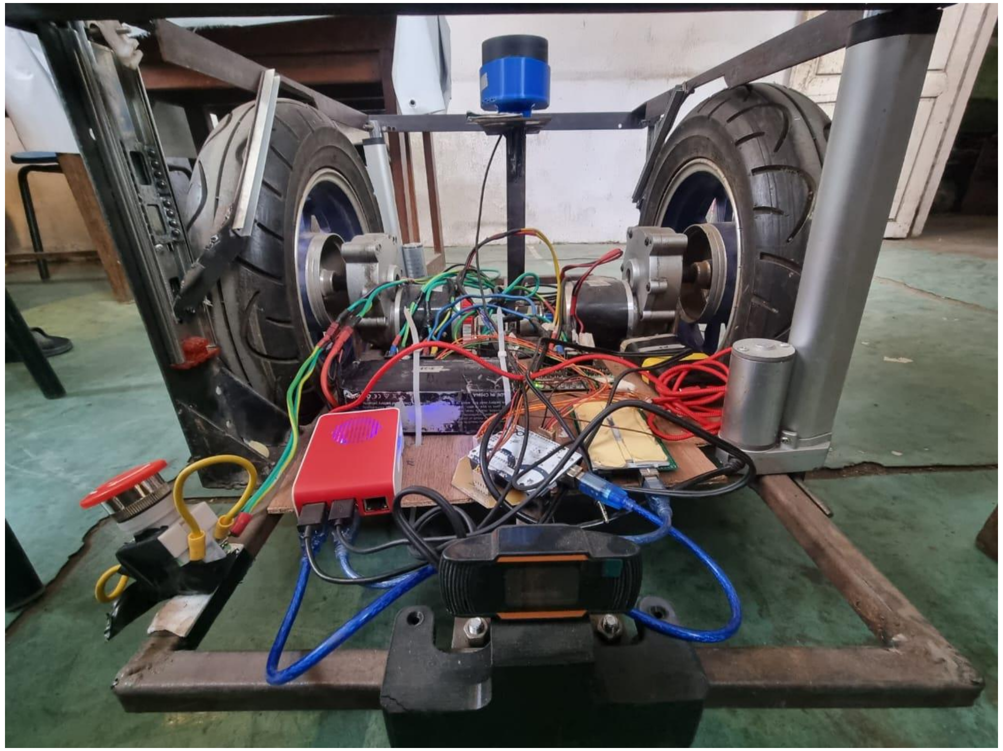
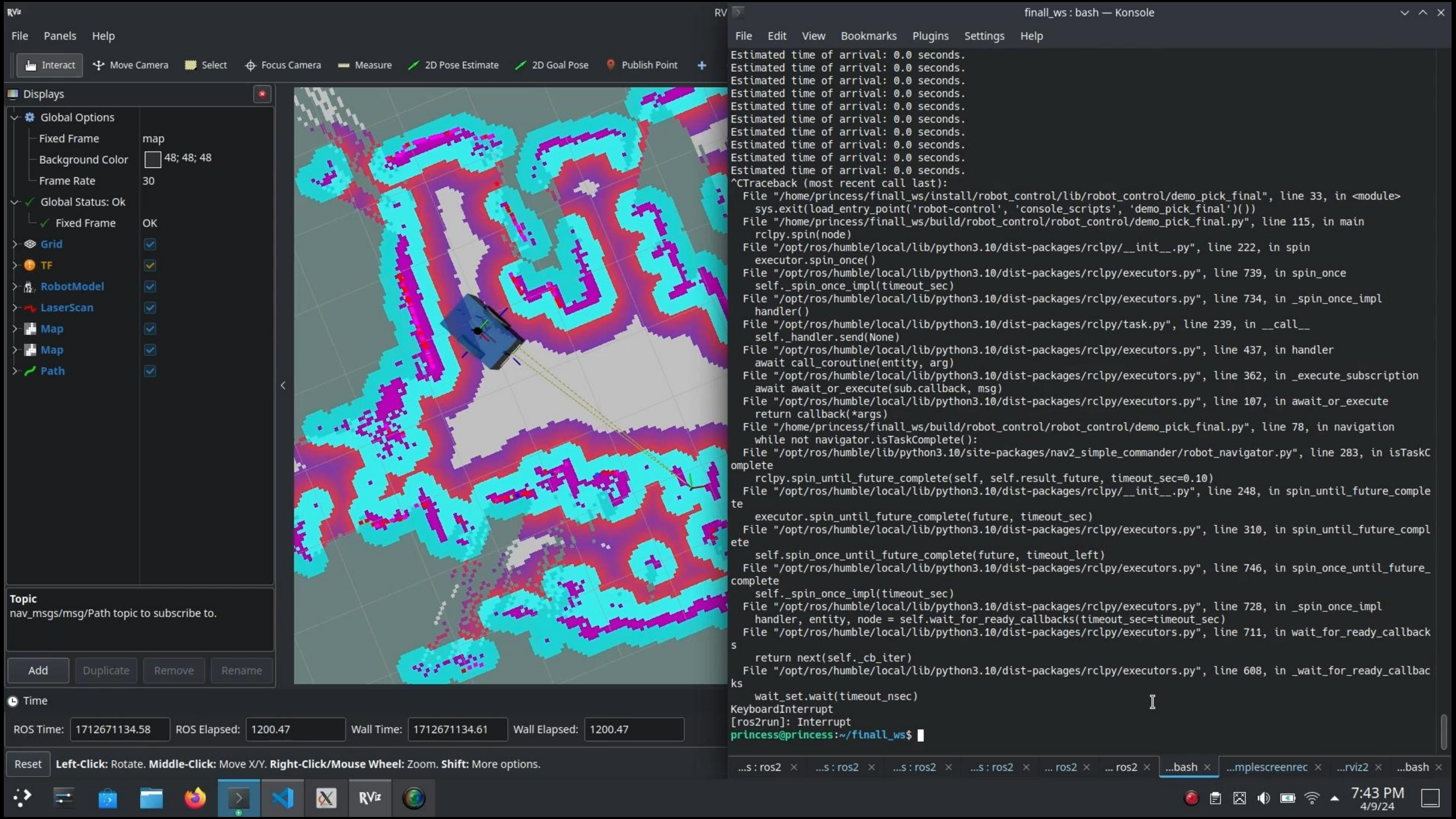


Figure 22 : Actual Control Circuit

OUTPUT: Autonomous Material Handling



Problems Faced

- Limited resources so if a problem occurred in a hardware, work would halt
- Failure in debugging errors which forced to start over
- Visual manager of Ubuntu for Raspberry PI not functioning
- Error in camera calibration using ArUco tag – discrepancy in distance obtained using CV and actual distance
- Major issue faced- Network Latency and Pi Crash
    - Localization delayed and as a result robot gets stalled or worse hits the obstacle
    - Real time video frame published lags and as a result not feasible in using camera feed if problem in navigation in remote location

Limitations

- The size of the AMR is constrained due to the use of available motors
- Different environments not considered for the operation of LIDAR
- Fleet of AMRs not considered which is the norm in the industry
- AMR is not capable of detecting obstacles below the lidar level
- Linear actuators operate in open loop
- The accuracy and precision of navigating and docking of AMR is affected due to network latency and camera calibration error respectively

Working Schedule

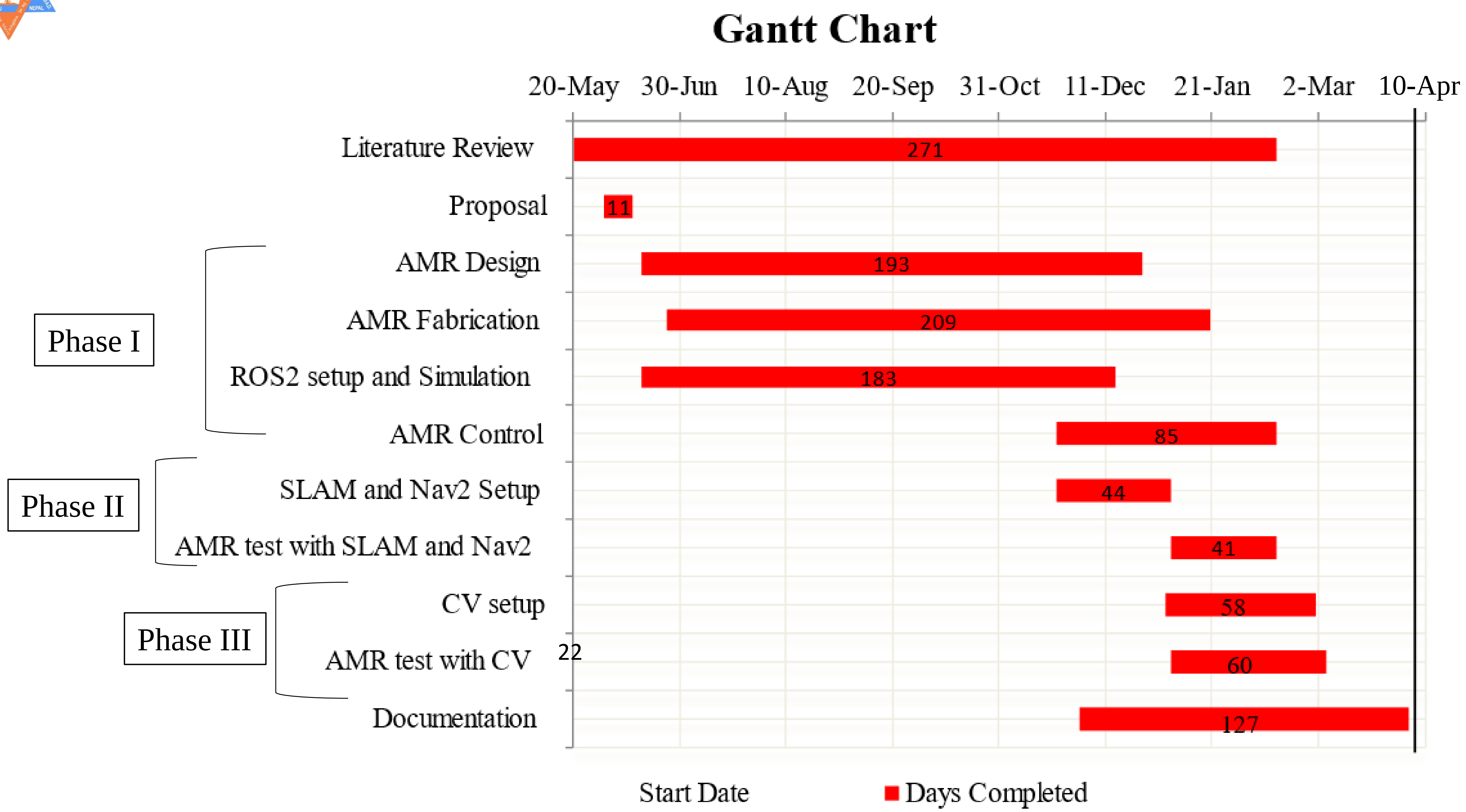


Figure 23: Project Gantt Chart

Budget analysis

Table 2: Items to be pledged

Table 1: Base Cost

S.N.	Description	Estimation in NRs
1	Manufacturing Cost	20,000
2	Logitech C270 Camera	3,895
3	Documentation	5,000
4	Miscellaneous	2,000
5	Raspberry Pi 4	20,000
Total		**50,895**

S.N.	Description	Estimation in NRs
1	Motor Drivers	3,000
2	Hub Motors	22,000
3	Arduino	2,380
4	LiDAR	15,000
5	Linear Actuator	8,000
Total		**50,380**

CONCLUSION AND FUTURE ENHANCEMENT

AMR developed and tested for material handling successfully in three phases:

Phase I : Transport

- AMR Chassis Design and Fabrication
- ROS2 Setup and Simulation
- ROS control integration and Teleoperation of AMR

Phase II : Improvement on Transport

- Integration of SLAM for mapping and Nav2 for navigation
- Testing of AMR for autonomous mobility

Phase III : Functionality to Transport

- AMR material handling design and fabrication
- Teleoperated material handling
- CV setup using ArUco tag detection and camera calibration
- Autonomous navigation, docking and material handling

Conclusion and Future enhancements

- Minimizing height of the robot using low radius wheel and high torque motor or use of ultrasonic sensors for low obstacle detection

- System incorporating fleets of robot can be developed and path and task planning optimization such as Dijkstra's algorithm can be implemented.

- Used of additional sensors like Inertial Measuring Unit (IMU) can be incorporated to enhance the odometry data.

- Improvement upon existing design such as round body instead of rectangular and suspension for caster wheels

- Use of Visual SLAM with cameras for improved pose estimation and 3D mapping

References

[1] G. Ullrich, "The History of Automated Guided Vehicle Systems," in Automated Guided Vehicle Systems: A Primer with Practical Applications, Berlin, Heidelberg, Springer Berlin Heidelberg, 2015, pp. 1--14.

[2] W. Grzechca, "Manufacturing in Flow Shop and Assembly Line Structure," International Journal of Materials, Mechanics and Manufacturing, vol. 4, no. 1, pp. 25-30, 2016.

[3] N. Zghair, A. Khaleq and A. S. Al-Araji, "A one decade survey of autonomous mobile robot systems," International Journal of Electrical and Computer Engineering, vol. 11, no. 6, p. 4891, 2021.

[4] L. Lynch, T. Newe, J. Clifford, J. Coleman, J. Walsh and D. Toal, "Automated Ground Vehicle (AGV) and Sensor Technologies- A Review," 2018 12th International Conference on Sensing Technology (ICST), Limerick, Ireland, 2018, pp. 347-352, doi: 10.1109/ICSensT.2018.8603640.

[5] S.Macenski, T. Foote, B. Gerkey, C. Lalancette, W. Woodall, “Robot Operating System 2: Design, architecture, and uses in the wild,” Science Robotics vol. 7, May 2022.

[6] Macenski et al., (2021). SLAM Toolbox: SLAM for the dynamic world. Journal of Open Source Software, 6(61), 2783. https://doi.org/10.321105/joss.02783

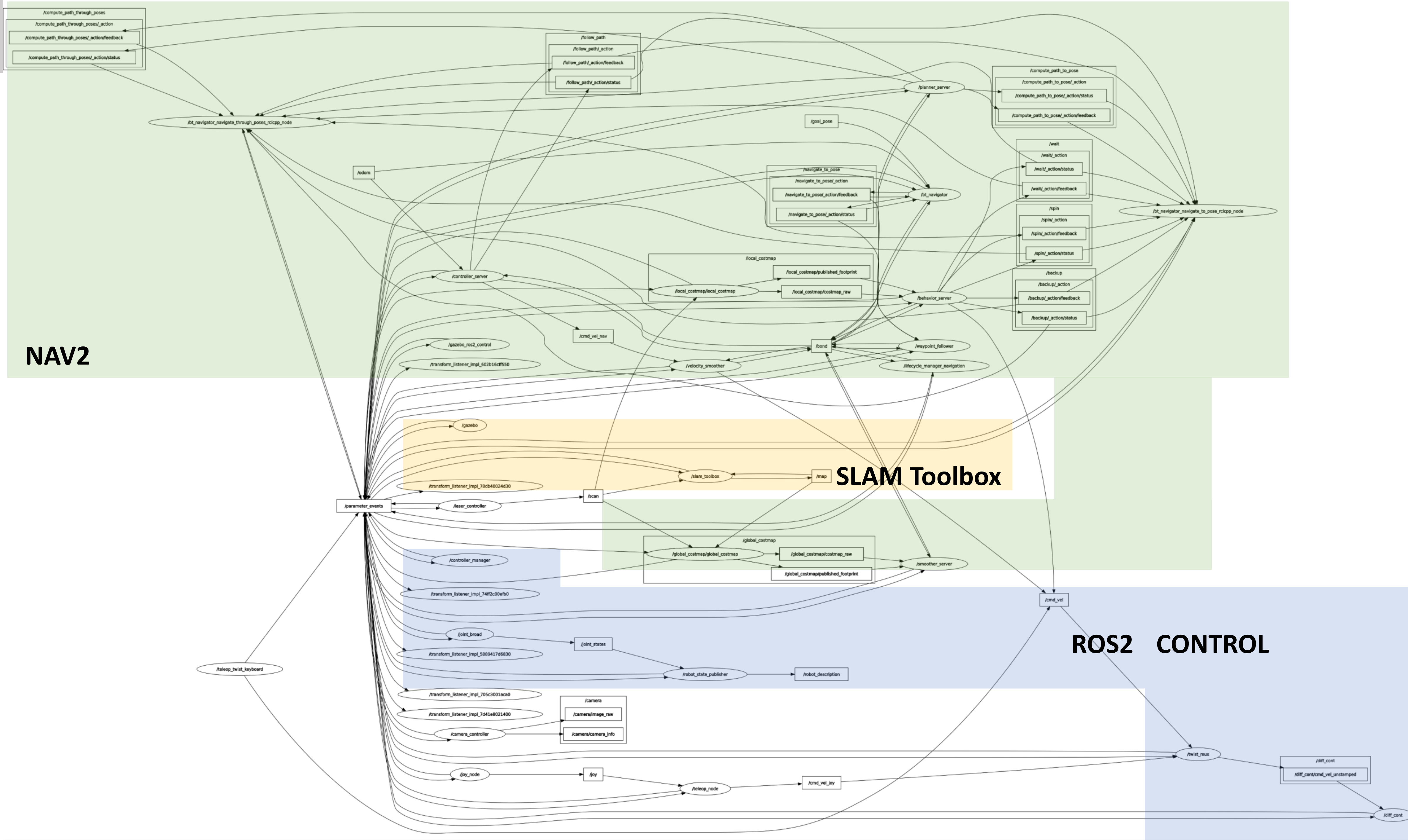
[7] G. Bradski, "The OpenCV Library," Dr. Dobb's Journal of Software Tools, 2000.

[8] J. Wubben, F. Fabra, C. T. Calafate, T. Krzeszowski, J. M. Marquez-Barja, J.-C. Cano and P. Manzoni, "Accurate Landing of Unmanned Aerial Vehicles Using Ground Pattern Recognition," Electronics, vol. 8, 2019.

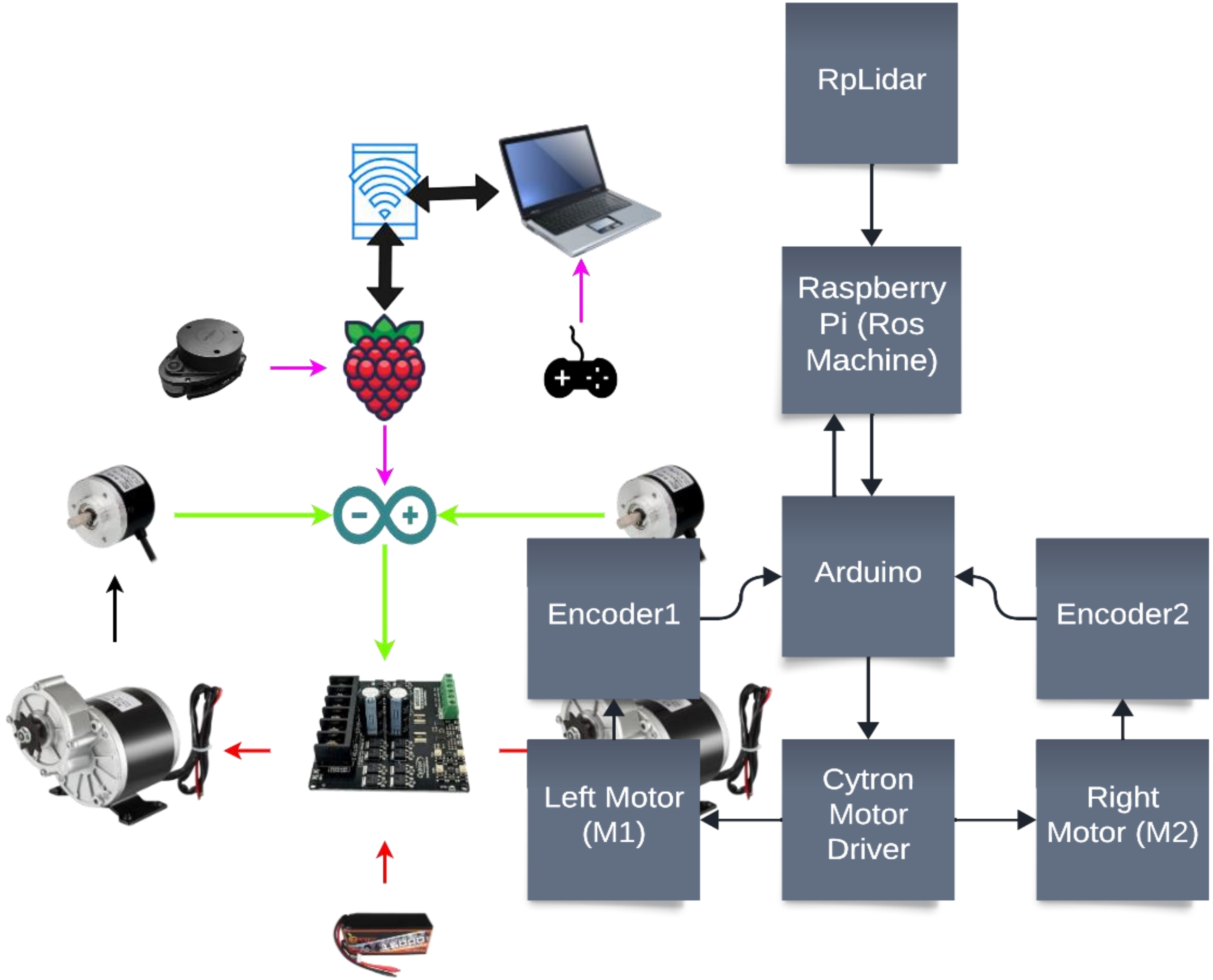
Acknowledgements

- Department of Mechanical and Aerospace Engineering, Pulchowk Campus and HOD Dr. Sudip Bhattarai
- Project Supervisor, Associate Prof. Dr. Surya Prasad Adhikari
- Deputy Head of Department and Assistant Professor Laxman Motra
- Deputy Head of Department and Assistant Professor Kamal Darlami
- Assistant Professor Aashish Karki
- Robotics Club, Pulchowk Campus; especially Ramesh Chaudhary sir, Alumni Nitesh Subedi and Mr. Sabin Shrestha
- To all the teachers, friends and seniors and juniors who have assisted us directly or indirectly in this project

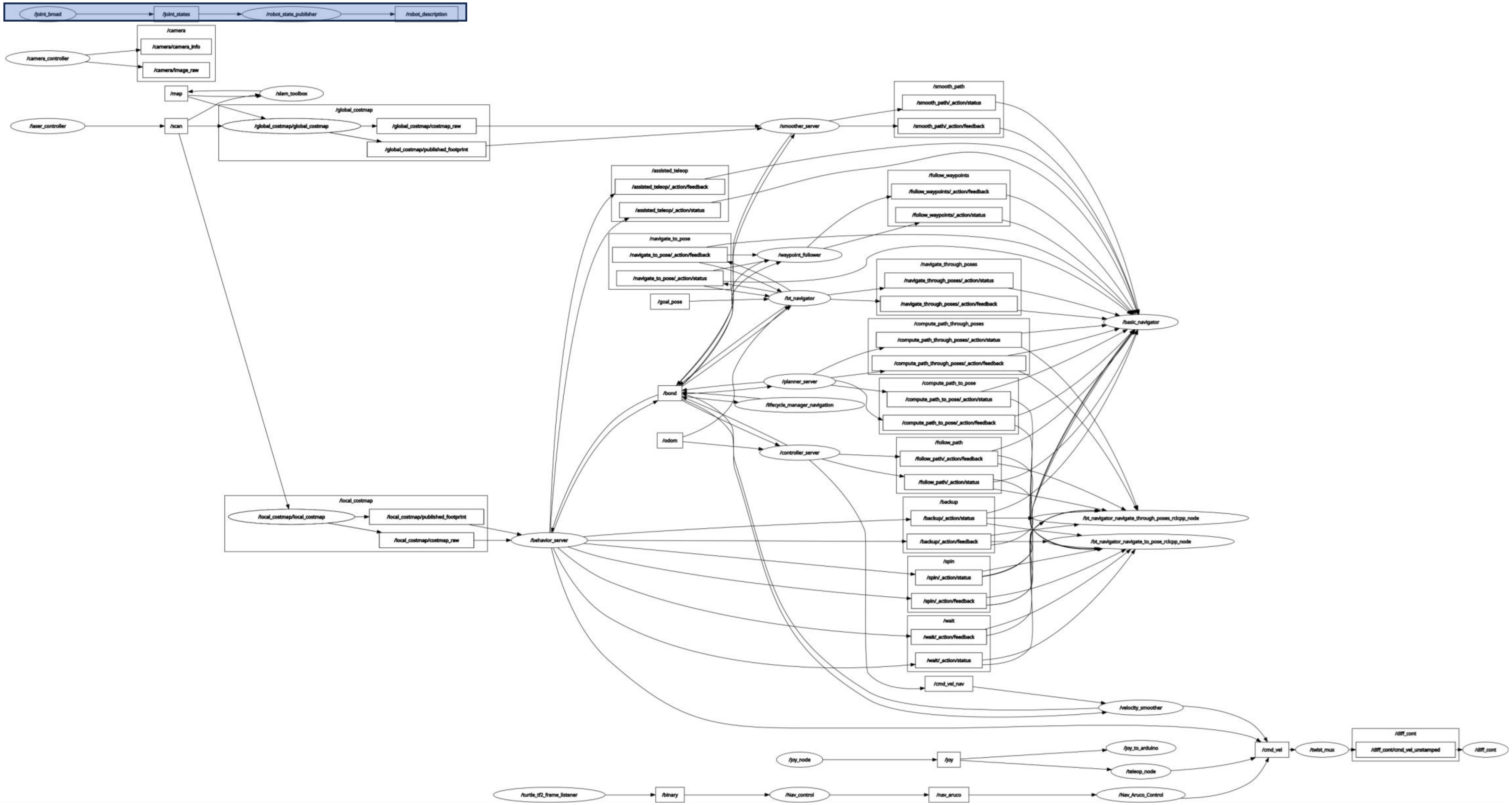
THANK YOU!


NAV2
SLAM Toolbox
ROS2 CONTROL
/compute_path_through_poses
/compute_path_through_poses/_action
/compute_path_through_poses/_action/feedback
/compute_path_through_poses/_action/status
/follow_path
/follow_path/_action
/follow_path/_action/feedback
/follow_path/_action/status
/planner_server
/compute_path_to_pose
/compute_path_to_pose/_action
/compute_path_to_pose/_action/status
/compute_path_to_pose/_action/feedback
/bt_navigator_navigate_through_poses_rclcpp_node
/goal_pose
/wait
/wait/_action
/wait/_action/status
/wait/_action/feedback
/odom
/navigate_to_pose
/navigate_to_pose/_action
/navigate_to_pose/_action/feedback
/navigate_to_pose/_action/status
/bt_navigator
/bt_navigator_navigate_to_pose_rclcpp_node
/spin
/spin/_action
/spin/_action/feedback
/spin/_action/status
/local_costmap
/local_costmap/published_footprint
/controller_server
/local_costmap/local_costmap
/local_costmap/costmap_raw
/behavior_server
/backup
/backup/_action
/backup/_action/feedback
/backup/_action/status
/cmd_vel_nav
/gazebo_ros2_control
/bond
/waypoint_follower
/transform_listener_impl_602b16cff550
/velocity_smoother
/lifecycle_manager_navigation
/gazebo
/slam_toolbox
/map
/transform_listener_impl_78db40024d30
/scan
/parameter_events
/laser_controller
/global_costmap
/global_costmap/global_costmap
/global_costmap/costmap_raw
/controller_manager
/smoother_server
/global_costmap/published_footprint
/transform_listener_impl_74ff2c00efb0
/cmd_vel
/joint_broad
/joint_states
/transform_listener_impl_5889417d6830
/teleop_twist_keyboard
/robot_state_publisher
/robot_description
/transform_listener_impl_705c3001aca0
/transform_listener_impl_7d41e8021400
/camera
/camera/image_raw
/camera_controller
/camera/camera_info
/twist_mux
/diff_cont
/diff_cont/cmd_vel_unstamped
/joy_node
/joy
/teleop_node
/cmd_vel_joy
/diff_cont

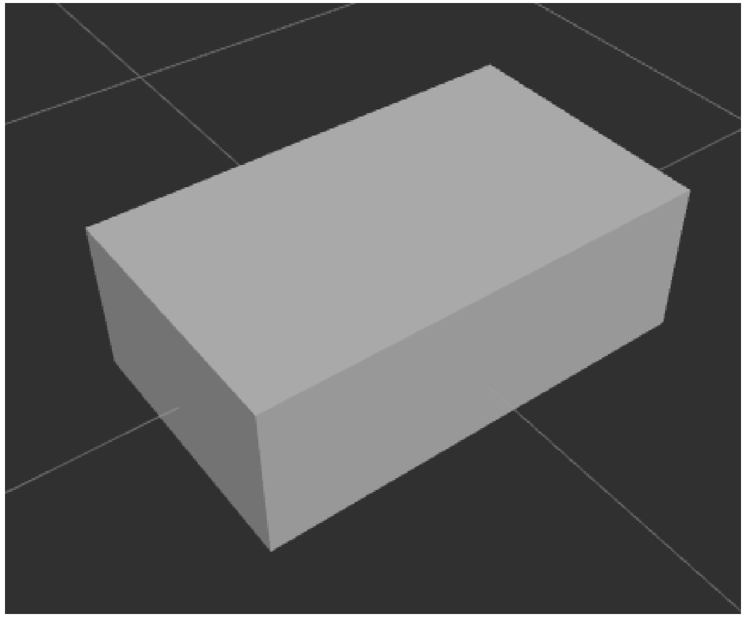
Teleoperated AMR Locomotion Control



OUTPUT: Autonomous Material Handling



Creation of Robot Model



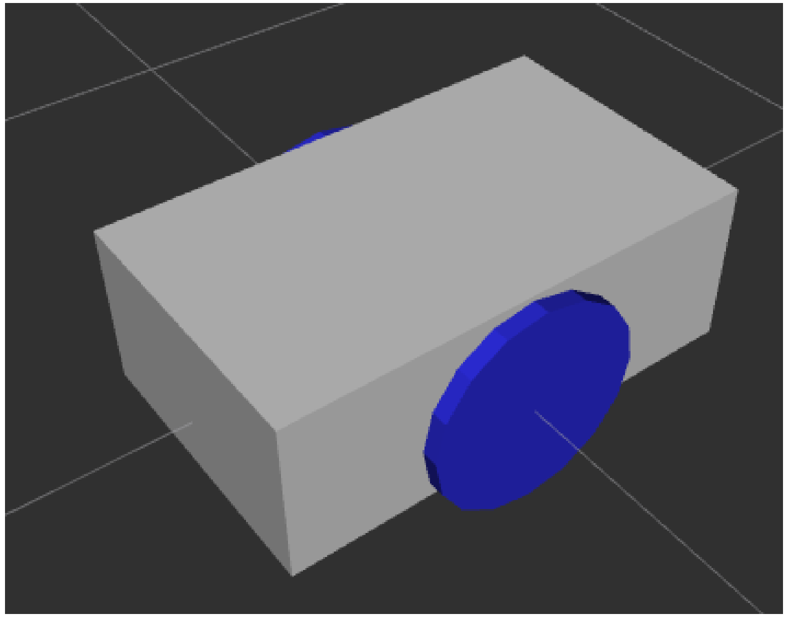
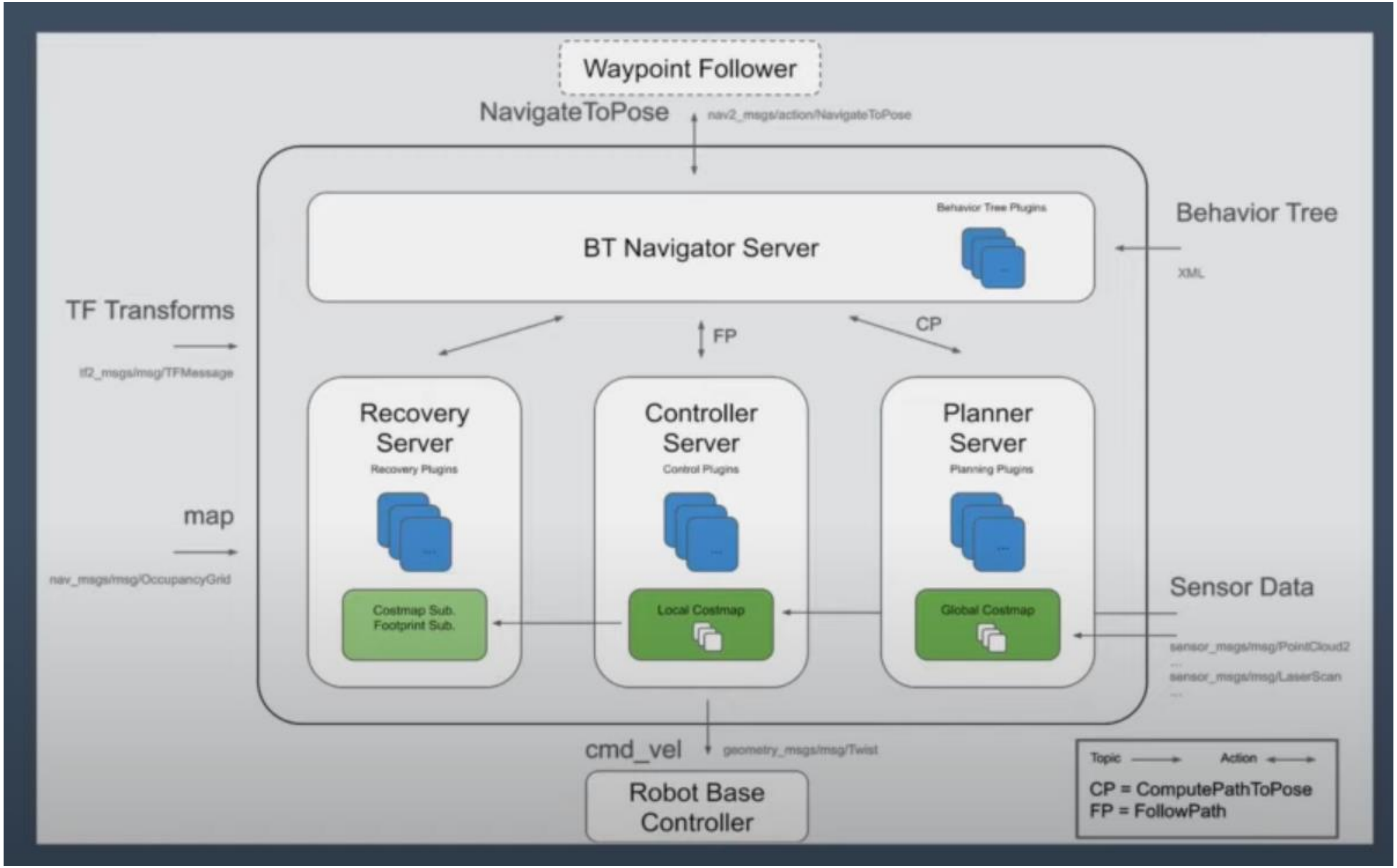
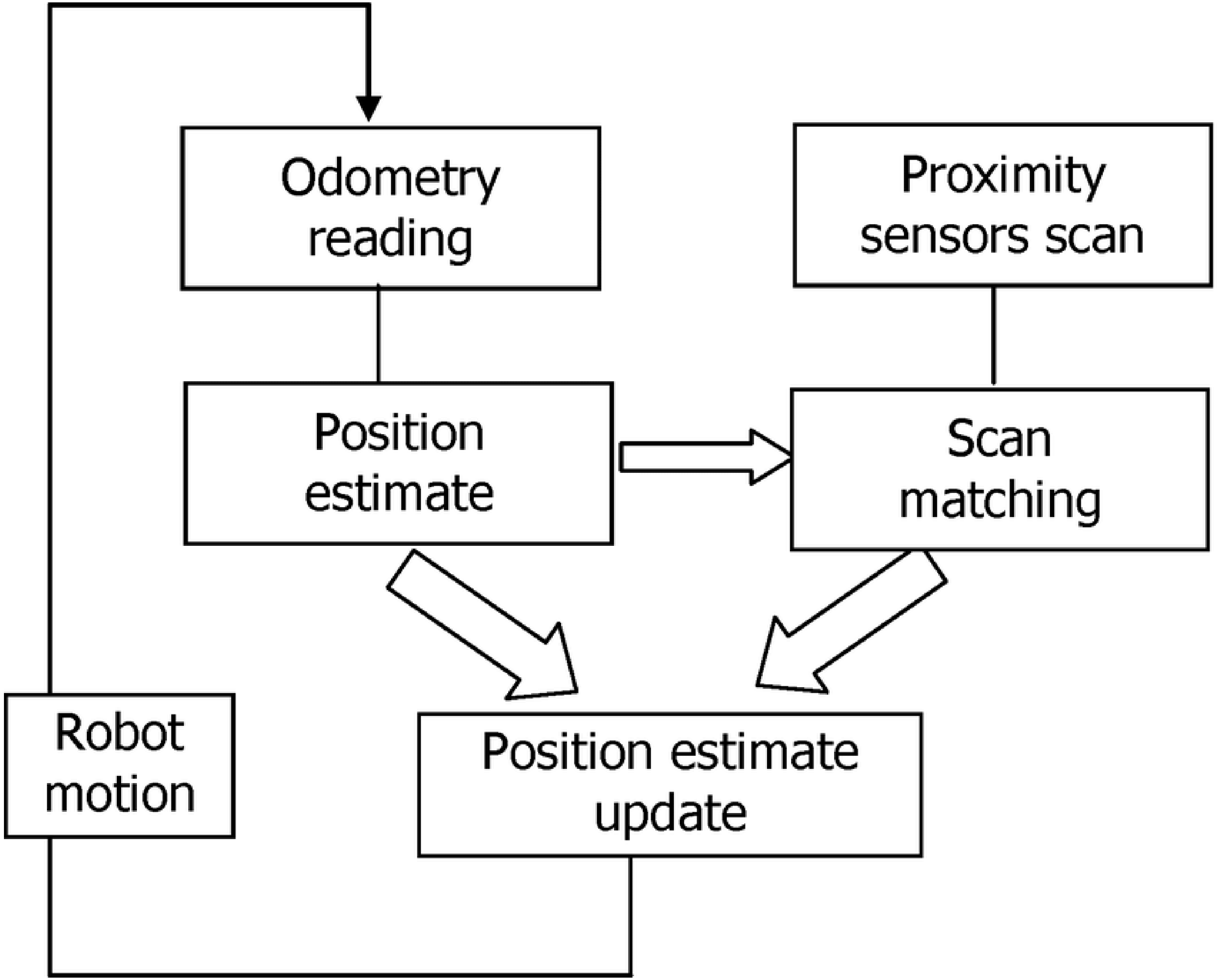


Figure 10 : Creation of simplified robot model

NAV2



Waypoint Follower
NavigateToPose
nav2_msgs/action/NavigateToPose
Behavior Tree Plugins
BT Navigator Server
Behavior Tree
XML
TF Transforms
tf2_msgs/msg/TFMessage
FP
CP
Recovery Server
Recovery Plugins
Controller Server
Control Plugins
Planner Server
Planning Plugins
map
nav_msgs/msg/OccupancyGrid
Costmap Sub.
Footprint Sub.
Local Costmap
Global Costmap
Sensor Data
sensor_msgs/msg/PointCloud2
sensor_msgs/msg/LaserScan
cmd_vel
geometry_msgs/msg/Twist
Robot Base Controller
Topic
Action
CP = ComputePathToPose
FP = FollowPath


Odometry reading
Proximity sensors scan
Position estimate
Scan matching
Robot motion
Position estimate update