E-PUCK MOTION CONTROL USING MULTI-OBJECTIVE PARTICLE SWARM OPTIMIZATION

Vikas Singh Panwar – Anish Pandey* – Md. E. Hasan

School of Mechanical Engineering, Campus-8, Kalinga Institute of Industrial Technology (KIIT) Deemed to be University, An Institute of Eminence, Bhubaneswar-751024, Odisha, India

ARTICLE INFO

***Article history*:**
Received: 07.05.2020.
Received in revised form: 18.10.2020.
Accepted: 19.10.2020.

Keywords:
Motion control,
Orientation control,
Multi-Objective Particle Swarm Optimization algorithm,
Virtual robot experimentation platform software, Infra-Red sensors

DOI: https://doi.org/10.30765/er.1647

Abstract:

This article describes the velocity-based motion and orientation control method for a differential-driven two-wheeled E-puck Robot (DDER) using the Multi-Objective Particle Swarm Optimization (MPSO) algorithm in the Virtual Robot Experimentation Platform (V-REP) software environment. The wheel velocities data and Infra-Red (IR) sensors reading make the multi-objective fitness functions for MPSO. We use front, left, and right IR sensors reading and right wheel velocity data to design the first fitness function for MPSO. Similarly, the front, left, and right IR sensors reading, and left wheel velocity data have been taken for making the second fitness function for MPSO. The multi-objective fitness functions of MPSO minimize the motion and orientation of the DDER during navigation. Due to the minimization of motion and orientation, the DDER covers less distance to reach the goal and takes less time. The Two-Dimensional (2D) and Three-Dimensional (3D) navigation results of the DDER among the scattered obstacles have been presented in the V-REP software environment. The comparative analysis with previously developed Invasive Weed Optimization (IWO) algorithm has also been performed to show the effectiveness and efficiency of the proposed MPSO algorithm.

1 Introduction

For the last two-three decades, researchers are giving their tremendous effort to minimize the motion and orientation of a wheeled robot among obstacles in the environment. But in the complex environment, it remains a challenging task yet. Researchers are implementing various types of deterministic and nondeterministic algorithms to address this issue. However, the use of nondeterministic algorithms for motion and orientation optimization of a wheeled robot is increasing in the last few years because these algorithms can quickly minimize the motion and orientation of a wheeled robot through fitness function. Nondeterministic algorithms like ant colony optimization algorithm, genetic algorithm, simulated annealing algorithm, bacterial foraging optimization algorithm, invasive weed optimization, particle swarm optimization algorithm, etc. have been implemented with a single objective fitness function.

These algorithms helped the wheeled robot to search the collision-free navigation path from the starting point to the goal between obstacles in the workspace. Almasri et al. [1] designed the fuzzy controlled sensor fusion based collision-free motion planning for a wheeled robot. Further on, twenty-four fuzzy fi-then rules were used for the robot navigation in the Webots simulation.

Pandey et al. [2] presented the navigation method for an autonomous wheeled robot between static and dynamic hurdles using multiple adaptive neuro-fuzzy inference system architectures. An improved version of a genetic algorithm has been implemented [3] for offline motion planning of a wheeled robot between

* Corresponding author
E-mail address: anish06353@gmail.com

static obstacles. Jhang et al. [4] designed a differential evolution algorithm tuned interval type-2 fuzzy neural controller (IT2FNC) and implemented this technique to make the leader and follower cooperative carrying and wall-following control for multiple wheeled robots. Euclidean distance-based single fitness function genetic algorithm has been presented in the article [5] for wheeled robot path planning.

Another interesting article can be found in [6], where the author applied IWO algorithm in multiple robots to search the optimal path from start to goal points among obstacles. However, the authors did not consider any wheel velocities based multi-objective functions in their work. An extensive review article on the swarm robotics navigation and their various control method was reported [7-8]. Besides, the authors provide a reference for new future research. McGuire et al. [9] presented a route planning architecture of an autonomous wheeled robot using a bug algorithm. In that work, the authors have tested the algorithm in the maze-like simulation environment. However, real-time implementation of that algorithm in a wheeled robot and experiments were not performed. Orozco-Rosas et al. [10] designed the membrane evolutionary artificial potential field method that provided the feasible path for a wheeled robot with a reasonable convergence rate between static and dynamic obstacles. However, Euclidean based single objective function has been used for motion control. Ant colony optimization and wind-driven optimization algorithm tuned fuzzy controller based wheeled robot motion and orientation control technique have been designed in the articles [11] and [12], respectively. The authors implemented some interesting multi-objective fitness-based various optimization algorithms [14-16] to solve the various engineering computation problems.

In our study, we observed that most researchers have used Euclidean distance-based single objective fitness function [3, 5, 6, 10] to search the collision-free and feasible motion and orientation control for a wheeled robot among obstacles in the environment. Nevertheless, for the velocity control based motion control of a differential-driven two-wheeled robot, we have to take multi-objective fitness functions for implementing any algorithm. Therefore, in the present article, we suggested and implemented the right and left wheel velocity based MPSO algorithm that optimizes the motion and orientation of the DDER during navigation from the starting to goal points among scattered obstacles.

The outcomes of navigation results of the DDER among the obstacles have been demonstrated in the V-REP software environment. This work's contributions are as follows: - Section 2 presents the kinematic study of a differential-driven two-wheeled E-puck robot. Section 3 provides a brief description of the MPSO algorithm's design for motion and orientation control of a DDER among obstacles in the V-REP software environment. The outcomes of the experiments and comparative analysis are organized and elaborated in Section 4. Finally, the Conclusion part of the present work is summarized in Section 5.

2 Kinematic Study of the Differential-Driven Two-Wheeled E-puck Robot

In this section, we have investigated the kinematic equations for a differential-driven two-wheeled E-puck robot, which controls the motion and orientation of DDER during navigation and obstacle avoidance in the V-REP software environment. Figure 1 illustrates the schematic representation of a DDER. The DDER is a differential-driven two-wheeled mobile robot, which is developed by École Polytechnique Fédérale de Lausanne.

The diameter and height of the DDER are 7 cm and 5 cm, respectively. The wheel diameter of the DDER is 4 cm; and the total weight of the DDER is 0.16 kg. The DDER can move in the forward direction and backward directions, as well as can turn with a top speed of 0.15 m/s. It includes 8 IR sensors with 8 LEDs. IR sensors of the DDER can read the obstacles between 1 cm to 6 cm range approximately. Figure 2 shows the top view of the DDER with the eight IR sensors' position from S_0 to S_8. In this article, we have used readings from only four sensors: S_0, S_1, S_6, and S_7. The sensor obstacle readings are taken as minima between two neighboring sensors: $D_f = \min(S_0, S_7)$, $D_l = \min(S_6, S_7)$, $D_r = \min(S_0, S_1)$, where indices f, l, r indicate forward, right, and left direction, respectively.

Moreover, the DDER consists of two independent driving wheels, which carry the mechanical chassis of DDER. The two driving wheels attached to the two independent stepper motors drive the DDER.

The V_L represents the left wheel velocity, and V_R indicates the right wheel velocity in the kinematic equations. The DDER moves in the solid plane, and a rigid chassis make it. The axes (x_c, y_c) are the current

posture of the DDER from origin O point in the global frame $\{O, X, Y\}$. θ denotes the orientation angle of the DDER with respect to an axis (O, X), b is the track width between the left and right wheel drive systems, r is the radius of the driving wheels, C is the center of the mass of the DDER system. The following kinematic equations control the velocities and steering angle of the DDER:

$$V_c = \frac{r}{2} \cdot (\omega_R + \omega_L) = \frac{V_R + V_L}{2} \tag{1}$$

$$\omega_c = \dot{\theta} = \frac{r}{b} \cdot (\omega_R - \omega_L) = \frac{V_R - V_L}{b} \tag{2}$$

where V_c and ω_c represent the center (mean) linear velocity and center angular velocity of the DDER, respectively. V_c and ω_c control the motion and orientation of the DDER in the V-REP software environment, respectively. Next, ω_R and ω_L indicate the angular velocities of the right wheel driving system and left wheel driving system, respectively. Additionally, the following equations express the velocity (linear and angular) with respect to time (t):

$$\frac{dx}{dt} = \dot{x}(t) = V_c \cdot \cos\theta = \frac{r}{2} \cdot (\omega_R + \omega_L) \cdot \cos\theta \tag{3}$$

$$\frac{dy}{dt} = \dot{y}(t) = V_c \cdot \sin\theta = \frac{r}{2} \cdot (\omega_R + \omega_L) \cdot \sin\theta \tag{4}$$

$$\frac{d\theta}{dt} = \dot{\theta}(t) = \omega_c = \frac{r}{b} \cdot (\omega_R - \omega_L) \tag{5}$$

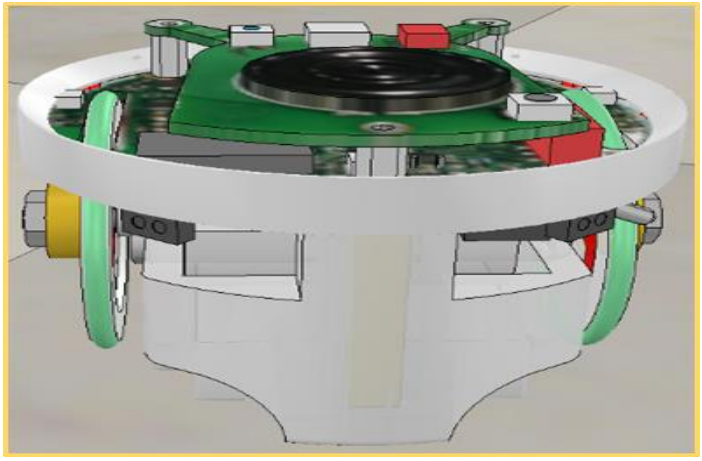

Figure 1. Differential-driven two-wheeled E-puck Robot (DDER).

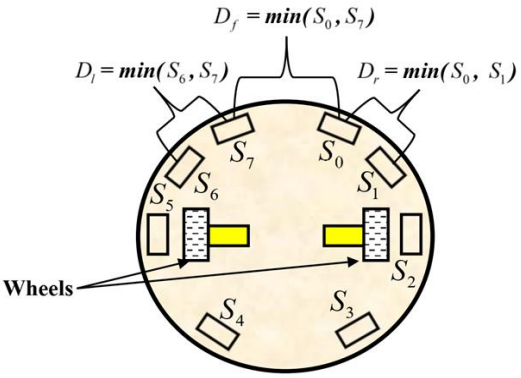

Figure 2. Top view of the DDER with the position of eight IR sensors.

3 Design of MPSO algorithm for Motion and Orientation Control of DDER Among Obstacles in the V-REP Software Environment

This section describes the brief description of the proposed MPSO algorithm and design of multi-objective fitness functions, which controls and minimizes the motion and orientation of the DDER during navigation among obstacles in the V-REP software environment.

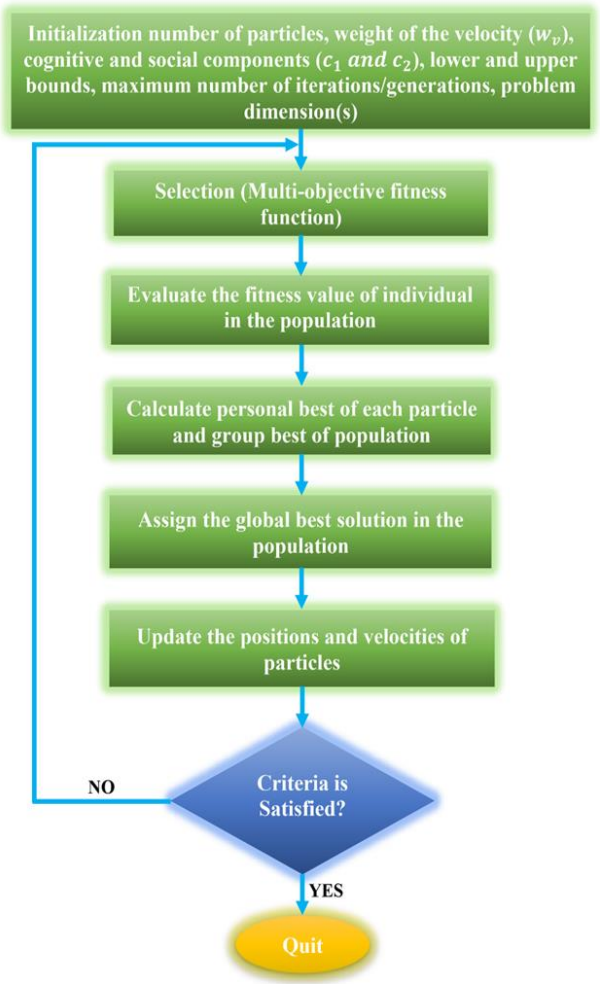

Figure 3. General flowchart of the MPSO algorithm.

3.1 Brief Description of the MPSO Algorithm

MPSO algorithm is a swarm-based nondeterministic or stochastic optimization algorithm, developed by Kennedy and Eberhart [13]. The social behavior of fish schooling and bird flocking inspires this algorithm. Figure 3 shows the general flow chart of the MPSO algorithm, which combines different mechanisms of the algorithm. Mainly MPSO algorithm can be categorized in the four steps: initialization, selection of fitness function, velocity, and position up-gradation of particles. In the initialization process, we choose particles' size, the w_v weight of the velocity, c_1 cognitive and c_2 social components, lower and upper bounds, a maximum number of iterations/generations, and problem dimension(s).

After the initialization process, we put the fitness function of the problem in algorithm's code. This study designs and selects the multi-objective fitness function for the MPSO algorithm. We used front, left, and right IR sensors reading as decision variables with the right and left wheel velocity data as predictors to design the first and second fitness functions, respectively.

Further, the generalized multi-objective fitness function for the MPSO algorithm can be written as $f(\bar{x}) = \left[f_1(\bar{x}),\ f_2(\bar{x}),\ldots,\ f_i(\bar{x}) \right]^T$, where $\bar{x} = \left(x_1, x_2, \ldots, x_m \right)$ indicates the position of particles. The next steps of the algorithms are velocity and position up-gradation of particles. All the particles move with some velocity in the search space, and particles have their position in search space. The position $\bar{x}$ and velocity $\bar{v}$ of the population of the n particles can be expressed as:

$$\bar{x}_n(j) = (x_{n,1}(j),\ x_{n,2}(j),\ldots,x_{n,m}(j)), \tag{6}$$

$$\bar{v}_n(j) = (v_{n,1}(j),\ v_{n,2}(j),\ldots,v_{n,m}(j)), \tag{7}$$

where n and j denote the number of particles and number of iterations, respectively, after completion of one iteration, i.e., $j+1$, the velocity and position of particles are updated by the following equations:

$$\begin{aligned} v_{n,m}(j+1) &= w_v \cdot v_{n,m}(j) + c_1 \cdot r \cdot \left(x_{pbest,\,n,m}(j) - x_{n,m}(j) \right) \ldots \\ &+ c_2 \cdot r \cdot \left(x_{gbest,\,n,m}(j) - x_{n,m}(j) \right), \end{aligned} \tag{8}$$

$$x_{n,m}(j+1) = x_{n,m}(j) + v_{n,m}(j+1), \tag{9}$$

where $v_{n,m}(j)$ and $x_{n,m}(j)$ represent the velocity and position of the n particles at j number of iteration, the w_v controls the weight of the velocity $w_v \in \{0,\ 1.2\}$. $x_{pbest,\,n,m}(j)$ denotes the individual best position of the n particles at j number of iteration. Similarly, the $x_{gbest,\,n,m}(j)$ is called the global best position, which is selected from the group of particles' population. $c_1 \in \{0,\ 2\}$ and $c_2 \in \{0,\ 2\}$ are known as cognitive and social components, respectively, that tune the level of influence of $x_{pbest,\,n,m}(j)$ and $x_{gbest,\,n,m}(j)$. The r generates the random number in the range $(0,\ 1)$ to run the algorithm.

3.2 Design of multi-objective fitness functions for the MPSO algorithm

In this section, the multi-objective fitness functions are designed to control the velocity of DDER. We have taken the senor data as inputs and velocity data as outputs in the Minitab software, and we have applied the general regression method to obtain multi-objective fitness functions. The multi-objective fitness functions for V_R and V_L to minimize the motion and orientation of the DDER are given below:

$$V_R = 0.0578646 + 0.906727 D_f \ldots + 0.881006 D_l - 0.562324 D_r \quad (10)$$

$$V_L = 0.0489124 + 0.934448 D_f \ldots - 0.11076 D_l + 1.0997 D_r \quad (11)$$

The ranges of lower and upper bounds are $0.01 \le D_f, D_l, D_r \le 0.06$ of decision variables in the meter for the multi-objective fitness functions of the MPSO algorithm. Equations 10 and 11 reveal the multi-objective fitness functions of the algorithm. These functions control and minimize the motion and orientation of the DDER during navigation among scattered obstacles in the V-REP software environment. Due to minimization of motion and orientation, the DDER covers less distance to reach the goal and also takes less time. The V_R and V_L ranges are fixed between 0.063 m/sec to 0.15 m/sec. Figure 4 shows the Pareto optimality graph of multi-objective fitness functions of V_R and V_L, considered minimization problems. Table 1 illustrates the selected values of the MPSO algorithm's parameters, used in experiments.

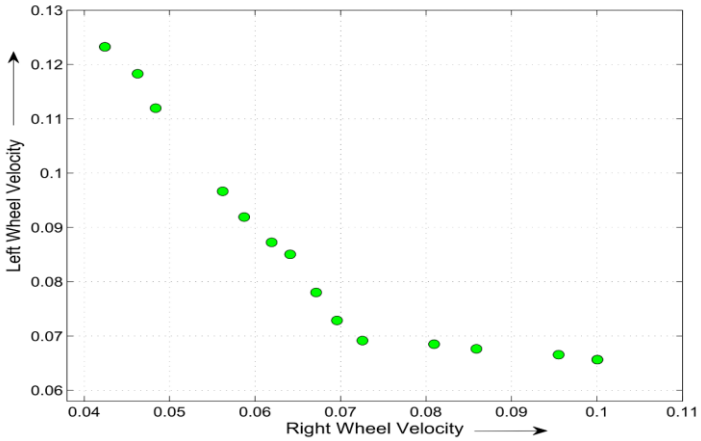

Figure 4. Pareto optimality graph of multi-objective fitness functions of V_R (Right Wheel Velocity) and V_L (Left Wheel Velocity).

Table 1. Selected parameters of the MPSO algorithm in this study and their values.

Parameter	Value
Maximum number of iterations	1000
Population size of particles	100
Cognitive component c_1	1.5
Social component c_2	1.5
Problem dimension(s)	3
Weight of the velocity w_v	1
Fitness function	Multi-objective
Lower bounds of decision variables	[0.01, 0.01, 0.01]
Upper bounds of decision variables	[0.06, 0.06, 0.06]

4 Experimental Results of DDER Among Three-Dimensional Scattered Obstacles and Comparison with Previously Developed Invasive Weed Optimization in the Two-Dimensional Scenario

This section presents the experimental results of DDER among obstacles in the V-REP software environment using the MPSO algorithm, and comparison with previously developed IWO algorithm [6] in the 2D scenario.

4.1 Experimental Results of DDER Among Three-Dimensional Scattered Obstacles Using MPSO algorithm

V-REP software-based 3D motion and orientation results of the DDER have been presented in this section using the MPSO algorithm. The kinematic equations and MPSO algorithm are mentioned in the form of a MATLAB script (m-file). The programming script uses the different parameters for the MPSO algorithm and their values from Table 1. The 3D environment and simulation result of DDER among scattered obstacles was designed and performed in the V-REP software. The remote Application Program Interface (API) functions are used to connect the MATLAB and V-REP software. After establishing a connection, we run the MATLAB script, and simultaneously we start the V-REP simulation. The script sends the navigation command to the DDER in the 3D V-REP software environment, and the MPSO algorithm controls the right and left wheel velocities by receiving the D_f, D_l, and D_r readings.

Furthermore, Figure 5 reveals 3D motion and orientation result of DDER among scattered obstacles in the V-REP software environment using the MPSO algorithm. In the Figure 5, the DDER starts motion from (10 cm, 10 cm), and reaches the goal that is placed at coordinates (200 cm, 200 cm). To show the motion and orientation control of DDER, blue color cuboid and red color cylindrical obstacles are randomly placed in the environment. At first, the DDER starts to move directly towards the goal until the obstacles are detected within the specified sensory range. Then MPSO is activated and sends the velocity control command to DDER to avoid the obstacles. Figure 6 shows the recorded angular velocities (degree/sec) of the right wheel (red color) and left wheel (green color) of DDER during navigation among scattered obstacles in Figure 5 (V-REP software environment).

Similarly, Figure 7 shows the recorded linear velocities (meter/sec) of the right wheel (blue color) and left wheel (cyan color) of DDER during navigation among scattered obstacles in Figure 5 (V-REP software environment). As we can see in Figure 5, the goal (purple color small cuboid) is placed at the right corner, and due to this, most of the time, the DDER takes a right turn to reach the goal. Therefore, in Figures 6 and 7, the left wheel's linear and angular velocity is high as compared to the right wheel. The DDER covers the distance of 125 cm to reach the goal from the starting point between scattered obstacles and takes 30 seconds.

Table 2. Navigation path length and time comparison data between the proposed MPSO algorithm and a previously developed IWO algorithm [6].

Name of Algorithm	Figure Number	Start Point (cm)	Goal (cm)	Navigation Path Length	Navigation Time	Navigation Path Length Error
				cm	Sec	cm
MPSO	Figure 8	(10, 10)	(200, 200)	125	30 s	2.46
IWO [6]	Figure 8	(10, 10)	(200, 200)	133	32 s	3.94

4.2 Comparison with Previously Developed Invasive Weed Optimization

This sub-section shows the comparative analysis between the proposed multi-objective fitness function based MPSO algorithm and previously developed Euclidean distance-based IWO algorithm [6] in the same 2D scenario between scattered obstacles. In the article [6], the authors implemented the IWO algorithm for optimal trajectory planning of Khepera-II wheeled robot in known and unknown environments. Figure 8 shows the 2D motion and orientation comparison result between the MPSO algorithm and IWO algorithm among scattered obstacles in the same 2D scenario. In Figure 8, the yellow gold color path reveals the MPSO algorithm controlled motion and orientation results. Similarly, the green pickle color trajectory presents the IWO algorithm-driven navigation results of the DDER. As we can see in Figure 8, the MPSO algorithm gives a smooth trajectory, and the IWO algorithm provides a rough path because the MPSO algorithm directly controls the wheel velocities of the DDER, and for the IWO algorithm, authors take the Euclidean distance based fitness function that searches the random coordinate during navigation.

Besides, Table 2 provides the path length and time comparison data between the proposed MPSO algorithm and a previously developed IWO algorithm. The MPSO algorithm running DDER takes 125 cm to reach the goal from the start with the navigation path length error of 2.46 cm, while the IWO algorithm running DDER covers 133 cm to reach the goal from the start with the navigation path length error of 3.94 cm. Therefore, the comparative analysis reveals that the velocity controlled MPSO algorithm gives a reasonable convergence rate with less path length error than the previously developed IWO algorithm.

5 Conclusion

We presented the optimization of motion and orientation of DDER through wheel velocities based multi-objective fitness functions using the MPSO algorithm in the V-REP software environment. The designed MPSO algorithm consists of two multi-objective fitness functions. The first and second fitness functions controlled and optimized the right wheel velocity and left wheel velocity of DDER. Due to the minimization of wheel velocities, the DDER covers less distance to reach the goal from the starting point among the scattered obstacles.

The programming of the MPSO algorithm and kinematic equations have been written in MATLAB scripts. Through the remote API functions, these scripts give the motion and orientation controlling command to the DDER in the V-REP software environment among scattered obstacles. The obtained 2D and 3D navigation results show that the multi-objection fitness functions of the MPSO algorithm successfully controls and minimizes the velocities of DDER in the tested environment. The comparative analysis in Section 4 reveals that the MPSO algorithm controlled robot covers 125 cm in 30 seconds to reach the goal from the starting point. Whereas the IWO benchmark algorithm controlled robot takes 133 cm in 32 seconds. Therefore, it can be concluded that the MPSO algorithm is more efficient than the IWO algorithm. Future work can include these multi-objective fitness functions for other non-deterministic optimization algorithms to search for a more accurate path and better convergence rate.

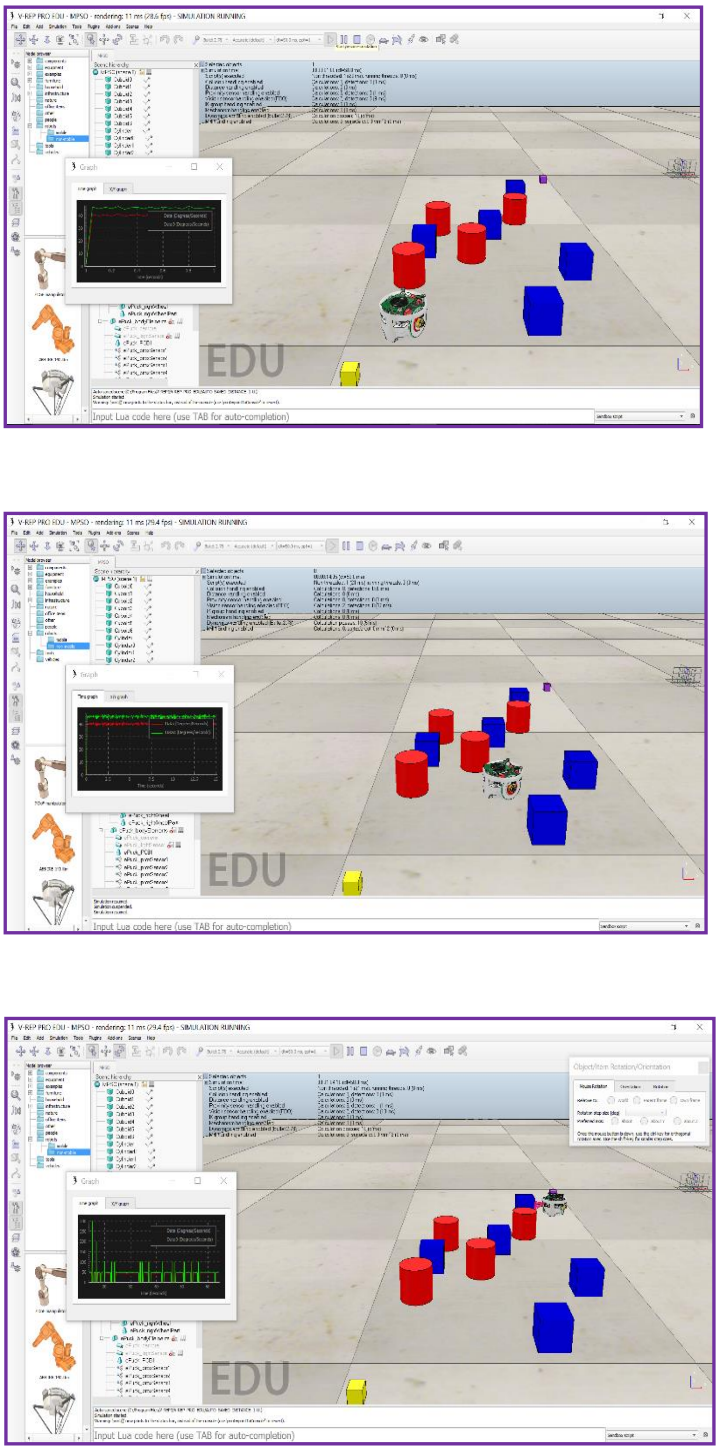

Figure 5. 3D motion and orientation results of DDER among scattered obstacles in the V-REP software environment using the MPSO algorithm.

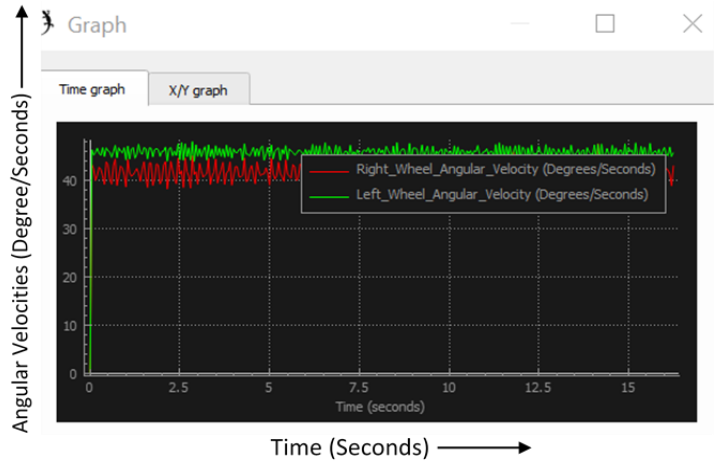

Figure 6. Recorded angular velocities (degree/sec) of a right wheel (red color) and left wheel (green color) of DDER during navigation among scattered obstacles in Figure 5 (V-REP software environment).

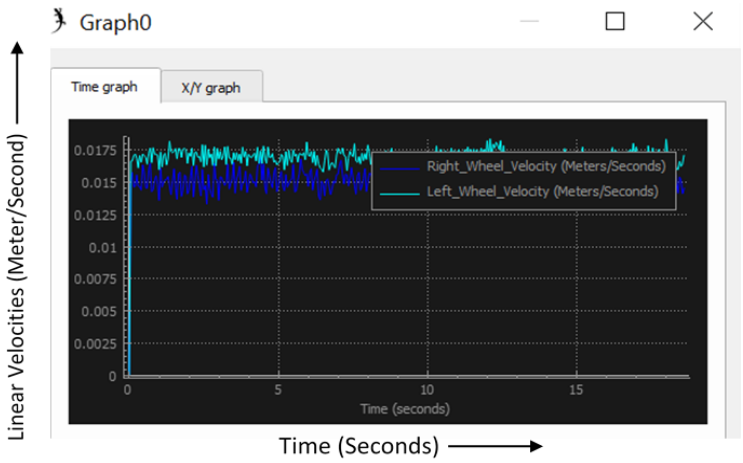

Figure 7. Recorded linear velocities (meter/sec) of a right wheel (blue color) and left wheel (cyan color) of DDER during navigation among scattered obstacles in Figure 5 (V-REP software environment).

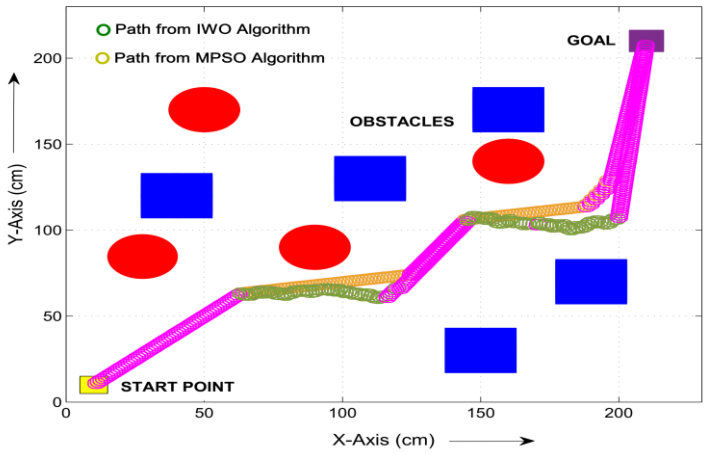

Figure 8. 2D motion and orientation comparison result between proposed MPSO algorithm and previously developed IWO algorithm [6] among scattered obstacles in the same scenario.

References

[1] Almasri, M., Elleithy, K., Alajlan, A.: *Sensor fusion based model for collision free mobile robot navigation*, Sensors, 16 (2016), 1, 1-24.
[2] Pandey, A., Kashyap, A. K., Parhi, D. R., Patle, B. K.: *Autonomous mobile robot navigation between static and dynamic obstacles using multiple ANFIS architecture*, World Journal of Engineering, 16 (2019), 2, 275-286.
[3] Jiang, M., Fan, X., Pei, Z., Jiang, J., Hu, Y., Wang, Q.: *Robot path planning method based on improved genetic algorithm*, Sensors & Transducers, 166 (2014), 3, 255-260.
[4] Jhang, J. Y., Lin, C. J., & Young, K. Y.: *Cooperative Carrying Control for Multi-Evolutionary Mobile Robots in Unknown Environments*, Electronics, 8 (2019), 3, 1-21.
[5] Song, B., Wang, Z., Sheng, L.: *A new genetic algorithm approach to smooth path planning for mobile robots*, Assembly Automation, 36 (2016), 2, 138-145.
[6] Panda, M. R., Das, P. K., Dutta, S., Pradhan, S. K.: *Optimal path planning for mobile robots using oppositional invasive weed optimization*, Computational Intelligence, 34 (2018), 4, 1072-1100.
[7] Osaba, E., Del Ser, J., Iglesias, A., Yang, X. S.: *Soft Computing for Swarm Robotics: New Trends and Applications*, Journal of Computational Science, 39 (2019), 1-4.
[8] Nedjah, N., Junior, L. S.: *Review of methodologies and tasks in swarm robotics towards standardization*, Swarm and Evolutionary Computation, 50 (2019), 1-26.
[9] McGuire, K. N., de Croon, G. C. H. E., Tuyls, K.: *A comparative study of bug algorithms for robot navigation*, Robotics and Autonomous Systems, 121 (2019), 1-17.
[10] Orozco-Rosas, U., Montiel, O., Sepúlveda, R.: *Mobile robot path planning using membrane evolutionary artificial potential field*, Applied Soft Computing, 77 (2019), 236-251.
[11] Castillo, O., Neyoy, H., Soria, J., Melin, P., Valdez, F. *A new approach for dynamic fuzzy logic parameter tuning in ant colony optimization and its application in fuzzy control of a mobile robot*, Applied soft computing, 28 (2015), 150-159.
[12] Pandey, A., Parhi, D. R.: *Optimum path planning of mobile robot in unknown static and dynamic environments using Fuzzy-Wind Driven Optimization algorithm*, Defence Technology, 13 (2017), 1, 47-58.
[13] Kennedy, J., Eberhart, R.: *Particle swarm optimization*. Proceedings of ICNN'95-International Conference on Neural Networks, Perth, WA, Australia, Australia, 1995, 1942-1948.
[14] Xie, N. G., Chen, Z., Cheong, K. H., Meng, R., Bao, W.: *A multiobjective game approach with a preferred target based on a leader-follower decision pattern*, Mathematical Problems in Engineering, 18 (2018), 1-12.
[15] Wang, C., Koh, J. M., Yu, T., Xie, N. G., & Cheong, K. H.: *Material and shape optimization of bi-directional functionally graded plates by GIGA and an improved multi-objective particle swarm optimization algorithm*. Computer Methods in Applied Mechanics and Engineering, 366 (2020), 1-25.
[16] Meng, R., Cheong, K. H., Bao, W., Wong, K. K. L., Wang, L., & Xie, N. G.: *Multi-objective optimization of an arch dam shape under static loads using an evolutionary game method*, Engineering Optimization, 50(2018), 1061-1077.