DOI: https://doi.org/10.33103/uot.ijccce.23.3.13

Real-Time Multi-Agent Mobile Robot based on A Client-Server Model

Ammar Abdul Ameer Rasheed[1], Ahmed Sabah Al-Araji[2], Mohammed Najm Abdullah[3], Hamed S Al-Raweshidy[4]

[1,2,3]*Computer Engineering Department, University of Technology, Baghdad, Iraq,*
[4]*Brunel University, London, UK*
[1]*Ammar.A.Rasheed@uotechnology.edu.iq,* [2]*Ahmed.s.alaraji@uotechnology.edu.iq,*
[3]*mohammed.n.abdullah@uotechnology.edu.iq,* [4]*hamed.al-raweshidy@brunel.ac.uk*

Abstract— *A client-server network is one of the most important topics in a computer network. In this work, a real-time computer control system is designed based on the Client-Server Model for a Multi-Agent Mobile Robot System (CSM-MAMRS) and is applied to a building that consists of (N) floors and uses one mobile robot on each floor that has specific actions in different types of environments. The new proposed CSM-MAMRS consists of four layers. The first stage manages the network communication for each agent. The second stage solves the major problems of path planning by using a proposed hybrid algorithm that combines the Rapidly Exploring Random Tree Star (RRT*) and the Particle Swarm Optimization (PSO) algorithms in order to provide the shortest and smoothest path with collision avoidance between the starting and the target points in a static and dynamic robot environments are used.. In the third stage, a velocity planner controller is based on an inverse kinematic mobile robot model. In the fourth stage, the Hypertext Transfer Protocol HTTP is used to send velocity values to real mobile robots via the Wireless Network Control Administration. The simulation results and experimental works successfully achieved to significant improvement in real-time when using three missions of three mobile robots on different static map floors in the building. The maximum tracking pose errors for three robots in the static enviormnets are 0.39 cm, 0.02 cm, 0.32 cm respectively, but the maximum tracking pose error in the dynamic environments are 2.94 cm and 2.8 cm only for two mobile robots along the maximum distance of 250 cm.*

Index Terms— *Real-time computer control, Mobile robot, Multi-agent system, Path planning algorithms.*

I. INTRODUCTION

The client-server model refers to an application's connection between interdependent programs, where clients request services and servers fulfill those requests. The client-server model is often known as the client-server [1,2]. The client-server model is one of the most subjects in the world and many studies and research discussed this subject in different applications filed, so here listing of Previous Works. E. Oussama proposed dynamic adaptive streaming (DASH) over an HTTP [3]. An experimental teaching model was created in conjunction with virtual reality, remote control, and an on-site lab to support the teaching of robotic kinematics [4]. Multiple cooperative robots that can move physically and must interact physically make up a multi-robot system (MRS). Individual nodes in multi-agent systems (MAS) remain stationary while the group of computers cooperates to perform a task. The agent in the MAS is software as opposed to a robot in the MRS. [5]. The MAS has including multiple applications, for example, communication networks, open-source software, machine learning, motion coordination, electronic learning, and an

Received 01/February/2023; Accepted 20/March/2023
© 2017 University of Technology, Iraq ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.3.13

unknown environment. T. Samad et al. developed and created a multi-agent framework for the intelligent cloud-based management of cooperative flight drones and quadcopters so that they can conduct a fast, yet efficient survey of the disaster site following an earthquake for which detection systems are still ineffective [6]. H. Du et al. presented a novel family of resolved consensus algorithms based solely on local knowledge, based on building fixed-time observers as well as fixed-time controllers consensus to solve the problems for a type of heterogeneous nonlinear multi-agent dynamical systems.[7]. K. E. Ehimwenma and S. Krishnamoorthy introduced the modeling (e-learning system) of a pre-assessment system with five interactive agents using the Prometheus Agent unified modeling languages (AUML) methodology [8]. E. Jones introduced a multi-agent framework open-source software. It was used to implement algorithms for multi-agent applications. [9]. Mobile robots are becoming more and more common in a variety of industries. A. Singhal et al**.** introduced a fleet management system for a community of autonomous mobile robots (AMR) based on cloud robotics platforms, and it was utilized in three configurations: single-master, multi-master, and cloud robotics platforms. [10]. D. G. Sáenz introduced a learning environment useful for educational systems that employ mobile robots, where the learning environment was made up of several inexpensive robots and a method for accurately localizing such robots using vision-based data gathering. [11]. C. F. Eduardo provided a framework of robotic swarm systems that combined blockchain technologies and swarm robotic systems, providing the capabilities needed to render robotic swarm operations more protected, autonomous, scalable, and even productive, by utilizing the robots as nodes within the network and enveloping their transactions in chains [12,13]. Path planning is any process of choosing an unobstructed path from the initial state (the start) to the final state (the goal) in order to keep the path safe and avoid obstacle collisions. The mobile robot navigation field considers path planning a crucial component of autonomous mobile robots and an important topic. [14,15,16]. Nevertheless, Many challenges still exist, including the robot's location, the creation of the environment map, the algorithm for path planning (which must take into account the characteristics and limitations of robot motion), and the paths (which should be designed more effectively and easily) [17]. The Q-learning algorithm is one of the most important methods in the field of reinforcement learning-based path planning. [18,19]. Numerous academics employed hybrid algorithms to improve the path-planning algorithm, such as the authors of [20,21]. W. Xiaolu et al. developed a global path-planning technique based on enhanced particle swarm optimization (PSO). The standard PSO and the A* method were combined in the enhanced path-planning algorithm to make up for the standard PSO's poor convergence rate [22]. The problem definition considers three points: the challenges of path planning with free navigation (short distance, smoothness, and free collusion) for each mobile robot in the global and local environment; the limitation of fast collecting and processing the data for MAMRS in real-time; the reduction of trajectory tracking error for each agent (mobile robot) to follow the desired path.

The Contribution of this paper is a new MAMRS creation based on a client-server model with a real-time computer control system leading to the successful mission of each mobile robot on different floors in the building with different types of environments (static and dynamic) through monitoring and motion controlling in real-time.

This paper is organized as follows: Section II describes kinematics models of differential wheeled mobile robots. Section III is the client-server model design. Section IV experimental works and the conclusions are given in section V.

Received 01/February/2023; Accepted 20/March/2023
© 2017 University of Technology, Iraq ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.3.13

II. KINEMATICS MODELS OF DIFFERENTIAL WHEELED MOBILE ROBOTS

The Wheeled Mobility Robot platform contains a multidirectional wheel known as a castor wheel fitted to the front or back of the platform of the Wheeled Mobility Robot, which has two wheels attached to a parabolic shaft for mobility and orientation. The castor wheel's function is to maintain the cart's stability when the mobile robot moves in various directions. According to *Fig. 1*, this platform of the Wheeled Mobility Robot is a fixed structure with wheels that functions on a horizontal plane. [23].

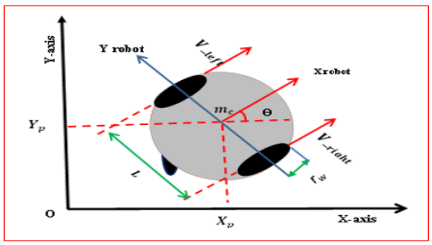


FIG. 1. STRUCTURE OF MOBILE ROBOT PLATFORM.

In addition, the mobile robot platform is made up of two DC motors that move the right wheel and left wheel, (m_c) is the mass center of the mobile robot, (***L***) is the distance between the right wheel and left wheel, and (r) the radius of the wheels. [0, X-axis, and Y-axis] are used to represent the world coordinate system. The coordinates of the point(mc) are Xp and Yp, (m_c) represents the mobile robot's direction angle as seen from the X-axis, and Ts stands for the sampling time. The configuration of the mobile robot is determined by the global coordinates. the equations of computer simulation equations are as follows: [24,25] where ks is a step of the movement of the mobile robot.

$$X_p(ks) = 0.5\left[\left(V_{\_left} + V_{\_right}\right) \times \cos\left(\theta(ks)\right) \times Ts\right] + X_p(ks-1) \quad (1)$$

$$Y_p(ks) = 0.5\left[\left(V_{\_left} + V_{\_right}\right) \times \sin\left(\theta(ks)\right) \times Ts\right] + Y_p(ks-1) \quad (2)$$

$$\theta(ks) = 0.5\left[\left(V_{\_left} + V_{\_right}\right) Ts\right] + \theta(ks-1) \quad (3)$$

III. THE CLIENT-SERVER MODELDESIGN

This section describes the proposed Client-Server Model based on Multi-Agent Mobile Robot Systems (CSM-MAMRS) in real-time applications, as shown in *Fig. 2*. It consists of (N) mobile robots distributed in a building with (N) floors. According to the maps of the various types of environments, each mobile robot has a specific mission on each floor. The process of data management and control for each mobile robot (agent) is built into the main server to successfully complete the mission of each mobile robot. *Fig. 3* shows the proposed CSM-MAMRS divided into two sides.

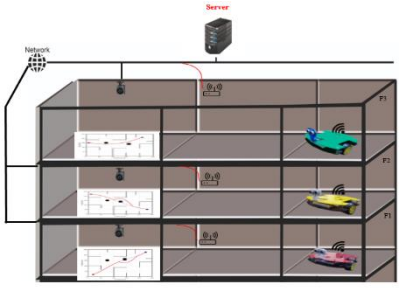


FIG. 2. THE CLIENT-SERVER MODEL PROPOSED IN THE BUILDING.

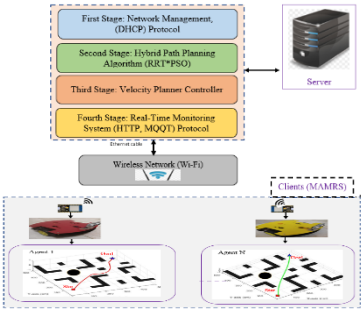


FIG. 3. THE PROPOSED CSM-MAMRS LAYERS.

Received 01/February/2023; Accepted 20/March/2023
© 2017 University of Technology, Iraq
ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.3.13

The first side is the server, which consists of four stage as follows: network management (DHCP) protocol, hybrid (RRT*PSO) path planning algorithm, velocity planner controller, and real-time monitoring system (HTTP, MQQT) protocol. The second layer is the client, which consists of the MAMRS and wireless network.

First Stage: (network management and DHCP), the main jobs of this layer can be described as follows: Firstly, the management of computer networks that are made up of several different types of hardware included in the proposed system, such as a mobile robot, an IP camera, and a wireless network. Secondly, the server supports the IP address of each device in the network by using the Dynamic Host Configuration Protocol (DHCP).

Second Stage: hybrid (RRT*PSO) path planning algorithm, the main goal of this layer is to create the best path between the starting node and the goal node. The proposed hybrid path-planning algorithm is based on the RRT* and PSO algorithms [26]. *Fig. 4* shows the flowchart of the proposed hybrid RRT*PSO algorithm for the generated optimal or near-optimal desired path of the mobile robot. This hybrid method was used in static and dynamic environments to achieve the smoothest and shortest path for the mobile robot and to avoid obstacles by reducing the number of iterations, function of evaluation, and processor unit execution time during the production of the desired path with free-navigation.

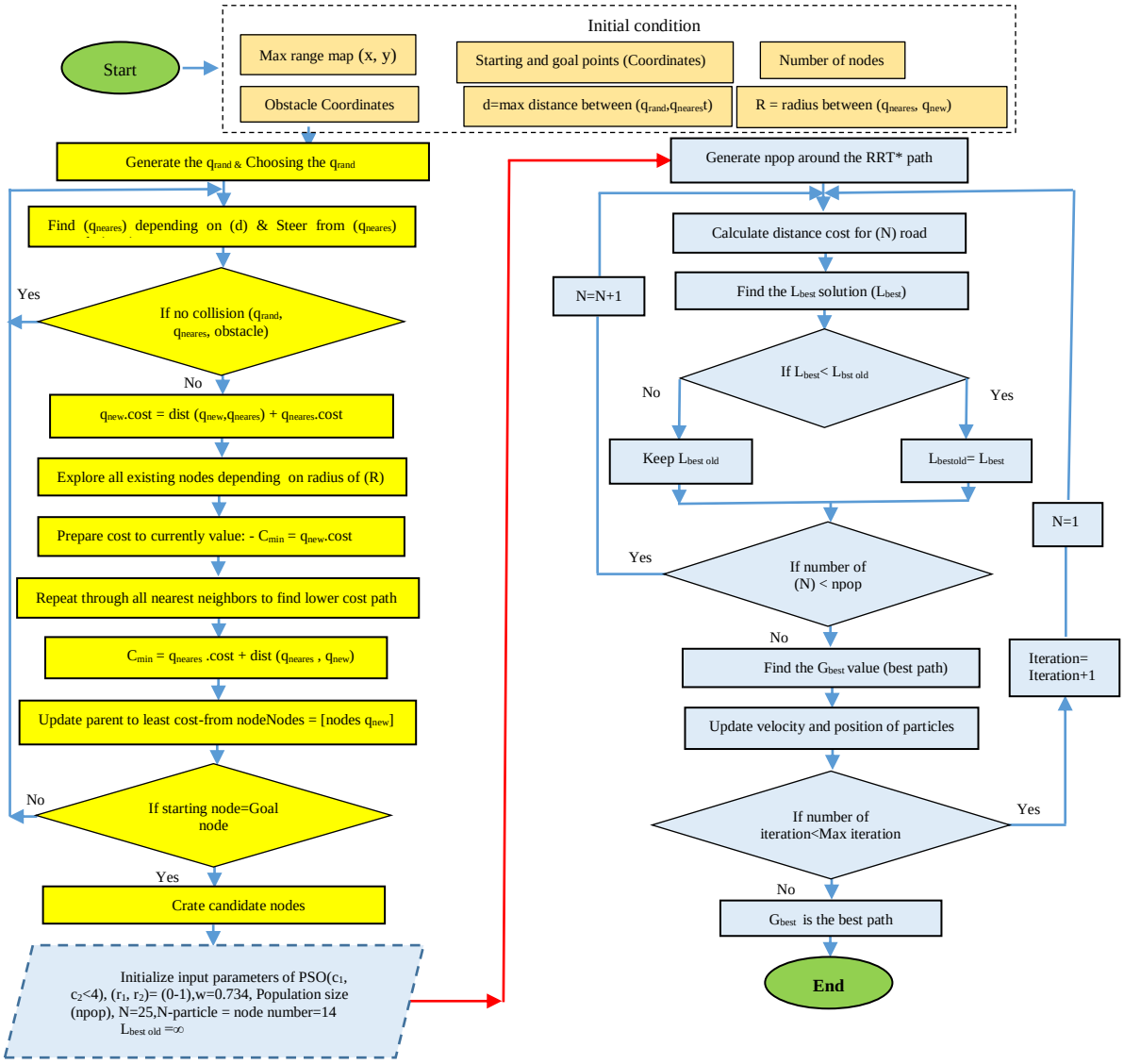


FIG. 4. FLOWCHART OF THE RRT*PSO HYBRID ALGORITHM.

Received 01/February/2023; Accepted 20/March/2023
© 2017 University of Technology, Iraq
ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.3.13

Third Stage: Velocity Planner Controller, this layer presents a velocity planner controller design that is based on the inverse kinematics model of the mobile robot. The input of the controller is the reference path equation of the mobile robot that was obtained from the 3rd-order polynomial equation that described the desired path based on the hybrid RRT*PSO algorithm. The output of the controller is the reference velocities of the platform of the mobile robot, where the right and left wheel linear velocities are calculated in order to track the desired path equation with a minimum tracking error in the pose. The reference linear velocity for the platform of a mobile robot can be calculated using equation (1), which depends on knowing the velocity of moving the cart in the direction of the x and y axes.

$$v_{\_ref(k+1)} = \sqrt{\left(\dot{x}_{\_ref(k+1)}\right)^2 + \left(\dot{y}_{\_ref(k+1)}\right)^2} \quad (4)$$

Where $v_{\_ref}$ is the reference linear velocity, $\dot{x}$ the reference velocity of the mobile robot for the x-axis,$\dot{y}$ is the reference velocity of the mobile robot for the y-axis.

$$w_{\_ref} = \frac{(\dot{y}_{\_ref} \times \dot{x}_{ref} - \ddot{x}_{\_ref} \times \dot{y}_{\_ref})}{\left((\dot{x}_{\_ref})^2 + (\dot{y}_{\_ref})^2\right)} \quad (5)$$

Where $w_{\_ref}$ is The reference angular velocity,$\ddot{x}$ is The reference accelerations of the mobile robot for the x-axis $\ddot{y}$ is The reference accelerations of the mobile robot for the y-axis. The proposed velocity control laws of the right and left wheels are shown in Equations (3) and (4), respectively.

$$v_{\_right} = \frac{(2v_{\_ref} + Lw_{\_ref})}{2} \quad (6)$$

Where $v_{\_right}$ is The linear velocities of the right wheel.

$$v_{\_left} = \frac{(2v_{\_ref} - Lw_{\_ref})}{2} \quad (7)$$

Where $v_{\_left}$ is The linear velocities of the right wheel.

Fig. 5 shows the block diagram of the proposed velocity planner controller.

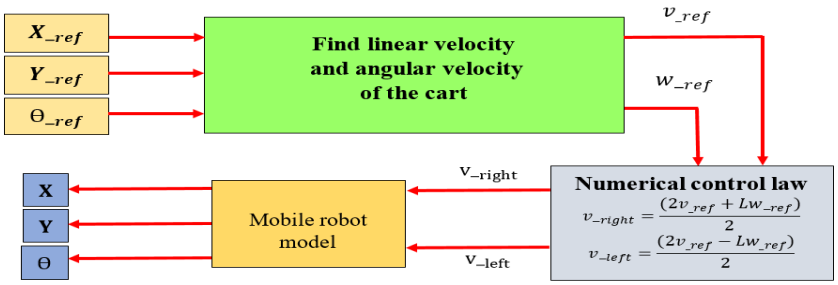


FIG. 5. VELOCITY PLANNER CONTROLLER.

Fourth Stage: Real-Time Data Management (HTTP and MQTT protocol), the first main target of this layer is to employ the Hypertext Transfer Protocol (HTTP) to create the application (webserver) to transmit the data between the server and client (agent), the web server created using the Hyper Text Markup Language (HTML). The administration used this webserver to control all agents (mobile robots), the administration sent the amplitude of the velocity which has been converted into period time (T_{on}) to the agent depending on the proposed equation (15).

$$T_{on} = \frac{T_{pwm} \times Liner\ velocity\ of\ wheel}{maximum\ velocity\ of\ the\ wheel} \quad (8)$$

Where T_{on} denotes the time duration of the DC motor's operation. T_{pwm} is the time of pulse width modulation and the maximum linear velocity of the wheel is (0.52 m/sec). The second main target of

Received 01/February/2023; Accepted 20/March/2023
© 2017 University of Technology, Iraq
ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.3.13

this layer is to use the Message Queuing Telemetry Transport protocol (MQQT) to send the read distance of the ultrasonic sensor from the mobile robot to the server. Multi-Agent Mobile Robot System (MAMRS): The proposed system at this stage depends on the number of mobile robots that are used and distributed in the building. The agent is defined as a mobile robot, which is a two-wheeled differential mobile robot type. A wireless network uses a wireless router device that provides Wi-Fi, it sends information from the server to the client (a mobile robot). Moreover, the server administrator can use the IPC to monitor the client in the workspace environment. The control administration interface (CAI) is a set of visible, interactive components for computer tools and software. A CAI presents information-conveying and action-representative items for the administration to interact with the alter their color, size, or visibility. The CAI is created by using HTML, cascading style sheets (CSS), and JavaScript, the admin of the server used it to view the information that comes from the agent through the HTTP. The CAI consists of the following, firstly the start button is presented to start the agent in the job (from starting node to goal node), secondly stop button is used to stop immediately, and thirdly speed meter to the right and left wheel. The second part is the wireless network that used Wi-Fi to transition data between the server and agents, any agent who can access the server via the wireless router. The admin of the server can access any agent by using CAI through the IP address. In addition, the wireless network used Wi-Fi to data transition between the server and agents. Finally, the CAI presents a real-time computing computer controller between the client (agent) and server, *Fig. 6* shows the CAI for CSM-MAMRSs.

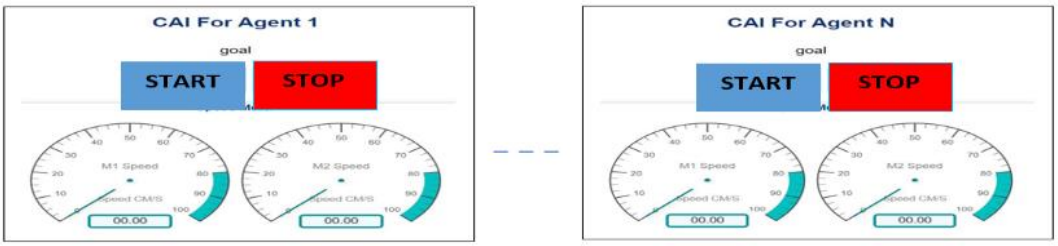


FIG. 6. THE CAI FOR CSM-MAMRSS.

IV. EXPERIMANTALS WORKS

Using three real mobile robots constructed for two scenarios depending on the workspace's static and dynamic environments experiments were conducted to verify the feasibility of the proposed methodology. To implement the real mobile robot motion in static and dynamic environments in real-time and apply the proposed algorithm on different floors in the building, it is used another map that has a dimension of the workspace [250×250] cm for each floor as shown in *Fig. 7*.

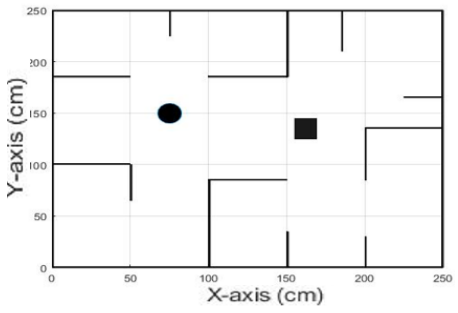


FIG. 7. THE PROPOSED WORKSPACE ENVIRONMENT WITH OBSTACLES.

Received 01/February/2023; Accepted 20/March/2023
© 2017 University of Technology, Iraq
ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.3.13

The first real mobile robot (red color) is applied on the first floor, the first scenario in a static environment the position information of each obstacle in the workspace is provided. The proposed RRT*PSO hybrid algorithm is used to find a collision-free path, the start node position of the mobile robot is at (20, 20) cm, and the goal node position is at (225, 225) cm. After applying the RRT*PSO hybrid algorithm and the input candidate node equal 12 nodes, to find the smooth and optimal shortest path as shown in *Fig. 8a*. The distance of the optimal path to have the best distance cost function was found in iteration number 50, as shown in *Fig. 8b* with a maximum iterations number equal to 100 iterations. The value of the hybrid algorithm optimal path distance is equal to (292.7 cm).

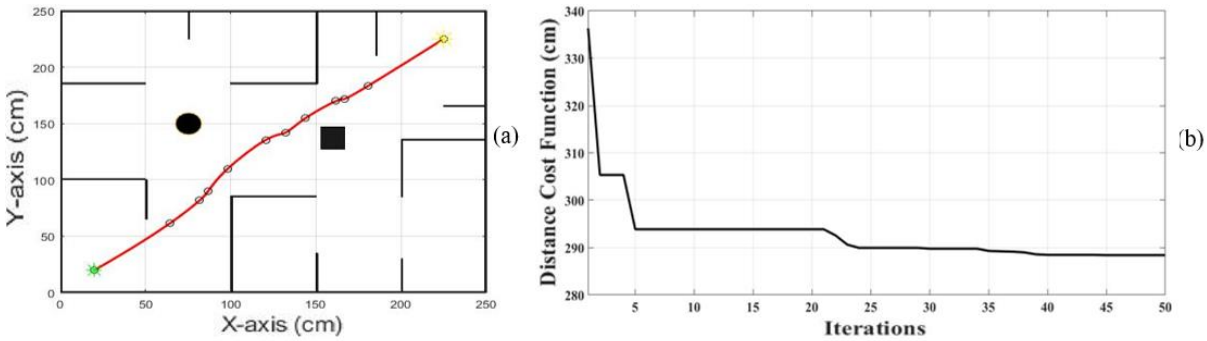


FIG. 8. THE HYBRID RRT*PSO ALGORITHM: (A) THE OPTIMAL PATH AND (B) THE DISTANCE BEST COST FUNCTION.

The mobile robot reference path equation obtained from the RRT*PSO hybrid algorithm was calculated using the third-order degree polynomial fitting function in the MATLAB package, as shown in Equation (9).

$$y_{-ref}\left(x_{-ref}\right) = 3.622 \times 10^{-7} \times x_{-ref}^{4} - 0.0001718 \times x_{-ref}^{3} + 0.02534 \times x_{-ref}^{2} - 0.1923 \times x_{-ref}^{1} + 17.02 \quad (9)$$

Fig. 9 shows the proposed RRT*PSO hybrid algorithm's optimum path has a smooth path, short distance, and free navigation with the presence of obstacles. The second scenario is in the dynamic workspace environment and moving obstacles in the workspace and let the velocity of an obstacle is (0.096 m/sec). Using the suggest proposed algorithm in dynamic obstacles to avoid the collision between the mobile robot and the dynamic obstacle. *Fig. 10* demonstrates the re-planning contour down a path

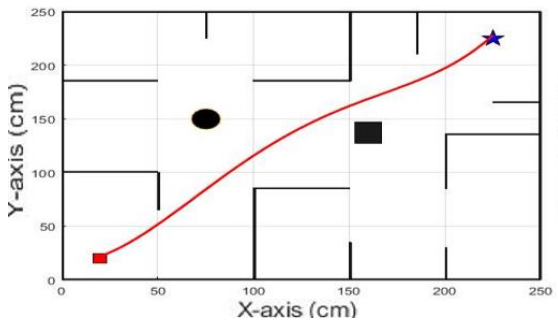


FIG. 9. OPTIMAL PATH OF THE (RRT*PSO) HYBRID ALGORITHM STATIC ENVIRONMENT.

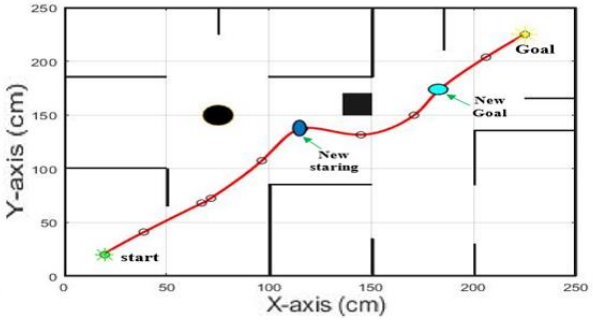


FIG. 10. THE RE-PLANNING CONTOUR DOWN A PATH USING THE (RRT*PSO) HYBRID ALGORITHM IN DYNMIC ENVIRONMENT.

Consequently, the reference path equation for the optimal path in a dynamic environment can be divided into three parts based on the second-order degree polynomial fitting function as follows: Part 1: from position (20, 20) cm to position (120,140) cm as

Received 01/February/2023; Accepted 20/March/2023
© 2017 University of Technology, Iraq

DOI: https://doi.org/10.33103/uot.ijccce.23.3.13

represented in Equation (9). Part 2: from position (120,140) cm to position (180,175) cm represented in Equation (10).

$$y_{-ref}\left(x_{-ref}\right) = 0.024 \times x_{-ref}^{2} - 6.7316 \times x_{-ref}^{1} + 599.99 \quad (10)$$

Part 3 is from nodes (180,175) to (255,255) represented in Equation (9).

Fig. 11 shows the optimal path of the mobile robot that is produced based on the reference path equation with free navigation. Then we applied these optimal reference path equations in the velocity planner controller to generate the reference linear velocity and reference angular velocity of the mobile robot platform. *Fig. 12* shows the reference (linear and angular) velocities of the platform of the mobile robot. It is clear from the responses that the best values of the velocities and smoothness were generated, but in second 3, the response of the angular velocity has a high spike which means there is a big change in the orientation of the mobile robot platform in order to follow the re-planning contour down the path and avoid the dynamic obstacle, and in second 5.1, the response of the angular velocity has a high spike that means there is a big change in the orientation of the mobile robot platform in order to return to the reference path equation. The motion of the mobile robot and following the reference path equation, the velocity planner controller produces the right wheel velocity and left wheel velocity as shown in *Fig. 13* which shows the greatest performance and smoothness of the right and left wheels velocities of the mobile robot to the trajectory the reference path but in second 2.97, the response of the right wheel velocity has a positive spike and the response of the left wheel velocity has a negative spike that means there is a great change in the angular velocity and the orientation of the mobile robot platform in order to follow the re-planning contour down a path and avoid the obstacle, and in second 5.2, the response of the right wheel velocity has a negative spike and the response of the left wheel velocity has a positive spike that means there is a big change in the angular velocity and the orientation of the mobile robot platform in order to return to the reference path equation. During the large change in the re-planning path, the response of both the linear velocities of the wheels remains below the saturation value (0.52 m/sec).

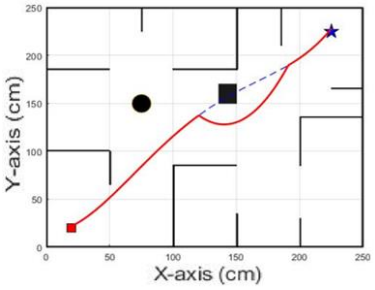


FIG. 11. OPTIMAL REFERENCE PATH OF THE MOBILE ROBOT.

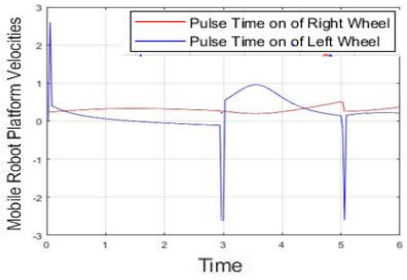


FIG.12. THE LINEAR AND ANGULAR VELOCITIES OF THE PLATFORM OF THE MOBILE ROBOT.

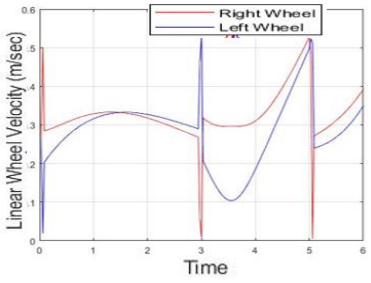


FIG.13. THE LINEAR VELOCITY OF THE RIGHT AND LEFT WHEELS OF THE MOBILE ROBOT PLATFORM.

Fig. 14 demonstrates the desired path and the actual path taken by the mobile robot from starting node (20, 20) cm to point (225, 225) cm under dynamic obstacle movement. The mobile robot tracks the desired path successfully, and the maximum distance error between them is 2.9417 cm. *Fig. 15* demonstrates the desired orientation and actual orientation of the platform of the mobile robot. It is clear from the smooth orientation response, but in second 3, the desired orientation response becomes a low spike which means there is a big change in the orientation of the mobile robot platform in order to follow the re-planning contour down a path and avoid the obstacle, and on second 5.2, the desired orientation response becomes a high spike that means there is a big change in the orientation of the mobile robot platform in order to return to the reference path equation. *Fig. 16* demonstrates the

Received 01/February/2023; Accepted 20/March/2023
© 2017 University of Technology, Iraq
ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.3.13

maximum position error between the desired position and the actual position of the cart of the mobile robot did not exceed 1.5 cm.

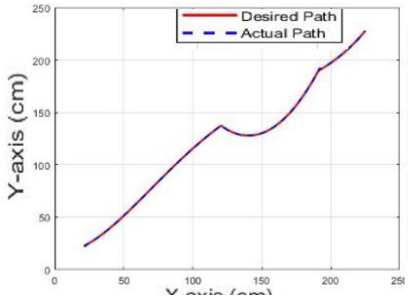


FIG. 14. THE DESIRED PATH AND ACTUAL MOBILE ROBOT PATH UNDER THE DYNAMIC OBSTACLE

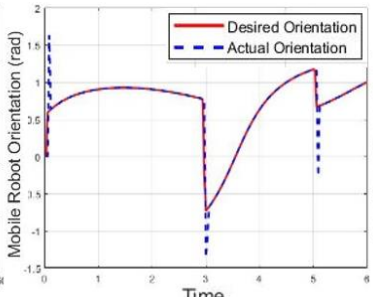


FIG. 15. THE MOBILE ROBOT ORIENTATION UNDER THE DYNAMIC OBSTACLE.

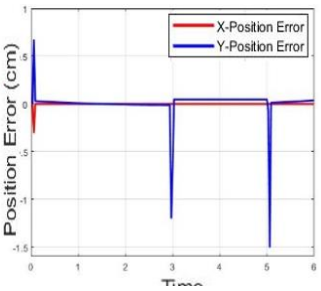


FIG. 16. THE POSITION ERROR OF THE MOBILE ROBOT UNDER THE DYNAMIC OBSTACLE.

In addition, *Fig. 17* shows the orientation error; these errors appeared when the mobile robot changed direction to avoid the dynamic obstacle and track the new path and did not exceed ±1 rad. The velocities values for the two wheels (right and left) of the mobile robot are converted to values appropriated with the work of the drive module for a DC motor of the wheels. Through HTTP, the values have been sent from (CAI) to the mobile robot. The proposed equation for converted values as known as installation sample pulse amplitude (ISPA) is shown in Equation (21).

$$ISPA = \frac{255 \times v_{right}, v_{left}}{\max \text{velocity}} \quad (11)$$

Where: max velocity equal to (0.52m/sec).

These values occurred between 0-255 (pulse amplitude) and they are equivalent to the velocity of 0 to 0.52 m/sec respectively, the velocities values for the two wheels (right and left) of the mobile robot are converted to values ISPA. Through HTTP, the values of ISPA have been sent from (CAI) to the mobile robot. *Fig. 18* shows the pulse amplitude of the right and left mobile wheels of the mobile robot, it shows the value of pulse amplitude is clear and smooth to responses and did not exceed the maximum value (255). Moreover, *Fig. 19* . These velocities are sent in a packet via a wireless network to the microcontroller, taking into account the direction of the wheel velocity (clockwise and counterclockwise). Then, these values are converted to a period (on/off) through (PWM) to operate the drive, taking into account the time period of the motor for the right wheel and the left wheel, shown in *Fig. 19*, and the time pulse duration did not exceed the 33 msec. depending on the sampling time of the motor, which represents (Ts of the motor). The proposed equation for the pulse time is shown in Equation (5.3).

$$pulse\ time = \frac{ISPA \times 33\ msec}{\max ISPA} \quad (12)$$

Where max $ISPA$ is equal to (255).

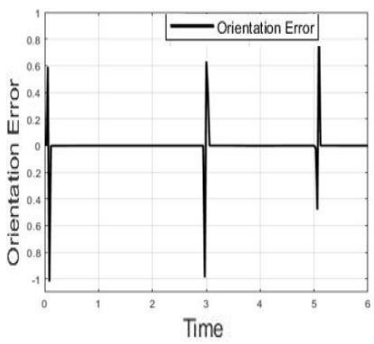


FIG. 17. THE ORIENTATION ERROR OF THE MOBILE ROBOT UNDER THE DYNAMIC OBSTACLE.

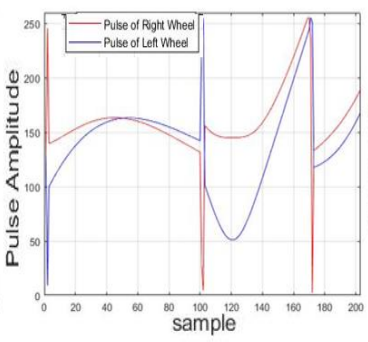


FIG. 18. PULSE AMPLITUDE OF THE RIGHT AND LEFT OF THE PLATFORM OF THE MOBILE ROBOT.

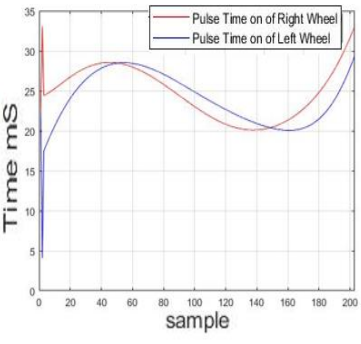


FIG. 19. TIME DURATION OF PULSE OF THE RIGHT AND LEFT WHEELS OF THE MOBILE ROBOT.

Received 01/February/2023; Accepted 20/March/2023
© 2017 University of Technology, Iraq
ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.3.13

The implementation of (red color) real mobile robot in a dynamic real workplace environment, from starting node (20,20) to goal node (225,255) path in a real workspace [250×250] as shown in *Fig. 20 a,b,c,d,e,f*, movement of the mobile robot is smooth, quick and free-navigation to avoid the dynamic obstacle which achieved the mission of the mobile robot is successfully achieved from the starting node to the goal node.

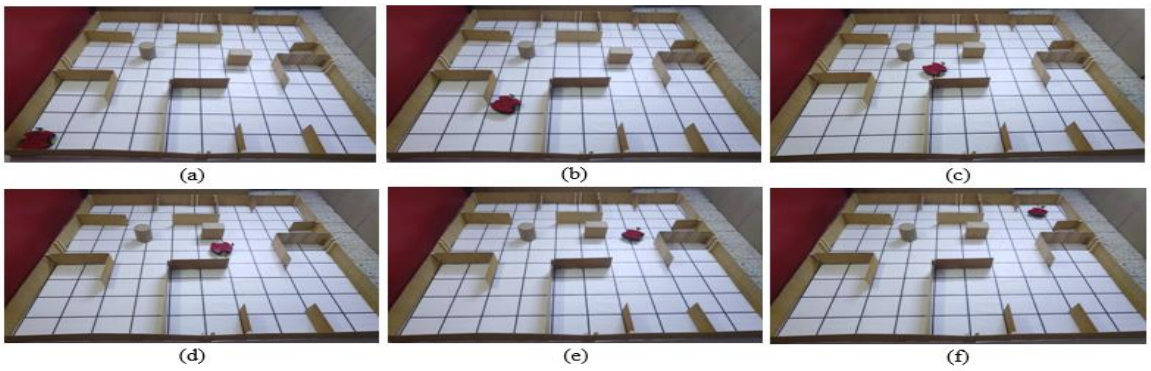
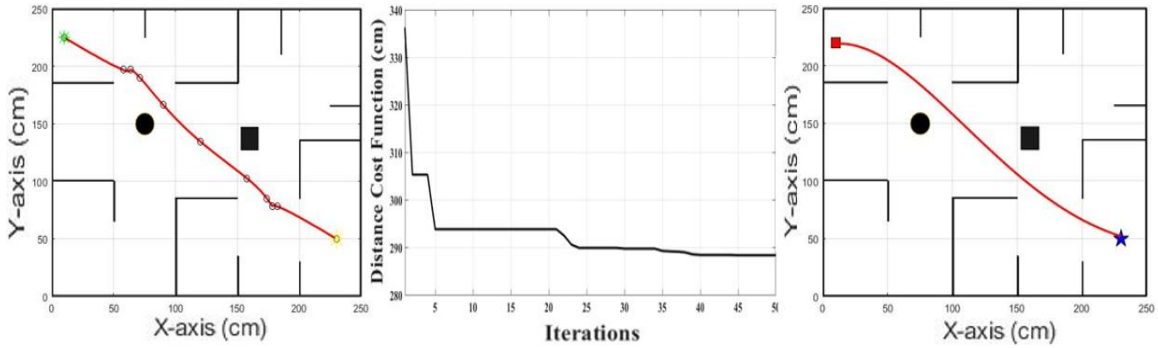


FIG. 20. A, B, C, D, E, F PATH OF RED COLOR REAL MOBILE ROBOT IN A DYNAMIC WORKPLACE ENVIRONMENT.

To implement the second real mobile robot (yellow color) on the second floor of the building, the same workspace environment shown in *Fig. 7* was used representing the implementation of static and dynamic obstacle environments with a workspace of [250×250] cm. However, the different start node and goal node were considered with the same computer hardware and package. The first scenario in a static environment used the proposed hybrid RRT*PSO algorithm to discover a collision-free path, with the mobile robot's start node at (10, 220) cm and the destination position node at (230, 50) cm. *Fig. 21a* shows the smooth and optimal shortest path after using the hybrid RRT*PSO algorithm and the input candidate node equals 11 nodes. The optimal path with the best cost function was discovered at iteration number 40, as shown in *Fig. 21b*, with a maximum number of iterations equals to 50. The best path distances calculated by the proposed hybrid algorithm are (285.51 cm). To generate the reference path equation to the optimal path that was obtained from the RRT*PSO hybrid algorithm, the fourth-order degree polynomial fitting function was used in the MATLAB package. The reference path equation is shown in Equation (8): *Fig. 21c* shows the proposed hybrid RRT*PSO algorithm's optimum path that has a smooth path, short distance, and free navigation with the presence of obstacles.

$$y_{\_ref}\left(x_{\_ref}\right) = -7.53315 \times 10^{-8} \times x_{\_ref}^{4} + 6.16866\text{e} - 05 \times x_{\_ref}^{3} - 0.0148712 \times x_{\_ref}^{2} + 0.357659 \times x_{\_ref}^{1} + 216.783 \quad (12)$$



(a) (b) (c)

FIG. 21. THE PROPOSED HYBRID RRT*PSO.

(a) the optimal path and (b) the best distance cost function (c) the optimal path of the proposed RRT*PSO hybrid algorithm.

Received 01/February/2023; Accepted 20/March/2023
© 2017 University of Technology, Iraq
ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.3.13

In particular, the reference path equation for the optimal path in a dynamic environment can be divided into three parts based on the second-order polynomial fitting function as follows: Part 1: from position (10, 220) cm to position (103,154.5) cm, as represented in Equation (12).Part 2: from position (103,154.5) cm to position (174,83) cm, represented in Equation (13).

$$y_{-ref}\left(x_{-ref}\right) = -0.0346 \times x_{-ref}^{2} + 8.5524 \times x_{-ref}^{1} - 357.68 \quad (13)$$

Part 3 is from the nodes (174,83) to (230,50), as presented in Equation (12). *Fig. 22* demonstrates the re-planning contour up path using hybrid algorithms (RRT*PSO) *Fig. 23* demonstrates the optimal path generated based on the reference path equation with free navigation by the mobile robot.

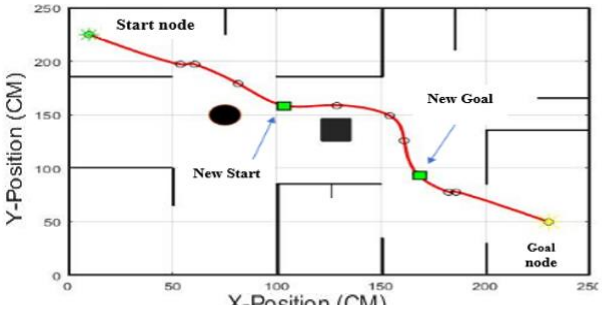


FIG. 22. THE RE-PLANNING CONTOUR UP PATH USING THE HYBRID (RRT*PSO) ALGORITHM.

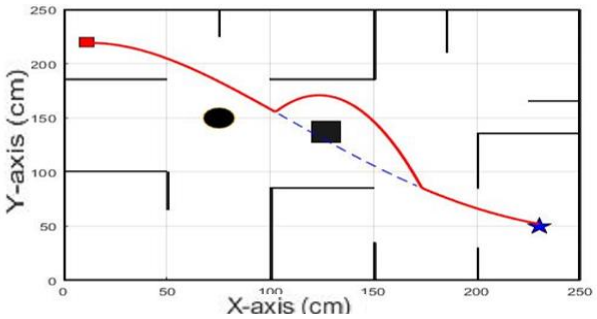


FIG. 23. OPTIMAL REFERENCE PATH OF THE MOBILE ROBOT.

Fig. 24a, b, c, d, e, f demonstrate the application of the real mobile robot (red color) movement in a dynamic workplace environment, from the starting node at (10,220) to the goal node at (230,50) path in a real workspace of [250×250], the path is smooth, fast with free-navigation to avoid the dynamic obstacle, which means that the mission of the mobile robot is successfully achieved from the starting node to the goal node.

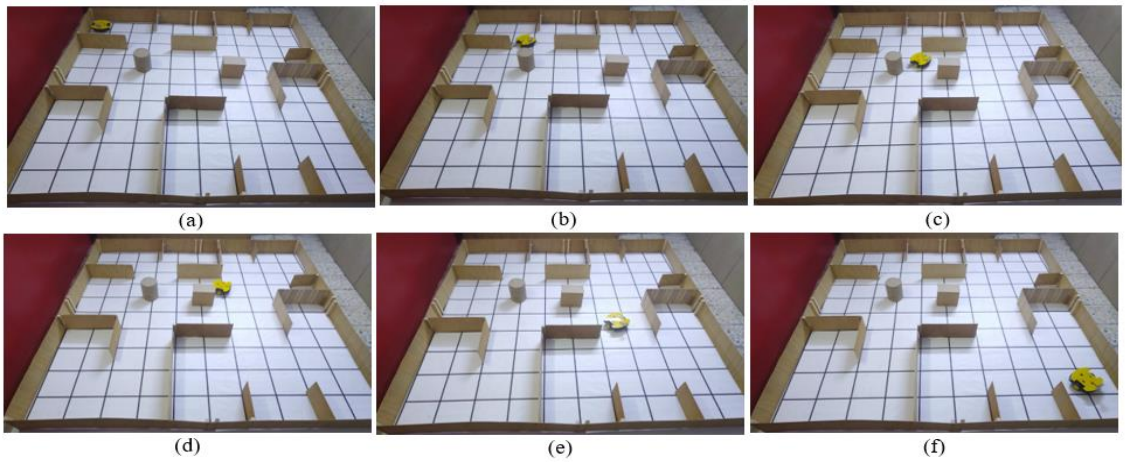


FIG. 24. A, B, C, D, E, F PATH OF THE REAL MOBILE ROBOT (YELLOW COLOR) IN THE DYNAMIC WORKPLACE ENVIRONMENT.

The third real mobile robot (green color) was applied on the third floor in a static environment, and the position information of each obstacle in the workspace is provided. The proposed hybrid RRT*PSO algorithm is used to find a collision-free path. The start node position of the mobile robot is at (10, 150) cm, and the goal node position is at (210, 200) cm. The hybrid RRT*PSO algorithm was applied and the input candidate node equals 11 nodes to find the smooth and optimal shortest path, as shown in *Fig. 25a*. The distance of the optimal path to h the best cost function was found in iteration number 45, as shown in *Fig. 25b* with a maximum iterations number equals to 50 iterations. The value of the

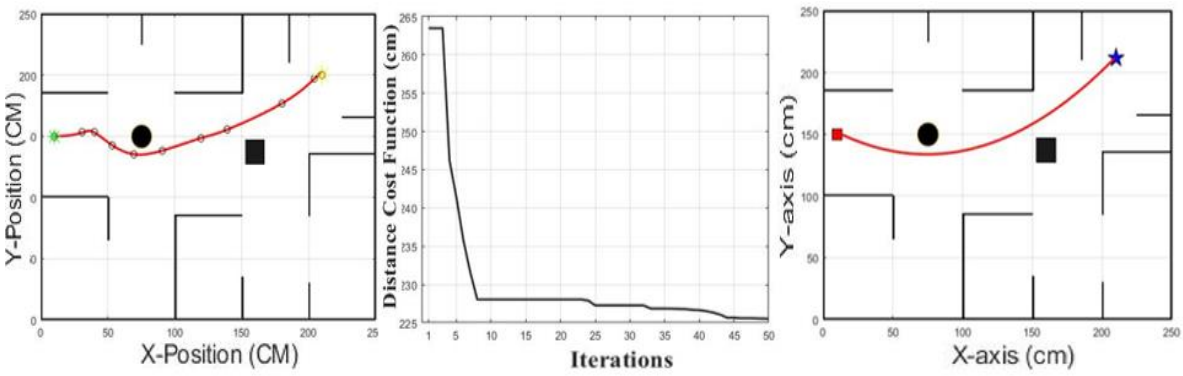
Received 01/February/2023; Accepted 20/March/2023
© 2017 University of Technology, Iraq

DOI: https://doi.org/10.33103/uot.ijccce.23.3.13

proposed hybrid algorithm optimal path distances is equal to (225.7 cm). Equation (14) illustrates how the reference path Equation was created using the second-degree polynomial fitting function in the MATLAB software to fit the optimal path that was produced by the hybrid RRT*PSO algorithm:

$$y_{-ref}\left(x_{-ref}\right) = 0.0034 \times x_{-ref}^{2} - 0.64 \times x_{-ref}^{1} + 157.53 \quad (14)$$

Fig. 25c illustrates the optimal path of the proposed hybrid (RRT*PSO) algorithm that has a short distance and a smooth path without colliding with any static obstacles in the map.



(a) (b) (c)

FIG. 25. THE PROPOSED HYBRID RRT*PSO: (A) THE OPTIMAL PATH AND (B) THE BEST DISTANCE COST FUNCTION. (C) THE OPTIMAL PATH OF THE PROPOSED HYBRID RRT*PSO ALGORITHM.

The application of the real mobile robot (green color) in a static environment that began from the staring node at (10, 150) to the goal node at (210, 200) in a real workspace of [250×250] as shown in *Fig. 26a,b,c,d,e,f*. The movement of the mobile robot is smooth, and quick and avoids the static obstacle, which means that the mission of the mobile robot is successfully achieved from the starting node to the goal node.

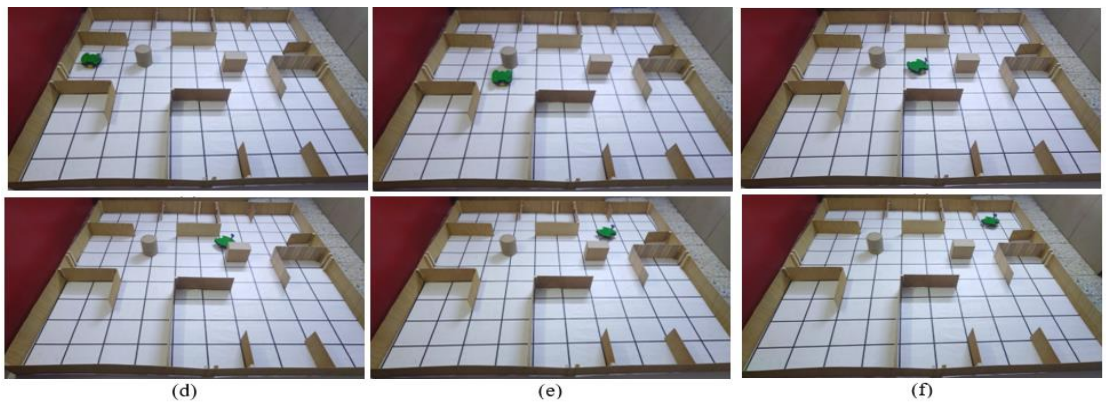


FIG. 26 A, B, C, D, E, F PATH OF THE REAL MOBILE ROBOT (GREEN COLOR) IN THE STATIC WORKPLACE ENVIRONMENT.

In addtion, Table I shows the path error between the simulations and experiments. In the experimental work, the real-time execution for the currently proposed system is shown in Tables II, III, and IV. CSM-MAMRS was implemented in (static and dynamic environments) to calculate the real-time execution of the mobile robot. It has been found that the execution time is < 30 msec (sample rate of the mobile robot). Therefore, three scenarios were achieved: the first one CSM-MAMRS was implemented in a building that has three floors each of which has a robot in a static environment. The real-time execution of the mobile robot is (22.89) msec, as shown in Table II the second one CSM-

Received 01/February/2023; Accepted 20/March/2023
© 2017 University of Technology, Iraq

DOI: https://doi.org/10.33103/uot.ijccce.23.3.13

MAMRS was implemented in a building that has three floors when two of them have a robot in a static environment and the third one has a robot in a dynamic environment. The real-time execution of the mobile robot is (27.207) msec, as shown in Table III. The third one SM-MAMRS was implemented in a building that has two floors each one has a robot in a dynamic environment. The real-time execution of the mobile robot is (24.134) msec, as shown in Table IV.

TABLE I. PATH ERROR BETWEEN THE SIMULATIONS AND EXPERIMENTS

Mobile robot	Static environment			Dynamic environment		
	Desired path(cm)	Actual path(cm)	Error (cm)	Desired path(cm)	Actual path(cm)	Error (cm)
mobile robot on the first floor	291.7287	292.1196	0.3910	309.5767	312.5184	2.9417
mobile robot on the second floor	280.2159	280.2382	0.0224	310.1544	312.9550	2.8005
mobile robot on the third floor	228.2956	227.9676	0.328	-	-	-

TABLE II. REAL-TIME EXECUTION OF THE MOBILE ROBOT IN A STATIC ENVIRONMENT

Parameter	Agent 1	Agent 2	Agent 3
Execution time for generated sample (k) msec	2.57	2.57	2.75
Time of sending packet (msec)	2.5	2.5	2.5
Time of receiving packet (msec)	2.5	2.5	2.5
Total time	22.89 << 30 msec (sample rate of the mobile robot)		

TABLE III. REAL-TIME EXECUTION OF THE MOBILE ROBOT IN A STATIC AND DYNAMIC ENVIRONMENT

Parameter	Agent 1	Agent 2	Agent 3
Execution time for generated sample (k) msec	2.57	2.57	2.57
Time of sending a packet (msec)	2.5	2.5	2.5
Time of receiving a packet (msec)	2.5	2.5	2.5
Time of Re-planning if is encountered a dynamic obstacle (msec)	4.497		
Total time	27.207 << 30 msec (sample rate of the mobile robot)		

TABLE IV. REAL-TIME EXECUTION OF THE MOBILE ROBOT IN DYNAMIC ENVIRONMENTS

Parameter	Agent 1	Agent 2
Execution time for generated sample (k) msec	2.57	2.57
Time of sending a packet (msec), in static obstacle	2.5	2.5
Time of receiving a packet (msec)	2.5	2.5
Time of Re-planning if is encountered a dynamic obstacle (msec)	4.497	4.497
Total time (msec)	24.134 << 30 msec (sample rate of the mobile robot)	

V. CONCLUSIONS

In this paper, the real-time application for a client-server model for a multi-agent mobile robot system (CSM-MAMRS) was designed and implemented. The remaining errors in the experimental results were generated by the difference between the simulations' and experiments' outcomes because the simulation results ignore the two-wheel differential drives of the mobile robot platform's real-world dynamics (torque, and changes mass), underlying friction current in the real system, drop in power (battery), particularly in a dynamic environment (changing the velocity), and modeling errors resulting

Received 01/February/2023; Accepted 20/March/2023
© 2017 University of Technology, Iraq
ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.3.13

from challenges in estimating or measuring the geometric, kinematic, or inertial parameters or from a lack of thorough knowledge of the system's components. Additionally, the simulation was not accurate due to certain incorrect readings induced by the ultrasonic sensors' calibration and alignment, which measure the distance between the mobile robot and obstacles in its path. The velocity planner controller was successfully able to generate the best and smoothest values of the right wheel velocity and the left wheel velocity in order to quickly track the desired path with a minimum tracking position error that did not exceed 2.9 cm along the maximum distance of the actual path of 250 cm. In the experimental work, CSM-MAMRS was implemented in both static and dynamic environments to successfully complete the mobile robot missions in real-time. It has been found that the execution time is less than 30 msec (sample rate of the DC motor of the mobile robot platform). Depending on the execution time, there are three scenarios that can be achieved as shown in Tables II, III, and IV. For future work, adding a sub-server to distribute the task manager to reduce the real-time execution that leads to increasing the number of robots in the real environment, also using another control algorithm for the mobile robot to follow the desired path.

REFERENCES

[1] K. Santosh, "A Review on Client-Server based applications and research opportunity," International Journal of Recent Scientific Research, vol. 10, No.7, pp.33857-3386, 2019.

[2] S. A. Hamid, R.A. Abdulrahman, R.A. Khamees,"What is Client-Server System: Architecture, Issues, and Challenge of Client-Server System, " HBRP Publication, vol. 8, No.18, pp.1-6, 2020.

[3] E. Oussama, Tarik Taleb, Mohamed Menacer, and Mouloud Koudil, "On improving video streaming efficiency, fairness, stability, and convergence time through client-server cooperation", IEEE Transactions on Broadcasting, vol.64, no.1, pp. 11-25,2017.

[4] X. Xiao, Peisen Guo, Jingmei Zhai, and Xianwen Zeng. "Robotic kinematics teaching system with virtual reality, remote control, and an on-site laboratory", International Journal of Mechanical Engineering Education, vol.48, no. 3, pp.197-220,2020.

[5] W. T. Botelho et al., "Toward an interdisciplinary integration between multi-agents systems and multi-robots systems: a case study," The Knowledge Engineering Review, vol. 35, pp. 1–24, Aug. 2020.

[6] T. Samad, S. Iqbal, A. W. Malik, O. Arif, and P. Bloodsworth, "A multi-agent framework for cloud-based management of collaborative robots," International Journal of Advanced Robotic Systems, vol. 15, no. 4, Jul. 2018.

[7] H. Du, G. Wen, D. Wu, Y.Cheng, and J. Lü, "Distributed fixed-time consensus for nonlinear heterogeneous multi-agent systems," Automatica, vol.113, no.1, pp. 108797. 2020.

[8] K. E. Ehimwenma and S. Krishnamoorthy, "Design and analysis of a multi-agent e-learning system using prometheus design tool," IAES International Journal of Artificial Intelligence (IJ-AI), vol. 10, no. 1, pp. 9–23, Mar. 2021

[9] E. Jones, D. Adra, and M. S. Miah, "MAFOSS: multi-agent framework using open-source software," in 2019 7th International Conference on Mechatronics Engineering (ICOM), pp. 1–6, Oct. 2019.

[10] A. Singhal, P. Pallav, N. Kejriwal, S. Choudhury, S. Kumar, and R. Sinha, "Managing a fleet of autonomous mobile robots (AMR) using cloud robotics platform," in 2017 European Conference on Mobile Robots (ECMR), pp. 1–6, Sep. 2017.

[11] D. G. Sáenz, P. Á. Benito, J. R.Azagra, and M. G. Martínez, "Adapted Laboratory for Mobile Robotics Teaching and its Application to Coordinated Control of Robots," SNE, vol. 32 no.2, June 2022.

[12] C. F. Eduardo, "The blockchain: a new framework for robotic swarm systems," In Proceedings of the future technologies conference. Springer, Cham., vol. 881,pp. 1037-1058, Nov. 2018.

[13] P. Ricardo, G. Carvalho, L. Garrote, and U. Nunes, "Sort and Deep-SORT Based Multi-Object Tracking for Mobile Robotics: Evaluation with New Data Association Metrics," Applied Sciences vol.12, no. 3, pp. 1319. 2022V.

[14] B. K. Oleiwi, R. Al-Jarrah, H.t Roth, and B. I. Kazem, "Integrated motion planning and control for multi objectives optimization and multi robots navigation," IFAC-PapersOnLine , vol.48, no. 10,pp. 99-104, (2015).

Received 01/February/2023; Accepted 20/March/2023

© 2017 University of Technology, Iraq ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.3.13

[15] Z. E. Kanoon, A. S. Al-Araji, and M. N. Abdullah. "Enhancement of Cell Decomposition Path-Planning Algorithm for Autonomous Mobile Robot Based on an Intelligent Hybrid Optimization Method." International Journal of Intelligent Engineering and Systems, vol. 15, No. 3, pp. 161-175, 2022.

[16] A.A.A. Rasheed, M.N. Abdullah, and A.S. Al-Araji, "A review of multi-agent mobile robot systems applications," International Journal of Electrical & Computer Engineering, vol.12, no.4, pp.2088-8708, 2022.

[17] L. Bai, and C. Du, "Design and Simulation of a Collision-free Path Planning Algorithm for Mobile Robots Based on Improved Ant Colony Optimization", Ingénierie des Systèmes d'Information , vol.24, No.3, pp.331-336, 2019.

[18] Y. Chao, and X. Xiang, "A path planning algorithm for UAV based on improved Q-learning", In 2018 2nd international conference on robotics and automation sciences (ICRAS), pp. 1-5. IEEE, 2018.

[19] L. Ya, W. Wei, Y.Gao, D.Wang and Z.Fan, "An improved path planning algorithm for mobile robots", Expert systems with applications, vol.152, p.113425, Aug.2020.

[20] Q. Li, F. Gama, A. Ribeiro, and A. Prorok, "Graph Neural Networks for Decentralized Multi-Robot Path Planning", In Proc. Of International Conference on Intelligent Robots and Systems (IROS), pp. 1785-1792, 2020.

[21] F. H. Ajeil, I. K. Ibraheem, M. A. Sahib, and A. J. Humaidi, "Multi-Objective Path Planning of an Autonomous Mobile Robot Using Hybrid PSO-MFB Optimization Algorithm", Applied Soft Computing, vol. 89, p. 106076, 2020.

[22] W. Xiaolu, C. Huang, and F. Chen, "An Improved Particle Swarm Optimization Algorithm for Unmanned Aerial Vehicle Route Planning", In Journal of Physics: Conference Series, vol. 2245, no. 1, p. 012013. IOP Publishing, 2022.

[23] O.A.R. A Wahhab, and A.S.Al-Araji, "An Optimal Path Planning Algorithms for a Mobile Robot", Iraqi Journal of Computers, Communication, Control & Systems Engineering, vol.21, no.2, pp.44-58, 2021.

[24] A. S. Al-Araji, K. E. Dagher, and B. A. Ibraheem, "An Intelligent Cognitive System Design for Mobile Robot based on Optimization Algorithm", In: Proc. of third Scientific Conference of Electrical Engineering (SCEE), pp.84-89, 2018.

[25] A. S. Al-Araji, M. F. Abbod, and H. S. Al-Raweshidy, "Design of a Neural Predictive Controller for Nonholonomic Mobile Robot based on Posture Identifier", In: Proc. of the Lasted International Conference Intelligent Systems and Control, Cambridge, United Kingdom, pp.198-207, 2011.

[26] A.A.A. Rasheed, and A.S. Al-Araji, M.N. Abdullah, "Static and Dynamic Path Planning Algorithms Design for a Wheeled Mobile Robot Based on a Hybrid Technique," International Journal of Intelligent Engineering and Systems, vol. 15, No. 4, pp. 167-181, 2022.

Received 01/February/2023; Accepted 20/March/2023

© 2017 University of Technology, Iraq *ISSN (Print) 1811-9212* *ISSN (Online) 2617-3352*