JISSE
ISSN: 2636-4425

Journal of International Society for Science and Engineering
Vol. 6, No. 3, 52-58 (2024)

JISSE
E-ISSN:2682-3438

An Onboard and Simple to Use Trajectory Programming Odometry System for Work Floor Mobile Robots

Mahmoud El-Bayoumi

Mechanical Engineering Department, Engineering and Renewable Energy Research Institute, National Research Centre, Egypt

ARTICLE INFO

Article history:
Received:06-10-2024
Accepted:10-10-2024
Online:11-10-2024

Keywords:
Mobile Robots
Wheeled Robot
Navigation
Indoor

ABSTRACT

Mobile robot applications are widely spreading and are expected to gain more ground, especially in production facilities and warehouses. They employ different techniques to initially define or decide their trajectories and then follow it. The outdoor robots employ trajectories based on radio-navigation systems, such as GPS, while indoor robots employ computer-generated trajectories. This study aims to introduce an onboard simple-to-use, yet efficient, odometry-based trajectory programming system for work floor mobile robots, which could be used by unskilled workers. The proposed system employs two indexed wheels fitted to the robot's non-steered robot wheels. The robot's trajectory is programmed by manually driving the robot or trailing it by a vehicle, on its required trajectory. The resulting signals from the indexed wheels are recorded by a control unit and used to construct the necessary robot's trajectory, using a specially formulated algorithm. The same indexed wheels could be further employed to facilitate the robot's awareness of its location, position, speed, and acceleration to facilitate its trajectory following process. The system performance was evaluated using a specially written simulation program. The simulation results show that the trajectory programming system is able to capture and construct the robot's required trajectories with considerable accuracy. It also pointed out the parameters that govern its accuracy such as the resolution of the indexed wheels and the robot's wheels track (robot's width). The proposed trajectory programming system was found suitable for facilitating the wider adoption of mobile robots' services.

1. Introduction

The mobile robot turned from science fiction movies to a sound reality not only rolling on wheels but also flying overheads. Although they have stirred a lot of worries and concerns, they have become a necessity, especially in production facilities. Yet, they require further development for extra services and more safety.

For robot trajectory planning the robots would be autonomous; taking the target location and deciding the best trajectory to follow to reach the target. These are more expensive robots and require high technical capacities for their installation, programming, and maintenance. Also, robots would be non-autonomous, hence requiring a feeding of the trajectories to their control units. They are less expensive than autonomous robots but they are still costly for developing economies and require substantial technological capacities for their efficient operation.

Both robots' operation and trajectory programming require the robot to employ equipment to detect its environment, to be able to adjust its location and orientation with respect to a given coordinate system. Some of this equipment would be carried on board the robot and some would be installed somewhere within the robot's environment.

Outdoor mobile robots make use of space-based radio-navigation systems such as the USA's GPS, Russia's GLONASS,

* Mahmoud El-Bayoumi, Mechanical Engineering Department, National Research Centre, Cairo, Egypt, +2 01151886211, el_bayoumi@hotmail.com

Chinese BeiDou [1], and European Galileo [2]. As these systems are satellite-based their performances suffer greatly indoors, and consequently are not to be relied on for work floor applications.

Work floor mobile robots are indoor mobile robots that serve very important and rapidly expanding roles in modern manufacturing facilities [3] such as in material handling [4] as well as handling of finished products. Any failure of the mobile robot's motion planning would lead to work floor congestion leading to a reduction in production volumes [5]. That would, in turn, increase product costs, reduce product market competitiveness, and reduce profitability. Accordingly, a lot of research is conducted towards their trajectory planning to ensure flawless traveling [6].

Indoor work floor mobile robots employ different systems to flow their programmed trajectories. Hence, the trajectories are based on the means of employed tracking systems. Most robots employ Inertial Navigation Systems (INS) [7], which is a type of Microelectromechanical systems (MEMS), for tracking systems. MEMS are advanced IC sensors that can measure their longitudinal, lateral, vertical, and angular accelerations and gyroscopes [8], or a mix of them. The control unit should employ these acceleration measurements with the aim of following the suitably programmed trajectory.

Some systems employ surveillance cameras fitted on the walls to detect the location and position of the robot and feed it to the robot wirelessly [9]. Some robots use ultrasonic sensors [10], onboard cameras [11], or light detection and ranging LiDAR [6] fitted on the robots to detect their surroundings and obstacles. Finally, calculate its own location and orientation with respect to them to navigate this environment.

Other robots are fitted with sensors that detect colored objects [12] or tracks on the floor, hence detecting their location and orientations [13] with respect to the track. Some detect magnetic field patterns on the work floor to track its location [14]. Some are fitted with metal detectors to sense metals guiding their path [4]. In some cases, sensors are based on simple mechanical switches to detect touching obstacles [4]. Trajectories for each system would be programmed in a manner that would facilitate the use of the employed tracking systems.

The main aim of the current proposed system is to facilitate, on-the-spot, trajectory programming for mobile robots. The system is low-cost and easy to use even by unskilled workers. It could also be used to control the robot's traveling on the required trajectory. The ease of trajectory programming should facilitate the wider adoption of mobile robots by developing countries and by low-tech firms and facilities.

2. The proposed trajectory programming system

2.1. System layout

The current system layout, Figure 1, is comprised of two indexed wheels with two sensors and an Arduino microcontroller control unit. The indexed wheels are fitted to the right and left shafts holding the non-steered wheels of the robot. The sensors are fitted against the indexed wheels to feel their rotation and produce consequent signals. Rotation signals are fed to the Arduino control unit. The system employs a dedicated algorithm to convert the two streams of data into a specially coded trajectory.

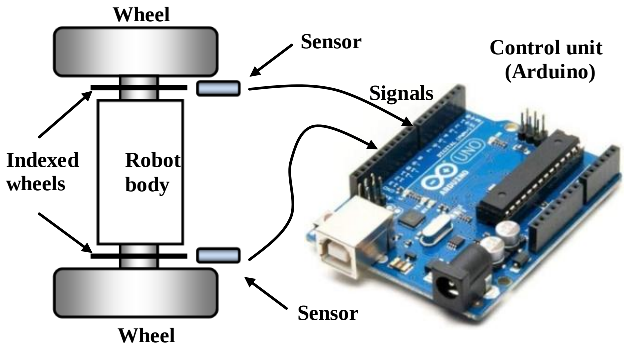

Figure 1: Proposed system layout

2.2. Indexed wheels and sensors

Indexed wheels have been used for a very long time, yet ample modern time applications. They are versatile, and robust and would need minimal care. With developments of sensors, indexed wheels blended in computerized applications.

Indexed wheels are thin metallic or plastic wheels that have some equally spaced angular physical features on them. These features could be holes, slots, magnets, magnetic material pads, protrusions, or patches with different colors or specific reflections. The number of index features could be just one up to a large number of microscopic features.

The indexed wheel operates by rotating in front of a sensor that senses the passing of the specific feature and produces a signal in response. The signals should inform the connected electronics to the passage of the indexed wheel feature. Some sensors are designed to sense the direction of the indexed wheel rotation as well; clockwise or counterclockwise.

These indexed wheel signals could be used for a wide range of applications. For the current work, the indexed wheels will be fitted to the robot's non-steered wheels (rear wheels) to track its travel for the sake of trajectory programming. This odometry application should find the right and left indexed wheel signals, while the robot is manually driven or trailed, reflecting the trajectory required to be programmed in the mobile robot's control unit. The frequency of the pulses should reflect, with respect to system parameters, the robot's speeds required.

2.3. Optical Indexed wheel

The optical indexed wheel employs an illumination source and an illumination sensor arranged in front of each other leaving a small gap between them. This sensor feels the change in opacity of its gab and generates a pulse in response. The arrangement required for this sensor is an equiangular slotted disk. Figure (2-a) shows a diagram of the optical indexed wheel and its sensor.

2.4. Mechanical indexed wheel

The mechanical sensor employs a pulse switch equipped with a liver. Raising the lever triggers the switch. The arrangement required for this sensor is a disk equipped with a set of cams arranged in an equiangular way. The cams are designed to trigger the switch that responds with a pulse for each cam. Figure 2-b shows a diagram of the mechanical sensor.

2.5. Magnetic indexed wheel

This sensor feels the change in a magnetic field and generates a pulse in response. The arrangement required for this sensor is a disk equipped with a set of iron pads. The pads are arranged at equal angles around the disk center. Figure (2-c) shows a diagram of the magnetic (proximity) sensor.

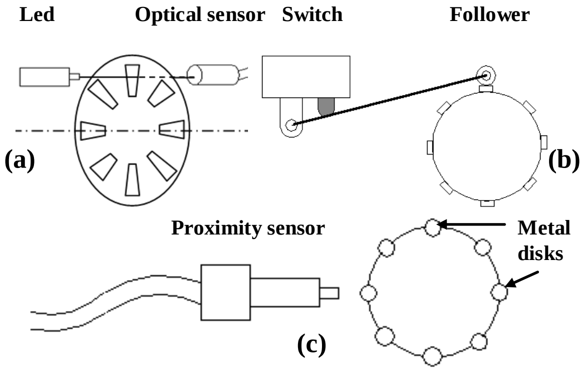

Figure 2: Diagram of indexed wheels (a) Optical (b) mechanical (c) Magnetic

Some types of optical and proximity sensors produce analog signals that represent the amount of excitation they experience, while some are turned on/off at a preset threshold. The on/off sensors would maintain compatibility with the on-board control unit, for Arduino microcontroller boards signals should ideally be 5 volts. The sensors should generate a stream of pulses each representing rotation of the indexed wheel by an angle equal to that between two adjacent indexes.

2.6. Signal conditioning

In some cases where the sensor output is not compatible with the Arduino input levels, a signal level shifter would be fitted between the sensor and the control unit. Also, if the sensor output is analog a signal conditioner would be required to convert it to logic levels, otherwise, the signal would be fed to an analog input on Arduino, which has Analog to analog-to-digital converter (ADC) limitations.

2.7. The mathematical approach

The speed difference between the inner and outer wheels is due to the difference between each radius of turn, as shown in Figure (3). As the indexing wheel with its n indexes causes the sensor to send n pulses per wheel revolution, the distance that is traveled by the wheel between the events of a sensor sending two consecutive signals is called the indexed distance I_d this distance is evaluated as a function of wheel diameter W_d and number on indexes of the indexed wheel n.

$$I_d = \frac{\pi W_d}{n} \tag{1}$$

For the current system, the two sensors are sending pulses in response to wheels rotations that are intrinsically not equal. This could be broken down into two conditions. In the first condition, the pulses are alternating from the right and left sensor and this happens when the robot is moving in a straight line or a very large diameter. For the current work alternating pulses are considered to represent a straight line moving forward for a distance of I_d. In the second condition, two consecutive pulses arrive at the control unit from the same sensor. This case is considered as a wheel center travel of distance $I_d/2$ in a direction having an angle of $\pm\theta$, based on which sensor. The robot itself rotates by $\pm 2\theta$ based on which sensor signal is repeated. The $I_d/2$ is an approximation as the $I_d/2$ should have represented an arc length and not the cord, as shown in Figure 4-b and 4-c, but this approximation is expected to pose no significant effect due to the small arc angle, especially at higher indexes number per the indexed wheels.

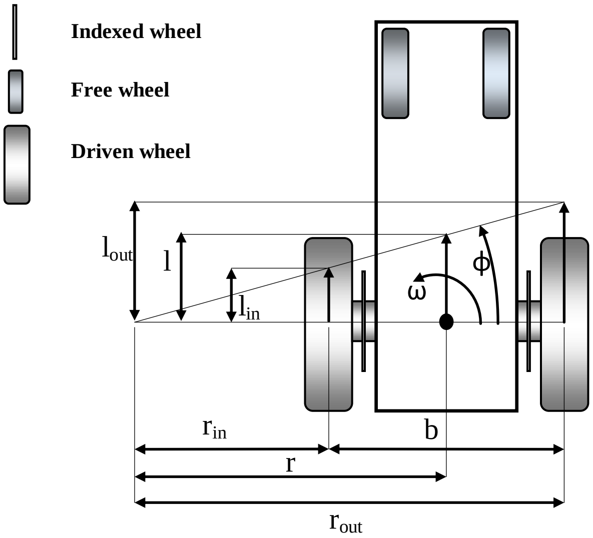

Figure 3: Schematic diagram of the wheeled robot turning

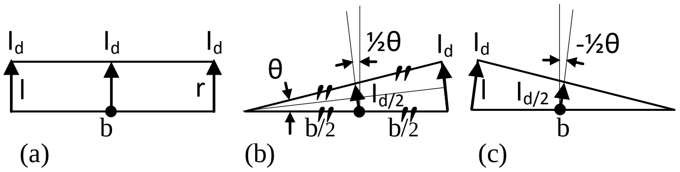

Figure 4: Interpretation of pulse sequencing (a) alternating pulses (b) extra right sensor pulse (c) extra left sensor pulse

To further simplify the process, the pulses in the current system are called 0 when they originate from the left sensor and 1 when they originate from the right sensor. This zero and one notation allows for compression of the trajectory map data as it would be written in a series of hexadecimal numbers.

The process of map construction searches the pulses for alternating trains (010101) to identify straight runs and searches for two (or more) consecutive pulses from the same sensor (00) or (11) for turns. An example of the process is given in Figure 5. The example applies the concept shown in Figure 4. For standardization of the process, the first pair will be considered the normal pattern that the bends will contradict it. Accordingly, for the example, the normal pattern is 01. This results in the first repeated zeros being the bends, while the second repeated ones being the bends. The bends always are multiplications of θ or -θ which are evaluated from the geometry of Figure 4-b as in equation (**2**). The technique should speed up and simplify robot trajectory programming to a great extent.

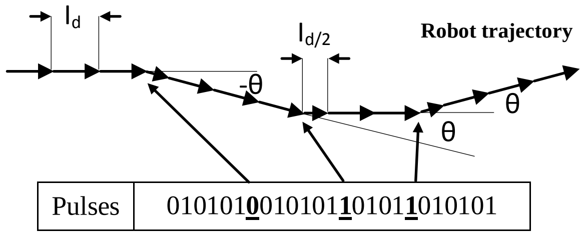

Figure 5: Example of trajectory construction using a pulse train

$$\theta = 2 * \sin^{-1}\left(\frac{l_d/2}{b}\right) \qquad (2)$$

For the current work, the moving distances are the main factor that is employed to capture the required trajectory. The speed of motion would be separately fed to the robot. The acceleration and the deceleration to and from the mean speed of robot travel could be embedded in the control unit rather than given as input to the system. However, the robot speed could be extracted from the programming process itself as given in equation (3), where t_n and t_{n+1} are the time of two consecutive pulses from one of the sensors during a period of alternating pulses; straight run.

$$v = \frac{l_d}{t_{n+1} - t_n} \qquad (3)$$

3. Results and discussion

To check the validity of the proposed system, a dedicated simulation program has been specially written using MATLAB 2009a. The program aims to evaluate an ideal robot path and compare it to the trajectory that would be captured by the proposed programming system. For the current simulation task, a number of robot parameters were fixed. The wheel diameter value was fixed at 20 cm, while the wheelbase (distance between right and left wheels) was fixed at 50 cm.

As the most challenging feat of the proposed system is capturing curved paths, the simulation considered a track of 90º bends for the robot runs. To evaluate the capacity of the proposed system at different curvatures the simulation was carried out against radius of turns of 50 cm, 100 cm, and 200 cm. These values are supposed to cover the range of bends of work floors. Although bends would have greater values, the smaller bends were considered the more challenging to the proposed system.

The trajectories were evaluated with an indexing number of 10 and were increased until a satisfactory match was achieved. This is meant not to overcrowd the figures and complicate their interpretation.

The simulation of the robot trajectory negotiating the 50 cm turn is shown in Figure 6. The results showed enhancement matching of the simulated trajectory to the ideal path as the indexing n values increased. At an n value of 30, the simulated trajectory overlapped the ideal path with a minor overshoot at the end.

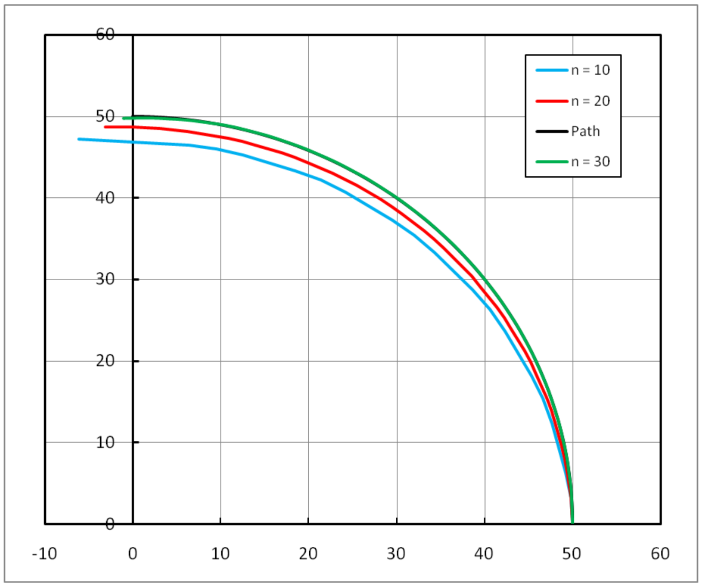

Figure 6: Robot path on a 50 cm radius turn and captured trajectories using indexed wheels with n = 10, 20, 30 and 50

The simulation of the 100 cm bend, shown in Figure 7 showed a match at a lower n value. This was expected as the smaller bends are more difficult due to the faster change of the robot's orientation angle φ. At n value of 20 the simulated trajectory followed closely the ideal path, with a smaller overshoot than that of the 50 cm bend trajectory at n equal to 30.

The results of the 200 cm bend, Figure 8, showed controversial results as it failed to deliver a trajectory with a good match until n reached 50. This could be attributed to longer distances traveled forward before the sensor would be able to capture a bend and send the expected signal. This in turn is caused by the right and left wheels rotating at closer speeds as the radius turn increases.

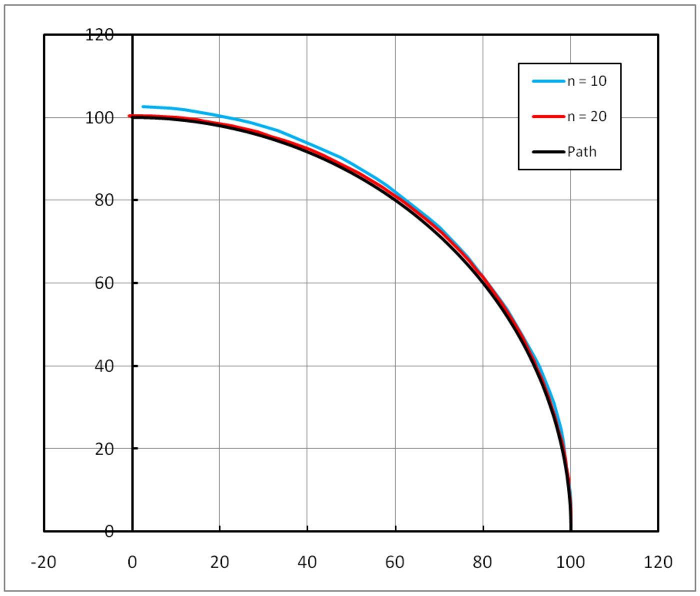

Figure 7: Robot path on a 100 cm radius turn and captured trajectories using indexed wheels with n = 10, 20, 30 and 50

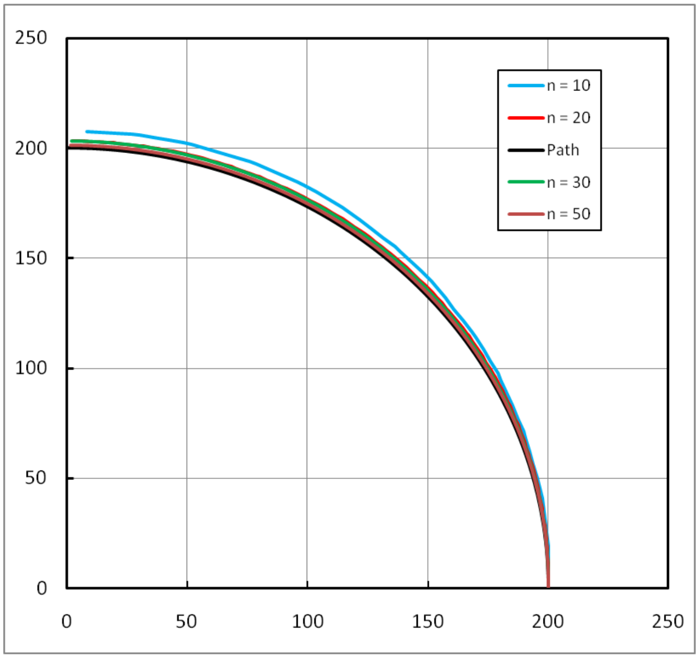

Figure 8: Robot path on a 200 cm radius turn and captured trajectories using indexed wheels with n = 10, 20, 30 and 50

As the system should be able to operate for all the bend radiuses expected in the work floor, in the current simulation, indexing n equal to 50 was considered for all the bend radiuses, and a numerical evaluation of the end position and orientation deviations was evaluated.

To further investigate the performance of the system objectively the x, y location, and orientation (φ) of the robot at the end of different trajectories were evaluated. The results were compared to the ideal values expected from the originally planned path. This comparison was carried out at different turn radiuses of 50 cm, shown in Table 1, 100 cm, shown in Table 2, and 200 cm, shown in Table 3. The results showed the expected general trend of the values getting closer to ideal values as the number of wheel indexes increased, with some exceptions.

Table 1: Ideal path and trajectory deviations at the bend of 50 cm

	Radius 50 cm				
	Path	n = 10	n = 20	n = 30	n = 50
x	**0**	**-6.17772**	**-3.09897**	**-1.0804829**	**-0.77666**
y	**50**	**47.19727**	**48.63726**	**49.73274016**	**49.75954**
φ	**180**	**176.457**	**180.0148**	**181.2066688**	**179.2823**

Table 2: Ideal path and trajectory deviations at the bend of 100 cm

	Radius 100 cm				
	Path	n = 10	n = 20	n = 30	n = 50
x	**0**	**2.376187**	**-0.66553**	**-0.16628**	**0.005711**
y	**100**	**102.759**	**100.4039**	**100.6945**	**99.99803**
φ	**180**	**176.457**	**180.0148**	**178.8065**	**179.2823**

Table 3: Ideal path and trajectory deviations at the bend of 200 cm

	Radius 200 cm				
	Path	n = 10	n = 20	n = 30	n = 50
x	**0**	**8.291878**	**2.184405**	**1.923926**	**1.450518**
y	**200**	**207.3244**	**203.2799**	**203.0742**	**201.279**
φ	**180**	**176.457**	**180.0148**	**178.8065**	**179.2823**

To investigate the trend of the results, deviations of all trajectories' ends were plotted against each other. The deviations of x, y, and orientation (φ) values were plotted in Figures 10, 11, and 12, respectively at n values of 10, 20, 30, and, 50.

Figures 9, 10, and 11 agree on the lower performance of n = 10 with respect to x, y, and φ deviation at ends, albeit its seemingly better performance in the 100 cm radius trajectory, Figure 7.

The enhancement trend as the n values increase was expected, but results on indexing value n = 20 were a surprise, especially when compared with indexing value n = 20 modest performance at 50 cm radius turn, Figure 6.

Indexing value n = 20 achieved the lowest deviation of orientation φ. Also, results showed equal φ deviations for the same n value's runs regardless of the turn radius. The reason behind this was investigated and it turned out that the proposed system building of the robot orientation as multiples of θ and –θ led to this phenomenon.

To investigate this result values of the robot's incremental rotation θ values were evaluated for different n values, using equation 2 for different n values, are shown in Table 4. Also, the expected undershoots and overshoots, as θ values accumulate starting from 90 (simulation start point) and surrounding 180 (simulation end point) for different n values. Also the expected φ errors in case of undershoot and overshoot are given.

The results show that θ value should be decreased to the maximum possible value. Given equations 1 and 2, to reduce θ value there are three options, to reduce the wheel diameter, to increase the wheelbase, or to increase the n value. Accordingly, the only feasible means to reduce θ value is to increase the n value.

The steady trend of error reduction as n increases shows that the system could be tailored to the required job of programming the robot's required track into the control unit using the proposed odometry system.

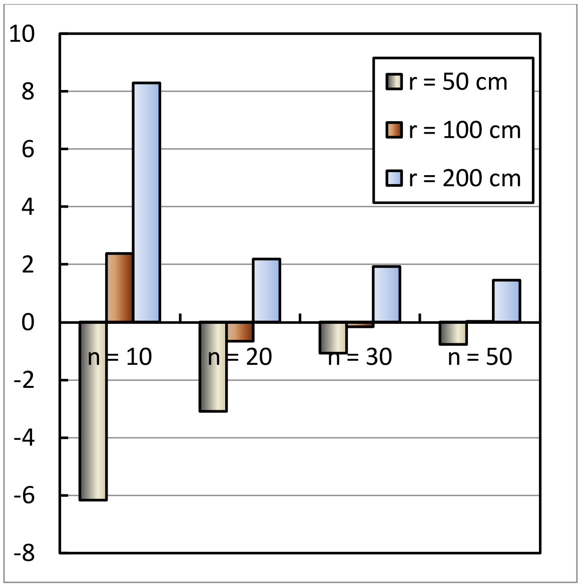

Figure 9: Trajectory's end, x deviation (cm) at n = 10, 20, 30 and 50

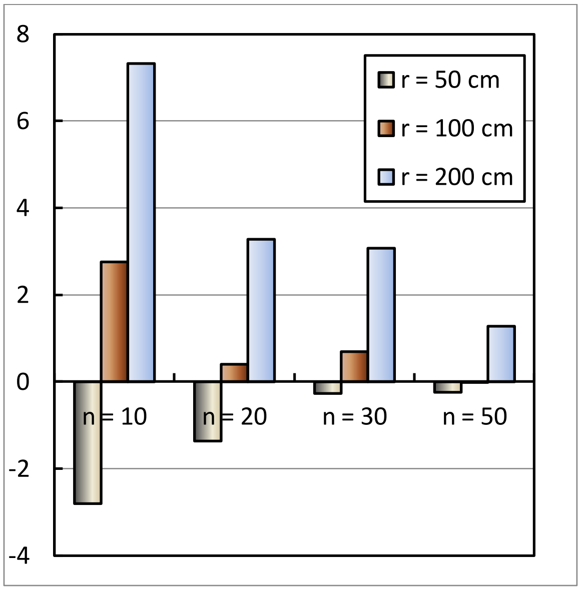

Figure 10: Trajectory's end, y deviation (cm) at n = 10, 20, 30 and 50

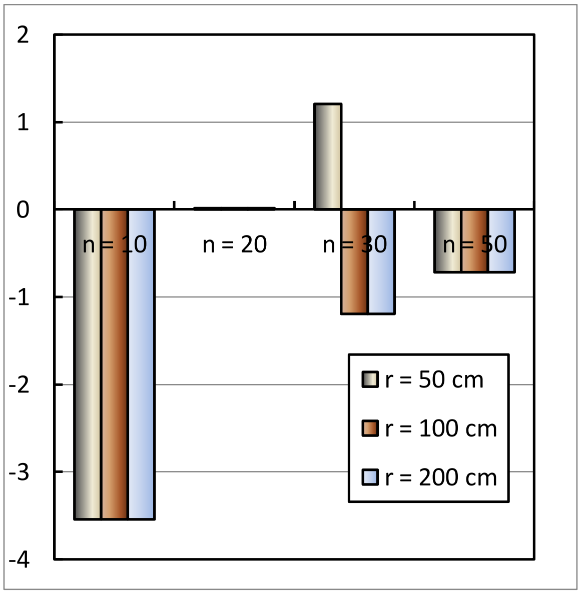

Figure 11: Trajectory's end, φ deviation of orientation (degree) at n = 10, 20, 30 and 50

Table 4: Value of robot orientation increment θ (degree)

	n = 10	n = 20	n = 30	n = 40	n = 50
θ (deg)	7.2047	3.6006	2.4002	1.8001	1.44
Higher than 180	176.4564	176.4144	178.8074	178.2049	179.2800
Lower than 180	183.6611	180.0150	181.2076	180.0050	180.7200
Undershoot error	-3.5436	-3.5856	-1.1926	-1.7951	-0.72
Overshoot error	3.6611	0.015	1.2076	0.005	0.72

The proposed system is simple enough to be fitted to older mobile robots or platforms, given that some types of indexed wheels would allow being fitted to the inner side of the robot's non-steered wheels, or even printed directly on the wheels, as in the case of reflective indexed wheels where patches of different reflections are fitted to or spray painted on the indexed wheels.

4. Conclusions

In the current study, an odometry-based system that would facilitate programming trajectories of mobile robots on the spot with ease by unskilled workers was introduced. The current system relies on manual driving or trailing of the mobile robot on the required trajectory to copy this trajectory to the systems control unit. Then this trajectory could be travelled upon as many times as required. This also means any worker would be capable of programming new trajectories as required with ease.

The proposed system is cheaper, simpler, and easier to implement and maintain than competitive systems. Also, the system is rather simple with two indexed wheels, two sensors, and an Arduino control unit.

The system performance was tested using a specially written MATLAB R2009a code and the results demonstrated the good potential of the current system as a mobile robot trajectory programming means.

The simulation also uncovered that its performance is highly governed by the increase of the number of indexes on its index wheels, as well as the properly selected value of its incremental orientation angle θ. The system benefits and expected ease of use demonstrate that it would allow its fitted mobile robots an edge and facilitate their penetration in low tech facilities and developing countries.

5. References

[1] D. Nemec, V. Šimák, A. Janota, M. Hruboš, and E. Bubeníková, "Precise localization of the mobile wheeled robot using sensor fusion of odometry, visual artificial landmarks and inertial sensors," *Robotics and Autonomous Systems*, vol. 112, pp. 168–177, Feb. 2019, doi: 10.1016/j.robot.2018.11.019.

[2] T. Xu *et al.*, "Galileo multi-day precise orbit determination using L-band and SLR data," *Geodesy and Geodynamics*, Sep. 2024, doi: 10.1016/j.geog.2024.08.002.

[3] H. Jo, S. Seo, J. Kim, and F. Bien, "A coreless track-type seamless wireless charging system using co-planar wires enabling quasi-free planar movements for mobile logistics robots," *Applied Energy*, vol. 375, p. 123943, Dec. 2024, doi: 10.1016/j.apenergy.2024.123943.

[4] S.-F. Wu, J.-S. Mei, and P.-Y. Niu, "Path guidance and control of a guided wheeled mobile robot," *Control Engineering Practice*, vol. 9, no. 1, pp. 97–105, Jan. 2001, doi: 10.1016/S0967-0661(00)00070-8.

[5] D. Suhardini, W. Septiani, and S. Fauziah, "Design and Simulation Plant Layout Using Systematic Layout Planning," *IOP Conference Series: Materials Science and Engineering*, vol. 277, p. 012051, Dec. 2017, doi: 10.1088/1757-899X/277/1/012051.

[6] N. Promkaew, S. Thammawiset, P. Srisan, P. Sanitchon, T. Tummawai, and S. Sukpancharoen, "Development of metaheuristic algorithms for efficient path planning of autonomous mobile robots in indoor environments," *Results in Engineering*, vol. 22, p. 102280, Jun. 2024, doi: 10.1016/j.rineng.2024.102280.

[7] Y. L. Wei and M. C. Lee, "Mobile robot autonomous navigation using MEMS gyro north finding method in Global Urban System," in *2011 IEEE International Conference on Mechatronics and Automation*, Aug. 2011, pp. 91–96. doi: 10.1109/ICMA.2011.5985637.

[8] A. Pajaziti and L. Gara, "Navigation of Self-Balancing Mobile Robot through Sensors," *IFAC-PapersOnLine*, vol. 52, no. 25, pp. 429–434, Jan. 2019, doi: 10.1016/j.ifacol.2019.12.576.

[9] J.-H. Shim and Y.-I. Cho, "A Mobile Robot Localization using External Surveillance Cameras at Indoor," *Procedia Computer Science*, vol. 56, pp. 502–507, Jan. 2015, doi: 10.1016/j.procs.2015.07.242.

[10] X. Chen, S. Wang, B. Zhang, and L. Luo, "Multi-feature fusion tree trunk detection and orchard mobile robot localization using camera/ultrasonic sensors," *Computers and Electronics in Agriculture*, vol. 147, pp. 91–108, Apr. 2018, doi: 10.1016/j.compag.2018.02.009.

[11] P. Guo, Z. Liang, X. Wang, and M. Zheng, "Adaptive trajectory tracking of wheeled mobile robot based on fixed-time convergence with uncalibrated camera parameters," *ISA Transactions*, vol. 99, pp. 1–8, Apr. 2020, doi: 10.1016/j.isatra.2019.09.021.

[12] M. Alymani, M. E. Karar, and H. I. Shehata, "A Practical Study of Intelligent Image-Based Mobile Robot for Tracking Colored Objects," *Computers, Materials and Continua*, vol. 80, no. 2, pp. 2181–2197, Aug. 2024, doi: 10.32604/cmc.2024.052406.

[13] O. Gumus, M. Topaloglu, and D. Ozcelik, "The Use of Computer Controlled Line Follower Robots in Public Transport," *Procedia Computer Science*, vol. 102C, pp. 202–208, Oct. 2016, doi: 10.1016/j.procs.2016.09.390.

[14] T. Zhang, L. Wei, J. Kuang, H. Tang, and X. Niu, "Mag-ODO: Motion speed estimation for indoor robots based on dual magnetometers," *Measurement*, vol. 222, p. 113688, Nov. 2023, doi: 10.1016/j.measurement.2023.113688.