Applied Computer Science 2025, vol. 21, no. 3, pp. 117–136
https://doi.org/10.35784/acs_8089

Submitted: 2025-07-11 | Revised: 2025-08-11 | Accepted: 2025-08-18

Keywords: differential wheeled robot, odometric drift characterization, external motion tracking, error analysis, educational robotics, LEGO EV3

Stanisław Piotr SKULIMOWSKI [1*], Szymon RYBKA [1], Bartosz TATARA [1], Michał Dawid WELMAN [1]

[1] Lublin University of Technology, Poland, s.skulimowski@pollub.pl, s104341@pollub.edu.pl, s103497@pollub.edu.pl, s103509@pollub.edu.pl
[*] Corresponding author: s.skulimowski@pollub.pl

Systematic drift characterization in differential wheeled robot using external VR tracking: Effects of route complexity and motion dynamics

Abstract

Industrial mobile robots face critical positioning challenges that impact manufacturing efficiency, warehouse automation productivity, and biomedical service delivery. This paper presents a reproducible framework for quantifying odometric drift in differential-drive robots, validated by consumer-grade, low-cost VR tracking. Applications include industrial automation calibration, warehouse logistics management, and precision biomedical device positioning. Through more than 750 automated experimental trials spanning a comprehensive matrix of motor configurations and path geometries, the results show that both path complexity and turn size significantly influence drift patterns. Specifically, routes with higher geometric complexity (12-15 segments) exhibited 22% greater position error than simpler paths. The analysis used advanced metrics such as the Normalized Drift Contribution Index. The results confirm robust, high-resolution drift analysis and provide a low-cost validation tool for robot calibration in manufacturing and medical instrumentation. The work provides actionable insights for optimizing robot programming, calibration, and curriculum design, and establishes a scalable protocol for benchmarking autonomous navigation systems in real-world scenarios. In addition, the methodology enables data-driven decision making for robot fleet management, reducing operational downtime compared to manual calibration methods, while providing quantitative performance benchmarks essential for industrial quality control standards.

1. INTRODUCTION

1.1. Background and motivation

Accurate positioning is critical in automated assembly lines, logistics vehicles, and surgical robots (Dong et al., 2023). However, systematic odometric drift undermines throughput, safety, and accuracy in these areas. Drift accumulation results from systematic errors such as unequal wheel parameters, encoder limitations, wheel slippage (Đuraševic & Milovanovic, 2021; Palacín et al., 2022; Xuying et al., 2017) and track contact surface uncertainties (Đurašević & Milovanović, 2017) and non-systematic errors, such as variations in surface friction (Jin & Chung, 2019). Surface irregularities (Palacín et al., 2022) or dynamic effects during acceleration and deceleration (Ortega-Contreras et al., 2023). The persistent systematic error, in autonomous systems that require continuous operation without external correction (Agrawal et al., 2025; Araujo et al., 2022), which accumulate over time, leading to significant inaccuracies that compromise mission success and safety (Wongsuwan & Sukvichai, 2015).

1.2. External motion tracking for robot validation

Traditional robot drift analysis using internal sensors (motor encoders, IMUs) suffers from self-referential bias - the same sensors that cause drift measure their own errors (Shi, 2020). External tracking systems such as the HTC Vive provide independent ground truth positioning, eliminating this bias for accurate drift characterization at a significantly lower cost than professional motion capture (Jung & Sukhatme, 2010; Merker et al., 2023).

VR tracking, originally developed for gaming, demonstrates millimeter precision suitable for robotics (Borges et al., 2018). Specifically, the Vive Tracker 3.0 has demonstrated measurement accuracy suitable for generating valid position feedback, with spatial differences as small as 10.4 mm ± 4.5 mm under optimal conditions (Hsiao et al., 2022). This consumer-grade hardware democratizes high-precision robotic validation, making systematic drift investigation accessible without specialized investment (Buchanan et al., 2023).

Despite extensive research in robot odometry, systematic drift characterization through external validation remains underexplored (Sapounidis et al., 2024). Current internal sensor fusion approaches cannot validate their own accuracy due to measurement circularity (Poma et al., 2024). The integration of external motion tracking into educational robotics offers a promising methodology to advance the understanding of fundamental navigation limitations (Urbano Lopes, de Santana Costa et al., 2024).

1.3. Applications of differential wheeled robots

Differential Wheel Robots (DWRs) are widely used in many industries, with many Automated Guided Vehicles (AGVs) and Autonomous Mobile Robots (AMRs) utilizing differential drive mechanisms for improved maneuverability and control accuracy. In these applications, uncorrected odometric drift can severely degrade performance and safety outcomes. Industrial AGV systems with differential drive configurations are widely used for automated material handling in warehouses and manufacturing facilities, where centimeter-level localization errors result in mispositioning, collisions, and lost throughput (Maga Vasile & Crenganis, 2024). The differential traction system is particularly favored in AGV design for its "high maneuverability" and ability to navigate tight spaces in controlled industrial environments.(Maga Vasile, 2024). Modern AGV implementations with differential drive, after proper calibration, achieve path errors of less than 0.02% and 0.03% for linear and angular motions, respectively (Ferreira et al., 2023).

Two-wheel differential drive AGVs are increasingly common in manufacturing applications, where "differential drive mobile robots are most commonly used in industrial applications among wheeled mobile robots" (Thai et al., 2022). These systems require precise path following for spot welding and assembly operations; unmeasured drift can cause weld misalignment or assembly defects, increasing scrap rates and rework costs (Nie et al., 2025).

Autonomous mobile robots (AMRs) with differential drive configurations perform equipment health monitoring in power plants and pipelines, where drift correction is essential to maintain map consistency in SLAM-based navigation and avoid missed inspection points (Šelek et al., 2023). The differential drive mechanism enables smooth, collision-free patrolling, which is essential for industrial monitoring and inspection tasks.

Industrial logistics applications are increasingly relying on differential-drive AGVs for "material handling, overcoming the challenges of traditional AGV systems" (Maga Vasile, 2024). These systems must navigate dynamic, multi-vehicle environments where accurate drift measurement and correction enable reliable localization, consistent mapping, and safe operation in obstacle-rich industrial environments (Simon et al., 2021). Assistive medical devices such as powered wheelchairs also rely on differential drive systems for maneuverability, and real-time drift estimation and compensation improves user safety by preventing unintentional drift, especially in confined clinical environments.

The critical importance of drift measurement in these applications is underscored by research showing that proper odometry calibration can reduce positioning errors by up to 35 times in differential drive systems operating under slip conditions (De Giorgi et al., 2023). By integrating external validation methods (e.g., VR tracking or advanced sensor fusion), systematic drift characterization supports robust control, optimized path planning, and compliance with stringent industrial accuracy requirements that are essential for AGV and AMR systems in modern manufacturing and logistics operations.

1.4. Point-to-point movement and discrete navigation

Point-to-point (PTP) motion is a fundamental DWRs navigation strategy: discrete commands with stops. This approach simplifies systematic error analysis by isolating waypoint errors, removing speed-dependent sources, and enabling static accuracy assessment (Jialong et al., 2020; Yao & Cao, 2020). Educational robotics favors PTP for its simplicity and ease of implementation (e.g. Pybricks OS), allowing focus on core navigation without complex continuous planning. Polygonal paths provide clear geometric references for drift analysis. Research confirms that discrete motion simplifies failure analysis and is practical, as industrial robots use similar PTP to perform precise tasks. The systematic nature of PTP allows controlled experiments to isolate and quantify drift contributions.

1.5. LEGO EV3 platform as valuable testbed

Despite lacking industrial-grade precision, LEGO Mindstorms EV3 remains a cornerstone of educational robotics and early-stage prototyping due to its accessibility, standardized hardware, and extensible software ecosystem (Kenny et al., 2025; Montés et al., 2021; Ribeiro et al., 2024). Widely used in university curricula and secondary school workshops, EV3 enables real-time deployment of control algorithms via MATLAB/Simulink and Pybricks, fostering hands-on learning in robotics, control theory, and algorithm design under conditions that mirror industrial workflows (Montés et al., 2018; Samara et al., 2021). The LEGO Mindstorms EV3, which supports real-time application development for education and research, is ideal for systematic drift analysis (Mitchell, 2015).

However, these popular robots exhibit significant drift that, if uncharacterized, can undermine experimental validity and educational outcomes (Kin et al., 2019). The current literature on EV3 performance mainly addresses programming, control, and educational applications, with limited attention to systematic position error characterization (Gavrilas & Kotsis, 2024). While the drift of gyroscopic sensors has been studied, a comprehensive analysis of overall positioning accuracy using independent measurement systems remains scarce.

Although the EV3's internal motors and encoders exhibit greater baseline drift than commercial AGVs, characterizing this error with external measurements (e.g., VR trackers) provides transferable insights into drift compensation techniques, calibration protocols, and validation pipelines that are directly applicable to industrial differential-drive vehicles. The resulting methodological framework bridges the gap between accessible educational platforms and real-world automation systems, ensuring that lessons learned on the EV3 are translated into robust industrial solutions.

1.6. Research objectives

This research investigates the systematic characterization of positioning drift in LEGO EV3 robots using external motion tracking validation, with a particular focus on the relationship between path geometric complexity and drift accumulation patterns (Bilal & Unel, 2021). The study uses a Vive VR motion tracker system to provide independent ground truth position data, eliminating the measurement bias inherent in internal sensor-based drift analysis (Dmytriv et al., 2024). By analyzing robot performance over systematically varied path complexities and turn characteristics, this research aims to quantify how geometric path properties influence positioning accuracy over time (Alici & Shirinzadeh, 2005).

The research objectives were:

- RO1 - To systematically quantify and characterize the drift error in the trajectory of a LEGO EV3 DWR performing discrete PTP motions, using an external motion tracking system as an independent ground truth.
- RO2 - To investigate how geometric complexity and average turn size of closed polygonal paths affect the magnitude and pattern of accumulated drift during robot navigation.
- RO3 - Evaluate the suitability and advantages of external VR-based motion tracking for accurate, reproducible assessment of mobile robot navigation accuracy in educational and research contexts compared to traditional internal sensor-based approaches.

1.7. Paper structure overview

The methods and materials chapter describes the experimental setup in detail, including the LEGO EV3 robot configuration, the use of the Vive VR motion tracker, and the discrete PTP motion protocol. The section also explains the design of the test tracks, data collection procedures, and statistical methods for drift analysis. The Results chapter presents the experimental results, including quantitative analysis of drift over various route complexities and curve characteristics. Visualizations and tables summarize the magnitude and patterns of drift observed using external tracking. The discussion chapter interprets the results in the context of the existing literature, explores the implications for robot navigation and educational practice, and evaluates the strengths and limitations of the external tracking approach. The paper concludes by summarizing the main findings, their significance for robotics research and education, and potential directions for future work, such as extending the methodology to other platforms or continuous motion scenarios.

2. METHODS AND MATERIALS

2.1. Comparison of the current and new measurement systems

The work described in this article is a direct continuation of research conducted in this area in previous years (Montusiewicz & Skulimowski, 2019). The table below compares the original and current systems (Table 1).

Tab. 1. Original and enhanced system versions comparison

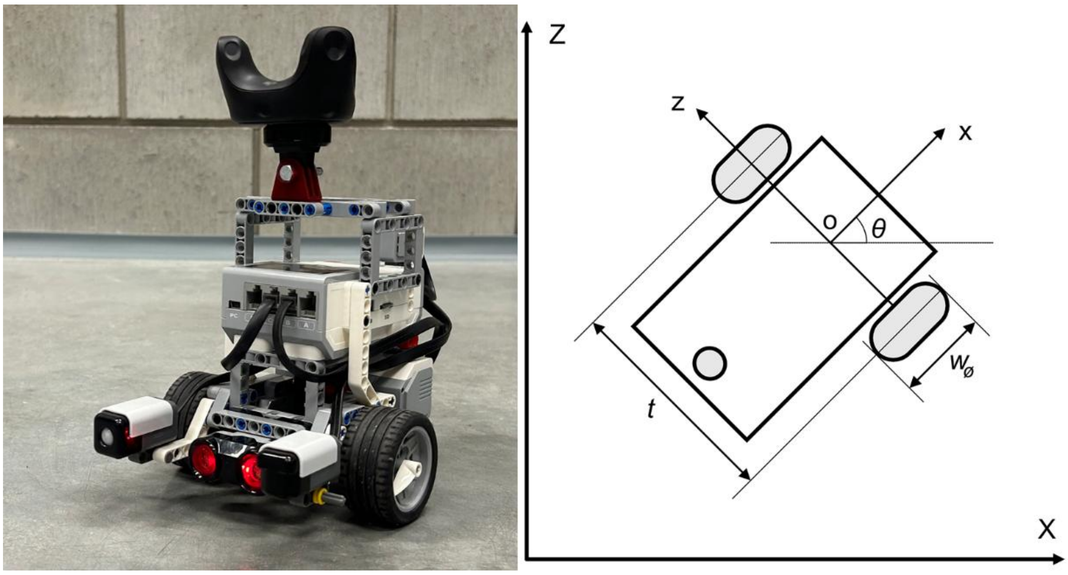
Factor	Original System (1.0)	Enhanced System (2.0)
Control system	LabView graphical programming	Pybricks OS based on microPython
Motion capture system	Video recording	Motion tracker with base station
Mechanism of conducting experiments	Manual – starting rides and recording from the camera	Automatic – sequential launch of runs according to the given queue with simultaneous launch of log writing threads
Route data acquisition mechanism	Manual – application of post-processing object tracking on video recordings and manual vectorization	Automatic – while the experiment is running
Measurement frequency	Up to 30 FPS (video)	Up to 120Hz for motion tracker
Measurement accuracy	Up to 1 cm	Up to 0.1 cm
Skill Barrier	Basic video editing	Python programming
Energy constraints	EV3 robot batteries	EV3 robot batteries, Motion Tracker batteries
Dependencies	LabView	Pybricks, openVR, SteamVR
Additional Cost (without robot)	$100 (camera only)	$500 (motion tracker, base station)

Moving from manual video analysis to consumer-grade VR tracking significantly improves measurement accuracy and reproducibility (Jackson et al., 2016). This approach democratizes access to high-precision robot performance analysis by significantly reducing the cost of traditional motion capture systems (Kuti et al., 2024). In addition, migration to Pybricks allows for more sophisticated parameter manipulation and data logging, which is critical for systematic drift analysis (Araujo et al., 2022).

While previous manual methods allowed for ad hoc changes, these automation improvements minimize human error, increase measurement consistency, and allow for larger scale experiments that would otherwise be impractical (Sigron et al., 2023). Although the new approach introduces dependencies on the VR ecosystem (SteamVR, OpenVR libraries) and potential future hardware compatibility issues (Niehorster et al., 2017). The basic methodology remains transferable to alternative external tracking systems.

2.2. Experimental setup and test scenarios design

A LEGO Mindstorms EV3 DWR robot was used for the tests (Fig. 1.). The robot operates with kinematic constraints that govern its motion. The robot's pose q is defined by its position (x,z) and orientation (θ) in the global coordinate frame.



a) **b)**

Fig. 1. DWR used in the experiment. a) Real DWR with a Vive motion tracker 3.0 attached on top, with 3D printed The Original LEGO Technic GoPro Mount by Pleky085 (LEGO Technic GoPro Mount by mrusk | Printables.com n.d.). b) Sketch of the DWR.

The robot has been programmed in MicroPython (LEGO, n.d.). Unlike graphical programming approaches, it provides the deterministic timing and precise parameter control necessary for systematic drift analysis. Based on the parameters that can be configured during initialization of the robot control program, the following kinematic model can be assumed:

$$q = \begin{bmatrix} \Delta x \\ \Delta \mathrm{z} \\ \Delta\theta \end{bmatrix} = \begin{bmatrix} d \times \cos\theta_0 \\ d \times \sin\theta_0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \theta_{turn} \end{bmatrix} \tag{1}$$

where:

- d is straight distance (mm)
- θ_{turn} is turning angle in radians = $\alpha \times \frac{\pi}{180}$
- α is angle of the turn (degrees)

In the experiments, two functions were used to change the robot's pose $q$corresponding to straight motion and zero-radius turns (around point o- Fig.1. right). In terms of kinematic update, for moving straight pose is updated by:

$$\begin{gathered} x_{new} = x_{old} + d \times \cos\theta_{old} \\ z_{new} = z_{old} + d \times \sin\theta_{old} \\ \theta_{new} = \theta_{old} \end{gathered} \tag{2}$$

For a zero radius turn, the pose will be updated by:

$$\begin{gathered} x_{new} = x_{old} \\ z_{new} = z_{old} \\ \theta_{new} = \theta_{old} + \alpha \times \frac{\pi}{180} \end{gathered} \tag{3}$$

For actual robot motion estimation, wheel rotation formulas can be specified:

$$N_{straight} = \frac{d}{\pi \times w_{ø}} \tag{5}$$

$$N_{turn} = \frac{(\pi \times t \times \alpha)}{360 \times (\pi \times w_{ø})} = \frac{t \times \alpha}{360 \times w_{ø}} \tag{6}$$

where:

- $N_{straight}$ is wheel rotations for straight
- N_{turn} is wheel rotations for turn in place (left wheel – forward by N_{turn}, right wheel – backward by N_{turn})
- $w_{ø}$ is wheel diameter (mm)
- t is axle track (mm)

The new version of the system requires the use of a motion tracker device as a separate measurement device. The authors used the Vive VR Motion Tracker 3.0 (VIVE Tracker (3.0) | VIVE European Union n.d.). The sensor requires an external infrared lantern for operation. This consumer-grade tracking technology provides sub-centimeter accuracy for ground truth position measurement, enabling detailed analysis of drift patterns without the need for expensive industrial measurement systems (Merker et al., 2023). The disadvantage of using a motion tracker is that it extends the chain of dependencies. To operate the sensor, it is necessary to use SteamVR (SteamVR and Steam n.d.)and the openVR library (ValveSoftware/openvr: OpenVR SDK n.d.)and an additional script to communicate with the tracker (snuvclab/Vive_Tracker: Using Vive Tracker to estimate 6DOF n.d.).

The set of test scenarios is a reflection of the Cartesian product of class categories describing the robot's motor settings and path characteristics. This allows the results to be compared in terms of class evaluation and clustering.

2.3. Motors parameters design methodology

The motor settings included four parameters: Straight speed ($\frac{mm}{s}$), Straight acceleration ($\frac{mm}{s^2}$), Turn rate ($\frac{deg}{s}$) and Turn acceleration ($\frac{deg}{s^2}$). The motor parameters were grouped into two composite categories – Motion Dynamics Profile and Rotational Dynamics Profile – with three classes each. The values on which these classes operate were prepared based on load tests, within which the real maximum speed and acceleration – $RV_{straight}^{max}$, $RA_{straight}^{max}$, RV_{turn}^{max} , RA_{turn}^{max} – and the real minimum speed – $RV_{straight}^{min}$, RV_{turn}^{min} . The Table 2. shows the specific limit values.

Tab. 2. List of the real maximum and minimum values of speed and acceleration

Extreme	Description	$RV_{straight}(\frac{mm}{s})$	$RA_{straight}(\frac{mm}{s^2})$	$RV_{turn}\ (\frac{deg}{s})$	$RA_{turn}\ (\frac{deg}{s^2})$
min	Values below which the robot does not start moving due to too much friction, or when the movement contains a lot of jitter	50	--	50	--
max	Values that, despite higher declarations in the program, cannot be exceeded	380	410	400	300

The Motion Dynamics Profile (MDP) combines linear velocity and acceleration (Table 3.). It is fundamental for odometry accuracy in differential drive systems and a core parameter in motion control systems throughout mobile robotics. The Rotational Dynamics Profile (RDP) combines rotational velocity and acceleration (Table 4.). It is critical for non-holonomic constraint management in DWRs and determines stability during direction changes in confined spaces. RDP is a key factor in simultaneous localization and mapping accuracy.

Tab. 3. Descriptions of MDP classes with supporting examples

MDP class	Description	Example
Low Dynamics	$V_{straight} = 30\% \times RV_{straight}^{max}$ $A_{straight} = 30\% \times RA_{straight}^{max}$	Delta-robot manipulators in cellular-level microsurgery often run at about 30% of their maximum speed to secure sub-micron positioning accuracy for optical biopsies (Huang et al., 2023).
Medium Dynamics	$V_{straight} = 60\% \times RV_{straight}^{max}$ $A_{straight} = 60\% \times RA_{straight}^{max}$	Collaborative lab robots operating at ~60% top speed achieve reliable 0.1 mm repeatability in pipetting and sample handling (Szczepanski et al., 2024).
High Dynamics	$V_{straight} = 90\% \times RV_{straight}^{max}$ $A_{straight} = 90\% \times RA_{straight}^{max}$	High-speed pick-and-place manipulators in electronics assembly operate at ≈ 90% of their maximum velocity, enabling over 5 000 picks/hour and sustaining > 99% placement success rates under production conditions (Khafagy et al., 2025).

Tab. 4. Descriptions of RDP classes with supporting examples

RDP class	Description	Example
Precision Rotation	$V_{turn} = 30\% \times RV_{turn}^{max}$ $A_{turn} = 30\% \times RA_{turn}^{max}$	In teleoperated surgery, slow rotational profiles (~30% speed) mirror surgeons' fine instrument orientation changes, following the speed-curvature power law in motor control (Zruya et al., 2022).
Balanced Rotation	$V_{turn} = 60\% \times RV_{turn}^{max}$ $A_{turn} = 60\% \times RA_{turn}^{max}$	Homotopy-based path planners for industrial arms use ~60% rotation speed to execute smooth obstacle avoidance within 50 ms response times (Velez-Lopez et al., 2022).
Aggressive Rotation	$V_{turn} = 90\% \times RV_{turn}^{max}$ $A_{turn} = 90\% \times RA_{turn}^{max}$	Delta-robot cells for electronic component pick-and-place employ aggressive rotation (≈ 90% max), reaching high angular accelerations for rapid part exchanges (Velez-Lopez et al., 2022).

The table below shows a summary of all the adopted configurations of settings for the robot's motors (Tab.5.). Each of the adopted settings was unchanged during a single pass.

Tab. 5. List of values associated with robot motors configurations

Drives configuration	Motion Dynamics Profile	Rotational Dynamics Profile	Straight Speed (mm/s)	Straight acceleration (mm/s/s)	Turn rate (deg/s)	Turn acceleration (deg/s/s)
D1	Low Dynamics	Precision Rotation	114	123	120	90
D2	Low Dynamics	Balanced Rotation	114	123	240	180
D3	Low Dynamics	Aggressive Rotation	114	123	360	270
D4	Medium Dynamics	Precision Rotation	228	246	120	90
D5	Medium Dynamics	Balanced Rotation	228	246	240	180
D6	Medium Dynamics	Aggressive Rotation	228	246	360	270
D7	High Dynamics	Precision Rotation	342	369	120	90
D8	High Dynamics	Balanced Rotation	342	369	240	180
D9	High Dynamics	Aggressive Rotation	342	369	360	270

2.4. Route design methodology

For the experiment, PTP routes were designed as sequences of straight-line movements (mm) and zero-radius turns (degrees), forming polygonal rather than curvilinear paths. The systematic path design methodology systematically varied geometric complexity and maneuver alternation (rate of change of motion commands). Each resulting class combination presented specific challenges to the robot's motion control system.

Route Geometric Complexity (RGC) classifies paths by curvature variation and direction changes. It is often used as a measure of the efficiency of motion planning algorithms (Table 6.). It models error propagation in dead reckoning systems. Route Maneuver Alternation (RMA) is a transition between maneuver types per distance (transition/meter). It measures temporal control demands through command sequencing (Table 7.).

Tab. 6. Descriptions of RGC classes with supporting examples

RGC class	Description	Example
Simple	Route has 4-6 segments	Two-wheel AGVs in automotive assembly execute simple routes with 4–6 straight segments for palletizing, maintaining < 2 cm path deviation over 100 m runs (Pantano et al., 2022).
Moderate	Route has 7-10 segments	Warehouse AMRs following medium-complexity loops of 7–10 waypoints achieve 30% collision-rate reduction in narrow aisles using RRT-based planners under real-world constraints (Luo et al., 2023).
Complex	Route has 11-14 segments	Robot-assisted endoscopic systems navigate 11–14-segment tool paths around anatomy, performing sub-millimeter-accuracy resections in soft-tissue surgery with integrated kinematic control (Zruya et al., 2022).

Tab. 7. Descriptions of RMA classes with supporting examples

RMA class	Description	Example
Low	Corresponds to low transition frequency, no more than 2 t/m	Beverage bottling-line gantry robots make < 2 turns/m to transfer bottles in straight lines, sustaining 120 bottles/min throughput with < 1% positioning error (Tao, 2023).
Medium	Corresponds to medium transition frequency, between 2 and 4 t/m	Automotive spot-welding robots execute ≈ 3 turns/m along curved panels, balancing speed and weld quality to reduce defect rates by 15% in high-volume production (Luo et al., 2023).
High	Corresponds to high transition frequency, higher than 4 t/m	Order-picking AMRs in e-commerce facilities perform > 4 turns/m to navigate dense rack aisles, cutting pick times by 20% while maintaining ± 5 cm localization accuracy (Luo et al., 2023).

Each route not only had to correspond to a combination of classes of the accepted categories, but also had to meet the necessary conditions:

- should be almost closed - the distance between the start and end points should not exceed 10 cm,
- must contain both left and right turns
- can be a self-intersecting chain of polygons or a complex, non-concave polygon
- had to be between 100 and 700 cm long,
- its bounding box area could not exceed 1 m^2.

Table 8. shows the values for each configuration. Figure 2. shows graphical representations of the routes.

Tab. 8. List of values associated with routes characteristics configurations

Route	RGC Class	RMA Class	Segments	Transitions per meter	Route Length (cm)	Bounding Box Area (m²)
T1	Simple	Low	6	1.961	306	0.25
T2	Simple	Medium	6	2.808	214	0.25
T3	Simple	High	6	4.638	129	0.11
T4	Moderate	Low	10	1.951	513	0.56
T5	Moderate	Medium	10	3.667	273	0.25
T6	Moderate	High	10	5.007	200	0.2
T7	Complex	Low	14	1.983	706	0.55
T8	Complex	Medium	14	3.956	354	0.25
T9	Complex	High	14	6.282	223	0.2

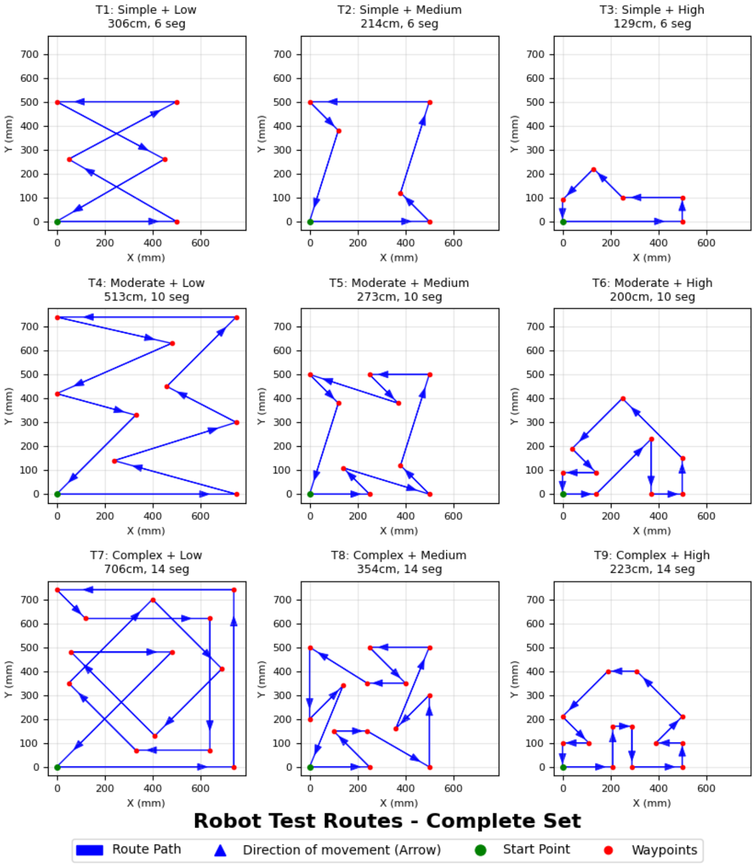


Fig. 2. Routes visualization

2.5. Data collection protocol

An automated client-server system was developed for experimental control via Bluetooth (Fig. 3.). The PC client managed motion tracker recording and sent scenario initiation commands to the EV3 robot server, which responded with readiness and completion signals for data acquisition. All control, motor, and path data was preloaded into the EV3's RAM, effectively reducing the turnaround time between successive runs. The system also allowed relative start positions to be determined after acquisition, eliminating the need to manually reposition the robot after each completed run, a requirement in previous research.

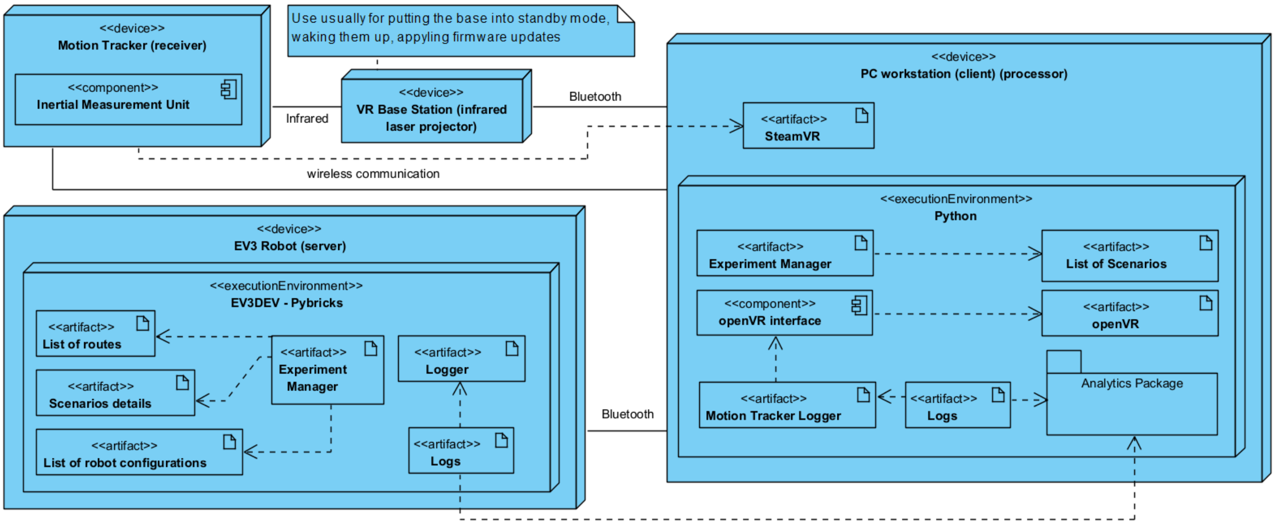


Fig. 3. Main software parts of the Deployment diagram

2.6. Data analysis methods

Once data acquisition is complete, data processing can begin, including cleaning, averaging (merging), synthesizing, and comparing (Fig.4.).

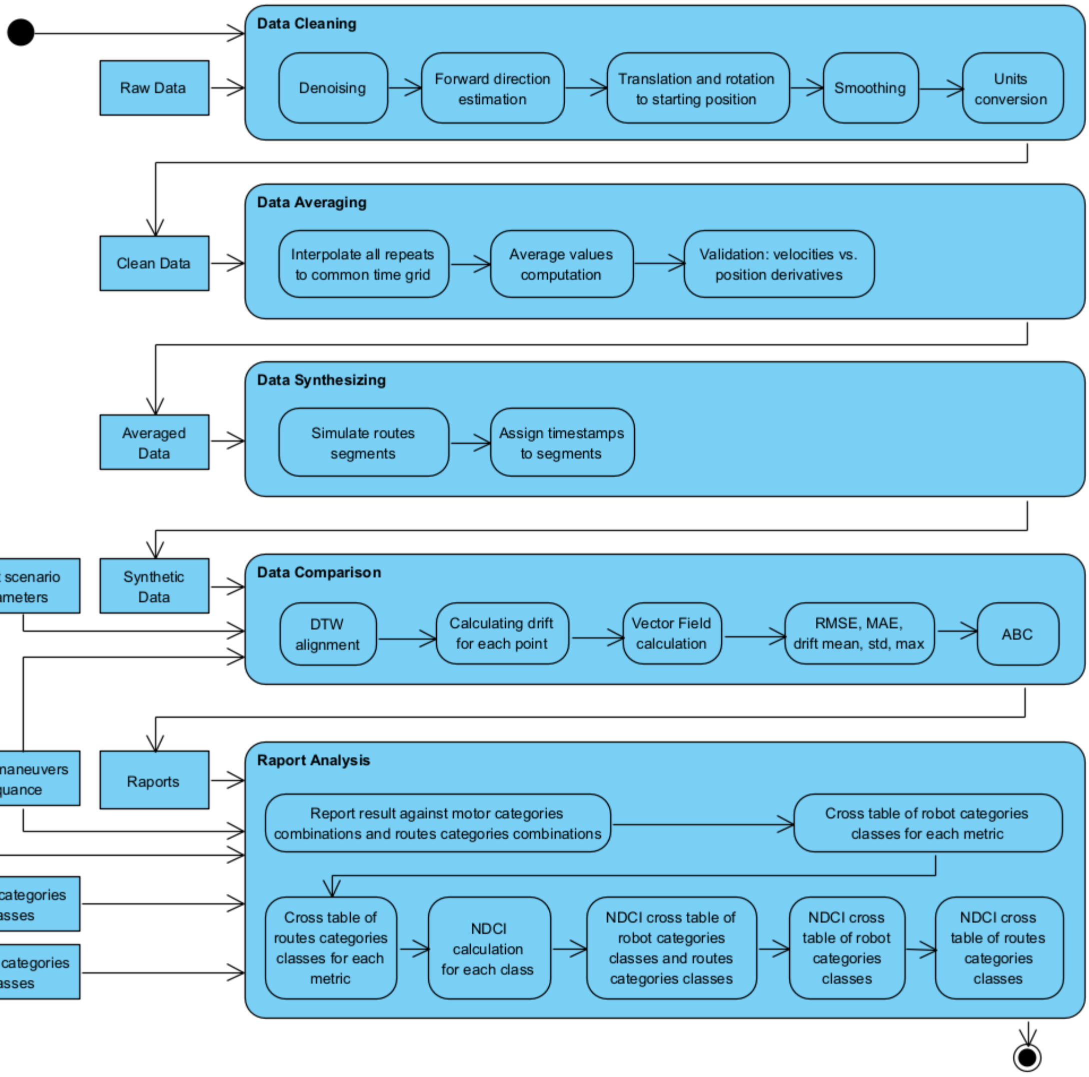

Data Cleaning
Raw Data
Denoising
Forward direction estimation
Translation and rotation to starting position
Smoothing
Units conversion
Data Averaging
Clean Data
Interpolate all repeats to common time grid
Average values computation
Validation: velocities vs. position derivatives
Data Synthesizing
Averaged Data
Simulate routes segments
Assign timestamps to segments
Data Comparison
Robot scenario parameters
Synthetic Data
DTW alignment
Calculating drift for each point
Vector Field calculation
RMSE, MAE, drift mean, std, max
ABC
Raport Analysis
Robot maneuvers sequance
Raports
Report result against motor categories combinations and routes categories combinations
Cross table of robot categories classes for each metric
Robot categories classes
Routes categories classes
Cross table of routes categories classes for each metric
NDCI calculation for each class
NDCI cross table of robot categories classes and routes categories classes
NDCI cross table of robot categories classes
NDCI cross table of routes categories classes

Fig. 4. Data processing Activity diagram

During the data cleaning phase, the data in the files is pre-processed taking into account the relative starting point, relative rotation and real pose. Then, the outliers were cut off and low-pass smoothed using the Savitzky-Golay filter.

Analysis of the data obtained requires prior averaging statistics for each scenario. The data averaging phase serves to eliminate noise and measurement errors. All repetitions of a given scenario were interpolated to a common time grid (Figure 5. Left).

Then, during the Data Synthesizing phase, based on the scheme of ideal, planned route and timestamps of averaged real trips, a synthetic run is prepared, representing the theoretical ideal positions of the vehicle at the same time moments as the averaged real trip.

The next step, the data comparison phase, is to compare the cleaned data with the synthetic run. As a result of this comparison, the following are created

- Error Vector Field (EVF) - Visualizes the spatial distribution and direction of errors between real and synthetic trajectories at any point. This provides a comprehensive spatial analysis of drift patterns, revealing how geometric complexity and curve characteristics affect positioning accuracy (Figure 5. Center).

- Area Between Curves (ABC) - Quantifies the total non-overlapping area between the two trajectories, effectively representing the area of the EVF and indicating overall spatial mismatch (Figure 5. Right).
- Error Vector Magnitudes (EVM) - Quantifies the positional deviation at each point along the trajectory. It is used to calculate the mean, standard deviation, maximum and 95th percentile along with the Root Mean Square Error (RMSE).

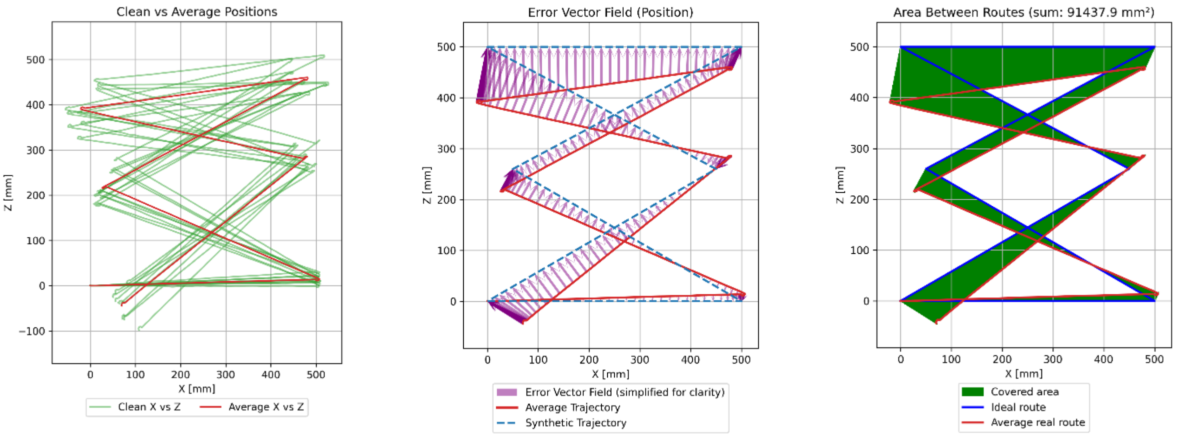


Fig. 5. Example of the analyses performed: (left) determination of the average route based on actual journeys, (middle) EVF being the result of comparing the average route with the synthetic one, (right) ABC of EVF boundaries.

During the Report Analysis phase, the results of the perturbation analysis of all scenarios are compared to each other, along with information about which classes the robot and path represented. The Normalized Drift Contribution Index (NDCI) is used to show how much each robot class or combination of robot and path classes contributes to the overall drift. The NDCI is calculated using the formula below:

$$NDCI_k = \frac{\left(w_{drift} \times \frac{\overline{\mathrm{EVM}_k}}{\overline{\mathrm{EVM}}}\right) + \left(w_{RMSE} \times \frac{\overline{RMSE_k}}{\overline{RMSE}}\right) + (w_{ABC} \times invABC_k)}{w_{drift} + w_{RMSE} + w_{ABC}} \tag{7}$$

$$invABC_k = 1 - \frac{\overline{ABC_k}}{\max{(ABC)}} \tag{8}$$

where:

- $\overline{\mathrm{EVM}_k}$ is mean EVM for all experiments that match the class combination (k)
- $\overline{\mathrm{EVM}}$ is the mean EVM value calculated across all experiments
- w_{drift} is weight of the variable
- $\max{(ABC)}$ is maximal value of ABC across all experiments

The following values were assigned to the weights: $w_{\mathrm{EVM}} = 0.4$, $w_{RMSE} = 0.3$, $w_{ABC} = 0.3$.

Based on the calculated NDCI for each class, agglomerative clustering is performed to determine whether each class independently has the greatest tendency to produce a high, medium, or low NDCI. The tendency is based on which cluster is most common for each class value.

3. EXPERIMENT

Experiments were conducted using a single function call queue that included all parameter sequences derived from the Cartesian product of drive configurations and routes. Details of the test bed preparation and start-up procedures are given in Appendices 1 and 2, respectively.

The robot operated on a pre-cleaned, flat surface to prevent slippage (Figure 6). Each setup combination was repeated five times to average out transient errors from motor slippage, surface irregularities, and sensor noise to ensure stable drift measurements. The number of repetitions was limited to reduce stress on the motors. All scenarios ran automatically, mostly without operator intervention. In addition, the relative starting point mechanism eliminated the need for manual robot repositioning, significantly accelerating data collection compared to the original system. For example, 810 measurements were completed in 10 hours.

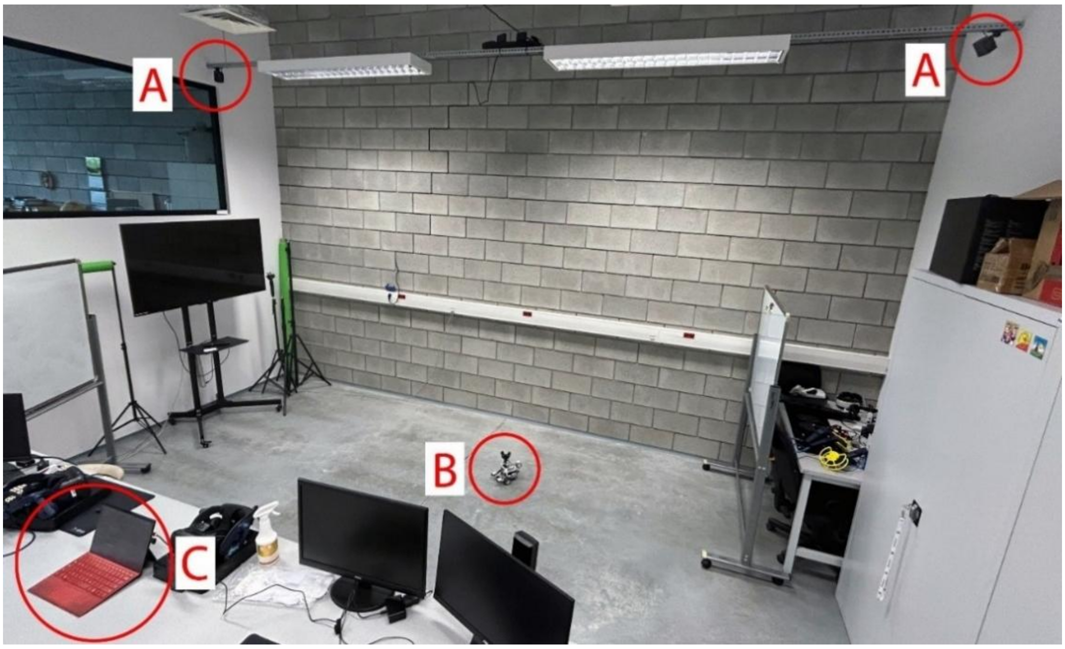


Fig. 6. Research setup. A- base station, B – robot (server), C – PC work station (client)

Although no single route had a bounding box larger than 1 m^2The accumulation of errors and the constant shifting of the starting point meant that the robot was operating in an area of approximately 9m^2. In addition, there were times when the base stations lost sight of the motion tracker, causing it to stop recording its position until the next run. The two main reasons were too great a distance from the base station (over 4 meters) or an obstacle between the base station and the sensor (such as a table top). As a result, some of the logs were incorrect, which had to be found and sorted out before analysis. As a result, out of 810 measurements performed, 773 (about 95%) were finally saved, with the number of repetitions of each scenario still not less than 5. Other experimental issues and observations are detailed in Appendix 3.

4. RESULTS

Based on the general results (Table 9.), it can be stated that higher number of rotations (High RMA) and moderate rotation speed ("Balanced Rotation" RDP) are associated with lower EVM (drift) mean, while lower number of rotations (Low RMA) and low rotation speed ("Precision Rotation" RDP) result in higher drift and error. In addition, it is not possible at this stage to determine the role of MDP, since "Medium Dynamics" is present in both the best and worst results.

Tab. 9. General results, best-worst combinations

Metric	Type	Combination	Value
EVM mean	Best (smallest)	('Medium Dynamics', 'Balanced Rotation', 'Moderate', 'High')	5.529
	Worst	('Medium Dynamics', 'Aggressive Rotation', 'Moderate', 'Low')	126.166
EVM std	Best (smallest)	('Medium Dynamics', 'Balanced Rotation', 'Moderate', 'High')	4.317
	Worst	('Medium Dynamics', 'Precision Rotation', 'Moderate', 'Low')	100.879
EVM max	Best (smallest)	('Medium Dynamics', 'Balanced Rotation', 'Moderate', 'High')	17.852
	Worst	('Medium Dynamics', 'Precision Rotation', 'Moderate', 'Low')	344.514
EVM 95th percentile	Best (smallest)	('Medium Dynamics', 'Precision Rotation', 'Simple', 'Medium')	13.904
	Worst	('Medium Dynamics', 'Precision Rotation', 'Moderate', 'Low')	325.992
RMSE	Best (smallest)	('Medium Dynamics', 'Balanced Rotation', 'Moderate', 'High')	7.0148
	Worst	('Medium Dynamics', 'Precision Rotation', 'Moderate', 'Low')	159.557

The NDCI matrix of robot configurations and path characteristics (Fig.7.) shows that the worst tracking (highest NDCI) occurs for "Moderate" or "Complex" paths with "Low" RMA and robots with "High Dynamics" and "Aggressive Rotation".

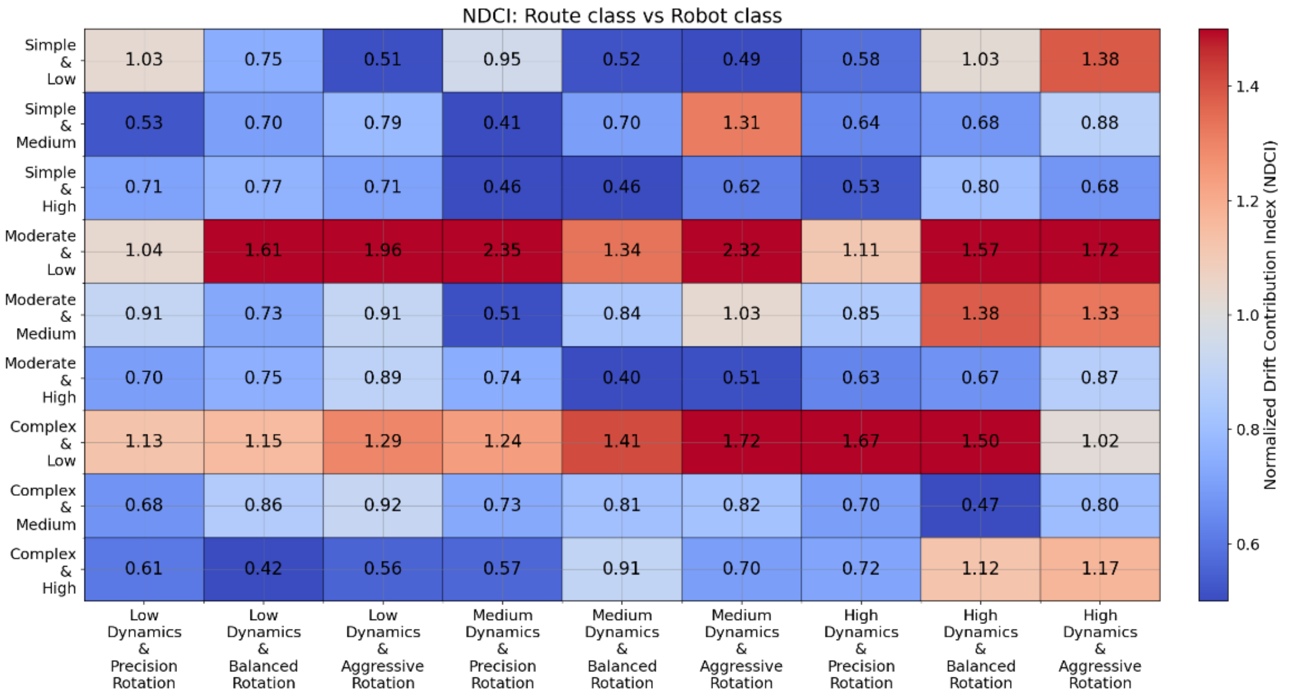


Fig. 7. The NDCI matrix for robot configurations and route characteristics

The NDCI Route Matrix (Fig. 8. left side) confirms that "Complex" and "Moderate" routes, especially with "Low" Turn Size (RMA), have the highest NDCI, indicating greater cumulative deviation and error. The same matrix shows that "Low" RMA consistently produces the highest NDCI across all route complexities.

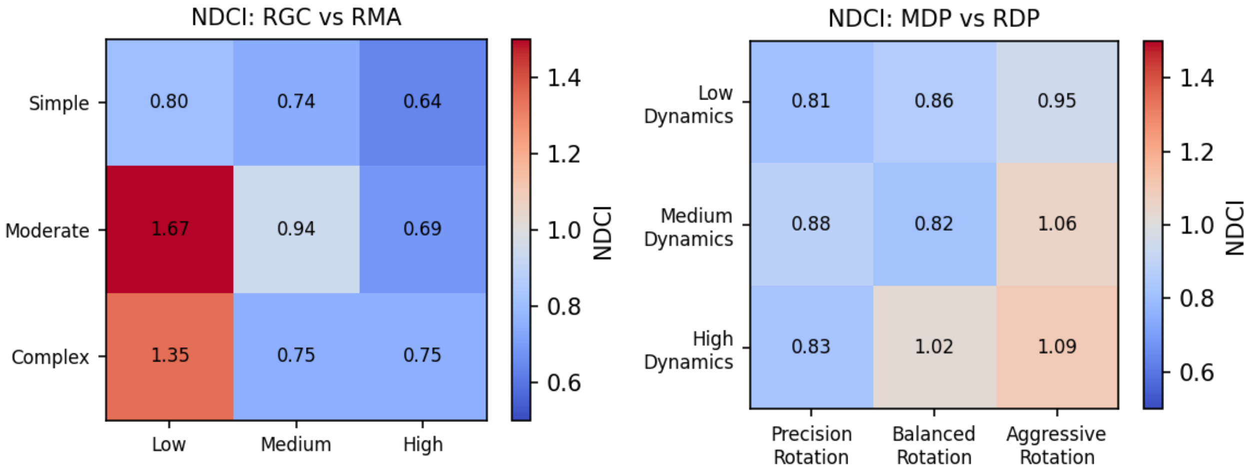


Fig. 8. NDCI matrixes: left – NDCIroute, right NDCIrobot

The NDCI robot matrix (Fig.8. right side) shows that the "High Dynamics" and "Aggressive Rotation" combinations have the highest NDCI, indicating more drift and error.

The clustering of the NDCI results (Fig.9.) into three clusters shows that the cluster with the largest mean NDCI contains the values of the widest range relative to the spectrum of values (cluster heights) of the other two clusters.

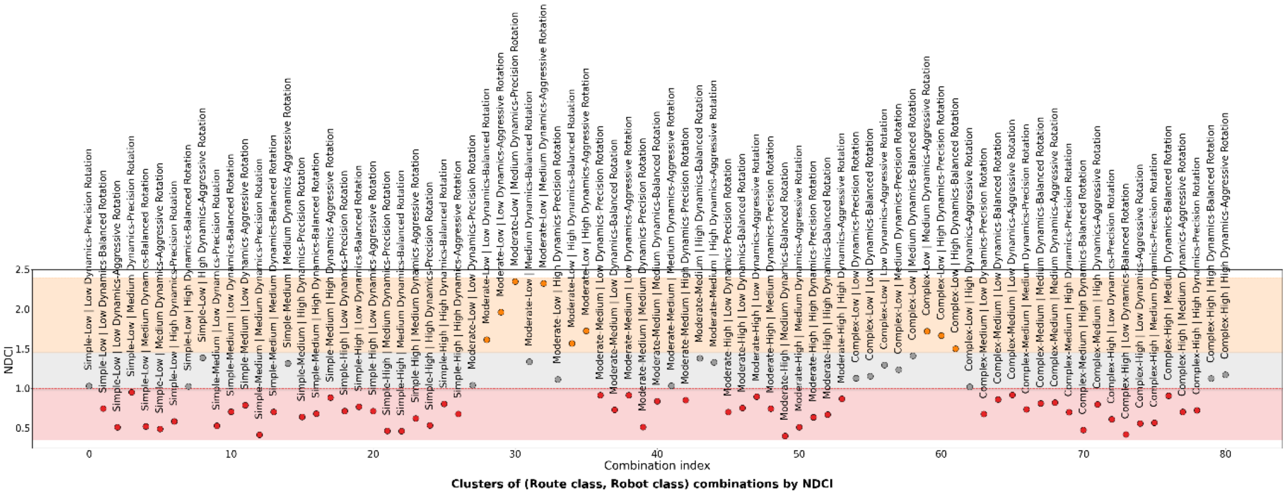


Fig. 9. Clusters of route classes and robot classes by NDCI value. Background colours refer to cluster borders

From clustering results (Tab. 10.) it can be found that "Low" RMA and "Moderate" RGC classes have the highest average NDCI, while "High" RMA and "Simple" RGC have the lowest.

Tab. 10. Class frequency matrix in clusters

Class type	Class value	Average NDCI	Most common cluster	Appearances in cluster 0	Appearances in cluster 1	Appearances in cluster 2
RMA	Low	1.273	2	6	9	12
RGC	Moderate	1.099	0	15	6	6
RDP	Aggressive Rotation	1.034	0	16	4	7
MDP	High Dynamics	0.982	0	15	4	8
RGC	Complex	0.951	0	16	3	8
MDP	Medium Dynamics	0.921	0	19	3	5
RDP	Balanced Rotation	0.901	0	18	3	6
MDP	Low Dynamics	0.874	0	20	2	5
RDP	Precision Rotation	0.842	0	20	2	5
RMA	Medium	0.812	0	23	0	4
RGC	Simple	0.727	0	23	0	4
RMA	High	0.692	0	25	0	2

5. DISCUSSION

Based on the results obtained, it can be stated that:

- More complex paths are harder to follow accurately, resulting in more drift and error accumulation.
- Small, frequent turns are more difficult for the robot to follow, resulting in greater drift accumulation.
- More dynamic and aggressive motor settings increase drift, especially when combined. The factor that had the greatest effect on drift was rotational acceleration.

5.1. Comparison with previous research

Although manual methods remain useful for small-scale educational explorations, the authors' automated VR-based approach offers superior precision, scalability, and labor efficiency for rigorous research, while manual tracking still aids in hypothesis development and low-cost training scenarios. The evolution from the original to the current methodology represents a major leap in experimental design, measurement accuracy, and scientific rigor. By using external motion tracking, we eliminate self-referential bias, greatly expand the experimental scope, and improve statistical reliability. This framework not only enhances educational robotics research, but also provides practical, cost-effective tools for industrial navigation challenges.

5.2. Practical implications

Combining route geometric complexity with turn size categories provides a standardized, low-cost calibration framework for a wide range of differential drive platforms-from industrial AGVs and AMRs to educational DWRs and LEGO EV3 testbeds. By abstracting the core behaviors of these systems, this protocol provides an affordable alternative to automotive-grade testing, enabling centimeter-level drift characterization without custom facilities. The comprehensive experimental design - with controlled variation of polygon complexity, turn frequency, and discrete point-to-point motion - serves a dual purpose. In industrial settings, AGV and AMR engineers can isolate drift contributions to optimize path planners, sensor fusion algorithms, and maintenance schedules to reduce misplacement and collision rates in warehouse and factory environments.

As an educational template, this reproducible methodology enables instructors and students on the LEGO EV3 and DWR platforms to conduct rigorous drift studies using consumer-grade external tracking (e.g., VR systems). Hands-on calibration exercises promote understanding of non-holonomic constraints and error compensation strategies, while insights are directly applicable to industrial AGVs/AMRs. Together, these features bridge academic and applied robotics, driving the development of improved algorithms and robust hardware in research, education, and real-world automation.

5.3. Limitations and future work

Although automation of data acquisition greatly streamlined the experiment, occasional manual intervention was required (e.g., sensor disconnections, collisions, communication failures; see Appendix 3). These problems are due to predictable factors-cable strain, uneven floors, tire wear, debris accumulation, and two-dimensional operating constraints-and can be mitigated by improved test-room preparation (e.g., cable management, levelled surfaces) and hardware refinements (e.g., tire guards, reinforced mounts).

Improving accuracy and expanding applicability will require:

- Integrate on-board logs. Fusing Vive Tracker data with EV3 motor encoder and IMU logs provides finer-grained drift profiles and enables multi-sensor drift estimation.
- Expand dynamic conditions. The introduction of variable-speed trajectories or acceleration-dependent maneuvers ("dynamic demand" category) quantifies system latency and control responsiveness critical to real-world AGVs and AMRs.
- Increase environmental realism. Design courses with repeated maneuver sequences, varying surface textures, and slopes to capture drift behavior under heterogeneous conditions similar to factory floors and field sites.
- Incorporate energy metrics. Adding power consumption to the performance evaluation will link drift compensation strategies with energy efficiency, informing the design of battery-powered mobile robots.
- Improve data synchronization. Embedding maneuver type and timestamp metadata in log files supports event-based fusion of external and onboard measurements.
- Communication latency assessment. A comparison of Bluetooth and Wi-Fi command exchanges will characterize control delays and guide network architecture choices for automated navigation systems.

These enhancements will bridge the gap between controlled laboratory studies and complex industrial, biomedical, and logistic environments, fostering robust, scalable drift characterization protocols for next-generation differential drive platforms.

The extension to continuous motion with circular arcs would investigate how sustained differential wheel velocities during constant curvature paths affect drift accumulation compared to discrete motion. The kinematic model predicts different error propagation characteristics for continuous turns where the angular velocity is constant rather than stepwise, potentially reducing integration errors while introducing centripetal force effects.

6. CONCLUSIONS

RO1 goal was fully realized. A reproducible framework for drift quantification was developed, achieving a mean positional accuracy of 0.55 cm over 773 trials.

The RO2 goal was fully realized. A strong correlation between the number of segments and the amount of drift was confirmed, with complex routes (12-15 segments) having 22% more drift than simple routes (4-7 segments).

The RO3 goal was fully achieved. The research demonstrated an 83% time reduction over manual video analysis (45 vs. 270 sec/trial processing, old vs. current methodology). This approach provides a scalable, cost-effective alternative to traditional motion capture for both educational and industrial robotics validation.

The automated framework presented supports continuous improvement cycles in lean manufacturing and medical device R&D, reducing downtime and human error by partially automating the data acquisition and analysis process.

NDCI provides a holistic, normalized measure of tracking performance that integrates drift, RMSE, and ABC. It allows quick identification of problematic robot/trajectory combinations and guides optimization of both DWR parameters and trajectory design to minimize drift.

Overall, the results provide educators with practical benchmarks for robotics curricula and the broader robotics community with a standardized protocol for validating autonomous navigation systems. The results inform trajectory planning, calibration, and error compensation strategies, supporting more robust and accessible robotics experimentation across contexts.

Funding

The work was co-financed by the Department of Computer Science Lublin University of Technology and Lublin University of Technology Scientific Fund FD-20/IT-3/015.

Acknowledgement

Part of the software was created using AI tools – Perplexity, Microsoft Copilot.
The initial translation of the article was done using AI tool – Google Translate.
The linguistic proofreading was done partly using AI – Language Tool.

Conflicts of interest

The authors declare no conflict of interest.

REFERENCES

Agrawal, D. R., Kim, T., Govindjee, R., Adeshara, T., Yu, J., Ravikumar, A., & Panagou, D. (2025). Certifiably-correct mapping for safe navigation despite odometry drift. *ArXiv, abs/2504.18713*. https://doi.org/10.48550/arXiv.2504.18713

Alici, G., & Shirinzadeh, B. (2005). A systematic technique to estimate positioning errors for robot accuracy improvement using laser interferometry based sensing. *Mechanism and Machine Theory*, *40*(8), 879–906. https://doi.org/10.1016/J.MECHMACHTHEORY.2004.12.012

Araujo, H. L., Gulfo Agudelo, J., Crawford Vidal, R., Ardila Uribe, J., Freddy Remolina, J., Serpa-Imbett, C., López, A. M., & Patiño Guevara, D. (2022). Autonomous mobile robot implemented in LEGO EV3 integrated with Raspberry Pi to use Android-based vision control algorithms for human-machine interaction. *Machines*, *10*(3), Article 193. https://doi.org/10.3390/MACHINES10030193

Bilal, D. K., & Unel, M. (2021). Increasing trajectory tracking accuracy of industrial robots using SINDYc. *IFAC-PapersOnLine*, *54*(4), 13–18. https://doi.org/10.1016/J.IFACOL.2021.10.003

Borges, M., Symington, A., Coltin, B., Smith, T., & Ventura, R. (2018). HTC Vive: Analysis and accuracy improvement. *IEEE International Conference on Intelligent Robots and Systems* (pp. 2610–2615). IEEE. https://doi.org/10.1109/IROS.2018.8593707

Buchanan, E., Alden, K., Pomfret, A., Timmis, J., & Tyrrell, A. M. (2023). A study of error diversity in robotic swarms for task partitioning in foraging tasks. *Frontiers in Robotics and AI*, *9*, Article 904341. https://doi.org/10.3389/FROBT.2022.904341

De Giorgi, C., De Palma, D., & Parlangeli, G. (2023). Online odometry calibration for differential drive mobile robots in low traction conditions with slippage. *Robotics*, *13*(1), Article 7. https://doi.org/10.3390/ROBOTICS13010007

Dmytriv, V., Dmytriv, I., Horodetskyy, I., Hutsol, T., Kukharets, S., Cesna, J., Bleizgys, R., Pietruszynska, M., Parafiniuk, S., Kubon, M., & Horetska, I. (2024). A method for simulating the positioning errors of a robot gripper. *Applied Sciences*, *14*(14), Article 6159. https://doi.org/10.3390/APP14146159

Dong, C., Povorozniuk, O., Topalov, A., Wang, K., & Chen, Z. (2023). Development of the control system for LEGO Mindstorms EV3 mobile robot based on MATLAB/Simulink elements. *Technology Audit and Production Reserves*, *1*(2), 30–35. https://doi.org/10.15587/2706-5448.2023.274846

Đurašević, S., & Milovanović, A. (2017). Correction of systematic errors in odometry model for position determination of mobile tracked robot. *60th International Scientific Conference on Information, Communication and Energy Systems and Technologies Proceedings*.

Đurašević, S., & Milovanović, A. (2021). Design for the mobile robots' position with the application of odometry. *International Journal of Web Applications*, *13*, 11–18. https://doi.org/10.6025/ijwa/2021/13/1/11-18

Ferreira, M., Moreira, L., & Lopes, A. (2023). Differential drive kinematics and odometry for a mobile robot using TwinCAT. *Electronic Research Archive*, *31*(4), 1789–1803. https://doi.org/10.3934/ERA.2023092

Gavrilas, L., & Kotsis, K. T. (2024). Development and validation of a survey instrument towards attitude, knowledge, and application of educational robotics (AKAER). *International Journal of Research and Method in Education*, *48*(1), 44-66. https://doi.org/10.1080/1743727X.2024.2358780

Hsiao, T., Sheu, S. J., & He, R. (2022). A multi-precision indoor localization strategy based on hybrid Vive and adaptive Monte Carlo method. *2022 International Automatic Control Conference (CACS 2022)* (pp. 1-6). IEEE. https://doi.org/10.1109/CACS55319.2022.9969805

Huang, X., Rendon-Morales, E., & Aviles-Espinosa, R. (2023). Towards cellular level microsurgery: Design and testing of a high precision delta robot for medical applications. *Proceedings of The 15th Hamlyn Symposium on Medical Robotics 2023*. UK-Robotics and Autonomous Systems (RAS) Network.

Jackson, B. E., Evangelista, D. J., Ray, D. D., & Hedrick, T. L. (2016). 3D for the people: Multi-camera motion capture in the field with consumer-grade cameras and open source software. *Biology Open*, *5*(9), 1334–1342. https://doi.org/10.1242/BIO.018713

Jialong, G., Sun, L., Xiang, X., & Li, Z. (2020). A real-time method on robot with continuous trajectory for low-drift odometry and mapping. *Journal of Physics: Conference Series*, *1576*(1), Article 012058. https://doi.org/10.1088/1742-6596/1576/1/012058

Jin, J., & Chung, W. (2019). Obstacle avoidance of two-wheel differential robots considering the uncertainty of robot motion on the basis of encoder odometry information. *Sensors*, *19*(2), Article 289. https://doi.org/10.3390/S19020289

Jung, B., & Sukhatme, G. S. (2010). Real-time motion tracking from a mobile robot. *International Journal of Social Robotics*, *2*(1), 63–78. https://doi.org/10.1007/S12369-009-0038-Y

Kenny, A., Christiand, & Kurniawan, R. (2025). Robot self-balancing berbasis LEGO Mindstorm EV3 untuk pembelajaran robotika. *Cylinder: Jurnal Ilmiah Teknik Mesin*, *11*(1). https://doi.org/10.25170/CYLINDER.V11I1.6675

Khafagy, R. R., Eltaib, M. E. H., & Abo-Shanab, R. F. (2025). Automated complex path extraction and optimization based on scalable vector graphics and genetic algorithm for industrial robot applications. *International Journal of Advanced Manufacturing Technology*, *139*(1), 1033–1052. https://doi.org/10.1007/S00170-025-15846-8

Kin, W. S., Ghani, N. M. A., Masrom, M. F., Jamin, N. F., & Razali, N. A. A. (2019). Control of a two-wheeled Lego EV3 robot using interval type-2 fuzzy logic with particle swarm optimization. *22nd International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines* (pp. 175-182). CLAWAR. https://doi.org/10.13180/CLAWAR.2019.26-28.08.35

Kuti, J., Piricz, T., & Galambos, P. (2024). A robust method for validating orientation sensors using a robot arm as a high-precision reference. *Sensors*, *24*(24), Article 8179. https://doi.org/10.3390/S24248179

LEGO Technic GoPro mount by mrusk. (n.d.). *Printables.com*. https://www.printables.com/model/334045-lego-technic-gopro-mount

LEGO. (n.d.). *EV3 devices – EV3 devices — ev3-micropython 2.0.0 documentation*. https://pybricks.com/ev3-micropython/ev3devices.html

Luo, S., Zhang, M., Zhuang, Y., Ma, C., & Li, Q. (2023). A survey of path planning of industrial robots based on rapidly exploring random trees. *Frontiers in Neurorobotics*, *17*, Article 1268447. https://doi.org/10.3389/FNBOT.2023.1268447

Maga Vasile, A. (2024). Development of a mobile platform with differential drive for industrial applications. *Acta Universitatis Cibiniensis. Technical Series*, *76*(1), 82–87. https://doi.org/10.2478/AUCTS-2024-0012

Maga Vasile, A., & Crenganis, M. (2024). Optimized dimensioning and encoder configuration for AGV traction systems. *Acta Universitatis Cibiniensis. Technical Series*, *76*(1), 75–81. https://doi.org/10.2478/AUCTS-2024-0011

Merker, S., Pastel, S., Bürger, D., Schwadtke, A., & Witte, K. (2023). Measurement accuracy of the HTC VIVE Tracker 3.0 compared to Vicon System for generating valid positional feedback in virtual reality. *Sensors*, *23*(17), Article 7371. https://doi.org/10.3390/S23177371

Mitchell, A. C. (2015). *Modeling and control of a motor system using the Lego EV3 robot*. [Doctoral dissertation]. https://doi.org/10.12794/METADC804943

Montés, N., Rosillo, N., Mora, M. C., & Hilario, L. (2018). Real-time Matlab-Simulink-Lego EV3 framework for teaching robotics subjects. *Advances in Intelligent Systems and Computing*, *829*, 230-240. Springer. https://doi.org/10.1007/978-3-319-97934-2_15

Montés, N., Rosillo, N., Mora, M. C., & Hilario, L. (2021). A novel real-time MATLAB/Simulink/LEGO EV3 platform for academic use in robotics and computer science. *Sensors*, *21*(3), Article 1006. https://doi.org/10.3390/S21031006

Montusiewicz, J., & Skulimowski, S. (2019). Multicriteria evaluation of the trajectory of a wheeled mobile robot – a case study. *MATEC Web of Conferences*, *252*, Article 09007. https://doi.org/10.1051/MATECCONF/201925209007

Nie, J., Zhang, X., Sheng, C., Wang, H., & Lu, X. (2025). Improved sliding mode path tracking control for automated guided vehicles with the disturbances compensation. *International Journal of Advanced Robotic Systems*, *22*(2). https://doi.org/10.1177/17298806251326424

Niehorster, D. C., Li, L., & Lappe, M. (2017). The accuracy and precision of position and orientation tracking in the HTC Vive virtual reality system for scientific research. *i-Perception*, *8*(3), 1–23. https://doi.org/10.1177/2041669517708205

Ortega-Contreras, J. A., Pale-Ramon, E. G., Vazquez-Olguin, M. A., Andrade-Lucio, J. A., Ibarra-Manzano, O., & Shmaliy, Y. S. (2023). Tracking a holonomic mobile robot with systematic odometry errors. *2023 International Conference on Control, Artificial Intelligence, Robotics and Optimization (ICCAIRO 2023)* (pp. 99–104). IEEE. https://doi.org/10.1109/ICCAIRO58903.2023.00023

Palacín, J., Rubies, E., & Clotet, E. (2022). Systematic odometry error evaluation and correction in a human-sized three-wheeled omnidirectional mobile robot using flower-shaped calibration trajectories. *Applied Sciences*, *12*(5), Article 2606. https://doi.org/10.3390/APP12052606

Pantano, M., Eiband, T., & Lee, D. (2022). Capability-based frameworks for industrial robot skills: A survey. *IEEE International Conference on Automation Science and Engineering* (pp. 2355–2362). IEEE. https://doi.org/10.1109/CASE49997.2022.9926648

Poma, A. L. H., Mendoza Puris, A. C., & Del Aguila Ramos, J. A. (2024). Proposal for the design of an educational robot to enhance learning and support children with attention deficit hyperactivity disorder from the fifth year of age at low cost-Nubox. *2024 4th International Conference on Robotics, Automation, and Artificial Intelligence (RAAI 2024)* (pp. 11–16). IEEE. https://doi.org/10.1109/RAAI64504.2024.10949547

Ribeiro, C. E., Palácios, R. H. C., & Todt, E. (2024). Evaluation of the impact of educational robotics on school engagement: Educators' perspectives. *2024 Brazilian Symposium on Robotics and 2024 Workshop on Robotics in Education (SBR-WRE 2024)* (pp. 255–260). IEEE. https://doi.org/10.1109/SBR/WRE63066.2024.10838106

Samara, E., Angelidis, P., Kotsiari, A., Kilintzis, P., & Giorgos, A. (2021). Robotics in primary and secondary education - The Lego Mindstorms EV3 implementation. *6th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM 2021)* (pp. 1-8). IEEE. https://doi.org/10.1109/SEEDA-CECNSM53056.2021.9566240

Sapounidis, T., Tselegkaridis, S., & Stamovlasis, D. (2024). Educational robotics and STEM in primary education: A review and a meta-analysis. *Journal of Research on Technology in Education*, *56*(4), 462–476. https://doi.org/10.1080/15391523.2022.2160394

Šelek, A., Seder, M., & Petrović, I. (2023). Smooth autonomous patrolling for a differential-drive mobile robot in dynamic environments. *Sensors*, *23*(17), Article 7421. https://doi.org/10.3390/S23177421

Shi, Z. (2020). *Identification and control of a LEGO EV3 self-balancing robot: An educational tool in automatic control classes* [Master's thesis]. Politecnico di Milano.

Sigron, P., Aschwanden, I., & Bambach, M. (2023). Compensation of geometric, backlash, and thermal drift errors using a universal industrial robot model. *IEEE Transactions on Automation Science and Engineering*, *21*(4), 6615-6627. https://doi.org/10.1109/TASE.2023.3328835

Simon, J., Trojanová, M., Hošovský, A., & Sárosi, J. (2021). Neural network driven automated guided vehicle platform development for Industry 4.0 environment. *Tehnicki Vjesnik - Technical Gazette*, *28*, 1936–1942. https://doi.org/10.17559/TV-20200727095821

snuvclab/Vive_Tracker: Using Vive Tracker for estimating 6DOF. (n.d.). GitHub. https://github.com/snuvclab/Vive_Tracker

SteamVR. (n.d.). Steam. https://store.steampowered.com/app/250820/SteamVR/

Szczepanski, R., Erwinski, K., Tejer, M., & Daab, D. (2024). Optimal path planning algorithm with built-in velocity profiling for collaborative robot. *Sensors*, *24*(16), Article 5332. https://doi.org/10.3390/S24165332

Tao, L. (2023). Application market of industrial robot in China. *Applied Science and Innovative Research*, *7*(1), 77. https://doi.org/10.22158/ASIR.V7N1P77

Thai, N. H., Ly, T. T. K., Hoang, T., & Dzung, L. Q. (2022). Trajectory tracking control for differential-drive mobile robot by a variable parameter PID controller. *International Journal of Mechanical Engineering and Robotics Research*, *11*(8), 614–621. https://doi.org/10.18178/ijmerr.11.8.614-621

Urbano Lopes, G. B., Costa, R. R. M. S., & Alves Filho, S. E. (2024). Robot Finder v2: Search and ingestion of educational robotics data from Youtube. *2024 Brazilian Symposium on Robotics and 2024 Workshop on Robotics in Education (SBR-WRE 2024)* (pp. 336–341). IEEE. https://doi.org/10.1109/SBR/WRE63066.2024.10837853

ValveSoftware/openvr: OpenVR SDK. (n.d.). GitHub. https://github.com/ValveSoftware/openvr

Velez-Lopez, G. C., Vazquez-Leal, H., Hernandez-Martinez, L., Sarmiento-Reyes, A., Diaz-Arango, G., Huerta-Chua, J., Rico-Aniles, H. D., & Jimenez-Fernandez, V. M. (2022). A novel collision-free homotopy path planning for planar robotic arms. *Sensors*, *22*(11), Article 4022. https://doi.org/10.3390/S22114022

VIVE Tracker (3.0). (n.d.). *VIVE European Union*. https://www.vive.com/eu/accessory/tracker3/

Wongsuwan, K., & Sukvichai, K. (2015). Development of visual odometry estimation for an underwater robot navigation system. *IEIE Transactions on Smart Processing & Computing*, *4*(4), 216–223.

Xuying, B., Ping, X., & Gao, W. (2017). Calibration of systematic errors for wheeled mobile robots. *International Journal of Scientific Engineering and Science*, *1*(9), 14–16.

Yao, W., & Cao, M. (2020). Path following control in 3D using a vector field. *Automatica*, *117*, Article 108957. https://doi.org/10.1016/J.AUTOMATICA.2020.108957

Zruya, O., Sharon, Y., Kossowsky, H., Forni, F., Geftler, A., & Nisky, I. (2022). A new power law linking the speed to the geometry of tool-tip orientation in teleoperation of a robot-assisted surgical system. *IEEE Robotics and Automation Letters*, *7*(4), 10762–10769. https://doi.org/10.1109/LRA.2022.3193485

Appendix 1

Preparing the Vive VR Motion Tracker for Use Without a VR Headset

1. Install the Visual Code environment
2. Install the OpenVR python library
3. Install Steam
4. Install SteamVR
5. In the [Steam installation directory]\Steam\steamapps\common\SteamVR\drivers\null\resources\settings\default.vrsettings file, set the value:
    a. “enable”: true,
6. In the [Steam installation directory]\ Steam\steamapps\common\SteamVR\resources\settings\default.vrsettings file, in the SteamVR section, set the values:
    a. "requireHmd": false,
    b. "forcedDriver": "null",
    c. "activateMultipleDrivers": true,
7. Launch the base station
8. Pair the motion tracker with your computer
9. Launch SteamVR and make sure that the motion tracker and base station have been detected
10. Adding a set of functions and dependencies to the project—https://github.com/snuvclab/Vive_Tracker
    a. Testing the operation of the motion sensor

Appendix 2

Preparing the Experiment Station

1. Make sure the robot is charged
2. Connect the base stations to the power
3. Start SteamVR
4. Start the EV3 robot with the SD card with the Pybricks system image
    a. After start, unplug and plug-in all cables to the EV3 brick.
5. Start the motion tracker
6. Start the Visual Code with the EV3 control server program project
    a. Connect to the robot
    b. Test the operation of the motors and sensors
7. Start the second instance of Visual Code with the motion tracker recording client program project
    a. Test the operation of the motion tracker
    b. Make sure what sequence of test scenarios is prepared

Starting the experiment

1. Select the scenario collection
    a. To check the operation of the communication, select a simple test scenario
    b. To run the full experiment, select the full test scenario collection
2. Select and run the server program main file in debug mode to upload the project to the robot's memory
    a. Wait until the data loading procedure and the initial robot configuration are finished
3. Select and run the main program file client
4. Monitor the communication between the client and the server

Completing the experiment

1. Exit the server and client programs
2. Download all log files from the robot
3. Save the log files from the robot and the motion sensor to a common archive
4. Turn off the robot and the motion sensor and disconnect the base stations from the power supply
5. Turn off all other programs and devices

Appendix 3

Problems observed during experiments

Problem	Description
Sensors and motors disconnecting	sometimes the plugs would come out, which would result in a critical error and stop the script. The reason could be: a faulty cable, a loose socket in the sensor, a loose socket in the ev3, pressure from structural elements on the cable causing it to break.
Hitting the wall and blocking further travel	due to accumulating drifts, the starting point kept moving and occasionally the robot, despite its original safe position, would hit obstacles at the walls of the room, which eliminated the need to repeat the travel.
Bluetooth communication disruptions	it happened that the Bluetooth communication of other devices located near the computer and the robot caused an error in communication with the robot. This resulted in disruption of the control application and the need to restart the system.