A Decentralized and PSO-Optimized Multi-Robot Motion Planning Approach for Dynamic Environments

Safa ZIADI [1], Ridha El HAMDI[2] and Mohamed NJAH[3]

Abstract— **In the context of mobile robot motion planning and Industry 4.0, decentralized and efficient control of dynamic environments remains a fundamental challenge. The successful implementation of these control methods is crucial for various real-world applications, including autonomous vehicles and automated guided vehicles (AGVs). This study introduces a decentralized multi-robot motion planning method optimized by Particle Swarm Optimization (PSO). The proposed method builds upon and enhances the "DC-F^2 (Decentralized Control, F^2 based)" approach through extensive testing in real-world scenarios. Experimental results demonstrate the approach's effectiveness in guiding multiple robots to their target destinations without collisions, regardless of environmental complexity—whether static or dynamic.**

Index Terms— **Mobile robots, Multi-robot motion planning, Optimal trajectory, Decentralized control, DC-F^2, Dynamic environments, Particle Swarm Optimization (PSO)**

I. Introduction

The evolution of the research domain of mobile robotics has provided a great service to several fields directly related to human life and participated in their evolution such as the field of Automated Guided vehicles (AGVs) [1], medicine [2], industry [2], agriculture [3], mining [4], space [5], the military domain [6], etc. A key element behind the success in this area is an efficient and intelligent planning generating safe and optimized motion plans for mobile robots moving autonomously in unfamiliar environments. However, these approaches' main drawback is that they are at risk of impractical local minima, particularly in complex scenarios.

The multi-robot coordination problem can be solved using one of two general strategies: centralized and decentralized. With centralized systems, all robot activities are planned by a single planner which continually gathers data from sensors to constitute a precise image about the current state of the environment and then sends its orders for the different robots so as to get the shortest, the securest and possibly the most rapid motion of the different robots under control [7]. In the opposite, decentralized methods do not utilize a central planner; instead, planning duties are divided among all the autonomously functioning robots that depend only on the data that each robot can access [7]. Our objective in this investigation is the implementation of an optimized decentralized motion planning approach for mobile robots which is easier to set up, it solves the problem without the need for a coordination system.

When moving through a working space, D. Wang et al. proposed the Force Field (F^2) mobile robot motion planning approach in 2005 [8]. The robot's condition determines the force field's magnitude and direction in this approach. The created force field is therefore constantly changing as the robot moves, and its parameters are influenced by the movement velocity, environmental conditions, dimensions, priority, and other considerations. An upgraded variant of the $F2$ technique for controlling the motions of mobile robots is the Dynamic Variable Speed Force Field (DVSF^2). Valls Miro et al.(2008) [9] applied DVSF^2 in the design of motion plans in unidentified environments.

After this, we proposed in [10] PSO-DVSF^2 a PSO optimized version of DVSF^2. In [11], we introduced the local version of PSO-DVSF^2 titled Autonomous PSO-DVSF^2. In this paper, we propose the mobile robot motion planning approach DC-F^2(Decentralized Control,F^2 based) which is really an improvement of the autonomous PSO-DVSF^2 approach to become a method of simultanous control of robots.

This paper is organized as follows: Section 2 presents the relevant work. The DC-F^2 motion planning approach and its key features are presented in Section 3. Section 4 presents the experimental results, taking into account both static and dynamic scenarios. The performance of a single robot and two robots operating in both static and dynamic scenarios is also evaluated in this section. The paper concludes at last in Section 5.

II. Related Work

Autonomous robots that operate without human control are needed in many applications such as urban mobility, industry and space exploration. Motion planning is divided into two categories: local (on-line) and global (off-line). If the motion planning is global, information about the environment is already available based on map, cells or grid. In this case, the global approach generates a complete path from the starting position until the destination once before navigation. If the motion planning is local, the robot does not have any information about its neighborhood and the robot sensors gather information about the environment at each position all along its travel until its destination. At each position, the motion planning algorithm searches for the best orientation that avoids nearby obstacles and directs it to its destination. In this paper, we deal with autonomous robots

[1]Advanced Technologies in Medicine and Signals (ATMS), National Engineering School of Sfax, University of Sfax, Sfax, Tunisia safa.ziadi@enis.tn

[2]Digital Research Center of Sfax, Technopole of Sfax, Tunisia hamdi.ridha@univgb.tn

[3]Advanced Technologies in Medicine and Signals (ATMS), National Engineering School of Sfax, University of Sfax, Sfax, Tunisia mohamed.njah@enis.tn

which assumes that the motion planning approach is local and the environment is initially unknown.

Many local approaches have been proposed in the literature and many global approaches have been improved to become local motion planning approaches. For example, in [12], an improved version of PSO, called Adaptive Particle Swarm Optimization (APSO), has been applied in the optimization of mobile robot paths for unknown environments. PSO was combined in [13] with Cuckoo Search (CS) to achieve an optimal path in an unknown complex environment. In [14], a continuous dynamic simulation environment was built to examine the path planning problem of mobile robot in unknown dynamic environments (UDE). The reinforcement learning theory with deep Q-network (DQN) was used to teach the mobile robot how to make the best judgements in order to create a collision-free path in UDE. A weighted reward mechanism was created to strike a balance between avoiding obstacles and moving towards the objective. Furthermore, it was discovered that the relative motion of robots and moving obstacles could result in anomalous incentives and ultimately induce a robot-obstacle collision. The trials demonstrated that the robot can successfully navigate all barriers and attain the goal by modifying the aberrant rewards by setting two reward thresholds in order to handle this issue. In [15], based on the improved COOT (ICOOT) optimization algorithm, this paper addresses the optimal path planning method for an autonomous mobile robot in an unknown environment with irregular static and dynamic obstacles and a static and dynamic target. The drawbacks of the standard COOT's unsteady searches were addressed by the ICOOT. The algorithm searches for the shortest collision-free path and the smoothest motion of the robot. By including the true dimensions of the mobile robot and its kinematic model, the ICOOT attempts to mimic reality. The issue of unstable motion state in obstacle avoidance planning arises in real-time motion planning when a car-like robot travels in an unstructured situation. In this study, a hybrid motion planning strategy based on the artificial potential field and timed-elastic-band (TEB) approach was presented in [16]. In an unstructured environment, various potential fields were created, and the virtual kinetic energy of the robot and the conversion function of the virtual potential energy of the superimposed potential field were used to plan the real-time velocity of the car-like robot. The local optimal path is planned by the optimized TEB approach, which also resolves issues with the non-reachable targets and local minimum region. Otherwise, an optimization approach was suggested to address the drawbacks of the A* algorithm's pathfinding process, including its lengthy calculation time and uneven path [17]. First, by substituting jump points for the expansion nodes in the A* algorithm, the amount of calculation was decreased thanks to a conjunction of A* with the jump point search algorithm (JPS). Next, a pruning method was employed to further eliminate superfluous nodes from the path by trimming the path that has been obtained. Finally, a quadratic bezier curve was used to smooth the path.

In order to create routes for decentralized robots that were intended to extract as much data as possible from unidentified spatial domains in [18], this research proposed a scalable planning algorithm. A density function is used to characterize how informative the data were from the spatial field. They provided an AsyncIGBO algorithm that effectively selects the next most informative site for a decentralized robot team. Then, within a constrained mission duration, they demonstrated a distributed reactive planning and control technique that generates continuous motions for robots to visit the informative places securely. They assessed our approach by examining its asymptotic no-regret characteristics conceptually in relation to a predetermined geographical field. Ultimately, they verified our system through a series of real-world and simulated trials involving numerous robots.

In order to enhance energy economy and safety for swarms of Autonomous Mobile Robots (AMRs) working in complex, dynamic industrial environments, this research [19] presented a decentralized navigation technique. Its main contribution was an Obstacle Persistence Evaluation system that assisted robots in avoiding pointless, energy-intensive detours brought on by transient impediments. The method used an Exponentially Weighted Moving Average (EWMA) to differentiate between more permanent blocks and transient obstacles. This approach, when paired with a conflict-free path planning system, enabled every robot to independently calculate safe, real-time trajectories, guaranteeing smooth operation, high efficiency, and lower system-wide energy consumption. Also, in [20], a decentralized Model Predictive Control (MPC) system that had been experimentally proven was presented for the cooperative movement of goods by several mobile robots. The system ensured accurate trajectory tracking and formation stability while taking into account nonlinear kinematics and physical coupling restrictions by addressing local optimization problems in real time. The approach's scalability and resilience for real-world industrial logistics, where centralized coordination was frequently impractical, were demonstrated by its implementation on physical TurtleBot3 platforms, which successfully managed dynamic barriers and sensor noise.

III. DC-F^2

This section presents the formulation of the mobile robot's mechanical behavior utilizing DC-F^2: the mobile robot's development in terms of position and orientation, the forces acting upon it, and its evolution in terms of linear and angular speeds.

A. Mobile Robot Motion Model

Kinematics is the most basic study of how mechanical systems behaves at time t. The motion model is needed to control the position and the orientation of a mobile robot. The kinematic model for the robot can be expressed as:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} cos(\theta) & 0 \\ sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_t \\ \omega_t \end{bmatrix} \tag{1}$$

The robot motion between two successive positions depends on the time step Δt and the robot's straight line speed and the rotational speed as follows:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{x_{t+1}-x_t}{\Delta t} \\ \frac{y_{t+1}-y_t}{\Delta t} \\ \frac{\theta_{t+1}-\theta_t}{\Delta t} \end{bmatrix} \tag{2}$$

For the modeling of motion in the Cartesian space given by control inputs, the formulation is as follows:

$$u_t = \begin{bmatrix} v_t \\ \omega_t \end{bmatrix} \tag{3}$$

$v_t(m/s)$ is the linear (or translational) speed and $\omega_t(rad/s)$ is the angular (or rotational) speed generated by the attractive force F_{att} and the repulsive forces $F_{rep,obs}$ as will be formulated in the next subsection. During the robot travel between the start and the target points, the control term u_t is computed at every moment t looking for a suitable collision-free path. Then, the robot motion can be represented in the time interval [t, t+1] as a function of the current location, orientation and control (eq.4):

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{bmatrix} = \begin{bmatrix} x_t + v_t \cdot \cos(\theta_t) \cdot \Delta t \\ y_t + v_t \cdot sin(\theta_t) \cdot \Delta t \\ \theta_t + \omega_t \cdot \Delta t \end{bmatrix} \tag{4}$$

The control inputs v_t and ω_t are computed according to the repulsive and attractive forces as will be formulated in the next subsection.

B. Linear and angular speeds computation

The evolution of linear and angular velocities depends on the attractive force (F_{att}) applied by the target on the robot and the set of repulsive forces applied by the obstacles ($F_{j_{rep-obs}}$). We start by computing the sum of the set of forces applied on the robot.

$$F_{Total} = F_{att} + \sum_{j=1}^{m} F_{j_{rep-obs}} \tag{5}$$

where m is the number of obstacles. The angular velocity is computed as follows:

$$\omega_t = \frac{\angle F_{Total} - \theta_t}{\Delta t} \tag{6}$$

and the linear velocity is computed as follows:

$$v_t = \frac{v_{max}}{1 + \frac{max(|F_{j_{rep-obs}}|)}{k_f}} \tag{7}$$

Where the attractive force F_{att} is a virtual force applied by the target in the center of the robot. It leads the robot to the goal. The module of the attractive force in the F^2 approach is constant.

$$|F_{att}| = Q \tag{8}$$

Furthermore, the attractive force orientation is given by:

$$\theta_{F_{att}} = atan2(\frac{y_{tg} - y_t}{x_{tg} - x_t}) \tag{9}$$

where (x_t, y_t) represents a robot position at time t and (x_{tg}, y_{tg}) displays the target position (Fig.1).

The $F_{j_{rep-obs}}$ repulsive force is virtual and is exerted by obstacle j. This force pushes the robot away from obstacle j. Its orientation is computed as follows:

$$\theta_{F_{j_{rep-obs}}} = atan2(\frac{y_t - y_{A_j}}{x_t - x_{A_j}}) \tag{10}$$

Where A(x_{A_j}, y_{A_j})(Fig.1) is the point of obstacle in which the largest repulsive force is reached. Its module is computed as follows:

$$|F_{j_{rep-obs}}| = \begin{cases} 0 & if \quad D_j > D_{max} \\ k_f \cdot \frac{D_{max}-D_j}{D_{max}-D_{min}} & if \quad D_{min} \le D_j < D_{max} \\ F_{max} & if \quad otherwise \end{cases} \tag{11}$$

Where D_j is the nearest distance in 2-D space from point (A_j) to the robot's edge. As D_j changes from D_{min} to D_{max}, the magnitude slowly changes from k_f to 0. k_f is a positive constant scalar which limits the magnitude of the repulsive force. F_{max} is the maximum repulsive force that allows the robot to decelerate as much as possible. D_{min} in (eq.12) is the distance at which the maximum repulsive force of this robot is attained. It provides the robot with a safe distance to avoid other objects coming into that area.

$$D_{min} = k_{dmin} \cdot D_{max} \tag{12}$$

Where, k_{dmin} is a positive fractional number verifying $0 < k_{dmin} < 1$ which strongly affects how close the robot may be to obstacles. D_{max} in (eq.13) is the maximal active distance of the force field of a robot, showing how far that robot may impact others in its vicinity.

$$D_{max} = \frac{k_r \cdot E_r \cdot R_r}{1 - E_r \cos(\theta)} \tag{13}$$

Where k_r is a positive multiplier that delimits a force field coverage area. R_r is the radius of a robot. Also, E_r in (eq.14) is a decimal positive fraction with $0 \le E_r < 1$.

$$E_r = \frac{v_r}{v_{max} \cdot k_v} \tag{14}$$

Where v_r is the absolute value of a robot's velocity, v_{max} is the maximum velocity of the robot and k_v is a positive constant which quantifies the environmental influence on the $DVSF^2$.

Also, θ in (eq.15) denotes the relative angle of this position to the direction of the robot.

$$\theta = \theta_0 - \theta_t \tag{15}$$

Where, θ_t is the orientation of the robot in a global coordinate system and θ_0 is the angle of this point in a global coordinate system. Searching the shortest and the safest path will therefore be in the range of 5 dimensions: k_r, k_v, k_f, Q and k_{dmin}.

In the next section we complete the presentation of our algorithm by presenting our DC-F^2 approach.

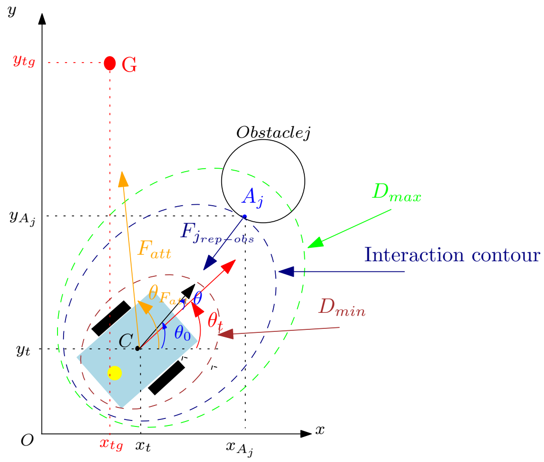


Fig. 1: The DVSF^2 attractive and repulsive forces

C. DC-F^2 approach

PSO is used in 'DC-F^2' to help the robot to select the shortest and the securest path to the target. As a consequence, we are in front of a multi-objective optimization. The first objective is the minimization of the distance between robot's positions and the target. The function to minimize to achieve this path length goal is formulated as follows:

$$Path(X_{i,it}(t)) = \sqrt{(x_{X_{i,it}}(t) - x_{tg})^2 + (y_{X_{i,it}}(t) - y_{tg})^2} \quad (16)$$

The second objective is the maximization of the distance between the nearest obstacle and the robot "$D_{nearest-robot}$". The function to minimize achieving this security objective is for each obstacle the following:

$$Dist_j(X_{i,it}(t)) = \frac{1}{\sqrt{(x_{X_{i,it}}(t) - x_{obs,j})^2 + (y_{X_{i,it}}(t) - y_{obs,j})^2}} \quad (17)$$

The fitness function is a weighted sum of *Path* and the different *Dist* terms for the m nearest obstacles. It is thus formulated as follows :

$$fitness(X_{i,it}(t)) = \phi_1 \cdot Path(X_{i,it}(t)) + \phi_2 \sum_{j=1}^{m} Dist_j(X_{i,it}(t)) \quad (18)$$

$$X_{i,it}(t) = [k_r, k_v, k_f, Q, k_{dmin}] \quad (19)$$

where ϕ_1 and ϕ_2 are two weighting factors such as $\phi_1+\phi_2=1$, m is the number of the nearest obstacles to the robot position, *it* is the PSO iteration , $X_{i,it}$ is the PSO particle i^{th} in the iteration *it* and t is the time step knowing that Δt=0.08s. Selected values for ϕ_1 and ϕ_2 in our simulations are 0.3 and 0.7, respectively [21].

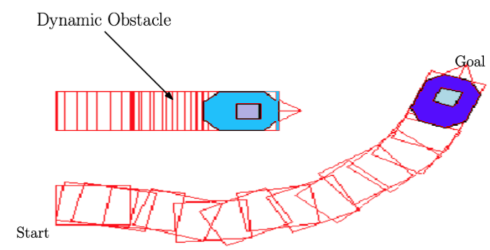
IV. Experimental Results

In this section, we present the results of the test of our approach 'DC-F^2' in the MobileSim virtual environment. Many static and dynamic environments of different levels of complexity are considered. The robot is characterized by: V_{max}=0.4m/s, F_{max}=30 and Δt=0.08s. 'DC-F^2' is tested, in the first subsection, in dynamic environments. In the secend subsection, the improved version of 'DC-F^2' is tested in a simultaneous control of a multiple number of mobile robots in static environments.

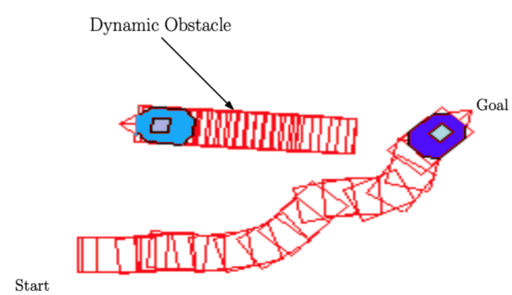
Our implementation consisted on developing a callback task control law. The task uses the robot's operating system named ARCOS (Advanced Robot Control and Operations Software). The client callback task is implemented by means of the provided ARIA SDK interface used for synchronization of packet transmission. A control interpretation task callback is then invoked by the robot's real time operating system via a communication interface every time cycle of 40ms in our simulations.

A. DC-F^2 for Single-Robot Control:

In these simulations, the environment is dynamic containing one dynamic obstacle. We consider two scenarios in which the robot starts from the origin of the map (0.0) and is asked to reach its destination (2.5,2). Figure 2a shows the robot's trajectory selected in the case of the first scenario and Figure 2b shows the robot's trajectory selected in the case of the second scenario.



(a) Case1



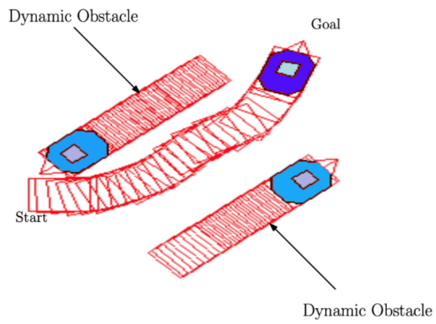
(b) Case2

Fig. 2: DC-F^2 controlling a robot in a dynamic obstacle scenario within the MobileSim environment

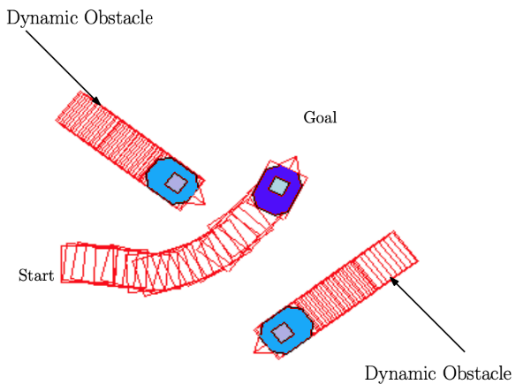
Two dynamic obstacles are present in the dynamic environment of these next simulations. We examine two situations where the robot is instructed to travel to its destination (2.5,2) from the map's origin (0.0). The robot's trajectory selected for the first scenario is shown in Figure 3a, and for the second scenario, it is shown in Figure 3b.

..

The two figures clearly demonstrate a mobile robot that navigates from a start place to a target while avoiding

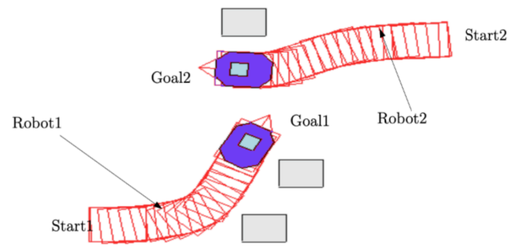


(a) Case1

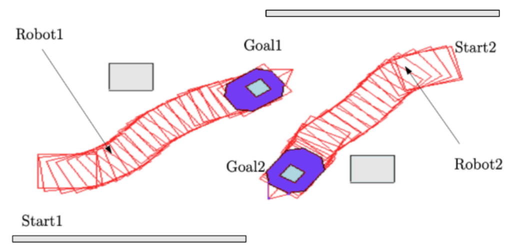


(b) Case2

Fig. 3: DC-F^2 controlling a robot in the MobileSim environment with two dynamic obstacles



(a) First scenario



(b) Second scenario

Fig. 4: DC-F^2 controlling multiple robots in a static environment within the MobileSim simulation

dynamic obstacles. Curved paths demonstrate that the robot follows a smooth and efficient motion while reacting to the movement of nearby obstacles, ensuring collision avoidance in a dynamic environment.

B. *DC-F^2 for Multi-Robot Control*

In these simulations, 'DC-F^2' controls simultanously a multiple number of mobile robots in environments with three and four static obstacles respectively. The robots navigate across environment with three or four static obstacles. The results displayed in the Figure 4a and in the Figure 4b demonstrate that both robots achieve their objectiveswithout any collision with each other and without any collision with obstacles following the shortest path.

Fig.4 illustrates the effectiveness of the navigation framework in handling multi-robot coordination in cluttered environments. In scenario (a), the robots follow smooth trajectories while safely avoiding obstacles and maintaining proper spacing. In scenario (b), even under more constrained conditions with complex obstacle arrangements and different goal positions, the robots successfully reach their destinations without collisions.

In this second simulation, 'DC-F^2' controls simultanously a multiple number of mobile robots in environments with one dynamic obstacle. The results displayed in the Fig.5 demonstrate that both robots achieve their objectives without any collision with each other and without any collision with obstacles following the shortest path.

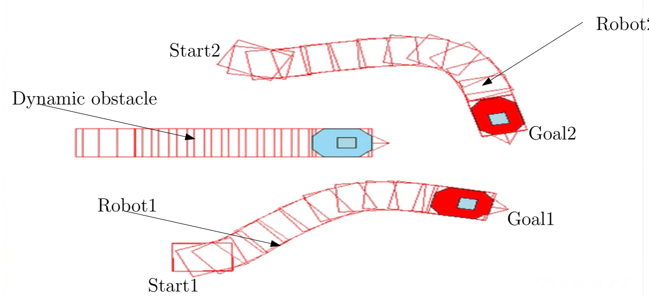


Fig. 5: DC-F^2 controlling multiple robots in a dynamic environment within the MobileSim simulation

The efficiency of the 'DC-F^2' motion planning approach to direct the two robots to their destinations in the environment with a dynamic obstacle is demonstrated in Fig.5.

In this third simulation, 'DC-F^2' simultaneously controls multiple mobile robots in environments with two static and one dynamic obstacle. The outcomes shown in Fig.6 show that both robots accomplish their goals by taking the shortest route and avoiding collisions with objects or one another.

Fig.6 illustrates the effectiveness of the 'DC-F^2' motion planning approach in guiding the two robots to their destinations in an environment with both dynamic and static obstacles.

V. Conclusion and Perspectives

In this study, we evaluated the local mobile robot motion planning approach, DC-F^2, originally introduced in our previous work [11], as a decentralized coordination system for

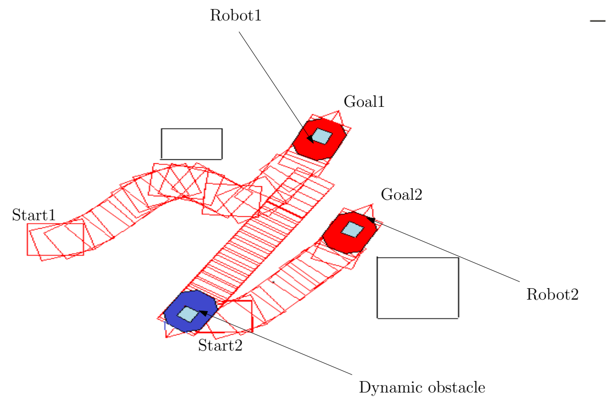


Fig. 6: DC-F^2 controlling multiple robots in a dynamic and static environment within the MobileSim simulation

guiding multiple robots simultaneously in various scenarios. The tests were conducted in the MobileSim environment, a multi-robot virtual platform that simulates real-world conditions. We started by testing our approach with a single robot in both static and dynamic environments that were previously unknown. Then, we applied DC-F^2 to the simultaneous control of multiple robots, addressing different multi-robot task scheduling scenarios. In all experiments, the robots were successfully guided to their destinations, regardless of the complexity of the environment. The results from each test demonstrated the effectiveness of DC-F^2 in identifying the safest and fastest routes to the desired locations.

For future work, the DC-F^2 approach could be enhanced by integrating swarm robotics for large-scale coordination and edge computing for real-time decision-making. Expanding its applications to human-robot interaction, autonomous vehicles, and sustainable energy-efficient navigation presents promising research directions. Additionally, bridging the gap between simulation and real-world implementation will further advance the system's versatility in dynamic environments.

References

[1] N. Rezaei and S. Darabi, "Mobile robot monocular vision-based obstacle avoidance algorithm using a deep neural network," *Evolutionary Intelligence*, pp. 1–16, 2023.

[2] N. Sharma, J. K. Pandey, and S. Mondal, "A review of mobile robots: Applications and future prospect," *International Journal of Precision Engineering and Manufacturing*, vol. 24, no. 9, pp. 1695–1706, 2023.

[3] M. S. A. Mahmud, M. S. Z. Abidin, A. A. Emmanuel, and H. S. Hasan, "Robotics and automation in agriculture: present and future applications," *Applications of Modelling and Simulation*, vol. 4, pp. 130–140, 2020.

[4] C. J. M. Martinez, R. C. Salguero, R. Palomares, and J. Cornejo, "Mechatronics development of terrestrial mobile robot for exploring and monitoring environmental parameters at mine analogue sites using iot platform," in *2020 IEEE XXVII International Conference on Electronics, Electrical Engineering and Computing (INTERCON)*, pp. 1–4, IEEE, 2020.

[5] M. Caruso *et al.*, "Design and simulation of multilevel interacting mobile robots for space and industrial applications," 2023.

[6] R. Raj and A. Kos, "A comprehensive study of mobile robot: history, developments, applications, and future research perspectives," *Applied Sciences*, vol. 12, no. 14, p. 6951, 2022.

[7] L. Poudel, S. Elagandula, W. Zhou, and Z. Sha, "Decentralized and centralized planning for multi-robot additive manufacturing," *Journal of Mechanical Design*, vol. 145, no. 1, p. 012003, 2023.

[8] D. Wang, D. Liu, X. Wu, and K. Tan, "A force field method for robot navigation," in *International Conference on Computational Intelligence, Robotics and Autonomous Systems*, CIRAS & FIRA Organising Committee, 2005.

[9] J. Valls Miro, T. Taha, D. Wang, and G. Dissanayake, "An adaptive manoeuvring strategy for mobile robots in cluttered dynamic environments," *International Journal of Automation and Control*, vol. 2, no. 2-3, pp. 178–194, 2008.

[10] S. Ziadi, M. Njah, and M. Chtourou, "Pso-dvsf^2: A new method for the path planning of mobile robots," in *16th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, pp. 445–451, IEEE, 2015.

[11] S. Ziadi, A. Benali, and M. Njah, "Autonomous pso-dvsf 2 in the control of real mobile robots in unknown environments," in *2023 9th International Conference on Control, Decision and Information Technologies (CoDIT)*, pp. 1442–1447, IEEE, 2023.

[12] H. S. Dewang, P. K. Mohanty, and S. Kundu, "A robust path planning for mobile robot using smart particle swarm optimization," *Procedia computer science*, vol. 133, pp. 290–297, 2018.

[13] S. Shukla, N. Shukla, V. K. Sachan, and S. Sharma, "Multi robot path planning based on cuckoo search and with pso algorithm in the complex and the unknown environment," *International Journal of Recent Technology and Engineering*, vol. 8, 2019.

[14] R. Huang, C. Qin, J. L. Li, and X. Lan, "Path planning of mobile robot in unknown dynamic continuous environment using reward-modified deep q-network," *Optimal Control Applications and Methods*, vol. 44, no. 3, pp. 1570–1587, 2023.

[15] J. A. Abdulsaheb and D. J. Kadhim, "Robot path planning in unknown environments with multi-objectives using an improved coot optimization algorithm.," *International Journal of Intelligent Engineering & Systems*, vol. 15, no. 5, 2022.

[16] X. Sun, S. Deng, T. Zhao, and B. Tong, "Motion planning approach for car-like robots in unstructured scenario," *Transactions of the Institute of Measurement and Control*, vol. 44, no. 4, pp. 754–765, 2022.

[17] X. Wang, Z. Liu, and J. Liu, "Mobile robot path planning based on an improved a* algorithm," in *International Conference on Computer Graphics, Artificial Intelligence, and Data Processing (ICCAID 2022)*, vol. 12604, pp. 1093–1098, SPIE, 2023.

[18] A. A. R. Newaz, M. Alsayegh, T. Alam, and L. Bobadilla, "Decentralized multi-robot information gathering from unknown spatial fields," *IEEE Robotics and Automation Letters*, vol. 8, no. 5, pp. 3070–3077, 2023.

[19] G. Triki, M. Mohammadpour, S. Kelouwani, and L. Zeghmi, "Energy efficient and conflict-free path planning for autonomous mobile robot fleets via obstacle persistence evaluation and smart zoning," *IEEE Access*, 2026.

[20] I. Muhammed, A. A. Nada, and H. El-Hussieny, "Real-time decentralized model predictive control for cooperative multi-robot object transport: experimental validation," *Scientific Reports*, 2026.

[21] S. Ziadi, M. Njah, and M. Chtourou, "Pso-cf2: A new method for the path planning of a mobile robot," in *Systems, Signals & Devices (SSD), 2015 12th International Multi-Conference on*, pp. 1–7, IEEE, 2015.