Int. j. inf. tecnol. (October 2021) 13(5):2101–2108
https://doi.org/10.1007/s41870-021-00731-w

ORIGINAL RESEARCH

Motor velocity based multi-objective genetic algorithm controlled navigation method for two-wheeled pioneer P3-DX robot in V-REP scenario

Vikas Singh Panwar[1] · Anish Pandey[1] · Md. E. Hasan[1]

Received: 2 April 2021 / Accepted: 11 July 2021 / Published online: 18 July 2021

Abstract This paper proposes the right and left motor velocity based multi-objective genetic algorithm controlled navigation method for Two-Wheeled Pioneer P3-DX Robot (TWPR) in Virtual Robot Experimentation Platform (V-REP) software scenarios. The first objective function is made by taking front, left, and right ultrasonic sensors data and right motor velocity. Similarly, the second objective function is designed using front, left, and right ultrasonic sensors data and left motor velocity. Next, the sensor data are considered independent variables or inputs. The velocities of the motors are chosen as dependent variables or outputs for making multi-objective fitness functions for genetic algorithm (GA). This multi-objective GA makes a sensor-actuator control architecture and helps the TWPR to avoid the obstacles in the simulated scenarios. Further, the successful simulation test results show that the multi-objective GA provided a collision-free smooth, and shortest path for TWPR compared to the previously developed benchmark single objective GA.

Keywords Multi-objective genetic algorithm · Navigation method · Two-wheeled pioneer P3-DX robot · Virtual robot experimentation platform · Sensor · Single objective genetic algorithm

✉ Anish Pandey
anish06353@gmail.com

[1] School of Mechanical Engineering, KIIT Deemed To Be University, Patia, Bhubaneswar PIN - 751024, India

1 Introduction

The modern wheeled robots or wheeled vehicles help the human and perform various tasks like assembly part transfer from one place to another place in industries, autonomous transportation, security surveillance, floor cleaning in the house, etc. The motion control problem in a wheeled robot can be classified mainly into four categories: obstacle avoidance, position stabilization, trajectory tracking, and line following [1]. Similarly, the direction, velocity, and orientation of a wheeled robot are controlled by either the differential drive concept [1, 2] or the car-like steer control concept [3, 4]. For car-like steer control concept wheeled robots, we can use a single objective function for motion planning. In contrast, we require at least multi-objective velocity functions to control the velocity and orientation for the differential drive control concept. Mainly the wheeled robot motion planning is done by using either vision sensors (camera) through real-time video (image) processing or by using distance measuring sensors like infrared and sonar sensors [5]. In [4], the multi-objective particle swarm optimization algorithm was implemented to solve the car-like wheeled robot's motion planning problem between rough terrain. In their work, the Euclidean distance between the two points of the robot was taken as an objective function for the presented algorithm. However, the authors have not considered any wheel velocities based multi-objective functions in their article. Multi-objective particle swarm optimization based global navigation method for a wheeled robot was presented in Ref. [6]. Grid map-based route planning of wheeled robot using objective function GA was presented by de Camargo et al. [7]. However, in their article, the authors have only performed computer simulations, and they have not done any physical or virtual experiments.

GA based type-2 fuzzy logic method was designed and implemented in Ref. [8] for an existing path tracking control of a wheeled robot. Castillo et al. [9] have been used Pareto optimality multi-objective GA for offline motion planning of a wheeled robot. In that paper, the authors have done a grid map based offline motion of a wheeled robot and did not use any real-time experiments. Obstacle distance data and heading (steering) angle based single objective function controlled GA method had been applied by Mohanta et al. [10] for multiple wheeled robot motion control and obstacle avoidance in the cluttered environment. In the manuscript, the authors have not mentioned how the velocity of wheels of robots was controlled. Euclidean distance-based single fitness function GA has been presented in Ref. [11] for wheeled robot path planning. Song et al. [12] tackle the problem of smooth trajectory planning for a wheeled robot using a GA and the Bezier curve. The authors of this manuscript used a grid-map-based simulation workspace to show the autonomous motion result of a two-wheeled robot. The presented method for the wheeled robot can only be applicable for computer simulation, and it can be tough to implement this method for the physical model. In another work [13], the GA tunes the membership functions of the type-2 fuzzy logic architecture, and this tuned type-2 fuzzy logic was used for dynamic torque control of an autonomous wheeled robot. Another interesting article can be found in [14], where Xiaolei et al. implemented an improved GA to tackle the navigation problem of a wheeled robot autonomously between complicated grid map workspace. The grid-map-based navigation may only be suitable for computer simulation because not all the real-world scenarios need to be grid-mapped. Fuzzy rule-based architecture has been implemented in an article [15] to control the motion of two co-operating wheeled robots to perform desired tasks in an unstructured environment. Wang [16] suggested the sensor-based GA to optimize the path of a wheeled robot during navigation among obstacles in grid-map-based workspace. However, the velocity control method for a robot has not been addressed and considered in the paper [16]. In [17], the fuzzy controller with an artificial potential field based hybrid method was implemented to control the steering angle of a mobile robot between C and H types complicated obstacle conditions. Single ultrasonic sensor-based autonomous object detection and collision avoidance method for a differential driven wheeled robot was studied in Ref. [18]. The simulation and experimental results of a wheeled robot had not been shown in that work. The authors of a manuscript [19] used PID (Proportional Integral Derivative) controller to adjust the speed of both the DC motors of a differential driven two-wheeled robot during its navigation.

After an extensive literature survey, the authors have found that most researchers have used the single objective function GA [7, 8, 10–12, 16, 20] for wheeled robot motion and orientation control between different platforms. Further, for controlling the left and right wheel velocities of any wheeled robot, at least multi-objective functions are required for smooth motion and path. However, using a car-like control concept in the wheeled robot, we can use a single objective function to control the steering angle. In the present work, the authors design and suggest the right and left motor velocity based multi-objective GA controlled navigation method; and implemented this method in V-REP based 3D (Three-Dimensional) real scenarios. This method guides and adjusts the motion and orientation of TWPR among obstacles and helps the TWGR to reach the target point. The primary aim of this research is to search the collision-free optimal path from the source point to the endpoint with minimum time, energy, and trajectory path length. In addition, we have compared this proposed multi-objective GA method with the previously developed single objective GA [20] and found that our developed multi-objective GA provides a smooth trajectory because it directly controls the wheel velocities of TWPR. The rest of the article has been sequenced in the following manner:—Sect. 2 contains the kinematics study of a differential driven TWPR. Section 3 displays a brief description of the multi-objective genetic algorithm design for velocity control. V-REP software-based three-dimensional (3D) experimental results and comparative study with the previously developed benchmark single objective GA [20] have been done in Sect. 4. Further on, the concluding remarks of the present study are summarized in Sect. 5.

2 Kinematics study of a differential driven two-wheeled pioneer P3-DX robot

The present work uses the differential driven Two-Wheeled Pioneer P3-DX Robot (TWPR) to perform experiments. The TWPR is a three-wheeled robot. Its two wheels are attached to the DC motors that control the motion and orientation of a TWPR, and the third wheel has a castor wheel. This castor wheel is used to provide stability, free orientation, and it also holds the chassis of the TWPR. The TWPR is 48.8 cm long, 38.1 cm in width, 21.7 cm height, and its total weight is 9 kg weight. The TWPR can move forward direction, backward direction, and it can travel with a top speed of 1.2 m/sec. In this work, the minimum and maximum speed of the TWPR have been fixed between 0.4 m/sec to 1.2 m/sec. It consists of 16 sonar sensors and 7 LEDs. These 16 sonar sensors can read the obstacles from 2 to 400 cm approximately. The TWPR is equipped with an inbuilt microcontroller, which makes the sensor-actuator system for the TWPR. This microcontroller sends the command to the motor controller to adjust the velocity, anti-clockwise, and clockwise directions of motors of the TWPR. Figure 1 reveals the schematic diagram of the differential driven TWPR in the (X, Y) plane. The axes (X_p, Y_p) are the present posture of the TWPR from the origin O point. θ is the orientation control angle, which denotes the turning of a TWPR about an axis (O, X). The following kinematic equations control the velocities and steering angle of the TWPR: -

$$u_c = \frac{r}{2}(u_r + u_l) \tag{1}$$

$$\omega_c = \dot{\theta} = \frac{r}{b}(u_r - u_l) \tag{2}$$

where u_c and ω_c are center (mean) linear velocity and center angular velocity of a TWPR, respectively. These u_c and ω_c control the motion and orientation of a TWPR in the computer programs. Next, u_r and u_l represent the right and left linear wheel velocities, respectively. The r is the radius of the driving wheels, and b is the track width between the two driving wheels. Figure 2 depicts the distribution of ultrasonic sensors in differential driven TWPR. The minimum value of the ultrasonic sensor numbers S_1 and S_2 gives the distance information of the Right Forward Object (RFO). Next, the minimum data of the sensor number S_3 and S_4 provides the distance information of the Front Forward Object (FFO). Similarly, the TWPR takes the distance information of Left Forward Object Detection (LFO) by taking the least value of sensor number S_5 and S_6. Further, the data of u_r, u_l, RFO, FFO, and LFO have been used for making the multi-objective fitness function for the GA in Sect. 3.

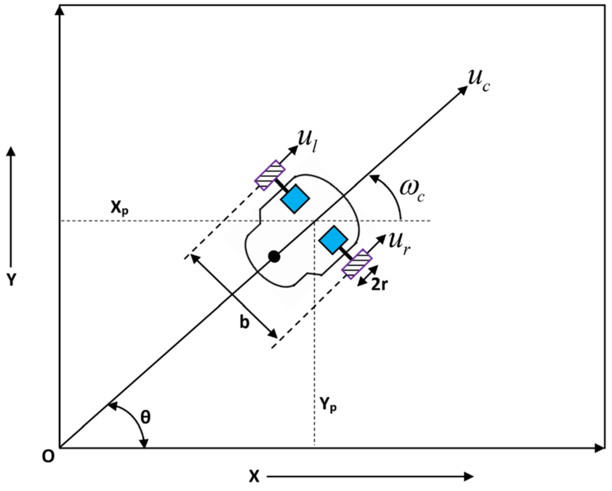

Fig. 1 Schematic diagram of the differential driven TWPR in the two-dimensional plane

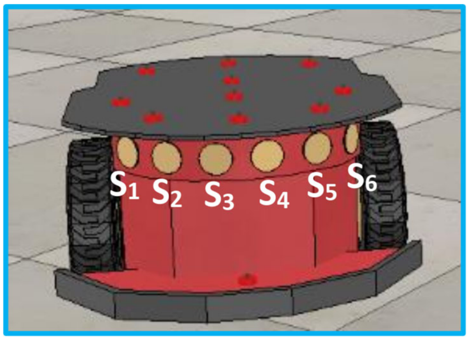

Fig. 2 Distribution of ultrasonic sensors in differential driven TWPR

Following conditions are taken for the straight motion and orientation of the TWPR: -

1. If $(u_r = u_l)$, then TWPR travels straight.
2. If $(u_r > u_l)$, then TWPR turns left.
3. If $(u_r < u_l)$, then TWPR turns right.

3 Multi-objective genetic algorithm design for velocity control

A genetic algorithm was introduced by John Holland in 1975 and is being used for solving various optimization problems. The GA is inspired by Charles Darwin's theory of natural evolution, selection, and genetics. Figure 3 reveals the basic flow chart of the GA, which is the combination of the different natural evolution processes. The GA has six main elements: initialization, selection of fitness function, crossover, mutation, evaluation, and generation. In the initialization step, we select the population size in a reasonable quantity, it may be between 20 to 1000, or it depends upon our requirements or the complexity of a problem. The selection of fitness function is a crucial part of the algorithm. In this study, the multi-objective fitness function was selected and designed using ultrasonic sensors data and the left and right motor (wheel) velocities. The crossover process has been done between the two parents to produce offspring. The crossover process is classified into four types, namely scattered, single-point crossover, two-point crossover, and multi-point crossover. In the mutation operation, some new genetic material has been added into the population to improve the performance of offspring. Mainly the ranges of crossover and mutation parameters in the multi-objective GA are kept between 0.6 to 0.9 and 0.001 to 0.01, respectively. After mutation operation, the algorithm evaluates the new property of mutated offspring and generates this mutated offspring as a solution (if the criteria satisfy; otherwise, the algorithm repeats the complete process).

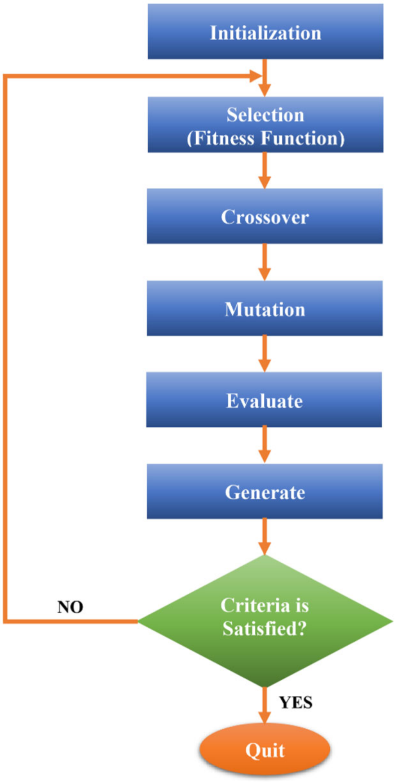

Fig. 3 Basic flow chart of Genetic algorithm

In Eqs. 3 and 4, the multi-objective fitness functions have been made by taking the real-time recorded data of independent variables like RFO, FFO, LFO, and dependent variables like u_r and u_l. These multi-objective fitness functions of the GA drive the TWPR and minimize the traveling path length for TWPR from the source point to the target point among obstacles in the various V-REP experimental scenarios. The minimum and maximum ranges of u_r and u_l have been taken as the same minimum and maximum speed of the motor (specified in Sect. 2). We have taken the senor data as inputs and velocity data as outputs in the Minitab software and have applied the general regression method to obtain multi-objective fitness functions. Figure 4 depicts the different phases of the TWPR motion controlling model through Eqs. 3 and 4. Figure 5 illustrates the Pareto optimality graph of multi-objective fitness functions of u_r (Right Wheel Linear Velocity) and u_l (Left Wheel Linear Velocity) as minimization-minimization problems. Similarly, Figs. 6, 7 reveal the fitness function training plot of u_r and u_l, respectively. Table 1 illustrates the selected values of various multi-objective GA parameters, which have been used in this study.

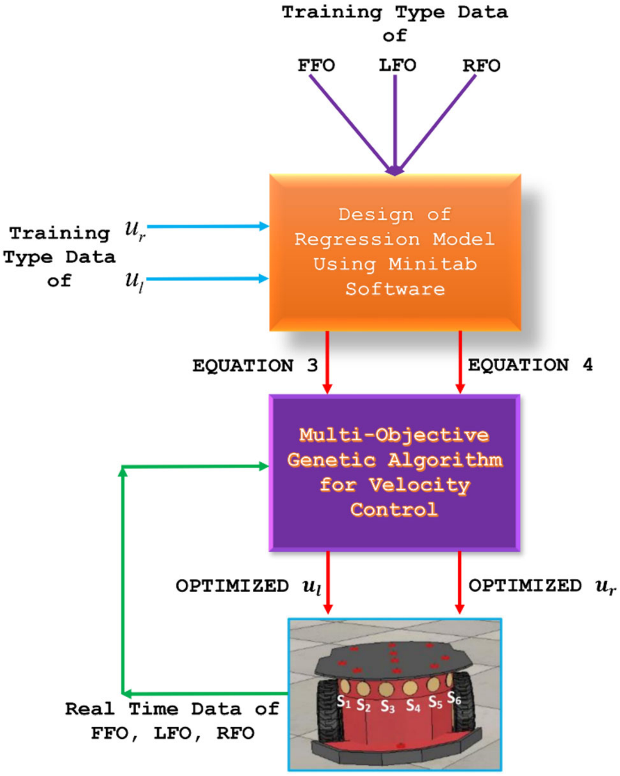
Training Type Data of
FFO
LFO
RFO
Training Type Data of
u_r
u_l
Design of Regression Model Using Minitab Software
EQUATION 3
EQUATION 4
Multi-Objective Genetic Algorithm for Velocity Control
OPTIMIZED u_l
OPTIMIZED u_r
Real Time Data of FFO, LFO, RFO
S1 S2 S3 S4 S5 S6

Fig. 4 Different phases of the TWPR motion controlling model through Eqs. 3 and 4

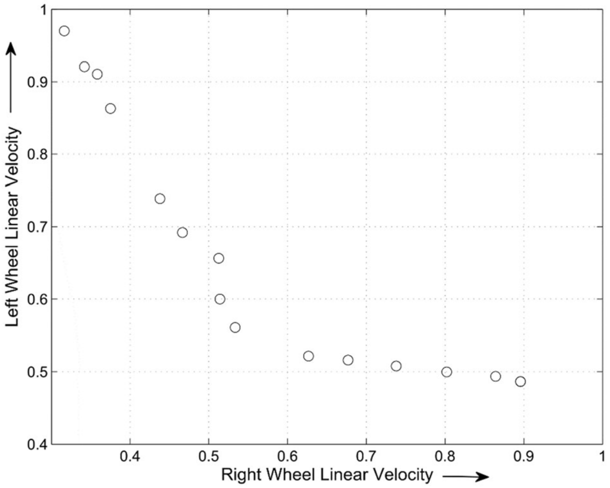

Fig. 5 Pareto optimality graph of multi-objective fitness functions of u_r (Right Wheel Linear Velocity) and u_l (Left Wheel Linear Velocity)

$$u_r = 0.461054 + 0.29154 \times FFO + 0.284199 \times LFO - 0.179613 \times RFO \tag{3}$$

$$u_l = 0.407061 + 0.289611 \times FFO - 0.036232 \times LFO + 0.342998 \times RFO \tag{4}$$

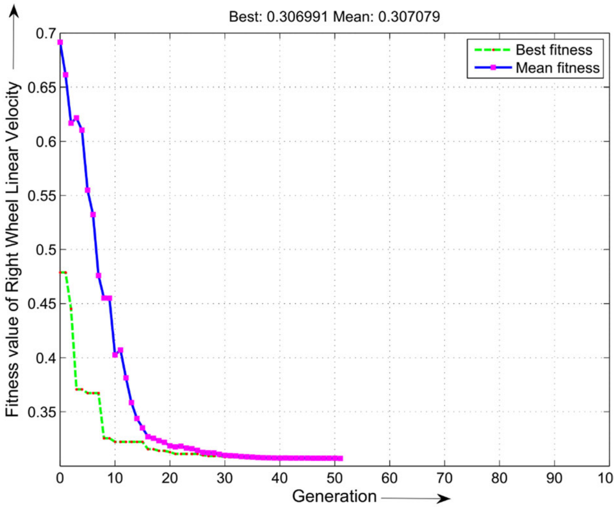

Fig. 6 Fitness function training plot of u_r (Right Wheel Linear Velocity)

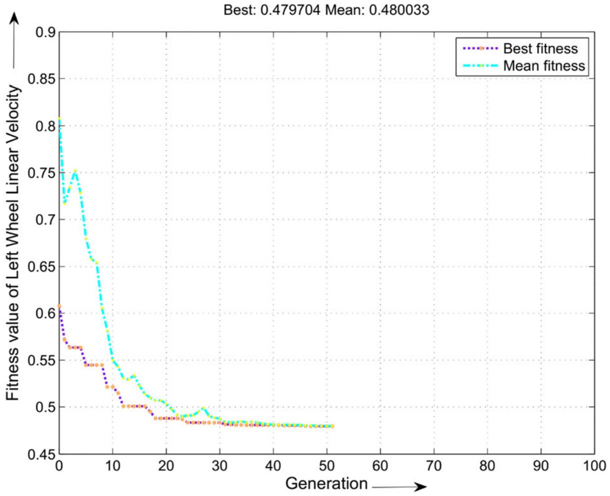

Fig. 7 Fitness function training plot of u_l (Left Wheel Linear Velocity)

Table 1 Selected parameters of multi-objective GA and their values

Parameter	Value
Population type	Double vector
Size of population	50
Selection function	Tournament
Tournament size	2
Value of crossover fraction	0.8
Function of mutation	Constraint dependent
Function of crossover	Scattered
Pareto font population fraction	0.35

The ranges of lower and upper boundaries of FFO, LFO, and RFO in the meter are $0.2 \leq FFO \leq 1.5$, $0.2 \leq LFO \leq 1.5$, and $0.2 \leq RFO \leq 1.5$.

4 V-REP based 3D experimental results and comparison analysis with previous benchmark single objective genetic algorithm [20]

In this section, the multi-objective GA controlled navigation method is applied for controlling the motion and orientation of TWPR in the V-REP scenario. The kinematic equations (listed in Sect. 2) and multi-objective GA with fitness functions (given in Sect. 3) have been written in MATLAB software programming script. The remote Application Programming Interface (API) functions of V-REP software have been taken to establish a connection between the V-REP and MATLAB software. After establishing the connection, the MATLAB running program script gives the controlling command to TWPR in the V-REP software scenario. The MATLAB receives the FFO, LFO, and RFO from the sensors (S_1 to S_6) of TWPR. According to the sensors data, the multi-objective GA provides the u_r and u_l to the TWPR. In the MATLAB script, some conditions (if-else) have been specified. If the TWPR does not find any object (within a specified range) during navigation, then it directly moves forward to reach the goal. Similarly, suppose the TWPR finds any object (within a specified range) at the front, left, and right sides during motion. In that case, the multi-objective GA gets activated and controls the u_r and u_l of TWPR in various V-REP scenarios.

In the current study, we have considered only a static environment for the experiments and have presented the outcomes of the navigation results of TWPR. Figure 8 reveals the 3D motion result of TWPR between the red color cube and blue color cylindrical object in the V-REP scenario. Figures 9, 10 show the real-time recorded right and left angular wheel velocities data (in Degree/Second) of TWPR, and real-time recorded absolute X and Y axes linear velocities (in Meters/Second) of TWPR, respectively. The TWPR starts motion from coordinates (20 cm, 20 cm), and the goal position is placed at coordinates (280 cm, 280 cm). In the V-REP scenario (see Fig. 8), firstly, TWPR starts the motion to reach the goal, and after moving some distance, the sensors find the object then TWPR turns right side that means the u_l increases and u_r decreases (see Fig. 9). The TWPR always turns right side because the goal is placed at the right corner of the scenario, and for taking a right turn, the multi-objective GA

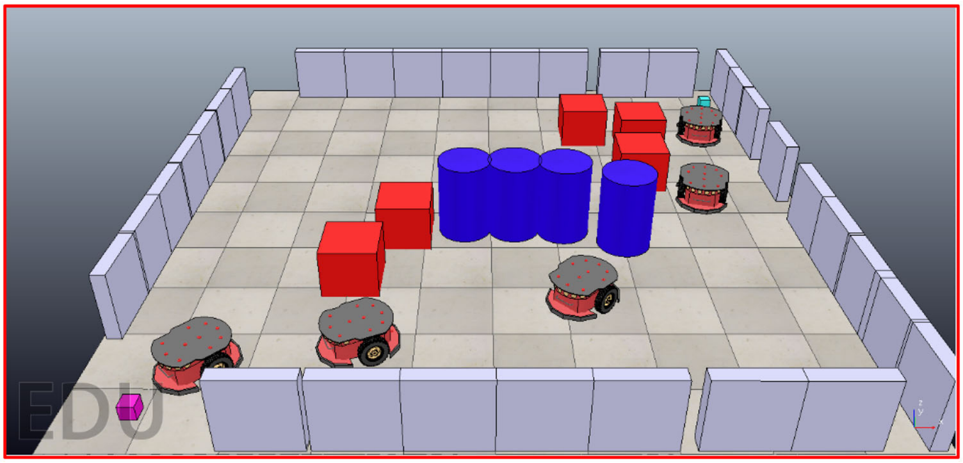

Fig. 8 3D motion result of TWPR among red color cube and blue color cylindrical object, start point (10 cm, 10 cm) and goal point (250 cm, 250 cm)

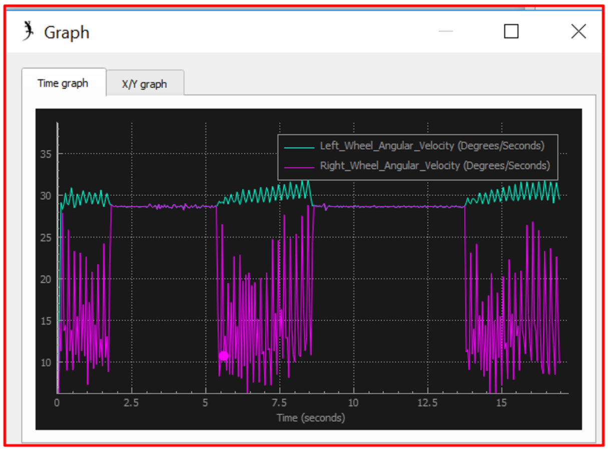
Graph
Time graph
X/Y graph
Left_Wheel_Angular_Velocity (Degrees/Seconds)
Right_Wheel_Angular_Velocity (Degrees/Seconds)
35
30
25
20
15
10
0
2.5
5
7.5
10
12.5
15
Time (seconds)

Fig. 9 Real-time recorded left and right angular wheel velocities data (in Degree/Second) of TWPR in Fig. 8

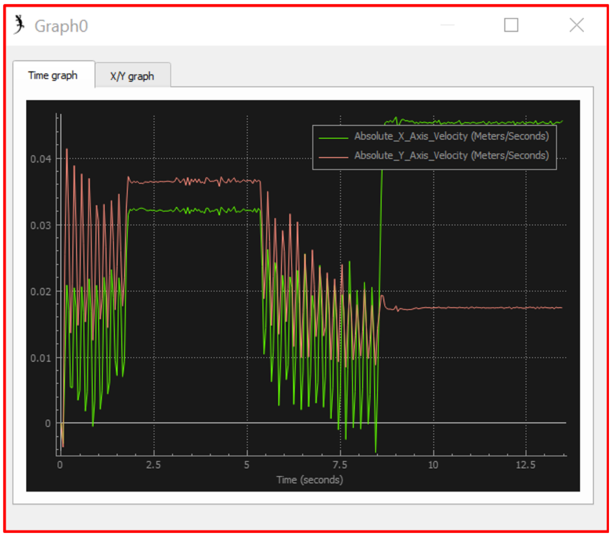
Graph0
Time graph
X/Y graph
Absolute_X_Axis_Velocity (Meters/Seconds)
Absolute_Y_Axis_Velocity (Meters/Seconds)
0.04
0.03
0.02
0.01
0
0
2.5
5
7.5
10
12.5
Time (seconds)

Fig. 10 Real-time recorded absolute X and Y axes linear velocities (in Meters/Second) of TWPR in Fig. 8

gives a command to increase the speed of u_l and decrease the speed u_r.

Figure 11 illustrates the MATLAB simulation result comparison of TWPR by applying a single objective GA [20] and the multi-objective GA in the same 2D (Two Dimensional) scenario. The size of the 2D scenario is 300 cm × 300 cm. In the 2D scenario, the TWPR navigates between the red color cube and blue color cylindrical objects (obstacles). In article [20], the authors apply the GA and use a single objective fitness function by taking the Euclidean distance between the two points of the robot. In Fig. 11, the black color circular trajectories reveal the motions of TWPR to reach the goal, and these motions are common in a single objective GA [20] and multi-objective GA.

Further, the yellow color circular trajectory represents the activation of a multi-objective GA to avoid the objects. Similarly, the blue color circular trajectory is obtained by

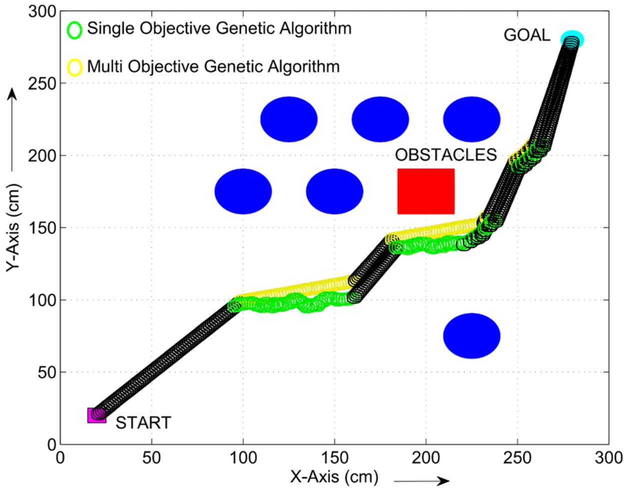

Fig. 11 MATLAB simulation result comparison of TWPR by applying single objective GA [20] and presented multi-objective GA in 2D scenario

Table 2 MATLAB simulation path length (from start to goal) comparison between single objective GA [20] and presented multi-objective GA in the same 2D scenario

Genetic Algorithm	Figure Number	Start Point	Goal Point	Trajectory Path Travel Distance (cm)	Final Trajectory Path Travel Distance (cm)
Single Objective [20]	Figure 11	(20 cm, 20 cm)	(280 cm, 280 cm)	152 cm	3.94 cm
Multi-Objective	Figure 11	(20 cm, 20 cm)	(280 cm, 280 cm)	141 cm	2.45 cm

the activation of single-objective GA. As we can see in Fig. 11, the presented multi-objective GA provides a smoother trajectory. Also, it takes less distance to reach the goal from the start position than the single objective GA. Table 2 shows the MATLAB simulation path length (from start to destination) comparison between the single objective GA and the multi-objective GA in a similar 2D scenario. The single objective GA running TWPR takes 152 cm to reach the goal with the path length error of 3.94 cm, while the multi-objective GA running TWPR covers 141 cm to reach the goal with the path length error of 2.45 cm. Besides, the developed multi-objective GA provides a smooth trajectory because it controls the wheel velocities of TWPR.

In contrast, the single objective GA works based on the Euclidean concept. The single objective GA calculates the distance between two points of TWPR randomly and then moves. Therefore, the probability of a smooth path is less in the single objective GA than wheel velocities controlled multi-objective GA. Moreover, we can see that the proposed method has given the optimal path in the static environment. Therefore, we can expect the same for the dynamic environment.

5 Conclusion

This article presents the multi-objective GA velocity controlled navigation method for TWPR in V-REP software scenarios. The multi-objective GA has given the velocity control commands to both the motors of TWPR for providing the forward, backward, and turning motions. The 2D and 3D experimental platforms with varying sizes of obstacles have been taken for testing of developed algorithm in the TWPR. The obtained experimental test results validate the effectiveness of the presented algorithm to control the motion of TWPR between obstacles. Compared to the previously developed benchmark single objective GA [20], this presented algorithm has given an optimal path. The developed multi-objective GA method can be implemented in different industries where motion-controlled automated guided robots and vehicles are extensively used. In the future, the same method with fitness functions can be used for navigation in the dynamic environment. Moreover, the developed multi-objective fitness functions can be used for other nature-inspired algorithms to obtain a more accurate path with a better convergence rate.

Declarations

Conflict of interest The authors declare that there is no conflict of interests.

References

1. Panahandeh P, Alipour K, Tarvirdizadeh B, Hadi A (2019) A kinematic Lyapunov-based controller to posture stabilization of wheeled mobile robots. Mech Syst Signal Process 134:106319–106337. https://doi.org/10.1016/j.ymssp.2019.106319
2. Pandey A, Kashyap AK, Parhi DR, Patle BK (2019) Autonomous mobile robot navigation between static and dynamic obstacles using multiple ANFIS architecture. World J Eng 16(2):275–286. https://doi.org/10.1108/WJE-03-2018-0092
3. Upadhyay S, Ratnoo A (2018) A point-to-ray framework for generating smooth parallel parking maneuvers. IEEE Robot Autom Lett 3(2):1268–1275. https://doi.org/10.1109/LRA.2018.2795655
4. Wang B, Li S, Guo J, Chen Q (2018) Car-like mobile robot path planning in rough terrain using multi-objective particle swarm optimization algorithm. Neurocomputing 282:42–51. https://doi.org/10.1016/j.neucom.2017.12.015
5. Bayat F, Najafinia S, Aliyari M (2018) Mobile robots path planning: electrostatic potential field approach. Exp Syst Appl 100:68–78. https://doi.org/10.1016/j.eswa.2018.01.050
6. Mac TT, Copot C, Tran DT, De Keyser R (2017) A hierarchical global path planning approach for mobile robots based on multi-objective particle swarm optimization. Appl Soft Comput 59:68–76. https://doi.org/10.1016/j.asoc.2017.05.012
7. de Camargo JTF, de Camargo EAF, Veraszto EV, Barreto G, Cândido J, Aceti PAZ (2019) Route planning by evolutionary computing: an approach based on genetic algorithms. Procedia Comput Sci 149:71–79. https://doi.org/10.1016/j.procs.2019.01.109
8. Martínez R, Castillo O, Aguilar LT (2009) Optimization of interval type-2 fuzzy logic controllers for a perturbed autonomous wheeled mobile robot using genetic algorithms. Inf Sci 179(13):2158–2174. https://doi.org/10.1016/j.ins.2008.12.028

9. Castillo O, Trujillo L, Melin P (2007) Multiple objective genetic algorithms for path-planning optimization in autonomous mobile robots. Soft Comput 11(3):269–279. https://doi.org/10.1007/s00500-006-0068-4
10. Mohanta JC, Parhi DR, Patel SK (2011) Path planning strategy for autonomous mobile robot navigation using Petri-GA optimisation. Comput Electr Eng 37(6):1058–1070. https://doi.org/10.1016/j.compeleceng.2011.07.007
11. Patle BK, Parhi DRK, Jagadeesh A, Kashyap SK (2018) Matrix-binary codes based genetic algorithm for path planning of mobile robot. Comput Electr Eng 67:708–728. https://doi.org/10.1016/j.compeleceng.2017.12.011
12. Song B, Wang Z, Sheng L (2016) A new genetic algorithm approach to smooth path planning for mobile robots. Assem Autom 36(2):138–145. https://doi.org/10.1108/AA-11-2015-094
13. Martínez R, Castillo O, Aguilar LT (2008) Intelligent control for a perturbed autonomous wheeled mobile robot using type-2 fuzzy logic and genetic algorithms. J Autom Mob Robot Intell Syst 2:12–22. https://doi.org/10.14313/JAMRIS/2-2008/32
14. Zhang X, Zhao Y, Deng N, Guo K (2016) Dynamic path planning algorithm for a mobile robot based on visible space and an improved genetic algorithm. Int J Adv Robot Syst 13(3):1–17. https://doi.org/10.5772/63484
15. Narasimhan GE, Bettyjane J (2020) Implementation and study of a novel approach to control adaptive cooperative robot using fuzzy rules. Int J Inf Technol. https://doi.org/10.1007/s41870-020-00459-z
16. Wang M (2021) Real-time path optimization of mobile robots based on improved genetic algorithm. J Syst Control Eng 235(5):646–651. https://doi.org/10.1177/0959651820952207
17. Teli TA, Wani MA (2021) A fuzzy based local minima avoidance path planning in autonomous robots. Int J Inf Technol 13(1):33–40. https://doi.org/10.1007/s41870-020-00547-0
18. Yasin JN, Mohamed SA, Haghbayan MH, Heikkonen J, Tenhunen H, Plosila J (2021) Low-cost ultrasonic based object detection and collision avoidance method for autonomous robots. Int J Inf Technol 13(1):97–107. https://doi.org/10.1007/s41870-020-00513-w
19. Khan H, Khatoon S, Gaur P (2021) Comparison of various controller design for the speed control of DC motors used in two wheeled mobile robots. Int J Inf Technol 13:713–720. https://doi.org/10.1007/s41870-020-00577-8
20. Jiang M, Fan X, Pei Z, Jiang J, Hu Y, Wang Q (2014) Robot path planning method based on improved genetic algorithm. Sensors & Transducers 166(3):255–260