

CHAPTER 5 – CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

- In this project, the system mostly utilizes Linorobot software however since Linorobot software is built based on ROS which results in only Linorobot firmware (Teensy level processing) is used in the system as it is ported to ROS2. It is reliable and beginner-friendly to use for AMR applications as most libraries are already inside such as robot odometry sensing, PID controller for actuation, and robot actuation itself. The only limitation of the system is most of the code needs to be ported from ROS to ROS2 as less documentation currently exists for ROS2.
- MicroROS which is used to connect Teensy 4.1 to ROS2 is considered not reliable it has synchronizing and initializing difficulties as to connect MicroROS to Teensy 4.1 requires several repetitions for it to succeed. However, when the devices connect, there will not be any disconnections during use.
- IMU MPU9250 is reliable to be used to collect headings data of an AMR with a measurement error of 0.01733% and accuracy reliability at 99.98267%. However, it still requires regular calibration at least once a week.
- Rotary Encoder HE50B is reliable to be used to calculate the angular velocity of the robot which will be used for robot odometry with an average error of 0.52% on the right motor and 1.29% on the left motor compared to the set value.
- GPS-RTK2 module is reliable especially used as an RTK-GNSS as it can have an accuracy ranging from one centimeter to four centimeters at different weather conditions as well as on long-period data taking. The base station which has a survey-in period of one hour will persist its accuracy after three hours with an average of 2.57 centimeters accuracy on the rover. Both the base station and the rover can receive accurate geospatial position during any kind of weather with an average of 2.57 centimeters on sunny weather, 2.77 centimeters on cloudy weather, 4 centimeters on rainy weather, and 2.81 centimeters at night. The only

limitations that RTK-GNSS can experience is when the antenna is covered with buildings or any other materials that blocked the antenna view, it will disrupt and output really bad readings on the geospatial position.

- RPLiDAR A2 is reliable to be used for object detection as it measures distance accurately with an average of 0.11 meters and a standard deviation of 0.32 centimeters. It also captures the shape of objects quite accurately compared to real objects. The limitations of LiDAR are however if the object is transparent like glass or reflective like a mirror, it will not be able to capture the reflections of the laser.
- Mapviz Visualization and GPS2XYZ package are reliable to be used for robot localization using RTK-GNSS geospatial position as it can accurately show the position of the robot on a map and calculate the distance the robot has traveled using geospatial position converted in the ENU coordinates.
- Robot actuation using microROS and Linorobot firmware is also reliable as it can actuate flexibly depending on the input that is given which is `/cmd_vel` topic of ROS2. As mentioned before, microROS will have no connection problem once the system is connected despite the difficulty to initialize it. However, the PID tuning for the system is not ideal and efficient thus robot maneuvering is not very responsive. The movements of the robot can also be visualized clearly on Mapviz Visualization which will ease navigation on AMR.
- Table 5-1 shows in summary the completion of the project and what requires to be developed in the future.

Test Result Summary	Tasks	Done?
	Utilize ROS2 in the system	✓
	Linorobot in ROS2	✓
	Utilize MicroROS for Teensy 4.1	✓
	Utilize IMU and rotary encoder in the system	✓
	Robot odometry using IMU and Rotary Encoder	✗
	Utilize RTK-GNSS in the system	✓
	Utilize Mapviz visualization in the system	✓
	Combine robot odometry, IMU, rotary encoder and RTK-GNSS with EKF for localization	✗
	Utilize LiDAR in the system	✓
	Utilize LiDAR for object detection	✗
	Utilize ultrasonic sensor for emergency brake in the system	✗
	Robot actuation with keyboard teleoperation	✓
	Robot actuation with automated navigation	✗

Table 5-1 Test Result Summary

5.2 Recommendation

- As ROS2 still requires a lot of development, more dedications in porting the ROS package to ROS2 is highly recommended. Codes and documentation of this project can be found in:

https://github.com/nicolasaw/outdoor_amr_ros2

It is hoped using the codes provided, further development of the AMR which is capable of navigating on command while selecting target locations on a UI (User Interface) can be achieved.

- As Linorobot software has not been fully utilized in ROS2, porting it to ROS2 is highly recommended as most AMR applications can be found in Linorobot software. In addition, further PID tunings or control can be added to the system as for 4WD driven with chain robot, the actuation is not smooth and its axis of rotation is not in the middle of the robot in which for 4WD with two motors driving the center of rotations for the robot supposed to be in the middle of the robot.
- As the current development is not capable of automated navigation yet, it is recommended to pursue it. Several ROS2 packages such as ROS Navigation2 which is the predecessor of ROS Navigation Stack back in ROS. This package emphasizes safety in navigation from point A to point B. It has AMCL for localization, Nav2 Planner to plan route path from one point to another, Nav2 Controller to control the robot actuation, and many more. (Macenski *et al.*, 2020) For more information on Nav2, more documentations are available in:

<https://navigation.ros.org>

- As Mapviz Visualization is used in this project, the whole infrastructure of Mapviz has not been fully utilized due to not all of its functions has been fully ported to ROS2. It is highly recommended to explore more about the plug-ins on Mapviz as it has several plug-ins that may assist the development of AMR or SDC such as GNSS data, images from the camera, laser scans from LiDAR or depth image, multi-resolution image for maps, robot odometry, and position,

navigation on UI as well as frame transformations. For more information on Mapviz, more documentations are available in:

<https://wiki.ros.org/mapviz>

- As the current development has not reached full automation and still relies on keyboard teleoperation for actuation, it is highly recommended to develop the robot until it can actuate automatically without human interventions in a Level 4 automation manner. As data for localization from RTK-GNSS, IMU, and rotary encoder has been obtained, combining all the three data are necessary, and using an EKF is recommended. Using geospatial position, targets can be set for the robot to move and Linorobot software will calculate the route plan for the robot. In addition, LiDAR will be used for object detection and if further improved, object recognition with 3D cameras can be added for better output in automated navigation.