International Journal on Robotics, Automation and Sciences

Simulation based Analysis of Encoder Resolution on Differential Drive AMR Odometry

Muhammad bin Hishamuddin, Lim Yun Ler*, Thangavel Bhuvaneswari and Min Thu Soe

***Abstract* – This research explores the impact of encoder resolution on the odometry accuracy and navigational performance of a differential-drive Autonomous Mobile Robot (AMR), using the Automated Trash Mobile Robot (ALTO) as a test platform. Encoder pulse-per-revolution (PPR) values ranging from 40 to 4096 were simulated in Gazebo. A custom encoder and odometry simulation algorithm were developed and integrated into the ROS1-based navigation stack. Controlled experiments—including straight-line, rotational, and dynamic path tests—were conducted in virtual environments to compare positional accuracy using /odom, /amcl_pose, /global_pose, and /world_pose. Results showed that higher PPR values improved odometry precision, particularly in orientation estimation, but had limited influence on global pose accuracy under AMCL-based sensor fusion. While lower resolutions caused noticeable drift, AMCL maintained robust localization. The findings offer practical guidance for optimizing encoder selection, balancing cost and performance in industrial AMR deployments.**

Keywords*— *Encoder, Resolution, Simulation, Robot Operating System (ROS), Autonomous Mobile Robot (AMR).

I. INTRODUCTION

Autonomous mobile robots (AMRs) are increasingly relied upon for navigation in various industrial and commercial applications, with odometry serving as a key method for estimating the robot's position and movement. Wheel encoders are integral to this process, as they measure wheel rotations to calculate the robot's displacement and velocities [1]. However, challenges such as wheel slip, encoder precision limitations, and cumulative errors from dead reckoning can lead to significant inaccuracies in navigation, especially in differential-drive robots [2], [3], [4]. While much research has focused on enhancing odometry through advanced techniques such as sensor fusion and visual localization [5], the impact of encoder resolution, measured in pulses per revolution (PPR) on odometry accuracy remains an underexplored area.

Since odometry estimates are derived directly from wheel encoder pulses, the resolution (PPR) determines how finely the robot can detect and respond to small movements. Inaccurate or low-resolution encoder data can cause cumulative drift in both position and orientation, particularly during long-distance travel or rotational maneuvers. Therefore, encoder resolution is a fundamental factor influencing the accuracy and stability of AMR localization systems over time. While much research has focused on enhancing odometry through advanced techniques such as sensor fusion and visual localization [5], [6], the impact of encoder resolution, measured in pulses

*Corresponding Author email: lim.yun.ler@student.mmu.edu.my, ORCID: 0009-0002-4757-5736
Muhammad bin Hishamuddin is with Centre of Robotics and Sensing Technologies, TM Research and Development Sdn Bhd, Cyberjaya, Malaysia (e-mail: muhammad@tmrnd.com.my).
Lim Yun Ler is with Faculty of Engineering and Technology, Multimedia University, Melaka, Malaysia. (lim.yun.ler@student.mmu.edu.my)
Thangavel Bhuvaneswari is with Centre for Advanced Robotics, Faculty of Engineering and Technology, Multimedia University, Melaka, Malaysia (e-mail: t.bhuvaneswari@mmu.edu.my).
Min Thu Soe is with Centre for Advanced Robotics, Faculty of Engineering and Technology, Multimedia University, Melaka, Malaysia (e-mail: stmin@mmu.edu.my).

International Journal on Robotics, Automation and Sciences (2025) 7, 3:87-96
https://doi.org/10.33093/ijoras.2025.7.3.11
Manuscript received: 30 Jun 2025 | Revised: 8 Aug 2025 | Accepted: 16 Aug 2025 | Published: 30 Nov 2025

Published by MMU PRESS. URL: http://journals.mmupress.com/ijoras

per revolution (PPR) on odometry accuracy remains an underexplored area.

Higher PPR encoders provide more detailed motion data, potentially leading to more precise odometry calculations [7], [8], [9]. However, the benefits of higher resolution may diminish beyond a certain threshold due to processing limitations and hardware constraints. Therefore, examining the performance of configurations with both high and low PPR can help identify the ideal balance between cost and performance [10]. This study investigates the effects of encoder resolutions ranging from 40PPR to 4096PPR on the navigation performance of a differential-drive AMR. The 40PPR encoder represents a low-resolution baseline, while the 4096PPR encoder explores the potential upper limit of improvement in odometry accuracy.

To evaluate these effects, the ROS Gazebo Simulation has been used to replicate real world conditions, and the robot performance is assessed in controlled scenarios. This approach offers a cost-effective, risk-free environment for optimization before physical hardware investments, reducing development costs and accelerating testing [11], [12]. The research gap is addressed by offering practical insights into how encoder resolution influences AMR navigation, specifically for improving the performance of the Automated Trash Mobile Robot (ALTO), the primary model examined at our facility.

This research contributes to AMR navigation studies by filling the gap in research surrounding encoder resolution's role in odometry. It also provides actionable recommendations for hardware upgrades, helping industrial users balance cost and performance in AMR deployment.

This paper is organized into four sections. Section II details the methodology, with II(A) focusing on the simulation environment, including the Gazebo worlds and ALTO's differential-drive model. Section II(B) describes the integration and development of a customizable encoder resolution package into the robot model. Section II(C) outlines the navigation stack, emphasizing planner tuning and AMCL parameter configuration. Section II(D) explains the experimental scenarios and data acquisition methods followed by the discussion of the obtained results in section III. The conclusion and future works are in section IV.

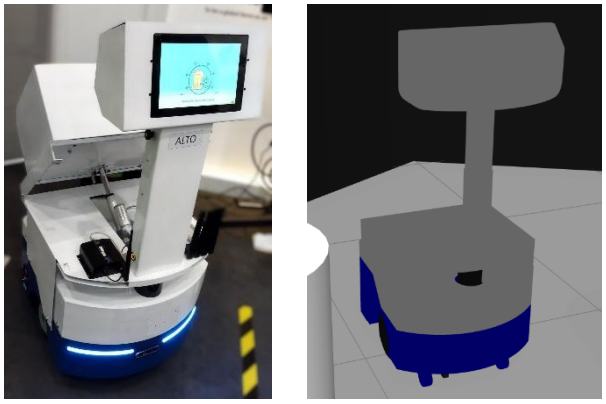
II. METHODOLOGY

This study investigates the impact of encoder resolution on the odometry accuracy and navigational performance of the ALTO robot. A series of simulations were conducted in Gazebo to create controlled test conditions, configure the AMR's navigation stack, and implement a custom encoder simulation algorithm. By simulating various encoder pulse-per-revolution (PPR) settings and with existing baseline 40PPR, the effects on robot localization and navigation are systematically assessed. The following sections describe the AMR platform, simulation setup, and encoder modeling approach.

A. AMR Platform

A detailed 3D model of the ALTO AMR robot was developed to replicate the real-world version in simulation. A comparison of the real-world ALTO and its simulated counterpart is presented in Figure 1, emphasizing their structural and functional similarities. Key specifications, including dimensions, weight, sensor configurations, encoder resolution, wheel diameter, and wheelbase width, are summarized in Table 1.

The ALTO platform is equipped with a differential drive system that estimates odometry using wheel encoders. Encoder input is provided by the hall sensors of the ZLTECH brushless motors, which generate a 90PPR pulse for wheel rotation measurement. This configuration offers a low-cost solution for AMR setups, eliminating the need for additional encoder hardware, though navigational performance remains a concern. Localization and mapping are supported by the Leishen 2D LiDAR sensor [13] with the Active Monte-Carlo Localization (AMCL) algorithm.



(a) (b)

FIGURE 1. (a) Original ALTO AMR robot (b) Gazebo model of ALTO AMR.

The robot operates using the ROS1 Noetic framework, ensuring compatibility with widely available navigation and localization packages.

TABLE 1. ALTO major hardware components.

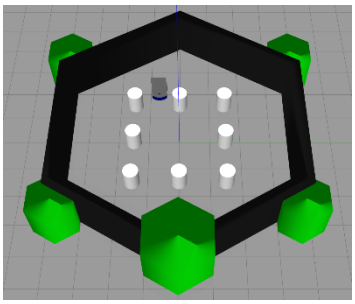
Item	Detail
ALTO AMR Platform	Physical Size: 0.5m x 0.6m x 1.16m Total Weight: 80kg Wheelbase Separation: 0.4m
Brushless Wheel Hub Motor and Encoder	Manufacturer: ZLTECH Model: ZLLG80ASM250-L V1.0 Wheel Diameter: 0.2m Motor Power: 250W Embedded Hall-Sensor (encoder): 90PPR
Lidar	Manufacturer: Leishen Model: N30103B-P Distance: 30m Accuracy: ± 3cm
Brushless Motor Controller	Roboteq SBLG2360T (Dual-channel drive)
ROS version	ROS1 - Noetic

B. Simulation Setup

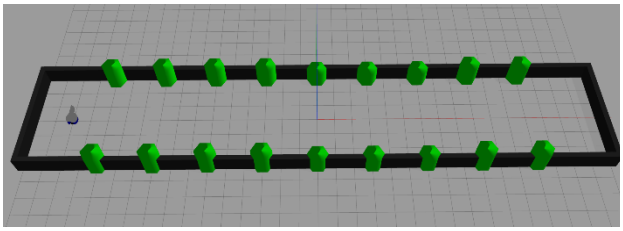
1) *Simulation Virtual World:* The simulation was conducted in Gazebo to replicate ALTO's operating conditions. Two distinct virtual test environments were created:

- *Turtlebot3 ROS Logo World:* A scaled-up version of the ROS logo world, measuring approximately 10m x 10m, with pole obstacles. as per Figure 2(a).
- *Straight Hallway:* A basic rectangular map with subtly complex walls, measuring 32m x 6m, replicating a straight hallway and designed for 20m straight-line navigation tests, as shown in Figure 2(b).

These environments were specifically designed to evaluate the effects of encoder resolution on navigational performance under controlled conditions.



(a)



(b)

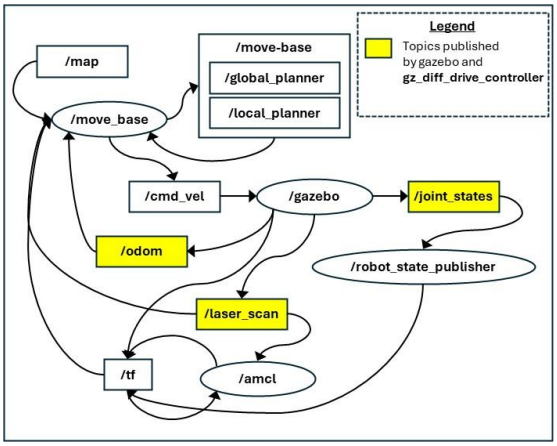
FIGURE 2. Gazebo virtual worlds for the experiment (a) A scaled-up version of ROS logo from Turtlebot3 package (b) A basic rectangular map with subtle complex walls for 20m straight-line test.

2) *Encoder Modelling in Gazebo:* Differential drive AMR simulations in Gazebo typically rely on the *gazebo_ros_diff_drive* (GRDD) plugin, a standard package for simulating motion in differential-drive robots. As illustrated in Figure 3(a), the plugin subscribes to the /cmd_vel ROS topic (rostopic) to receive velocity commands and calculates odometry data, including linear and angular velocities, which are published to the /odom rostopic. It also broadcasts the transform (TF) between /odom and /base_link frames. By default, the GRDD plugin assumes an idealized model where encoder data is perfectly accurate, without accounting for real-world factors such as resolution limits, sensor noise, or wheel slip. As a result, odometry outputs, including position updates, lack the inherent inaccuracies observed in physical AMR robots.

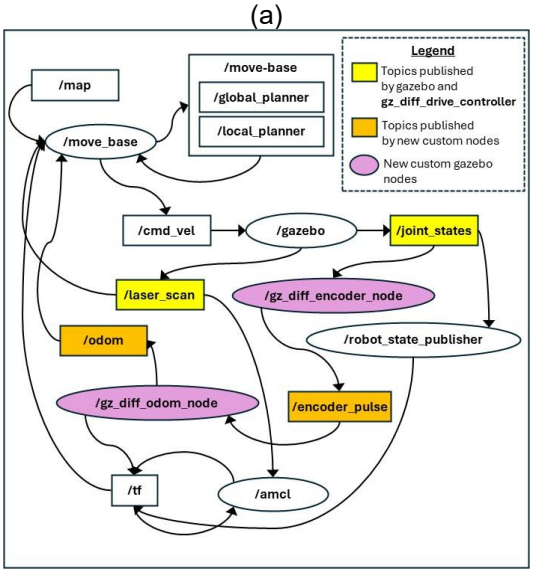
To address these limitations, a custom encoder simulation algorithm (rosnode), *gz_diff_encoder_node* (GDEN), was developed to better represent real-world conditions. As illustrated in Figure 3(b), the GDEN subscribes to the */joint_states* topic, which provides rotational position data of the virtual ALTO wheels in radians. The encoder pulses are calculated using (1), where the pulses per revolution (PPR) value is specified in the *config.yaml* file to simulate the desired encoder resolution.

$$Encoder\ Pulses = round\ down\left(\frac{\theta \times PPR}{2\pi}\right) \quad (1)$$

Here, θ refers to the rotational position data from the */joint_states* rostopic, and 2π corresponds to one full wheel rotation. The computed encoder pulses are then published to the */encoder_pulse* rostopic at a rate of 20Hz, matching the Roboteq motor controller's USB-serial interface for hall sensor encoder data. This alignment confirms the success of the simulated encoder model in replicating the behavior of the physical system.



(a)



(b)

FIGURE 3. ROS navigation stack for ALTO gazebo simulation (a) Using gz_diff_drive_controller plugin. (b) Using gz_diff_drive_controller plugin and new custom encoder plugin rosnode.

A second custom odometry rosnode *gz_diff_odom_node* (GDON), was subsequently developed to subscribe to the newly generated */encoder_pulse* rostopic. This node computes and

publishes a more realistic */odom* rostopic, and a new TF, utilizing the commonly used differential-drive kinematic equation [14], [15], [16], [17], which is adjusted to correspond with the variable PPR settings used in the simulation.

The GRDD plugin was then retained solely to simulate motor-wheel motion and to publish wheel rotation data to the */joint_states* rostopic. The */odom* rostopic generation by the GRDD plugin was disabled, allowing the more realistic */odom* rostopic from GDON to be used instead. The remainder of the navigation stack was configured as shown in Figure 3(b).

C. Navigation Stack Parameter Configurations

The navigation stack parameters were configured based on corresponding research on AMRs and adjusted for the specific hardware setup [9], [18]. These configurations, including Gmapping SLAM, LiDAR simulation, AMCL, and the local planner, were optimized for both ideal and realistic conditions. Detailed parameter settings are provided in Table 2, with references to relevant studies included in the following section.

TABLE 2. Tuned gazebo simulation and navigation parameters.

Item	Parameters	Valuea	Valueb
Gazebo Lidar Parameters	samples: minimum range: maximum range: noise_type: noise_mean: noise_stddev:	1000 0.15 6.0 Gaussian 0.0 0.01	1000 0.15 6.0 Gaussian 0.0 0.03
AMCL Parameters	max_particles laser_max_range: laser_max_beams: transform_tolerance: laser_z_hit: laser_z_short: laser_z_rand: laser_sigma_hit: kld_z: kld_err:	-c	3000 3.5 180 0.4 0.9 0.05 0.5 0.2 Default 0.02
DWA Planner Parameters	xy_goal_tolerance: yaw_goal_tolerance: goal_distance_bias: max_vel_x: max_vel_theta: acc_lim_x: acc_lim_theta:	-c	0.05 0.052 15.0 0.45 1.0 0.7 4.9

For Gmapping configuration
For Autonomous Navigation configuration
Not required for Gmapping

1) Gmapping SLAM: A one-time 2D occupancy grid map was generated for each virtual world, using the Gmapping algorithm and ideal odometry data from the GRDD plugin to ensure that the map data would not affect navigational accuracy performance tests. Odometry was later switched to the custom encoder package (GDEN) and custom odometry package (GDON) for the navigation tests.

2) LiDAR Configuration: For Gmapping, the gazebo LiDAR simulation plugin was configured with the *std_dev* parameter set to 0.01 equivalent to ±1 cm accuracy to ensure an ideal map was produced. For navigational tests, the LiDAR plugin was then configured to mimic the actual LiDAR condition, where the *std_dev* parameter was set to 0.03, equivalent to ±3 cm accuracy, to be able to simulate the actual ALTO condition.

3) AMCL Configuration: The AMCL parameter configuration, was adopted from [9], in which was used for an industrial use case. However, not all parameters were identically used, as some had to be tuned slightly off to match our hardware limitations as summarized in Table 2. The *laser_max_range* parameter was tuned to 3.5 meters to ensure that the effective LiDAR range is reduced to simulate limited range of an actual AMCL condition.

4) Path Planner Configuration: The global planner utilized the default package for path planning. The Dynamic Window Approach (DWA) planner was used for local planning to generate the velocity commands, and tuned parameters were adopted from [18] with slight tuned variation as in Table 2.

D. Experimental Design

1) Testing Scenarios: The experiment was designed to test the accuracy performance of the navigation stack using four different simulated encoder resolutions: 40PPR, 90PPR, 1024PPR, and 4096PPR. The encoder PPR values were selected to match the available specifications of ALTO's brushless motor wheel hub vendor product line from ZLTECH.

Three types of navigational tests were performed to evaluate the robot's positional accuracy with the different encoder resolutions:

a) Static-Line Test (SL): A 20-meter straight-line navigation task was conducted in the custom straight hallway world from Figure 2(b), with a single navigational goal point set exactly 20 meters straight ahead in the x-direction from the initial position, without any variance in the y-direction. This test provided a controlled scenario in which the robot moved along a known, predictable path, offering a baseline for comparing the accuracy of different localization methods. The effects of the robot's internal navigation system were isolated, eliminating the complexities introduced by dynamic obstacles or changes in the environment [19].

b) Static Multi-Turn Test (SMT): A static rotational accuracy test was conducted in the ROS logo world from Figure 2(a) by commanding the robot to perform five continuous 360-degree rotations in place while positioned stationary in the simulation. The test aimed to evaluate odometry stability and highlight rotational drift caused by encoder inaccuracies. Additionally, it served to identify systematic errors in the robot's motion model, such as biases or calibration issues, and assess the responsiveness of the navigation stack during continuous rotational commands [20]

c) Dynamic Long-Distance Test (DLD): A dynamic long-distance navigation test was conducted in the ROS logo world from Figure 2(a), with the robot navigating approximately 95 meters around the map, through straight paths, corners, and obstacles. The test involved following a predefined path in the Gazebo simulation, incorporating multiple turns and uniformly distributed waypoints to ensure dynamic, uninterrupted movement. This approach avoided the localized biases often introduced by traditional point-to-point navigation and provided a more accurate representation of real-world long-distance navigation scenarios [9].

2) Data Collection and Assessment: For each test scenario, thirty trials were performed to ensure statistically reliable results [18]. Using the wheelbase and wheel diameter from Table 1, the differential-drive kinematic equation [17] was applied to estimate the expected linear and angular accuracy based on encoder tick (pulse) distances, as summarized in Table 3. Additionally, the LiDAR accuracy was considered. Linear accuracy was the primary concern and was selected to meet the requirements of all subjects, finalizing at ±0.05 m. The angular accuracy was set at approximately ±3 degrees (equivalent to ±0.052 radians). These tolerances became the reference for the DWA planner accuracy parameters:

- *xy_goal_tolerance*: ±0.05 meters
- *yaw_goal_tolerance*: ±0.052 radians

TABLE 3. Accuracy tolerance estimation.

Sources	Expected Linear Accuracy	Expected Angular Accuracy
Lidar	±30mm	-
Encoder 40PPR	±15.7mm	±4.5 degrees
Encoder 90PPR	±7.98mm	±2.0 degrees
Encoder 1024PPR	±0.61mm	±0.18 degrees
Encoder 4096PPR	±0.15mm	±0.045 degrees

Positional data (X-Y coordinates and orientation angle) were collected from the following rostopics:

- ***/odom***: Robot positional data derived from direct encoder-based odometry.
- ***/amcl_pose***: Robot positional data estimated by AMCL localization algorithm, which is relative to the map frame. This rostopic does not update at a consistent frequency, rather it updates only when certain linear or angular motion is applied [21].
- ***/global_pose***: A custom rostopic generated from a custom rosnode, which extracted positional data from the TF tree between */map* and */base_link*. This topic served as a baseline reference to detect discrepancies in the */amcl_pose* topic, which does not update frequently. This rostopic was set to publish an update at a consistent 20Hz.
- ***/world_pose*:** The ground-truth positional data of the robot model to the virtual simulation world, extracted from the */gazebo/model_states* rostopic. This data represented the robot's true position in the simulated world frame.

In an ideal robot, all four data sources would match 100%. However, in an actual robot, discrepancies are inevitable due to the different localization algorithms on which each data source relies. The performance and accuracy of these four data sources were analyzed to form the basis of this experiment.

III. RESULTS AND DISCUSSION

Three simulation scenarios for autonomous navigation were conducted using four different encoder PPR settings. Initially, slight issues with navigation were observed, where the simulated ALTO robot missed some navigation goal points. However, these issues were resolved after tuning the DWA planner parameters, as detailed in Table 2. The ALTO robot successfully completed all courses with minimal abnormalities.

Pose data for the final goal point from each navigation test were recorded, and the positional and orientation errors were extracted and compared to the set tolerances of ±0.05 meters and ±3 degrees, respectively. The final statistical error data for goal position errors are presented in three tables: Table 4 for odometry data from */odom*, Table 5 for ground-truth data from */world_pose*, and Table 6 for AMCL data from */global_pose*. Quaternion orientation data were converted to degrees, and the minimum error, maximum error, standard deviation, and accuracy percentage were recorded.

The results from the SL test showed consistent and predictable patterns for x-error and y-error. The SL test supported the hypothesis that a good standard deviation for orientation errors could be achieved under the ±3degree tolerance condition, with the 40 PPR encoder achieving a standard deviation of 1.518 and the 4096 PPR encoder scoring 0.797. Orientation errors were minimal in this controlled situation. The SMT test also showed expected results, where higher PPR encoders provided more consistent and accurate performance. The DLD test, however, provided more interesting data for analysis, as reflected in Figure 4(c). The data distribution analysis clearly indicated performance differences across tests, allowing for the selection of the most relevant data for further comprehensive analysis.

The */amcl_pose* and */global_pose* data were found to match with only negligible differences, as shown in the tabulated data in Figure 4. This outcome was anticipated, as the AMCL algorithm updates the robot_model TF directly. However, the */amcl_pose* topic updates at a slower pace, triggered by minimum linear or angular motion defined by AMCL parameters such as *transform_tolerance*. For this analysis, the */global_pose* data were considered effectively identical to the AMCL output, with minor variations attributed to the data update process.

TABLE 4. Odometry statistics of navigational goal reaching performance (odom pose).

		40PPR			90PPR			1024PPR			4096PPR		
		x-error	*y-error*	*θ-error*	*x-error*	*y-error*	*θ-error*	*x-error*	*y-error*	*θ-error*	*x-error*	*y-error*	*θ-error*
SL	min[a]	-0.126	-0.280	-2.250	-0.169	-0.305	-1.000	-0.058	-0.117	-1.494	-0.055	-0.094	0.088
	max[b]	0.010	0.528	4.500	-0.002	0.305	4.000	0.026	0.059	2.373	0.001	0.056	3.669
	std-dev[c]	0.028	0.214	1.518	0.028	0.158	1.306	0.018	0.039	0.859	0.015	0.033	0.797
	%[d]	76.7	23.3	93.3	90.0	26.7	96.7	93.3	73.3	100.0	90.0	86.7	96.7
SMT	min	-0.251	-0.146	-4.500	-0.087	-0.028	-4.000	-0.001	-0.001	-2.461	-0.001	-0.001	-2.791
	max	0.094	0.188	78.750	0.004	0.175	0.000	0.001	0.001	-0.176	0.000	0.000	0.044
	std-dev	0.077	0.075	14.674	0.016	0.033	0.900	0.000	0.000	0.451	0.000	0.000	0.676
	%	76.7	56.7	83.3	96.7	96.7	96.7	100.0	100.0	100.0	100.0	100.0	100.0
DLD	min	-0.179	-0.420	-5.490	-0.112	-0.315	-2.740	-0.106	-0.315	-2.519	-0.114	-0.198	-2.409
	max	0.373	0.314	3.510	0.256	0.182	3.260	0.241	0.090	0.821	0.242	0.133	0.557
	std-dev	0.099	0.127	2.108	0.094	0.114	1.489	0.073	0.076	0.814	0.068	0.057	0.805
	%	13.3	43.3	76.7	16.7	16.7	93.3	10.0	13.3	100.0	3.3	10.0	100.0

[a] Minimum error from the goal point
[b] Maximum error from the goal point
[c] Standard deviation
[d] Percentage of data within the assumed goal tolerance

TABLE 5. Ground-truth statistics of navigational goal reaching performance (world pose).

		40PPR			90PPR			1024PPR			4096PPR		
		x-error	*y-error*	*θ-error*	*x-error*	*y-error*	*θ-error*	*x-error*	*y-error*	*θ-error*	*x-error*	*y-error*	*θ-error*
SL	min	-0.127	-0.043	-1.752	-0.187	-0.012	-1.324	-0.065	-0.011	-1.159	-0.059	-0.008	0.034
	max	0.019	0.061	3.109	-0.008	0.047	3.200	0.009	0.070	2.542	-0.005	0.042	3.022
	std-dev	0.030	0.021	1.212	0.030	0.013	1.025	0.017	0.016	0.826	0.014	0.010	0.708
	%	93.3	96.7	96.7	86.7	100.0	96.7	83.3	93.3	100.0	80.0	100.0	96.7
SMT	min	-0.260	-0.148	-4.357	-0.106	-0.019	-2.901	-0.106	-0.019	-2.901	0.000	0.000	-2.819
	max	0.105	0.187	80.341	0.002	0.165	0.304	0.002	0.165	0.304	0.002	0.001	0.008
	std-dev	0.079	0.077	14.917	0.020	0.030	0.843	0.020	0.030	0.843	0.000	0.000	0.670
	%	70.0	56.7	80.0	96.7	96.7	100.0	96.7	96.7	100.0	100.0	100.0	100.0
DLD	min	-0.105	-0.069	-1.490	-0.049	-0.034	-0.009	-0.063	-0.019	0.168	-0.063	-0.026	0.145
	max	0.011	0.057	4.932	0.008	0.069	3.551	0.001	0.041	2.677	-0.005	0.035	2.479
	std-dev	0.022	0.021	1.569	0.014	0.023	0.982	0.015	0.016	0.650	0.014	0.015	0.624
	%	90.0	93.3	86.7	100.0	96.7	90.0	90.0	100.0	100.0	96.7	100.0	100.0

TABLE 6. Amcl statistics of navigational goal reaching performance (global pose).

		40PPR			90PPR			1024PPR			4096PPR		
		x-error	*y-error*	*θ-error*	*x-error*	*y-error*	*θ-error*	*x-error*	*y-error*	*θ-error*	*x-error*	*y-error*	*θ-error*
SL	min[a]	-0.040	-0.046	-1.872	-0.046	-0.033	-1.891	-0.045	-0.034	-1.286	-0.049	-0.035	-0.155
	max[b]	0.047	0.014	2.750	0.030	-0.003	2.907	0.032	0.032	2.245	-0.001	-0.001	3.431
	std-dev[c]	0.027	0.014	1.380	0.016	0.006	1.185	0.017	0.014	0.746	0.010	0.007	0.829
	%[d]	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	96.7
SMT	min	-0.039	-0.043	-2.645	-0.026	-0.047	-2.729	-0.026	-0.047	-2.729	-0.013	-0.021	-2.108
	max	0.049	0.018	78.763	0.031	0.038	1.056	0.031	0.038	1.056	0.001	0.005	0.135
	std-dev	0.025	0.019	14.516	0.014	0.019	0.845	0.014	0.019	0.845	0.004	0.006	0.496
	%	100.0	100.0	96.7	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
DLD	min	-0.029	-0.046	-0.233	-0.013	-0.047	0.267	-0.006	-0.048	0.241	-0.005	-0.048	0.603
	max	0.029	0.040	4.159	0.029	-0.011	3.526	0.030	-0.030	3.120	0.025	-0.027	2.704
	std-dev	0.010	0.017	1.502	0.010	0.009	0.829	0.008	0.005	0.591	0.006	0.006	0.479
	%	100.0	100.0	73.3	100.0	100.0	93.3	100.0	100.0	96.7	100.0	100.0	100.0

[a] Minimum error from the goal point
[b] Maximum error from the goal point
[c] Standard deviation
[d] Percentage of data within the assumed goal tolerance

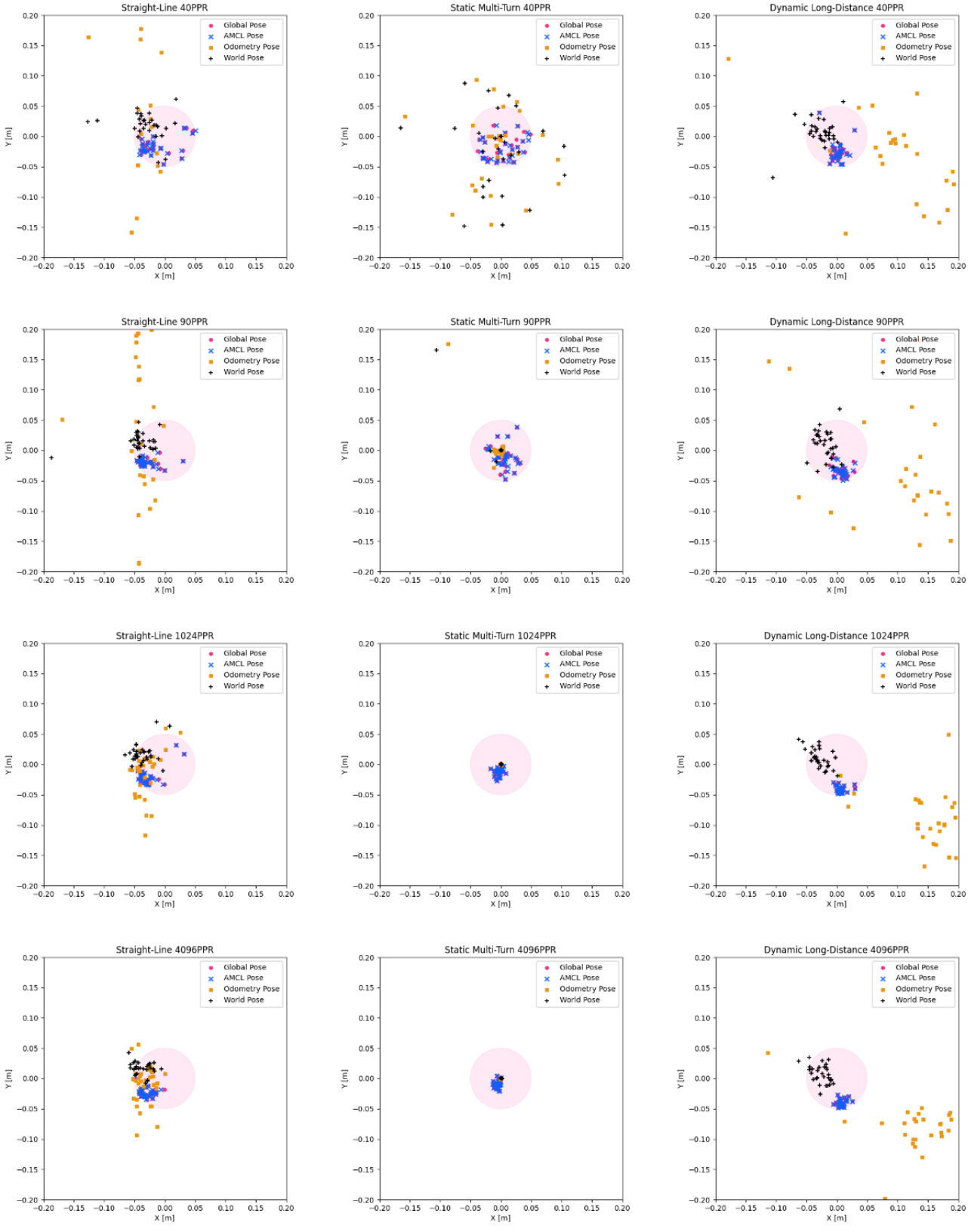
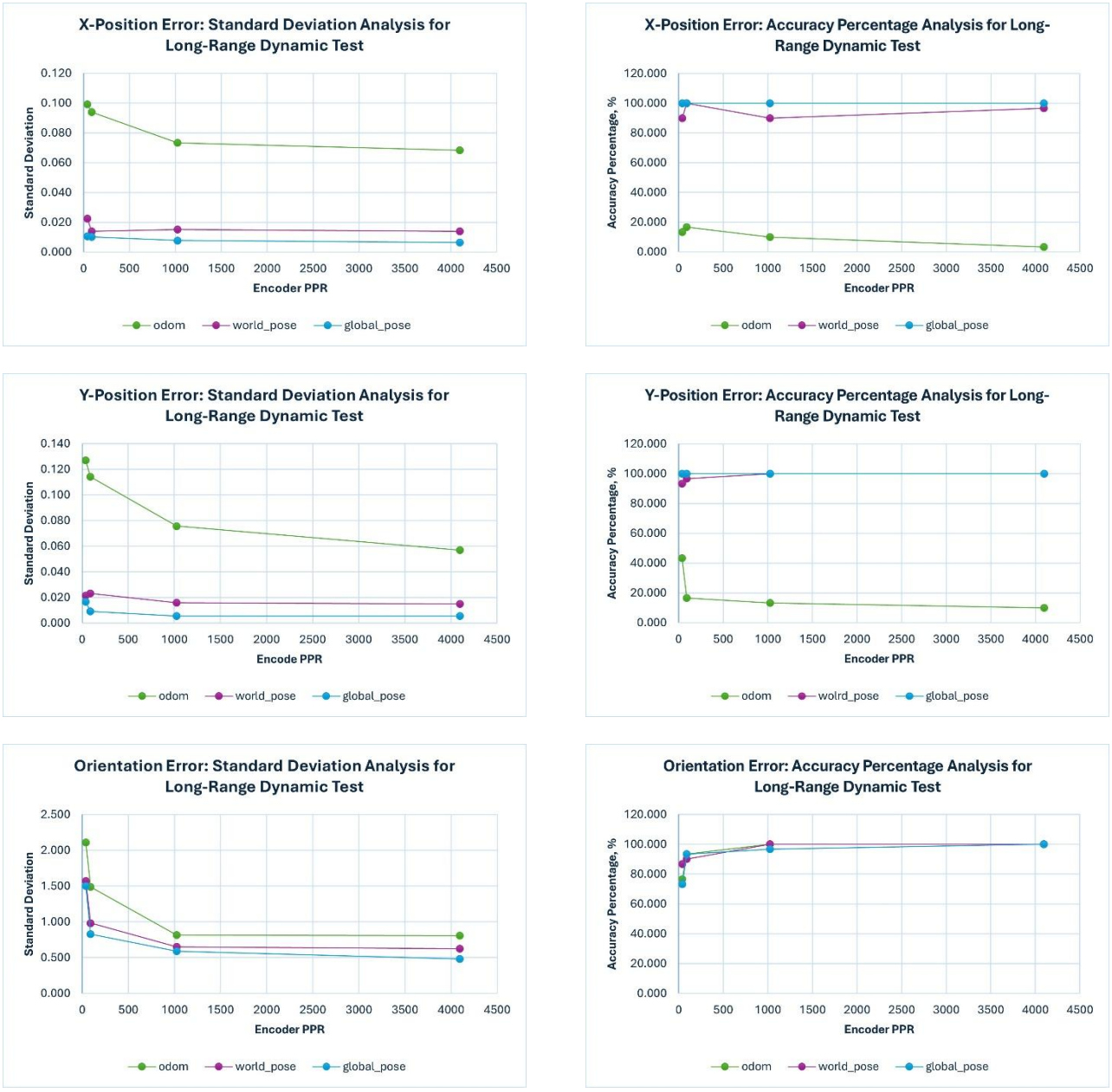


FIGURE 4. **XY positional errors of ALTO navigation simulation (a) (first column) Straight-Line Test (ST) (b) (second column) Static Multi-Turn Test (SMT) (c) (third column) Dynamic Long-Distance Test (DLD).**



(a) (b)

FIGURE 5. XY positional errors of ALTO navigation simulation (a) (first column) Straight-Line Test (ST) (b) (second column) Static Multi-Turn Test (SMT) (c) (third column) Dynamic Long-Distance Test (DL).

Following the analysis of the initial simulation results, a more detailed comparison of the accuracy performance across different encoder PPR settings was conducted. The raw error data was visualized in Figure 4, where it was observed that the lower 40PPR encoder produced more scattered error distributions around the tolerance circle, while higher PPR encoders demonstrated a more focused distribution, with a noticeable reduction in standard deviation. To assess the goal-reaching performance of navigation, both the standard deviation of errors and the accuracy percentage were considered, highlighting the impact of varying encoder PPRs. Positional deviations were measured as x-error and y-error, with x-error corresponding to forward and backward position error, and y-error representing sideways position error of the robot. It was found that greater control over positional accuracy was exhibited in terms of x-error, while higher variability was observed in y-error, which is more associated with path-following trajectories. Surprisingly, favorable accuracy percentages and standard deviations for both x-error and y-error were recorded across all encoder types, as shown in Figure 5(a) and 5(b).

Further analysis of the odometry data revealed poor performance, with the standard deviation exceeding 0.05 for all encoder types within the ±0.05m tolerance range. Specifically, during the DLD test, the x-error standard deviation was recorded as 0.099, 0.004, 0.073, and 0.068 for the 40 PPR, 90 PPR, 1024 PPR, and 4096 PPR encoders, respectively. Similar trends were observed in the y-error data, and it was noted that none of the encoder types achieved more than 20% accuracy in odometry, indicating limitations

in odometry performance across all encoder resolutions.

Despite these poor odometry results, 100% X-Y linear accuracy was recorded for the AMCL (*global_pose*) data for all encoder types, suggesting that the AMCL algorithm successfully estimated the robot's position relative to the 2D occupancy grid map, even with suboptimal odometry data from the 40 and 90 PPR encoders. Slight discrepancies in final positioning accuracy were noted when comparing AMCL estimates with the ground-truth *world_pose* data, with accuracy varying between 85% and 100%. These findings suggest that relying solely on AMCL for localization, particularly with low-resolution encoders, may still result in accuracy performance issues.

In terms of orientation accuracy, it was found that the lower PPR encoders (40PPR and 90PPR) produced orientation error accuracy percentages of less than 90%, while the higher PPR encoders (1024PPR and 4096PPR) achieved results above 90%. This indicates that encoder resolution has a significant impact on orientation accuracy, particularly in more complex navigational environments.

Despite the relatively accurate ground-truth errors, the 40 PPR encoder which the existing baseline performance could still be acceptable for AMRs, especially when AMCL is used for localization. However, discrepancies in orientation accuracy were observed, and it was inferred that, for industrial applications requiring precise orientation such as docking procedures, equipping the AMR with at least a 1024 PPR encoder is recommended. In use cases where high orientation accuracy is not critical, lower PPR encoders may be deemed sufficient.

IV. Conclusion

The simulation model successfully completed all experimental routes, validating the robustness of the custom encoder simulation framework. The study explored the effects of different encoder resolutions on the odometry performance of ROS-enabled differential drive AMRs. Results showed that higher resolutions (1024 and 4096 PPR) improved odometry stability, reducing drift and errors, but had limited impact on global pose accuracy, which remained largely unaffected by encoder resolution due to the AMCL localization method, except for orientation accuracy. These findings suggest that upgrading to higher PPR encoders enhances odometry stability, though the overall navigation accuracy may not significantly improve when using AMCL or similar global localization methods. The simulation environment, however, did not account for real-world factors such as wheel slippage or uneven terrain, which could affect odometry accuracy [3].

This research provides valuable insights for developers considering encoder upgrades, as it demonstrates that higher PPR encoders can improve odometry precision, but that global pose accuracy relies on sensor fusion methods like AMCL. The custom Gazebo package developed for encoder resolution simulation fills a gap in current tools and can be extended to other drive systems like Ackermann or skid-steer. Future studies should incorporate real-world testing to evaluate encoder performance under dynamic conditions and integrate additional sensors like IMUs or visual odometry to address challenges like slippage. Overall, this research offers actionable guidance for optimizing encoder resolution in AMRs and paves the way for further advancements in sensor integration and real-world testing.

Acknowledgment

We thank the anonymous reviewers for the careful review of our manuscript.

Funding Statement

The author acknowledges TM Research and Development Sdn. Bhd. for providing a research grant for this project, under project code RDTC/241127.

Author contributions

Muhammad bin Hishamuddin: Conceptualization, Data Curation, Methodology, Validation, Writing – Original Draft Preparation;

Lim Yun Ler: Conceptualization, Data Curation, Methodology, Validation, Writing – Original Draft Preparation;

Thangavel Bhuvaneswari: Project Administration, Supervision, Writing – Review & Editing;

Min Thu Soe: Project Administration, Supervision, Writing – Review & Editing.

Conflict of interests

No conflict of interests was disclosed.

Ethics statements

Ethical approval was not applicable to this research since it did not involve human participants, animals, or sensitive data.

References

[1] M. Sotnikova, E. Veremey and N. Zhabko, "Wheel Angular Velocity Stabilization using Rough Encoder Data," *2014 14th International Conference on Control, Automation and Systems (ICCAS 2014)*, pp. 1345–1350, 2014. DOI: https://doi.org/10.1109/ICCAS.2014.6987765

[2] D.V. Prakash, A. Elango and D. Vidhyaprakash, "Studies on Affecting Factors of Wheel Slip and Odometry Error on the Performance of Wheeled Mobile Robots-A Review," *International Journal of Robotics and Automation (IJRA)*, vol. 5, no. 1, pp. 6–16, 2016. DOI: http://doi.org/10.11591/ijra.v5i1.pp6-16

[3] M. Smieszek, M. Dobrzanska and P. Dobrzanski, "The impact of load on the wheel rolling radius and slip in a small mobile platform," *Autonomous Robots*, vol. 43, no. 8, pp. 2095–2109, 2019. DOI: https://doi.org/10.1007/s10514-019-09857-0

[4] K. Lyu, K. Wang, L. Ling, Y. Sun, Z. Shi and W. Zhai, "Influence of wheel diameter difference on surface damage for heavy-haul locomotive wheels: Measurements and simulations," *International Journal of Fatigue*, vol. 132, 2020. DOI: https://doi.org/10.1016/j.ijfatigue.2019.105343

[5] P.K. Panigrahi and S.K. Bisoy, "Localization strategies for autonomous mobile robots: A review," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 8, pp. 6019–6039, 2022. DOI: https://doi.org/10.1016/j.jksuci.2021.02.015

[6] Y. Liu, S. Wang, Y. Xie, T. Xiong and M. Wu, "A Review of Sensing Technologies for Indoor Autonomous Mobile Robots," *Sensors*, vol. 24, no. 4, p. 1222, 2024. DOI: https://doi.org/10.3390/s24041222

[7] I. Zunaidi, N. Kato, Y. Nomura and H. Matsui, "Positioning System for 4-Wheel Mobile Robot: Encoder, Gyro and Accelerometer Data Fusion with Error Model Method," *CMU. Journal*, vol. 5, no. 1, pp. 1–14, 2006. URL:https://www.thaiscience.info/journals/Article/CMUJ/10325060.pd (Accesed: 19 Dec 2024)

[8] J. Zhao, S. Liu and J. Li, "Research and Implementation of Autonomous Navigation for Mobile Robots Based on SLAM Algorithm under ROS," *Sensors,* vol. 22, no. 11, 2022. DOI: https://doi.org/10.3390/s22114172

[9] I.H. Savci, A. Yilmaz, S. Karaman, H. Ocakli and H. Temeltas, "Improving Navigation Stack of a ROS-Enabled Industrial Autonomous Mobile Robot (AMR) to be Incorporated in a Large-Scale Automotive Production," *International Journal of Advanced Manufacturing Technology*, vol. 120, no. 5–6, pp. 3647–3668, 2022. DOI: https://doi.org/10.1007/s00170-022-08883-0

[10] RLS, "Encoder resolution, accuracy and repeatability: What's the difference?" URL: https://www.rls.si/eng/encoder-handbook/resolution-accuracy-repeatability (Accessed: 10 Dec 2024)

[11] G.D. Wijaya, W. Caesarendra, M.I. Petra, G. Królczyk and A. Glowacz, "Comparative study of Gazebo and Unity 3D in performing a virtual pick and place of Universal Robot UR3 for assembly process in manufacturing," *Simulation Modelling Practice and Theory*, vol. 132, 2024. DOI: https://doi.org/10.1016/j.simpat.2024.102895

[12] M. Marian, M.-T. Georgescu, D. Popescu, F. Stîngă, H. Roibu and F. Manta, "A ROS-based Control Application for a Robotic Platform Using the Gazebo 3D Simulator," *21th International Carpathian Control Conference (ICCC)*, p. 1, 2020. DOI: https://doi.org/10.1109/ICCC49264.2020.9257256

[13] LsLidar, "N301 - Navigation & Obstacle Avoidance Lidar." URL: https://www.lslidar.com/product/n301-navigation-obstacle-avoidance-lidar/ (Accessed: 9 Dec 2024)

[14] T. Marcel and D.S. Ioan, "Design of an Electronic Direct Drive Automotive Differential using Axial Flux PSMS with Sensorless Direct Torque and Flux Control," *EPE 2022 - Proceedings of the 2022 12th International Conference and Exposition on Electrical and Power Engineering*, pp. 78–83, 2022. DOI: https://doi.org/10.1109/EPE56121.2022.9959740

[15] F.A. Salem, "Dynamic and Kinematic Models and Control for Differential Drive Mobile Robots," *International Journal of Current Engineering and Technology*, vol. 3, no. 2, pp. 253-263, 2013. URL:https://inpressco.com/kinematics-and-dynamic-models-and-control-for-differential-drive-mobile-robots/

[16] A. Topiwala, "Modeling and Simulation of a Differential Drive Mobile Robot," *International Journal of Scientific and Engineering Research*, vol. 7 , no 7, pp. 1410-1415, 2016. URL:https://ijser.org/researchpaper/Modeling-and-Simulation-of-a-Differential-Drive-Mobile-Robot.pdf (Accessed: 9 Dec 2024)

[17] U. Ruiz, "Capturing a Differential Drive Robot With a Dubins Car," *IEEE Access*, vol. 11, pp. 42124–42134, 2023. DOI: https://doi.org/10.1109/ACCESS.2023.3271592

[18] B. Cybulski, A. Wegierska and G. Granosik, "Accuracy comparison of navigation local planners on ROS-based mobile robot*," *2019 International Workshop on Robot Motion and Control*, 2019. DOI: https://doi.org/10.1109/RoMoCo.2019.8787346

[19] J. Palacín, E. Rubies and E. Clotet, "Systematic Odometry Error Evaluation and Correction in a Human-Sized Three-Wheeled Omnidirectional Mobile Robot Using Flower-Shaped Calibration Trajectories," *Applied Sciences*, vol. 12, no. 5, p. 2606, 2022. DOI: https://doi.org/10.3390/app12052606

[20] Muhtadin, R.M. Zanuar, K.E. Purnama and M.H. Purnomo, "Autonomous Navigation and Obstacle Avoidance for Service Robot," *2019 International Conference of Computer Engineering, Network, and Intelligent Multimedia (CENIM)*, pp. 125–132, 2019. DOI: https://doi.org/10.1109/CENIM48368.2019.8973360

[21] W.P.N. dos Reis, G.J. da Silva, O.M. Junior and K.C.T. Vivaldini, "An extended analysis on tuning the parameters of Adaptive Monte Carlo Localization ROS package in an automated guided vehicle," *International Journal of Advanced Manufacturing Technology*, vol. 117, no. 5–6, pp. 1975–1995, 2021. DOI: https://doi.org/10.1007/s00170-021-07437-0