Received: 21 August 2023 | Revised: 7 October 2023 | Accepted: 8 December 2023
DOI: 10.37917/ijeee.20.2.21 Vol. 20 | Issue 2 | December 2024

Open Access

Iraqi Journal for Electrical and Electronic Engineering
Original Article

Efficient Path Planning in Medical Environments: Integrating Genetic Algorithm and Probabilistic Roadmap (GA-PRM) for Autonomous Robotics

Sarah Sabeeh[1], Israa S. Al-Furati*[2]
[1]Computers Engineering Department, University of Basrah, Basrah, Iraq
[2]Electrical Engineering Department, University of Basrah, Basrah, Iraq

Correspondance
*Israa S. Al-Furati
Electrical Engineering Department,
University of Basrah, Basrah, Iraq
Email:israa.sabri@uobasrah.edu.iq

Abstract
Path-planning is a crucial part of robotics, helping robots move through challenging places all by themselves. In this paper, we introduce an innovative approach to robot path-planning, a crucial aspect of robotics. This technique combines the power of Genetic Algorithm (GA) and Probabilistic Roadmap (PRM) to enhance efficiency and reliability. Our method takes into account challenges caused by moving obstacles, making it skilled at navigating complex environments. Through merging GA's exploration abilities with PRM's global planning strengths, our GA-PRM algorithm improves computational efficiency and finds optimal paths. To validate our approach, we conducted rigorous evaluations against well-known algorithms including A*, RRT, Genetic Algorithm, and PRM in simulated environments. The results were remarkable, with our GA-PRM algorithm outperforming existing methods, achieving an average path length of 25.6235 ***units and an average computational time of*** 0.6881 ***seconds, demonstrating its speed and effectiveness. Additionally, the paths generated were notably smoother, with an average value of*** 0.3133***. These findings highlight the potential of the GA-PRM algorithm in real-world applications, especially in crucial sectors like healthcare, where efficient path-planning is essential. This research contributes significantly to the field of path-planning and offers valuable insights for the future design of autonomous robotic systems.***

Keywords
Genetic Algorithm, Medical Robotics, Path-planning, Probabilistic Roadmaps, Robot Navigation, Static and Dynamic Obstacles.

I. Introduction

In the field of robotics, path planning is essential for autonomous robots to make smart decisions while navigating complex terrains. This skill becomes increasingly important as robots are used in various industries, including search and rescue, environmental monitoring, and space exploration. Path planning not only affects efficiency but also safety, as it allows robots to avoid obstacles and collisions [1]. In the medical field, robots play a crucial role in supporting healthcare professionals by performing tasks like transporting supplies and assisting with diagnostics. Advanced path planning is vital for these robots to navigate crowded and dynamic medical environments efficiently. Integrating AI-powered path planning into healthcare robotics can reshape workflows, enhance patient care, and advance medical technology [2].

This research focuses on developing an innovative path planning algorithm tailored for robots operating in medical environments. The algorithm aims to adapt to dynamic obstacles, minimize path length, and ensure smooth robot movements. It combines Genetic Algorithms and Probabilistic Roadmaps to optimize path planning and outperforms conventional methods in extensive simulations. Overall, this work contributes to the advancement of autonomous robotics in various applications, from exploration to surveillance and transportation tasks. The key contributions of this paper can be briefly summarized in three primary aspects:

1. **Novel path-planning algorithm.** The paper introduces a new path-planning algorithm that combines genetic algorithms (GA) and probabilistic roadmaps (PRM) to enhance autonomous robot navigation. This novel approach improves path-planning efficiency and quality compared to traditional algorithms.
2. **Real-time performance and robustness.** The focus is on real-time performance and adaptability in dynamic environments. The algorithm operates in real-time and includes dynamic obstacle avoidance, allowing robots to respond quickly to environmental changes and navigate safely around moving obstacles. This robustness makes it suitable for real-world applications.
3. **Comparative analysis and scalability.** The paper also conducts a comprehensive comparison with existing path-planning methods like A*, RRT, genetic algorithms, and PRM. The main objective of this comparison is to gain insights into the advantages and shortcomings of these methods about three essential performance measures: average path length, average computational time, and average smoothness.

The paper reviews existing algorithms, details the new algorithm's components, and explains the system model in sections II. , III. , and IV. , respectively. Section V. explains the experimental setup, while Section VI. presents results with clear evaluation and analysis. The conclusion emphasizes the algorithm's significance and suggests future research directions.

II. Literature Review

In this part, we perform an extensive examination of existing research on the development of path-planning algorithms for robotics and autonomous systems. We will examine multiple recent research papers to gain a deep understanding of the current methods and innovations in this area. Through analyzing and synthesizing these discoveries, we aim to pinpoint areas where improvements can be made, identify challenges, and uncover opportunities for enhancing path-planning algorithms.

A. Existing Path-Planning Algorithms

Path-planning is a vital challenge in robotics, involving finding the best path while avoiding obstacles. Over time, researchers and engineers have developed numerous path-planning algorithms, each with its strengths and suitability for different scenarios. This overview discusses the top 10 algorithms known for their efficiency and effectiveness in real-world applications, highlighting their ability to handle complex path-planning tasks accurately.

- **Probabilistic Roadmaps (PRM):** Probabilistic Roadmaps (PRM) is a popular sampling-based path-planning algorithm used in robotics and motion planning [3]. It constructs a graph by randomly sampling points in the configuration space and connecting them through collision-free paths. PRM offers advantages in complex and high-dimensional spaces and is widely applied in real-world robot motion planning tasks.
- **Rapidly Exploring Random Trees (RRT):** Rapidly Exploring Random Trees (RRT) is a rapidly growing tree-based path-planning algorithm [4]. It starts with a single vertex representing the initial state and incrementally explores the search space by adding new vertices connected to the existing tree. RRT is particularly effective in exploring large spaces efficiently and is well suited for applications involving real-time obstacle avoidance and dynamic environments.
- **A* (A-star):** A* (pronounced as ”A-star”), a widely recognized graph search algorithm, is employed to locate the shortest path connecting two nodes within a weighted graph [5].It employs a heuristic function to guide the search towards the goal, making it both complete and optimal. A* is widely used in various domains, including robotics, computer games, and route planning applications.
- **D* (D-star):** D* (D-star) is an incremental heuristic search algorithm designed for dynamic environments where both the cost of the path and the terrain change over time [6]. It efficiently updates the path based on new information and avoids recomputing the entire path, making it suitable for scenarios with uncertain and changing conditions.
- **Dijkstra's Algorithm:** Dijkstra's algorithm is a classical shortest-path algorithm used to find the minimum-cost path in a weighted graph [7]. It iteratively explores neighboring nodes from the start node, keeping track of the minimum distance to each node. Dijkstra's algorithm guarantees the shortest path but may become computationally expensive in large graphs.
- **Wavefront Propagation:** Wavefront Propagation is a simple and widely used algorithm for grid-based path-planning [8]. It operates on a grid map where each

cell represents an obstacle or free space. The algorithm starts from the goal position, assigns wavefront values, and propagates them outward until it reaches the start position, creating a path.

- **Genetic Algorithms:** Genetic Algorithms (GA) are population-based optimization techniques inspired by the process of natural selection [9]. In path-planning, GA involves evolving a population of candidate paths using genetic operations such as crossover and mutation. Paths with higher fitness, determined by a fitness function, have a higher chance of being selected for the next generation, gradually improving the quality of the paths.
- **Artificial Potential Fields:** Artificial Potential Fields (APF) is a reactive path-planning approach that models the environment as attractive and repulsive forces [10]. The robot navigates by following the gradient of the potential field, moving towards the goal while avoiding obstacles. APF is well suited for real-time applications, but it may suffer from local minima and oscillations.
- **Ant Colony Optimization (ACO):** Ant Colony Optimization (ACO) is a metaheuristic inspired by the foraging behavior of ants [11]. In path-planning, ACO involves simulating the movement of virtual ants that deposit pheromones on the paths they explore. The algorithm utilizes pheromone trails to guide other ants in finding shorter and more efficient paths to the goal.
- **Harmonic Functions-Based Methods:** Harmonic Functions Based Methods use potential functions derived from harmonic functions to compute optimal paths [12]. These methods represent the environment as a graph and solve partial differential equations, ensuring that the computed paths follow the laws of physics and offer smooth trajectories.

In addition to the aforementioned algorithms, there are several other path-planning techniques used in various scenarios. For instance, Visibility Graphs are used to compute paths in environments where direct line-of-sight visibility between points can be guaranteed. They connect visible points with straight line segments, creating a graph that simplifies path planning [13]. Cell decomposition is a common technique used to simplify the path planning process, especially in structured environments. It is used to divide the robot's environment into smaller, manageable regions or cells. Each cell represents a portion of the environment, and these cells are typically of a uniform size and shape [14]. Finally, the Minkowski sum is used, in the context of robotics and path planning, to calculate the space that encompasses all possible positions of a robot when it moves in a particular environment while considering its geometry and size. When taking the Minkowski sum of the robot's shape and the obstacles in the environment, we can create a new shape that represents the free space in which the robot can navigate without colliding with obstacles. This free space is essential for path planning algorithms to find collision-free paths for the robot [15].

B. Path-Planning in Healthcare Settings

In recent times, path planning in healthcare has seen significant progress, with intelligent algorithms playing a vital role in various medical applications. Healthcare facilities' complexities have led to the development of advanced path planning methods that ensure smooth navigation. These solutions, powered by cutting-edge algorithms and technology, aim to enhance patient care, improve robotic assistance, and optimize logistical operations in the medical sector. This section explores path planning in healthcare, emphasizing the transformative impact of AI-driven approaches on medical workflows and seamless movement in critical healthcare environments.

For instance, in a notable study, Z. D. Hussein, M. Z. Khalifa, and I. S. Kareem [16] improved the performance of a Laparoscope surgical robot with seven degrees of freedom. They used a genetic algorithm and a MATLAB-based program to find the best path, optimizing distance while avoiding obstacles in dynamic environments. Real-world tests were conducted at Al-Sader educational hospital and the Research Unit of Automation and Robotics, University of Technology, using the Lab-Volt Servo Robot System Model5250 (RoboCIM5250). This advancement has the potential to enhance surgical procedures and reduce patient risks. However, the findings' practical relevance may vary depending on each hospital's robot and environment characteristics.

In another recent study [17], Fang et al. introduced advanced technology called SLAM (Simultaneous Localization and Mapping) for robots in healthcare facilities. This tech uses images to make hospitals run smoother and reduce COVID-19 risks. It is good at handling moving things in hospitals by understanding pictures. Combining this with a knowledge graph helps robots know where they are better. This study offers several benefits, addressing the challenges of frequent changes in hospital layouts through mapping and specialized descriptors. However, it may pose computational demands for map creation and image quality enhancement.

A notable advancement is the MKR (Muratec Keio Robot), an advanced robot designed for healthcare [18]. It uses omnidirectional wheels to move safely and efficiently, avoiding collisions with obstacles. The robot uses virtual potential fields to navigate globally, considering both stationary and moving obstacles. Experiments in a hospital demonstrated its ability to navigate successfully, avoiding collisions. This innovation has the potential to transform healthcare robot mobility, making operations smoother and more efficient.

However, it is crucial to consider potential limitations, including the need for accurate dynamic modeling, sensor data, and challenges in real-world hospital integration. Further testing in various hospital settings and conditions is necessary to assess its practicality.

A new path-planning algorithm was introduced in [19] for autonomous electric wheelchairs in hospitals, aiming to ensure patient safety by considering constraints on body acceleration and navigating appropriate routes within healthcare facilities. This algorithm takes into account wheelchair characteristics, user input, and wheel behavior, making it adaptable to different wheelchair systems. Through thorough numerical simulations and testing in real hospital environments, the algorithm proves its ability to meet body acceleration constraints and find suitable paths within hospitals. This promising solution enhances the performance of autonomous electric wheelchairs in healthcare settings.

In a university research project, D. P. Romero-Martí, J. I. Núnez-Varela, C. Soubervielle-Montalvo, and A. Orozco-de-la-Paz [20]. introduce a service robot that uses a map of a building and reinforcement learning to learn the best routes between locations. They incorporated a Roomba robot and created a user-friendly control system. This research has promising implications for using service robots in various settings like homes, hospitals, and offices, where they can assist with a variety of tasks. However, limitations exist, such as potential inaccuracies in the robot's maps and its applicability depending on the robot type.

In a recent study [21], a new algorithm for multi-robot path planning in hospital environments is introduced by X. Huang, Q. Cao, and X. Zhu. This method combines corridor and room navigation strategies using graph-based approaches for corridors and artificial potential fields for flexible movement in rooms. Simulations show that this algorithm significantly improves robot speed in corridors and enables flexible navigation within rooms. This innovation holds promise for enhancing multi-robot coordination in complex healthcare settings. However, there are concerns about its accuracy in modeling environments and performance in dynamic conditions, requiring further testing.

A different paper [22] introduces a new path-planning method designed for smart wheelchairs. This method uses the adaptive polymorphic ant colony algorithm and includes strategies to handle challenges and find better paths. Compared to other ant colony algorithms, it performs exceptionally well, providing efficient and optimal solutions for healthcare and smart wheelchair navigation. The method is good at finding the best overall paths without being stuck in local problems. Nevertheless, there are also some downsides to think about. It might be a bit hard for computers to use because it is a bit complex, and it might not work perfectly in real-life situations. Moreover, it might not work well in different types of places.

In a healthcare surface disinfection robot study [23], I. T. Kurniawan and W. Adiprawita created three key modules: one for finding its location (Augmented Monte Carlo Localization), one for planning its path (Rapidly Exploring Random Tree*), and another for covering surfaces (Spanning Tree Coverage). These robots can autonomously disinfect surfaces using ultraviolet-C lamps, achieving high sterilization rates without human involvement. The robot's intelligent decision-making ensures safety by minimizing infection and radiation risks. Nevertheless, concerns remain, including limited real-world testing and potential performance variations based on usage conditions.

In recent research [24], S. Wan, Z. Gu, and Q. Ni explored the latest developments in mobile healthcare robots with a focus on strong and fast communication. They worked on improving tasks that need quick responses and a lot of communication by offloading some tasks, making healthcare robots more effective. They split the robot's functions into edge and core tasks, emphasizing technologies like human-robot interaction, navigation, and AI. They also tackled challenges in wireless communication for these robots and highlighted AI's vital role in ensuring safety and reliability in healthcare services. This approach benefits healthcare service users by enabling quick-response, communication-intensive tasks. However, it necessitates robust communication and advanced AI for managing radio signals, movement, and service delivery efficiently.

A groundbreaking approach was introduced to transform the creation of dexterous robotic tools, especially for concentric tube robots, as explained in [25]. This innovative method combines image-based path planning, robot design, and 3-D printing technology. It uses preoperative ultrasound images to plan safe paths and set critical parameters for the robot. The goal is to improve access to diseased areas, especially in pediatric patients, during minimally invasive medical procedures. This new technique shows great promise for advancing healthcare procedures. However, it has limitations like needing accurate pre-surgery imaging, potential challenges in customizing robots for each patient, and the need for validation studies in real clinical settings.

The SERROGA research project focuses on creating a robot companion for older individuals to help with their health needs at home [26]. This paper presents the robot's architecture, abilities, and important services as a health assistant. It also introduces a new way to measure and evaluate how well the robot navigates in apartments. The research includes tests in 12 apartments with project staff and seniors, as well as case studies with nine seniors living at home. The study looks at both practical and emotional aspects of the robot's assistance, and it shows that seniors found value in its health-related help and formed emotional bonds with it. This research can help elderly people, and it introduces a new method to assess how well robots navigate homes. However, there is still work needed to improve and test the robot's reliability in different home settings.

In the context of real-world applications, M. Gillham et al. [27] highlight the importance of human-assistive devices, especially in situations where collision avoidance is crucial. The approach focuses on giving users control while providing helpful assistance. It uses a new technique to quickly recognize specific situations using basic sensor data, even when the data is uncertain. This innovative method could be a valuable tool for pattern recognition in human-assistive devices. However, it might face challenges in complex real-world settings with diverse obstacles. In addition, relying solely on simple sensor data might limit its ability to make precise decisions in uncertain situations, potentially leading to suboptimal path planning results.

Additionally, a recent study [28] introduces an innovative telemedicine method that utilizes robots for certain medical procedures. This pioneering approach involves the use of automated robotic systems, reducing operation time and the requirement for extensive robot training. With only a few actions, this automated method improves the quality of medical procedures, offering exciting possibilities for telemedicine in robot-assisted healthcare. This research has potential benefits in telemedicine and robotic-assisted healthcare, but limitations like adapting to different patients, concerns about accuracy, and needing more real-world testing and refinement exist.

In the context of the COVID-19 pandemic, P. Manikandan et al. [29] highlighted the crucial role of robotics in healthcare. This research emphasizes the importance of medical robots in various medical tasks. Medical robots help reduce human-to-human contact, improve cleaning and sterilization, and provide support in quarantine areas, thereby reducing risks for healthcare workers. The proposed system aims to aid healthcare professionals in delivering essential supplies to those who require them. The robot offers benefits like patient monitoring and precise medication delivery, but it has drawbacks, including limited human interaction and potential technical problems. The cost of implementing such robots in healthcare settings is also a concern.

C. Gap in the Current Research on Path Planning for Robots

There is ongoing research in this area to address the challenges discussed earlier and develop more robust and efficient path-planning algorithms for dynamic environments. One approach uses machine learning to improve path-planning in dynamic environments [30]. Machine learning can be used to learn the behavior of dynamic obstacles and to predict future changes in the environment. This can help path-planning algorithms to adapt to changes in the environment more quickly and effectively. Another approach uses distributed path-planning algorithms [31]. Distributed path-planning algorithms have the potential to break down the path-planning task into smaller, independent subtasks. This approach enhances the scalability of path-planning algorithms, making them better suited for navigating extensive and intricate environments. A third approach uses hybrid path-planning algorithms [32] and [33]. Hybrid path-planning algorithms merge various path-planning techniques to enhance the efficiency of path-planning in environments that are in continuous change. For instance, a hybrid approach could involve employing a sampling-based algorithm for swift exploration of the search area, complemented by a local planner to fine-tune the path.

However, there are a number of issues with current studies on path-planning approaches in terms of handling dynamic obstacles. For instance, lack of real-time adaptability and responsiveness [34]. Many path-planning algorithms are not able to adapt to changes in the environment in real time. This can be a problem in dynamic environments, where the environment is constantly changing. A second issue is the inability to deal with uncertainty and unpredictability [35]. Many path-planning algorithms are not able to deal with uncertainty and unpredictability in the environment. This can be a problem in dynamic environments, where the environment is often uncertain and unpredictable. A third issue is the inability to balance collision avoidance and smoothness [36]. Many path-planning algorithms are not able to balance collision avoidance and smoothness. This can lead to paths that are either too safe (or slow) or too risky (and fast). A fourth issue is the computational complexity [37]. The environment in dynamic settings is often uncertain and unpredictable, which means that the path-planning algorithm must be able to make decisions based on incomplete information [36]. This can be a difficult task, as it requires the algorithm to be able to estimate the probability of different outcomes and to make decisions that minimize the risk of collisions. Many path-planning algorithms are computationally expensive. This can make it difficult to find a path in real time or for large environments.

The most significant gap that proposed GA-PRM algorithm is trying to address is the need for efficient and adaptable path-planning algorithms in healthcare environments. This gap is significant as it directly influences the practical application of robotics in healthcare, which requires a unique set of capabilities compared to other industries. While the literature review highlights various advancements in robotics and path planning in healthcare, it also underscores the complexity and specific challenges of healthcare settings. GA-PRM algorithm is attempting to fill the gap of providing a robust and adaptable path-planning solution that can navigate healthcare environments efficiently and safely. Dynamic obstacles, the need for smooth and precise movements, and the requirement for patient safety characterize healthcare environments.

III. METHODOLOGY

In this section, the methodology employed for the development and implementation of the novel path-planning algorithm is discussed. A comprehensive description of the algorithm is provided, clarifying its complexities and fundamental components. Core techniques utilized in the approach are detailed, underscoring their essential role in effective path-planning. The seamless integration of genetic algorithms and probabilistic roadmaps is explained, displaying how this combination enhances the algorithm's performance and efficiency. Modifications or enhancements to existing algorithms are highlighted, spotlighting how these innovations and advancements refine the approach.

A. Techniques

The GA-PRM algorithm is a path-planning algorithm that combines the strengths of two other algorithms: Genetic Algorithm (GA) and Probabilistic Roadmaps (PRM).

The GA component of GA-PRM is inspired by natural selection. It uses a population-based search approach to iteratively evolve a set of candidate paths. Each candidate path is represented as a string of waypoints in the robot's configuration space. The fitness of each candidate path is evaluated based on its length and how well it avoids obstacles. The fittest individuals from each generation are selected to contribute to the next generation, which helps the algorithm converge to optimal solutions.

The PRM component of GA-PRM constructs a roadmap of the workspace. The roadmap is a graph that represents all the feasible paths between random configurations in the workspace. Random configurations are sampled uniformly across the workspace, and collision checking ensures that they avoid obstacles. The roadmap is built by connecting configurations through collision-free edges.

The GA-PRM algorithm leverages the strengths of both the GA and PRM algorithms. The GA component enables the algorithm to search for optimal paths in a large and complex search space, while the PRM component ensures that the algorithm finds feasible paths even in cluttered environments. The combination of these two algorithms results in a path-planning algorithm that is both efficient and effective.

The core components and techniques used in the algorithm are as follows:

1. **Roadmap generation:** A roadmap is a graph that represents the possible paths that a robot can take in an environment. Randomly sampling configurations within the workspace and then connecting them with collision-free paths generate it. The roadmap construction accounts for the presence of static obstacles, ensuring that the robot can traverse through open paths.

2. **Genetic algorithm:** The genetic algorithm is a potent tool for optimizing robot path planning. It iteratively improves the path by evolving potential routes through generations. It starts with a population of path segments, each representing a possible way for the robot to reach its goal while avoiding obstacles. These segments are evaluated based on how well they work. Parents are selected from the population based on their effectiveness and used to create new paths through crossover and mutation. Crossover involves parents exchanging parts of their paths to make new ones, while mutation makes small changes to paths. This process continues over generations until a suitable path is found, allowing the robot to reach its goal efficiently.

3. **Handling dynamic obstacles:** To ensure robots navigate safely in dynamic environments, there are two key methods: potential field-based path-planning and genetic algorithms. Potential field-based planning calculates forces guiding the robot towards its goal and away from obstacles, allowing it to adapt its path in real-time as conditions change. Genetic algorithms, on the other hand, use iterations to find the best paths by considering various factors like distance and efficiency, making them suitable for navigating complex spaces with both static and moving obstacles. For instance, in a busy hospital, robots can first use potential field-based planning to avoid patients and staff, and then optimize their route with genetic algorithms for safe and efficient navigation. In healthcare settings, ultrasonic sensors can complement infrared (IR) sensors by detecting various types of moving objects and equipment, ensuring collision-free movement.

4. **Path smoothing:** Path smoothing is a crucial part of path-planning, especially in complex and dynamic settings. It reduces noise and uncertainties caused by sensors and moving obstacles, making the path smoother and more reliable. Kalman filtering is a statistical technique used for this purpose. It estimates the robot's position and speed by combining previous estimates and noisy measurements. This process repeats to refine the estimate, resulting in a more accurate path.

5. **Performance metrics:** The performance of the new

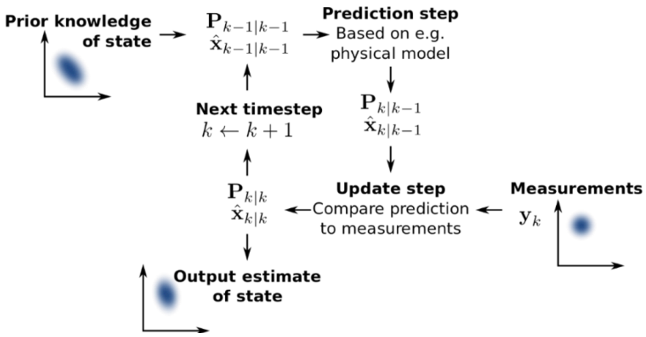


Fig. 1. Basic concept of Kalman filtering.

algorithm is evaluated using three metrics: average path length, average computational time, and average smoothness. These metrics offer insights into the efficiency and effectiveness of the algorithm across various scenarios.

B. Kalman Filtering

The proposed algorithm, GA-PRM, effectively addresses the issue of balancing collision avoidance and smoothness in path-planning, especially in dynamic environments like healthcare settings, through the implementation of the Kalman filter [38] for path smoothness. Kalman Filtering is a widely used technique in the GA-PRM algorithm to improve path planning accuracy. It is a systematic approach, which continuously updates and refines its understanding of a system's performance using sensor data and their uncertainties. This method is valuable because it combines noisy sensor data with dynamic models to provide a more reliable assessment of the system's status over time. Kalman Filtering operates in two stages: prediction and adjustment based on new measurements, making it particularly useful for real-time applications.

Figure 1 illustrates the core stages of Kalman Filtering, encompassing prediction and update. It visually represents how the filter not only monitors the average state value but also estimates the degree of variation. As shown in this figure, the Kalman filter maintains information about the system's estimated state and the level of uncertainty associated with this estimate. This estimation is refined through the utilization of a model depicting how the state changes over time and the inclusion of measurements. Specifically, denotes the estimate of the system's state at a given time step k before considering the k-th measurement y_k, while represents the corresponding level of uncertainty.

C. Algorithm Description

In terms of its contribution to the operation of the GA-PRM algorithm, the Kalman filter is crucial in balancing a robot's path to avoid obstacles while maintaining smooth motion. The genetic algorithm (GA) can be overly cautious or too risky, affecting navigation speed. To address this, the GA-PRM

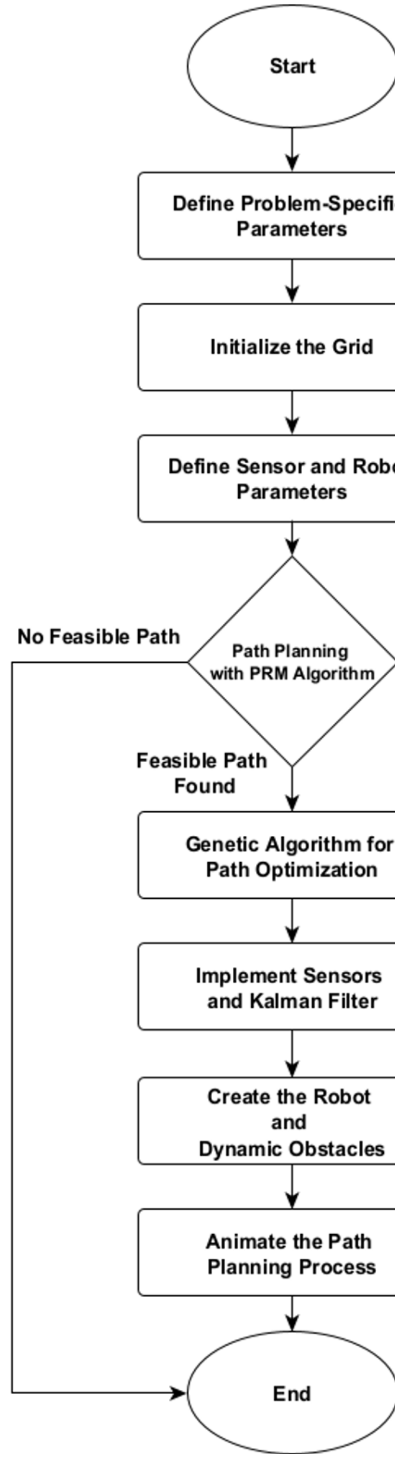


Fig. 2. Flowchart describing the systematic execution of the GA-PRM algorithm.

algorithm uses the Kalman filter. This filter uses sensor data to accurately estimate the robot's position, reducing errors. Smoothing the path allows the GA-PRM find a balance between avoiding collisions and smooth movement. The Kalman filter updates the robot's position based on sensor data, ensuring the robot can navigate safely and efficiently, even in changing environments, without putting safety at risk or disrupting medical tasks.

The GA-PRM algorithm, as described in the flowchart in Figure 2, follows a step-by-step process for efficient path planning. It begins with initialization and then utilizes the Probabilistic Roadmap (PRM) technique to create a roadmap by randomly sampling and connecting configurations. Subsequently, it generates a path segment from the current position towards the goal, considering forces to navigate obstacles. The generated path's feasibility is checked for goal reachability. To optimize the path, a Genetic Algorithm is employed, creating and refining potential paths within a population. Continuous goal checking occurs throughout execution. Dynamic obstacles (people) are introduced by randomly moving them within the workspace to simulate dynamic environments. The Kalman Filter enhances path estimation accuracy and minimizes sensor measurement noise. The algorithm concludes its execution, providing efficient path planning while accommodating obstacles and dynamic scenarios.

D. Modifications and Improvements

In this section, the modifications and improvements made to existing path-planning algorithms to develop a novel approach are described. These enhancements aim to address specific limitations and leverage the strengths of the individual algorithms. The modifications are designed to work together and perfectly within the framework of the proposed algorithm.

1. **Combining Strengths:** The algorithm blends Genetic Algorithm (GA) and Probabilistic Roadmaps (PRM) in a unique way. GA explores possibilities, while PRM carefully maps valid paths. They work together: GA suggests paths, and PRM refines them. This combination improves path quality and speeds up finding the best path.

2. **Faster Computing:** Smart data handling, like the R-tree index, speeds up PRM's work, making it suitable for quick path planning. Additionally, advanced methods like the Kalman filter improve sensor data processing, reducing the workload and making it even better for real-time tasks.

3. **Balancing Goals:** The algorithm considers various factors like path length, smoothness, and computation time all at once. This flexible approach gives different options, allowing users to balance these factors as needed.

Using an R-tree data structure [39] significantly boosts essential tasks like finding nearby positions and checking for collisions. This unique data structure, the R-tree index, smartly organizes spatial data. It does this by creating an efficient system where information is stored in a way that allows extremely fast access. This dynamic indexing method is a significant advancement, especially for real-time applications. Unlike older indexing methods, which struggle with objects of varying sizes in complex, multi-dimensional spaces, the R-tree brings a simplicity that greatly enhances the robot's ability to navigate swiftly and accurately.

IV. Problem Formulation

This propose approach aims to find the best path for a mobile robot in a workspace with both stationary and moving obstacles. This involves finding a sequence of grid cells that connect the start and end points while avoiding collisions with obstacles. This section formally defines the path-planning problem and describes the system model.

A. Path-Planning Formalization

Path-planning is the process of finding a safe and efficient path for a robot to move from one point to another in an environment. The environment may contain obstacles, such as walls and furniture, as well as dynamic obstacles, such as people and vehicles. The path-planning problem can be broken down into the following steps:

- **Representation of the workspace:** The workspace is represented as a grid-based environment. This means that the environment is divided into a grid of cells, and each cell represents a specific location in the workspace. This representation makes it easier for the robot to plan its path and avoid obstacles.

- **Definition of the start and goal points:** The start point is the robot's initial position, and the goal point is the desired destination. The path-planning algorithm must find a path that connects the start and goal points without colliding with any obstacles.

- **Identification of static obstacles:** Static obstacles are obstacles that do not move during the planning process. Their coordinates on the grid identify these obstacles, and they are marked as such in the grid representation.

- **Accounting for dynamic obstacles:** Dynamic obstacles are obstacles that move during the planning process. These obstacles are represented by their respective motion models, which describe how they move over time. The path-planning algorithm must take into account the motion models of dynamic obstacles when planning the robot's path.

- **Consideration of motion constraints:** The robot's motion is subject to specific constraints, such as maximum velocity, acceleration, and turning radius. These constraints must be taken into account when planning the robot's path, ensuring that the path adheres to the robot's physical limitations.
- **Optimization of the path:** The goal of the path-planning algorithm is to find an optimal path, which is a path that minimizes the total distance traveled from the start point to the goal point while avoiding collisions with all obstacles. The path-planning algorithm may use a variety of techniques to optimize the path, such as genetic algorithms or probabilistic roadmaps.

The initial positions of the dynamic obstacles are randomly generated within the workspace. The number of dynamic obstacles is determined is set to 50 in our code. This means that when we run the simulation, 50 dynamic obstacles will be randomly placed within the workspace at the beginning of the simulation. On the other hand, the speed of dynamic obstacles is set to 0.3 in the code, which represents the maximum velocity that dynamic obstacles can have. The velocity for each obstacle is randomly selected from the range of 0 to $velocity_{max}/2$. This randomization adds variability to the dynamic obstacle movement, making their speed dynamic and unpredictable.

B. System Model

In this section, we examine the use of a two-wheel differential mobile robot for path planning. This particular robot possesses the ability to control the speed of its two driving wheels, granting it the flexibility to execute a range of trajectory movements, such as moving in straight lines, making turns, and performing circular maneuvers. We establish a kinematic model by tracking the robot's consecutive positions, as illustrated in Figure 3.

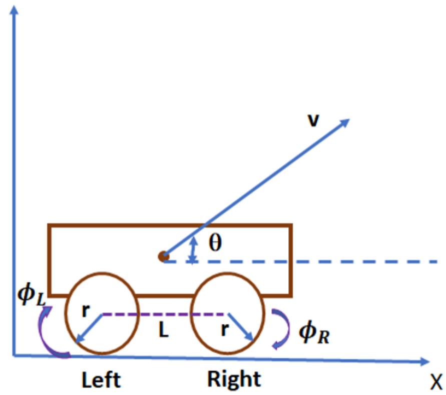


Fig. 3. Illustrated example of a path-planning robot diagram.

1) The Motion Model of Mobile Robotic System

In the field of mobile robotics, kinematics plays a pivotal role in comprehending how a robot's mechanical components, such as wheels, sensors, and DC motors, operate and position themselves within their environment. It provides insights into how a robot interacts with its surroundings and is indispensable for various tasks, including path planning, obstacle avoidance, and robot control. Kinematics enables engineers and researchers to establish precise models of a robot's behavior and make predictions about its actions in diverse scenarios. Leveraging mathematical and geometrical principles, kinematic models are formulated to illustrate the relationship between the robot's movements and the commands it receives. Moreover, robot kinematics proves vital for exacting tasks like navigation, self-localization, and object manipulation, serving as the foundation for algorithms that empower robots to reach specific destinations, evade collisions, and execute tasks with precision and efficiency. In the proposed robot kinematics model, a differential drive robot with odometry updates is considered. The robot's motion can be described using equations (1) to (3):

$$\frac{dx}{dt} = \frac{r}{2}\left(\frac{d\phi_L}{dt} + \frac{d\phi_R}{dt}\right)\cos(\theta) \tag{1}$$

$$\frac{dy}{dt} = \frac{r}{2}\left(\frac{d\phi_L}{dt} + \frac{d\phi_R}{dt}\right)\sin(\theta) \tag{2}$$

$$\frac{d\theta}{dt} = \frac{r}{2L}\left(\frac{d\phi_R}{dt} - \frac{d\phi_L}{dt}\right) \tag{3}$$

Where:
(x,y): Robot coordinates in the global frame.
θ: Orientation angle of the robot.
r : Wheel radius L : Wheelbase (distance between the wheels).
$d\phi_L/dt, d\phi_R/dt$: Angular velocities of the left and right wheels.

Equations (1) to (3) represent the kinematic equations for the motion of a differential-drive mobile robot. These equations describe how the robot's position and orientation change over time based on the angular velocities of its left and right wheels.

It is worth noting that the orientation angle of the robot gives the direction in which the robot is pointed. Typically, θ is measured relative to some reference, such as the positive x-axis. For example, if θ is 0 degrees, the robot is facing directly along the positive x-axis. If θ is 90 degrees, the robot is facing along the positive y-axis. If θ is 180 degrees, the robot is facing directly along the negative x-axis, and so on. In addition, the velocity variables in Equation (3) are crucial

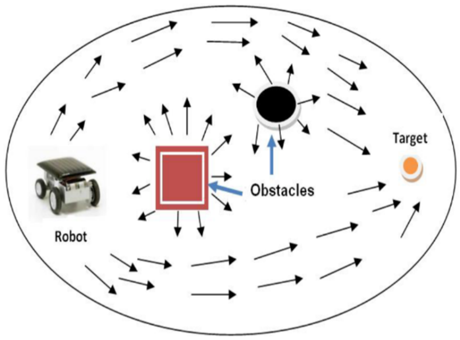


Fig. 4. Repulsive field method for a mobile robot.

for controlling the robot's motion. By adjusting the angular velocities of the left and right wheels $(d\phi_L/dt, d\phi_R/dt)$, the mobile robot can be controlled to move forward or turn.

2) *The Repulsive Field Method*

The Repulsive Field Method [40] is a form of potential field-based path planning in robotics. In this method, the environment is represented as a field of repulsive forces, which push the robot away from obstacles. As shown in Fig. 4, each obstacle in the environment generates a repulsive field, and the overall repulsive force experienced by the robot is the sum of these fields. The robot navigates by moving in the direction of the resultant force, attempting to minimize the potential energy associated with the repulsive fields. This approach allows the robot to avoid obstacles while reaching its destination. The repulsive field method usually consists of essential elements, which encompass settings to regulate repulsion intensity, methods for identifying and pinpointing obstacles in the surroundings, and forces or fields with an attractive nature employed to direct the robot towards a desired destination. However, it is important to note that although the repulsive field method can prove successful in specific situations, it does have constraints. These limitations include challenges in dealing with intricate environments characterized by narrow pathways and the possibility of the robot becoming trapped in localized low-performance states.

3) *Obstacle Avoidance Model*

A repulsive potential field method is used to avoid obstacles. For a circular obstacle located at (ox, oy) with a radius r_{obs}, the potential field is given by equation (4) [41]:

$$U_{\text{obs}} = k_{\text{obs}} \left(\frac{1}{\sqrt{(x-ox)^2 + (y-oy)^2}} - \frac{1}{r_{\text{obs}}} \right)^2 \quad (4)$$

Where:
U_{obs}: The potential energy associated with obstacle avoidance.
k_{obs}: Gain parameter for obstacle avoidance.
(x, y): Robot coordinates.
(ox, oy): Obstacle coordinates.

This potential field creates a repulsive force that pushes the robot away from the obstacle. In equation (4), the first part calculates the inverse of the distance between the robot's current position (x, y) and the center of the circular obstacle (ox, oy). It represents the strength of the repulsive force; shorter distances lead to higher repulsion. Finally, the square of the difference is taken to enhance the repulsive effect and ensure it is always positive.

Fundamentally, equation (4) calculates a repulsive potential field, U_{obs} , that increases, as the robot gets closer to the circular obstacle. The gain parameter, k_{obs}, controls the strength of this repulsion, and the distance-based terms determine the shape of the repulsive force field. This potential field is commonly used in robotics for obstacle avoidance, where robots move away from obstacles by following the gradient of this field.

4) *Dynamic Obstacle Model*

When dealing with a moving obstacle, its position can be dynamically described as a function of time, denoted as T. This dynamic position is calculated using equations (5) and (6):

$$ox_{\text{dynamic}}(T) = ox_{\text{initial}} + v_{\text{obstacle}} \cos(\theta_{\text{obstacle}})T \quad (5)$$

$$oy_{\text{dynamic}}(T) = oy_{\text{initial}} + v_{\text{obstacle}} \sin(\theta_{\text{obstacle}})T \quad (6)$$

Where:
$(ox_{initial}, oy_{initial})$: These represent the initial coordinates of the moving obstacle, specifying where the obstacle is located at the start of its movement.
$v_{obstacle}$: This parameter signifies the velocity of the moving obstacle. It defines how fast the obstacle is moving through space.
$\theta_{obstacle}$: The initial orientation of the moving obstacle is represented by this angle. It indicates the direction in which the obstacle is initially facing.

Equations (5) and (6) enable to track the dynamic position of a moving obstacle over time, allowing for effective collision avoidance and path planning in dynamic environments.

5) *Optimal Trajectory*

An optimal trajectory in mobile robot path planning is the most efficient and effective path the robot can take to reach its destination while meeting specific criteria. "Optimal" implies that this path is the best choice based on defined metrics. The key aspects of optimal trajectories include efficiency, obstacle avoidance, smoothness, consideration of dynamic constraints, task-specific objectives, global vs. local optimization, and real-time adaptation [42].

Efficiency is a critical criterion, focusing on minimizing factors like time, energy consumption, or distance traveled [41]. Obstacle avoidance ensures safe navigation around obstacles. Smooth trajectories with minimal changes in direction and velocity enhance stability and reduce physical deterioration. Dynamic constraints account for limits on acceleration or deceleration. Task-specific objectives vary depending on the mission, while trajectory optimization can be global or local. In dynamic environments, real-time adaptation may be necessary.

6) *Objective Function for Optimal Trajectory*

In the pursuit of finding the optimal trajectory for a mobile robot, an objective function that plays a pivotal role in balancing various objectives is employed, such as minimizing time while simultaneously avoiding obstacles. This objective function, denoted as J, is defined in equation (7).

$$J = \omega_{time} T + \omega_{obstacle} U_{obs} \tag{7}$$

Where:
ω_{time} and $\omega_{obstacle}$: These are weighting factors assigned to the time and obstacle avoidance components of the objective function, respectively. They allow to prioritize one aspect over the other based on their relative importance in the context of the task. T: This represents the total time required for the robot to reach its goal along a specific trajectory. It is a crucial metric in scenarios where minimizing the time of traversal is a primary objective.

The overarching aim is to minimize the value of the cost function J. Through optimizing the trajectory, a balance between minimizing the time taken to reach the goal while also considering the importance of avoiding obstacles is a stroke to ensure the safety and efficiency of the robot's path.

V. Experimental Setup

In this section, we explain how we set up our experiments to test different path-planning algorithms. We start by describing the simulation environment we used for testing and then discuss how we configured the performance evaluation. We also explain the metrics and criteria we used to measure how well the algorithms worked. This experimental setup is crucial because it ensures that the results we collect from the evaluation are reliable and unbiased.

A. *Simulation Environment*

The code was tested in a 2D healthcare simulation with static and moving obstacles. The robot used sensor data including ultrasonic and IR readings to plan its path efficiently, and avoiding collisions. This setup simulated real-world healthcare scenarios, evaluating the robot's ability to navigate a hospital environment safely while avoiding obstacles. The simulation environment was carefully tailored to meet the specific requirements of the path-planning problem, making use of the following adjustments:

- **Robot Model:** The robot utilized in the simulation is a wheeled mobile robot designed for efficient workspace navigation. It features motorized wheels for motion control and an array of sensors, including ultrasonic, infrared (IR), temperature and humidity sensors. The robot's kinematics guarantee smooth and precise movements, and its dynamic capabilities enable it to adapt paths based on sensor inputs and potential field calculations during planning.
- **Obstacles:** Both static and dynamic obstacles emulate real-world situations in the simulation. Static obstacles are immovable rectangular blocks carefully placed to create challenging navigation scenarios. In contrast, dynamic obstacles represent moving individuals (people) within the workspace. The motion of dynamic obstacles is randomized to replicate unpredictable human movement, requiring the robot to skillfully navigate while avoiding collisions with them.
- **Sensor Simulation:** The simulation incorporates sensor emulation to enable the robot's sensing and navigation abilities. The ultrasonic sensor provides short-range distance measurements, detecting nearby obstacles, while the IR sensor offers medium-range distance readings. The temperature and humidity sensor supply environmental data, allowing the robot to adapt its behavior to the prevailing conditions. The sensor simulation ensures the robot effectively perceives and responds to its surroundings.

B. *Performance Metrics*

The effectiveness of the algorithm was evaluated through an examination of three critical performance metrics: average path length, average computational time, and average smoothness. These metrics were computed using straightforward mathematical formulas. • Average path length: measures the average length of the paths generated by each algorithm. A shorter average path length means that the robot can reach its destination using the minimum distance possible. This is more efficient, as it reduces energy consumption and overall travel time.

- **Average computational time:** measures the average time it takes each algorithm to calculate a feasible path from the starting point to the goal. Faster computational times are especially important in dynamic environments, as they allow for real-time or near-real-time path-planning. A more efficient algorithm can reduce planning delays, allowing the robot to quickly respond to changes in the environment and dynamic obstacles.

- **Average smoothness measures:** the continuity and absence of abrupt changes in the robot's motion during navigation. Paths with smoother trajectories lead to more stable and comfortable robot movements. This is especially important when the robot is interacting with humans or in the presence of dynamic obstacles. A smoother path reduces the risk of collisions, improves user comfort, and ensures safer robot navigation in complex environments.

VI. Results

A. Result of Simulation

Fig. 5 visually represents the robot's movement in the simulated workspace. The workspace is divided into four sections, each corresponding to a different quadrant. Subfigure (2-A) shows the robot's position in quadrant 1, (2-B) in quadrant 2, (2-C) in quadrant 3, and (2-D) in quadrant 4. Each part of the diagram illustrates the robot's path as it moves through its respective workspace section. Gray boxes represent stationary obstacles, while small black dots depict moving obstacles.

B. Result of Comparison

Fifty tests were conducted to compare the performance of four path-planning algorithms: A*, RRT, Genetic, and PRM. In each test, the robot had to move from the starting point to the target point in a predefined area. The same specific parameters, workspace size, grid size, fixed and moving obstacles, and sensor settings were used for all four algorithms. The choice of comparing the GA-PRM algorithm with the A*, RRT, Genetic, and PRM algorithms is based on their well-established effectiveness and relevance in motion plansning and optimization. A* stands out for its efficiency in finding the shortest paths in grid-based environments [43]. RRT excels in high-dimensional spaces with dynamic obstacles due to its probabilistic completeness and fast convergence [1]. Genetic algorithms offer versatility in optimizing complex spaces, making them valuable for benchmarking global optimization by the GA-PRM algorithm [44]. PRM, a sampling-based method, is known for its simplicity and efficiency in roadmap construction, making it a suitable comparison for evaluating the GA-PRM algorithm's roadmap generation performance in high-dimensional spaces [3]. Other algorithms, while fundamental and well-established path-finding algorithms, may have characteristics that make them less suitable for direct comparison in the context of my research. For example, Dijkstra's Algorithm is known for its optimality in finding the shortest path in static environments. However, it does not naturally handle dynamic obstacles or provide probabilistic roadmaps for path planning [45]. Table I provides a comparison between the GA-PRM algorithm and four other path-planning algorithms (A*, RRT, Genetic algorithm, and PRM). This comparison is based on three performance metrics: average path length, average computational time, and average smoothness.

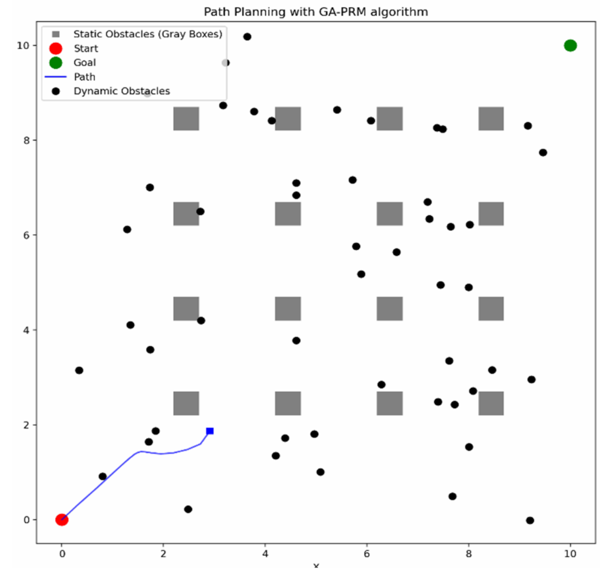
TABLE I. Performance Comparison Between GA-PRM Algorithm and A*, RRT, Genetic, and PRM Algorithms

No.	Algorithm	APL	ACT	AS
1	Proposed GA-PRM	25.6235	0.6881	0.3133
2	A*	29.1758	0.7452	0.0803
3	RRT	36.2037	0.6209	0.2911
4	Genetic	37.43	0.7147	1.5308
5	PRM	26.8700	0.9962	1.8543

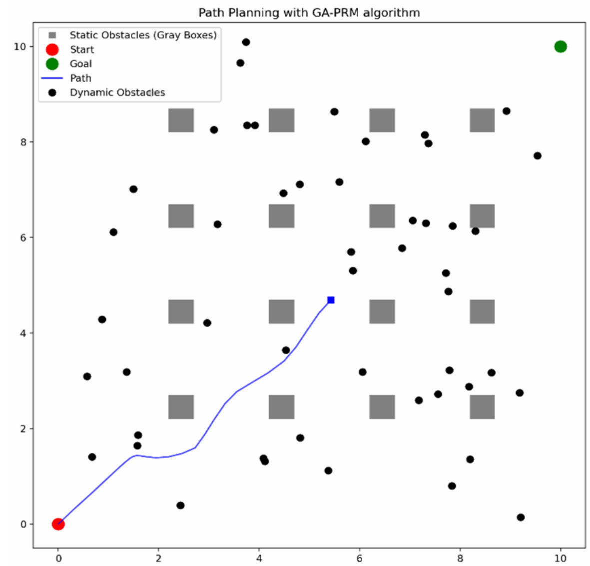
As can be seen from Table I, the GA-PRM algorithm achieves the shortest average path length among all the algorithms, with an APL of 25.6235 units. This indicates that the current algorithm is successful in finding paths that are, on average, shorter than those generated by the other algorithms. While GA-PRM excels in shortest average path length, it requires a moderate amount of computational time, with an ACT of 0.6881 seconds. This indicates that it strikes a balance between path length and computational efficiency. In addition, GA-PRM produces paths with a relatively high average smoothness (AS of 0.3133). This suggests that it achieves a good balance between path length and smoothness, resulting in paths with less abrupt changes in direction.

The GA-PRM algorithm demonstrates notable strengths that make it well suited for deployment in a hospital environment. Firstly, it excels in generating shorter and smoother paths, which can be pivotal in healthcare settings where precision and patient safety are paramount. These characteristics contribute to minimizing the time taken for robots to navigate through hospital corridors and reduce the risk of unexpected obstacles. Secondly, the algorithm's ability to optimize path smoothness ensures that robotic movements are fluid and less likely to cause disruptions or discomfort to patients, staff, and visitors. Although it may exhibit slightly longer computation times, the trade-off is justifiable in healthcare, given the emphasis on safe and efficient navigation within a controlled and predictable environment. Overall, the GA-PRM algorithm aligns well with the requirements of a hospital setting by prioritizing path quality and patient well-being.

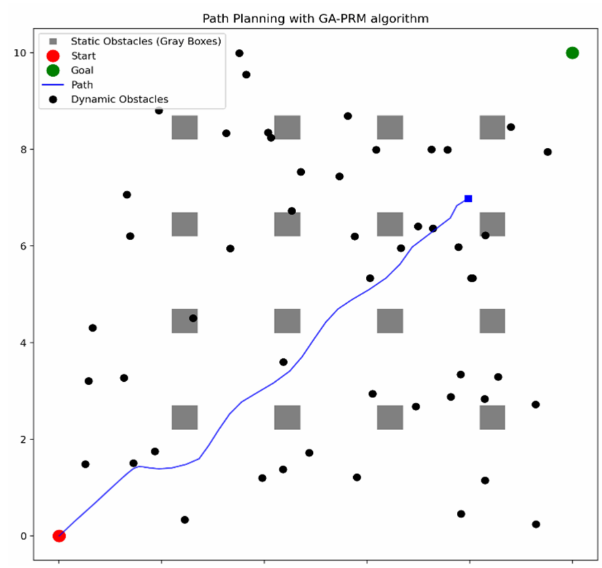
In Fig. 6, a comparison is made among various path-planning algorithms using three key performance measures given above. The algorithms under evaluation include GA-PRM, A*, RRT, Genetic, and PRM. The chart illustrates these



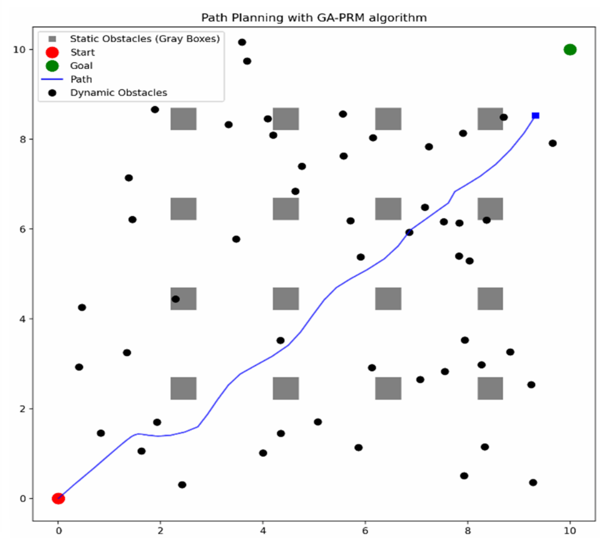
(a) Robot in quadrant 1.



(b) Robot in quadrant 2.



(c) Robot in quadrant 3.



(d) Robot in quadrant 4.

Fig. 5. Path-planning with GA-PRM algorithm.

performance metrics for each algorithm with three distinct lines. Each line corresponds to one of the performance measures, and the *x*-axis displays the names of the algorithms. On the *y*-axis, we can see the average value for each metric. To make it clearer, circular markers are used to denote the data points for each algorithm. This visual representation enables an easy understanding of how each algorithm performs in terms of these metrics.

VII. Discussion

The comparison experiment between our new path-planning method and existing ones, presented in Table I, reveals important insights. These results provide valuable information about how our approach could be applied in practical, real-world situations and how it might advance the field.

1. **Path length:** The GA-PRM algorithm showed an average path length of 25.62 units. This indicates that the robot's paths planned by the GA-PRM algorithm were

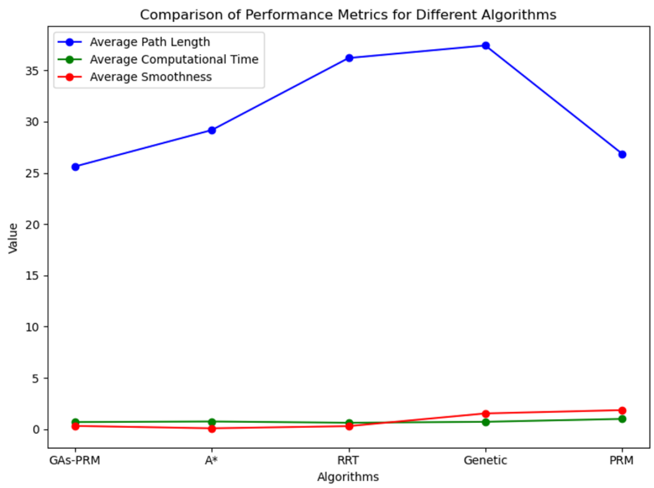


Fig. 6. Path-planning with GA-PRM algorithm.

more direct and efficient in reaching the goal while avoiding obstacles. The GA-PRM's combination of genetic algorithms and probabilistic roadmaps contributed to the algorithm's ability to explore the workspace effectively and find shorter paths. This characteristic is of utmost importance in various autonomous robotic applications, as shorter paths translate to reduced energy consumption and faster completion of tasks, which can be crucial for time-sensitive missions.

2. **Computational time:** The GA-PRM algorithm demonstrated an average computational time of (0.6881), which was faster in generating path plans as compared to the other algorithms (except for the RRT algorithm, which took 0.6209 seconds). The efficiency of the GA-PRM algorithm can be attributed to its use of genetic algorithms and probabilistic roadmaps, which allowed for effective exploration of the configuration space while keeping the computation time low. This characteristic makes the new algorithm suitable for real-time applications where prompt decision-making is imperative.

3. **Smoothness:** The GA-PRM algorithm achieved a moderate level of smoothness (0.3133) in its planned paths. While the PRM algorithm had the smoothest paths with the highest average smoothness (1.8543), the GA-PRM algorithm managed to strike a balance between path smoothness and path length, leading to an overall more optimal solution. Smoothness in the path is crucial to guaranteeing stable and controlled movement of the robot, particularly in situations with strict safety demands.

Although the proposed GA-PRM algorithm has shown promising performance in experiments, there are still areas for improvement.

First, the algorithm's performance depends heavily on the choice of parameters, such as the mutation rate and population size. These parameters can significantly influence the quality of the generated paths. Fine-tuning the genetic algorithm's parameters and exploring alternative genetic operators could further improve the algorithm's convergence and solution quality.

Second, as the environment becomes more complex and the count of dynamic obstacles rises, the algorithm's execution time also experiences an increase. Although genetic algorithms and probabilistic roadmaps naturally bring in elements of randomness and adaptability, it remains crucial to investigate strategies that can enhance computational efficiency when dealing with larger and dynamic settings.

Another aspect to consider is how to handle dynamic obstacles with unpredictable trajectories. The current algorithm models dynamic obstacles as random walkers within the workspace. However, incorporating predictive methods or learning algorithms to anticipate the future trajectories of these obstacles could lead to more predictive and preemptive path-planning behavior for the robot.

Finally, the proposed algorithm currently assumes known and fixed sensor ranges for ultrasonic, IR, and other sensor types. Incorporating adaptive sensor models that can dynamically adjust their ranges based on the environment's characteristics could enhance the algorithm's robustness in handling varying obstacle densities.

VIII. Conclusion

The GA-PRM algorithm holds significant importance in robotics and path planning, particularly in dynamic and complex environments such as healthcare settings. Through combining the power of genetic algorithms with the efficiency of probabilistic roadmaps, GA-PRM excels in finding adaptable and obstacle-aware paths for robots. This characteristic is crucial for ensuring safe and efficient navigation in environments where obstacles and conditions are subject to frequent changes, ultimately contributing to the reliable and effective deployment of robots in real-world scenarios, including healthcare, logistics, and more.

Future research could optimize GA-PRM parameters, integrate advanced sensors for context-aware planning, explore multi-robot systems, and leverage hardware advancements for real-time implementation.

Conflict of Interest

The authors have no conflict of relevant interest to this article.

References

[1] S. M. LaValle and J. J. Kuffner Jr, "Randomized kinodynamic planning," *The international journal of robotics research*, vol. 20, no. 5, pp. 378–400, 2001.

[2] M. Cardona, F. Cortez, A. Palacios, and K. Cerros, "Mobile robots application against covid-19 pandemic," in *IEEE ANDESCON*, pp. 1–5, 2020.

[3] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[4] S. LaValle, "Rapidly-exploring random trees: A new tool for path planning," *Research Report 9811*, 1998.

[5] F. Duchoň, A. Babinec, M. Kajan, P. Beňo, M. Florek, T. Fico, and L. Jurišica, "Path planning with modified a star algorithm for a mobile robot," *Procedia engineering*, vol. 96, pp. 59–69, 2014.

[6] A. Stentz, *The D* algorithm for real-time planning of optimal traverses*. Carnegie Mellon University, the Robotics Institute, 1994.

[7] E. W. Dijkstra, "A note on two problems in connexion with graphs," in *Edsger Wybe Dijkstra: His Life, Work, and Legacy*, pp. 287–290, 2022.

[8] A. Al-Jumaily and C. Leung, "Wavefront propagation and fuzzy based autonomous navigation," *International Journal of Advanced Robotic Systems*, vol. 2, no. 2, p. 10, 2005.

[9] J. H. Holland, "Genetic algorithms," *Scientific american*, vol. 267, no. 1, pp. 66–73, 1992.

[10] C. W. Warren, "Global path planning using artificial potential fields," in *1989 IEEE International Conference on Robotics and Automation*, pp. 316–317, 1989.

[11] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE computational intelligence magazine*, vol. 1, no. 4, pp. 28–39, 2006.

[12] S. Garrido, L. Moreno, D. Blanco, and F. Martín Monar, "Robotic motion using harmonic functions and finite elements," *Journal of intelligent and Robotic Systems*, vol. 59, pp. 57–73, 2010.

[13] L. Blasi, E. D'Amato, M. Mattei, and I. Notaro, "Path planning and real-time collision avoidance based on the essential visibility graph," *Applied Sciences*, vol. 10, no. 16, p. 5613, 2020.

[14] R. Gonzalez, M. Kloetzer, and C. Mahulea, "Comparative study of trajectories resulted from cell decomposition path planning approaches," in *IEEE 2017 21st International Conference on System Theory, Control and Computing (ICSTCC)*, pp. 49–54, 2017.

[15] J.-M. Lien, "Hybrid motion planning using minkowski sums," in *Proceedings of Robotics: Science and Systems IV*, 2008.

[16] Z. D. Hussein, M. Z. Khalifa, and I. S. Kareem, "Optimize path planning for medical robot in iraqi hospitals," *Engineering and Technology Journal*, vol. 33, no. 5, pp. 1009–1022, 2015.

[17] B. Fang, G. Mei, X. Yuan, L. Wang, Z. Wang, and J. Wang, "Visual slam for robot navigation in healthcare facility," *Pattern Recognition*, vol. 113, p. 107822, 2021.

[18] M. Takahashi, T. Suzuki, H. Shitamoto, T. Moriguchi, and K. Yoshida, "Developing a mobile robot for transport applications in the hospital domain," *Robotics and Autonomous Systems*, vol. 58, no. 7, pp. 889–899, 2010.

[19] Y. Jung, Y. Kim, W. H. Lee, M. S. Bang, Y. Kim, and S. Kim, "Path planning algorithm for an autonomous electric wheelchair in hospitals," *IEEE Access*, vol. 8, pp. 208199–208213, 2020.

[20] D. P. Romero-Martí, J. I. Núnez-Varela, C. Soubervielle-Montalvo, and A. Orozco-de-la Paz, "Navigation and path planning using reinforcement learning for a roomba robot," in *IEEE 2016 XVIII Congreso Mexicano de Robotica*, pp. 1–5, 2016.

[21] X. Huang, Q. Cao, and X. Zhu, "Mixed path planning for multi-robots in structured hospital environment," *The Journal of Engineering*, vol. 2019, no. 14, pp. 512–516, 2019.

[22] Z. Jiao, K. Ma, Y. Rong, P. Wang, H. Zhang, and S. Wang, "A path planning method using adaptive polymorphic ant colony algorithm for smart wheelchairs," *Journal of Computational Science*, vol. 25, pp. 50–57, 2018.

[23] I. T. Kurniawan and W. Adiprawita, "Autonomy design and development for an ultraviolet-c healthcare surface disinfection robot," in *IEEE International Symposium on Electronics and Smart Devices (ISESD)*, pp. 1–6, 2021.

[24] S. Wan, Z. Gu, and Q. Ni, "Cognitive computing and wireless communications on the edge for healthcare service robots," *Computer Communications*, vol. 149, pp. 99–106, 2020.

[25] T. K. Morimoto, J. J. Cerrolaza, M. H. Hsieh, K. Cleary, A. M. Okamura, and M. G. Linguraru, "Design of patient-specific concentric tube robots using path planning from 3-d ultrasound," in *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 165–168, 2017.

[26] H.-M. Gross *et al.*, "Robot companion for domestic health assistance: Implementation, test and case study under everyday conditions in private apartments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5992–5999, 2015.

[27] M. Gillham, B. McElroy, G. Howells, S. Kelly, S. Spurgeon, and M. Pepper, "Weightless neural system employing simple sensor data for efficient real-time round-corner, junction and doorway detection for autonomous system path planning in smart robotic assisted healthcare wheelchairs," in *IEEE Third International Conference on Emerging Security Technologies*, pp. 161–164, 2012.

[28] Y. Zhu, A. Smith, and K. Hauser, "Informative path planning for automatic robotic auscultation," in *ICRA 2021 Workshop on Impact of COVID-19 on Medical Robotics and Wearables Research*, 2021.

[29] P. Manikandan, G. Ramesh, G. Likith, D. Sreekanth, and G. D. Prasad, "Smart nursing robot for covid-19 patients," in *IEEE International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, pp. 839–842, 2021.

[30] B. Wang, Z. Liu, Q. Li, and A. Prorok, "Mobile robot path planning in dynamic environments through globally guided reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6932–6939, 2020.

[31] K. J. O'Hara and T. R. Balch, "Distributed path planning for robots in dynamic environments using a pervasive embedded network," in *IEEE the Third International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2004*, pp. 1538–1539, 2004.

[32] X. Zhong, J. Tian, H. Hu, and X. Peng, "Hybrid path planning based on safe a* algorithm and adaptive window approach for mobile robot in large-scale dynamic environment," *Journal of Intelligent & Robotic Systems*, vol. 99, pp. 65–77, 2020.

[33] Z. Du, D. Qu, F. Xu, and D. Xu, "A hybrid approach for mobile robot path planning in dynamic environments," in *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1058–1063, 2007.

[34] C. Zhou, B. Huang, and P. Fränti, "A review of motion planning algorithms for intelligent robots," *Journal of Intelligent Manufacturing*, vol. 33, no. 2, pp. 387–424, 2022.

[35] K. Karur, N. Sharma, C. Dharmatti, and J. E. Siegel, "A survey of path planning algorithms for mobile robots," *Vehicles*, vol. 3, no. 3, pp. 448–468, 2021.

[36] S. M. LaValle, *Planning algorithms*. Cambridge University Press, 2006.

[37] P. Navya and R. Ranjith, "Analysis of path planning algorithms for service robots in hospital environment," in *IEEE International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pp. 1–6, 2021.

[38] S. Chen, "Kalman filter for robot vision: a survey," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 11, pp. 4409–4420, 2011.

[39] A. Guttman, "R-trees: A dynamic index structure for spatial searching," in *Proceedings of the 1984 ACM SIGMOD international conference on Management of data*, pp. 47–57, 1984.

[40] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control*. John Wiley & Sons, 2020.

[41] H. Choset, K. M. Lynch, S. Hutchinson, G. A. Kantor, and W. Burgard, *Principles of robot motion: Theory, algorithms, and implementations*. MIT Press, 2005.

[42] M. Wahde, "Introduction to autonomous robots." Lecture Notes from the course Autonomous Agents, Chalmers University of Technology, 2012.

[43] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[44] J. R. Sampson, "Adaptation in natural and artificial systems (john h. holland)." Society for Industrial and Applied Mathematics, 1976.

[45] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.