Journal of Artificial Intelligence and Data Science (JAIDA)
e-ISSN: 2791-8335
https://dergipark.org.tr/pub/jaida

Vol. 3, No. 2, pp. 124-137, 2023
Research Article
Received: 05/11/2023 Accepted: 14/12/2023

Navigating in Complex Indoor Environments: A Comparative Study

Alper Hüseyin DOĞAN[1,*], Tarık Veli MUMCU[2]

Abstract

Autonomous robots face significant challenges in path planning and continuous motion planning in indoors due to their ability to navigate within these complex spaces. These complex problems arise in a wide range of application environments, including indoor areas such as corridors, rooms, and similar spaces. This study presents a comparative simulation analysis of path-finding techniques employed for indoor autonomous robot navigation. Conventional path-finding techniques, including Voronoi diagram and potential field, have been selected to illustrate these established methods. However, they were found to be unreliable and insufficient in coping with the intricacies of real-world situations characterised by non-linearity. Various artificial intelligence techniques were evaluated to showcase the superiority of artificial intelligence over conventional methods. The methods included genetics algorithm and neural networks. The use of these artificial intelligence methods proved their ability to handle complex navigation tasks with greater ease and strength, highlighting their vital contribution in overcoming obstacles. Additionally, we utilize the well-known A* algorithm as a benchmark to evaluate and compare the performance of filtering techniques, particularly Kalman and particle filters in the context of path tracking under diverse conditions, including scenarios with gaussian and exponential noise. Through these analyses, we shed light on the performance of Kalman and particle filters when applied in conjunction with the A* algorithm for path tracking, offering valuable insights into their effectiveness in real-world, noisy environments.

Keywords: Adaptive Path Planning; autonomous robot navigation; path tracking.

1. Introduction

Today, autonomous robots are capable of executing various tasks with exceptional accuracy and speed while maintaining safety. An entirely autonomous robot has the capacity to perceive and analyse its surroundings, identify objects, and execute an action or manipulation within that setting. Autonomous robots have shown great potential and efficiency, particularly when compared to automated guided vehicles, due to their capability to navigate without reliance on guidance devices or centralized control [1]. Obstacle avoidance requires directing a robot's movement to navigate predictable or unpredictable obstacles. The application of a diverse range of algorithms including soft computing, image processing, control systems, statistical techniques, and conventional techniques allow for effective autonomous navigation, even in dynamic environments [2]. The robot's current location and sensor data affect its motion. When given a map and an intended goal point, path planning is necessary to determine a path from the starting position to the goal position [3]. However, there might be several potential paths available, considering the unrestricted area in which the robot can operate. Path planning algorithms aim to obtain the optimal path or an acceptable approximation to it. The optimal path generally is defined as the one with the lowest computational cost [4].

Conventional methods include Voronoi diagram, potential field and the like. Voronoi diagram propose a mathematical approach to define obstacles and optimize robot path [5]. Potential field use potential field calculations to determine robot paths and avoid obstacles [6]. Soft computing refers to the use of inexact calculations to solve computationally complex tasks. This approach allows for solutions to problems that may be too difficult or time-consuming to solve using current hardware. Soft computing is different from hard computing as it tolerates imprecision, partial truth, uncertainty, and approximation. The human mind serves as a role model for soft computing [7]. Much research has been conducted on applying soft computing techniques such as neural networks, genetics algorithm, and more to path planning and navigation problems. Genetics algorithm use natural selection principles to solve optimization problems and improve paths [8]. Artificial neural networks (ANNs) enhance the learning capabilities of robots and help them better understand their surroundings [9]. The A* algorithm has proven to be a reliable approach for solving pathfinding problems [10].

This paper discusses navigation techniques in indoors, path planning methods. The techniques analysed are divided into two categories: conventional methods and artificial intelligence (AI) methods. In this study, we also examine path tracking algorithms to comprehensively analyse navigation specifically the Kalman filter and particle filters. The Kalman filter has proven successful in tracking moving objects in real-time scenarios [11]. Particle filter is a robotics technique that utilises the Monte Carlo method. Its principal goal is to update the robot's position distribution by incorporating new observations [12].

The remainder of the paper is organized as follows: Section 2 addresses the complications of navigating in ever-changing surroundings that autonomous robots face and highlights the significance of simultaneous localization and mapping (SLAM). Section 3 an examination of a range of conventional techniques for path planning and navigation, encompassing Voronoi diagram, and potential field. In Section 4, analysis of AI techniques, including genetics algorithm and neural networks. Section 5 showcasing filtering techniques for path tracking, particularly Kalman and particle filters. Section 6 discusses the advantages and disadvantages of AI techniques in comparison to conventional methods and reports performance metrics for Voronoi diagram, potential field, A* algorithm, genetics algorithm and neural networks. Finally, the paper is summarized and concluded by Section 6.

2. Explanation of Autonomous Robot Navigation Problem

2.1. Mobile robots

Mobile robots are capable of locomotion and are not limited to a single physical location. They can be designed to operate in vast workspaces and explore uncharted territories, enabling them to successfully navigate their environment, whether it is in the air, on land or underwater. Additionally, the term “mobile robot” refers to robots that have the ability to move, as opposed to stationary robots. These robots can be remotely controlled via guidance systems to carry out specific tasks. Alternatively, they can operate autonomously. Mobile robots have the capability of navigating unfamiliar environments without requiring human interaction or guidance devices. The key components of a mobile robot typically include a controller, which is typically a microprocessor or personal computer, control software, sensors tailored to the specific requirements of the robot, and actuators that allow the robot to move within its environment. Mobile robots can be manufactured in any size, depending on their intended tasks and the location where they will operate [13-15].

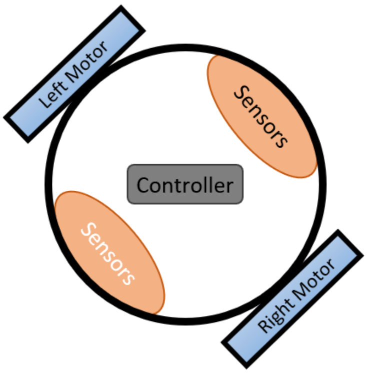


Figure 1. *A simplified model of mobile robot.*

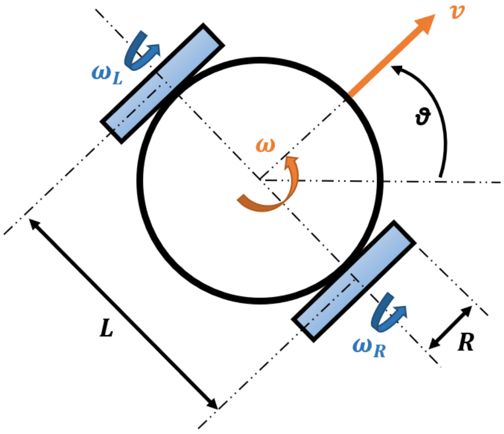


Figure 2. *DD visualization of the parameters.*

Figure 1 shows a simplified model of a mobile robot with mounted motors on its chassis, facilitating its mobility. The choice of the motors depends on the planned tasks for the robot. The robot is capable of moving in forward or reverse directions, altering its movement by adjusting the speed differential of its wheels. This

*Corresponding author

ALPER HÜSEYİN DOĞAN; Istanbul University-Cerrahpasa, Institute of Graduate Studies, Electrical and Electronics Engineering Department, Türkiye; e-mail: alperhuseyin.dogan@ogr.iuc.edu.tr; 0000-0002-1782-0465

TARIK VELİ MUMCU; Istanbul University-Cerrahpasa, Faculty of Engineering and Architecture, Computer Engineering Department, Türkiye; e-mail: tarik.mumcu@iuc.edu.tr; 0000-0002-8995-9300

approach is known as the differential drive (DD) and can be implemented using two separately wheeled robots. DD is a popular and easily controlled approach. Figure 2 visualises the parameters of DD approach.

In the DD method, the robot is capable of movement along both the x and y axes, as well as changing direction on this coordinate plane. Consequently, the robot possesses three degrees of freedom (DoF). The matrix presents the DoF in Eq. (1).

$$q = \begin{bmatrix} x \\ y \\ \Theta \end{bmatrix} \tag{1}$$

Θ, rotation of the robot about z-axis. (x, y), position of the robot on the x-y coordinate plane. For the robot to change direction, it is essential that both wheels turn at different speeds. Additionally, the radius of the wheels plays a significant role. Eqs. (2) and (3) demonstrate the correlation between the angular velocity of the wheels and their corresponding linear velocity.

$$v \cdot R = r \cdot \omega \cdot R \tag{2}$$
$$v \cdot L = r \cdot \omega \cdot L \tag{3}$$

r, wheel radius in meters. ω, left and right wheel speeds, in radians per second. Eqs. (4) and (5) illustrate the results, presenting the speed v in meters per second and the angular velocity ω in radians per second.

$$v(t) = \frac{v \cdot R(t) + \ v \cdot L(t)}{2} \tag{4}$$

$$\omega(t) = \frac{v \cdot R(t) - \ v \cdot L(t)}{r \cdot c} \tag{5}$$

rc, wheelbase in meters. All of these equations provide us with the mathematical depiction of this system Eq. (6).

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \Theta(t) \end{bmatrix} = \begin{bmatrix} cos\big(\Theta(t)\big) \cdot v(t) \\ sin\big(\Theta(t)\big) \cdot v(t) \\ \omega(t) \end{bmatrix} \tag{6}$$

Eq. (7) depicts a kinematic model of the system.

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \Theta(t) \end{bmatrix} = \dot{q} = \begin{bmatrix} cos(\Theta) & 0 \\ sin(\Theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} = j(\Theta) \cdot v \tag{7}$$

$\dot{q}$, derivation of the state vector is the outcome of alterations in both x-axis and y-axis speed and angular velocity [16-18].

2.2. Navigation

One of the primary areas of research in the field of robotics concerns applications of autonomous robot navigation. The intelligent robotic navigation system refers to the robot's ability to locate a path while avoiding obstacles and reach the target point from a predetermined starting point by means of sensors, control and navigation software, all without any human intervention. The robot must update its position constantly during its journey, monitor its surroundings, and develop strategies to achieve its objectives, whilst simultaneously executing these tasks. Moreover, these determinations need to be made in real-time [19].

Typically, an odometer is used to calculate position in solving an autonomous robot navigation issue. However, this creates the need for a map. Localisation through odometry information obtained from wheel velocities involves determining the position, direction and angle of the robot using an encoder with the coordinate plane as reference. The rotational speed of the wheels is used to calculate the direction of the robot, which is essential for localisation. However, relying solely on the robot's own position calculations does not suffice for navigation purposes. The robot must be aware of its starting point, current location, and destination. A map is necessary for conveying this information, and the starting point should be indicated on the map for the robot's guidance.

2.3. Simultaneous localisation and mapping

The autonomous robot must safely perform various tasks with a high degree of autonomy. The crucial aspect is that the robot follows the desired approach. However, for the robot to achieve this objective, answering two core questions is paramount: "Where am I?" and "What does the outside world look like?" The first question is connected to the issue of localisation, which demands that the robot knows its position to attain its purpose [20].

On the other hand, the second query concerns environmental data gathering. The robot seeks to identify the appearance of the external environment. This task is known as mapping in scientific literature. In this process, the robot endeavours to create a map or layout of the surrounding external world by relying on its sensory data [21].

In addition to the previously stated information [16], the localisation and mapping problem involves determining the location of a robot and acquiring environmental information. The SLAM problem was first introduced in a paper [22] and has gained popularity among researchers and mobile robotics communities over the past twenty-five years [23].

Since its inception, SLAM has been extensively applied in various field. In the past, the extended Kalman filter (EKF) was widely employed to address this issue [24]. Subsequently, particle filter solutions were introduced [25]. Currently, Kalman filters continue to be a popular method for solving the SLAM problem. However, visual SLAM techniques are gaining widespread recognition in the field [26].

3. Conventional Techniques Used in Navigation Applications

3.1. Voronoi diagram

Numerous researchers have employed the Voronoi diagram as an effective tool in their previous studies. The Voronoi diagram is defined as the division of a plane with n points into distinct areas, wherein each area contains only one generating point, and all of the points within a given area are closer to its generating point than any other. A sensor-based system capable of determining the safest path for robots within their operating environment have been introduced [27]. An alternative approach for safe navigation in the presence of obstacles is founded the generalised Voronoi diagram (GVD) [28]. The drawback of this technique is the computation of the entire GVD, which is very expensive and is often used for offline applications and global path planning [29]. Alternatively, it is feasible to estimate the GVD through specific algorithms applied to the Voronoi tessellation of the sensor measurements [30]. Another option is to discretize the known map into cells and use computer vision techniques [31] based on distance transforms [32]. However, these methods necessitate a balance between accuracy and computational load, particularly in cluttered and limited environments.

3.2. Potential field

Potential field methods are commonly utilised within the field of mobile robotics for the purposes of both path planning and motion control. Potential field method introduced, which incorporates a repulsive potential field around the obstacles and forbidden zones to enable collision avoidance, while employing an attractive potential field around the end goal to guide the robot towards its target. Path planning can be categorized into two subgroups: universal planning, where the robot has full knowledge of its environment via a pre-existing map before initiating movement, and local planning, where the robot builds a map of its surroundings as it moves towards its target. The potential field method can be applied to both universal and local planning techniques [33].

The potential field method provides effective and straightforward solutions to navigation issues. Nevertheless, a significant drawback is that robots may also encounter local minima because an obstacle usually generates a repulsive field locally, which can negate desirable target behaviours. Currently, extensive research is underway to address this challenge actively. Demonstrated the efficacy of simulated annealing in overcoming this problem, a previous study showed promising results [28]. Full or partial details of obstacles can be used to change the potential field surrounding them, in order to avoid local minima [34-36], or by modelling obstacles geometrically through envelopes [37]. Another enhancement to avoid local minima involves including multiple attractive points in the target instead of just one point [38] or virtual obstacles can be added to move away from local minima [39]. The artificial potential field can be improved by introducing a potential density alongside obstacles to aid in path planning using a quasi-geodesic curve [40]. Strategies for navigating local minima can also incorporate behavioural tactics, including wall tracking [41].

4. Artificial Intelligence Techniques Used in the Navigation Applications

4.1. Genetics algorithm

The genetic algorithm is a widely used search technique for optimising results. Its inspiration comes from the study of evolutionary biology encountered in nature. The algorithm searches for the best possible solution through natural selection in a complex multidimensional space. In the genetic algorithm, the population generated at first is random and the possible solution to the problem is referred to as a population. The fitness value of each solution is determined by the objective function of the problem. After selecting more suitable solutions to the problem, genetic operations such as mutation and crossover are applied to produce a new population comprised of better candidates. This process is repeated until the maximum number of generations have been generated or until an acceptable fitness value is obtained. This method is utilised to generate and analyse optimised outcomes in various applications.

Genetic algorithm has been proposed for motion planning in robots, specifically targeting a single point instead of the whole workspace [42]. A motion planning procedure has also been proposed for load-carrying robots, utilizing GA to find optimal solutions and fuzzy logic for plan implementation [43]. An evolutionary algorithm has been proposed to address the issue of autonomous robot navigation in an environment where there are several unknown polygonal obstacles [44]. Genetics algorithm (GA) are potent search techniques, grounded in natural selection and featuring crossover, mutation, and favourable gene selection processes. They have the potential to cover large spaces unlike standard search and optimisation methods.

The term “chromosomes” is used to describe potential solutions to a problem. The population comprises these chromosomes collectively. Each population undergoes fitness function calculation one at a time. The fitness function enables the creation of a selection process, challenging the chromosomes to survive until the end of this process. Usually, the chromosomes that suit best have higher chances of selection. Finally, genetic operations such as crossover and mutation are applied to these new solutions. After applying genetic operators to chromosomes, a new population is generated [45]. The success of the new population is then recalculated, taking into account the suitability of each new chromosome. Finally, the best chromosome is identified, resulting in an optimal or near-optimal solution. GA is currently being utilised to solve path planning problems, with examples of such studies found in the literature.

Path planning task using genetics algorithm has been successfully executed in an unfamiliar environment [46]. The simulation results indicate the efficacy of genetics algorithm in resolving navigation issues in dynamic settings.

The primary limitation of genetics algorithm is their inability to determine when an exact solution has been reached, as solutions of near-optimum quality are accepted instead. Consequently, the solution obtained is not necessarily optimal. Furthermore, executing the algorithm again on the same problem with identical input can produce substantially different outcomes. As a result, it is challenging to discern the best solution for these reasons [47].

4.2. Artificial neural networks

ANNs are able to learn through rapid data processing [48]. ANNs have greatly contributed to the development of autonomous robots in the field of robotics, particularly in kinematic and dynamic modelling and path planning. An autonomous robot is capable of moving around in its operating environment. Through the use of ANNs, robots can acquire decision-making abilities that were previously exclusive to humans. However, the robot requires physical capability to execute its decisions. In accordance with a study [49], a path planning system founded on dynamic ANNs has been proposed by a researcher. This system has the capacity to navigate a robot on a level area encircled by stationary or dynamic obstacles. The ANN employed for path planning has been taught using an extended back propagation approach, whilst an artificial potential field technique has been utilised to ensure the avoidance of collisions. Other ANNs have been employed to improve the avoidance of stationary obstacles. This approach was suggested when mobile robot navigation systems were first comprehended [50, 51]. A multi-layer feedforward ANN architecture was developed by researchers, who transformed the problem into a classification issue to simplify the output. Training samples for the ANN controller were accumulated using reinforcement Q-learning technique. The ANN controller runs in the background and trains itself once more by using its backup signal during the interaction with the environment. The result of this study indicates that a hybridised technique, converging these two methods, offers a superior solution to this type of problem than applying either technique in isolation.

ANNs comprise highly concurrent building blocks that exhibit clear design principles. This enables ANNs to be utilised in complex systems. Whilst ANN architectures cannot rival conventional methods for performing error-free numerical operations, they offer superior solutions to problems involving ambiguity, prediction, or classification.

ANNs have numerous advantages. One benefit is the ability to update weight coefficients between agents to perform specific tasks. For example, related weight coefficients are updated until the neural network model performs in the desired manner. Consequently, learning is completed by this neural network model. ANNs are commonly used in efficient learning problems and pattern recognition applications.

4.2.1. A Algorithm*

The A* algorithm is a particularly effective approach to solving the pathfinding problem. This algorithm aims to find the shortest path from a starting point to a destination. The A* algorithm combines two key components: actual cost and estimated cost. The actual cost represents the total cost of travelling from the starting point to a node. The estimated cost, on the other hand, represents the estimated cost of getting from that particular node to the destination node. These two costs allow the A* algorithm to select the most appropriate node at each step [52].

Over time, significant improvements have been made to the algorithm and numerous studies have been carried out. Iterative deepening versions have been developed that make the A* algorithm run faster and avoid excessive memory usage [53]. Epsilon optimisation refers to an improvement that allows the A* algorithm to find an optimal solution within a specified tolerance range in addition to the shortest path [54]. Research focusing on the development of heuristics that improve the performance of the A* algorithm is crucial [55]. A parallel version of the A* algorithm has been defined to run quickly on multi-processor systems [56]. The hybrid A* algorithm is used to solve larger and more complex problems by combining the A* algorithm with other search methods [57]. In this study, the A* algorithm is preferred to make the comparisons more understandable, as it’s a well-known and fundamental path planning algorithm.

5. Filtering Techniques for Path Tracking

5.2. Kalman filter

In navigation applications, one of the most popular methods is the Kalman filter [58]. Due to its capability to extract advantageous information from disruptive data and its minimal computational and memory demands, it is utilised within numerous application domains like spacecraft navigation, robotics motion planning, wireless sensor networks, and signal processing [59]. Filters can be thought of as designs that reduce input data and parse it into desired output data in navigation applications. The Kalman filter assumes that the system is linear. Therefore, all system dynamics and measurements can be expressed as continuous-time linear differential equations and discrete-time difference equations. The uncertainty in state values and measurement updates follows a zero-mean Gaussian distribution. When the system is perfectly linear and all noise follows a zero-mean Gaussian distribution, the most efficient state observer is the Kalman filter [60]. In Eq. (8), the discrete time difference equation is presented:

$$x_{k+1} = Fx_k + Bu_k + w_k \tag{8}$$

x represents the state vector of estimated values. The matrix F determines changes in future states based on the current state, while matrix B determines changes based on input data. W is a random variable with a Gaussian distribution of zero mean ($w_k \sim N(0, Q_k)$), and U represents the input value. When this equation is time dependent rather than time independent, it can be expressed as follows:

$$x_k = F_k \cdot x_{k-1} + B_k \cdot u_k + w_k \tag{9}$$

As the matrices F and B take different values at varying time intervals, they are incorporated in the Eq. (9) with index k. These shifts can either depend on time or may be a result of external data. The F_k matrix is known as Φ in literature [61, 62]. The equation for updating the measurement can only be expressed in discrete time and is as follows:

$$z_k = H_k \cdot x_k + v_k \tag{10}$$

The measured values z_k correspond to the state values x_k through the matrix H_k. A random variable, vector v_k, follows a white Gaussian distribution ($v_k \sim N(0, R_k)$) and influences the measured value.

The Kalman filter uses the above equations to estimate the mean of state x_k and calculate the covariance matrix P_k for this estimation. The initial estimate for the mean is x_0, while P_0 represents the initial covariance matrix. The filter then generates new states by applying the dynamic equations.

5.3. Particle filters

Particle filters are estimation techniques that deploy the Sequential Monte Carlo (SMC) method in non-linear and non-Gaussian Markovian state spaces. They are utilised in cases where the Kalman filter would be ineffective because of non-Gaussian or non-linear parameters. The issue involves utilizing known state probabilities and predicting future states $p(x_t|x_{t-1})$ based on current state values. Additionally, measurement likelihood $p(y_t|x_t)$ is obtained through measurement prediction equations [63]. Based on these functions, the equation for state estimation can be expressed as follows [64]:

$$p(x_t|y_{1:t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})dx_{t-1} \tag{11}$$

Using the previous measurements $(y_{1:t-1})$ Eq. (11) calculates the probability of the state at the future time interval x_t. The update equation is then applied to include the present measurement values y_t into the problem [65].

$$p(x_t|y_{1:t}) = \frac{p(y_t|x_t)p(x_t|y_{1:t-1})}{\int p(y_t|x_t)p(x_t|y_{1:t-1})dx_t} \tag{12}$$

Eq. (12) updates the probability from the previous time interval to the current time interval with the latest measurement, y_t.

Particle filters are advantageous in determining nonlinear, non-Gaussian distributed systems. One drawback of particle filters is that they necessitate a large quantity of particles when used to portray multidimensional state spaces.

6. Simulation Studies & Results

In this paper we have carried out a comparative analysis of path planning algorithms, dividing them into two main categories: conventional and AI techniques. The algorithms under consideration encompass a diverse set, including potential field, Voronoi diagram, A* algorithm, genetic algorithm, and neural networks. The motivation behind this study is to evaluate the performance of these algorithms in a discrete environment, similar to a grid world. This assumption simplifies the representation of the environment and the robot's movements. The robot under consideration is holonomic, meaning that it is free to move in any direction without constraints. This assumption allows us to explore a wide range of path planning algorithms without worrying about non-holonomic constraints. The initial position and the target point are known in advance, and the robot is equipped with complete knowledge of the map of the environment.

The map shown in the Figure 3 simulates an indoor environment with a structured layout designed to challenge path planning algorithms. The dimensions of the map are 120 metres in width and 60 metres in height, with a grid size of 1 metre per cell. A notable feature of this map is the inclusion of seven different rooms, each contributing to the complexity of the path planning task. These seven rooms are arranged in a grid-like pattern running from the top left to the bottom right of the map. Room 1, located in the top left corner, serves as the robot's starting position, culminating in Room 7, located in the bottom right corner, which serves as the target point. Each room is connected by corridors and separated by walls or obstacles, presenting a complex navigation challenge. With a robot radius of 1 metre, precision is required to traverse the map effectively while avoiding obstacles and finding an optimal path, making this map a realistic and complex testbed for evaluating path planning algorithms in indoor environments.

The performance of path planning algorithms is assessed according to factors such as path length and execution time, providing useful insights into their effectiveness and efficiency in navigating these complex environments. We will now present a detailed analysis of the algorithm's functionality in an intricate indoor setting. Our analysis employs both quantitative data and visual representations to provide a comprehensive overview of the algorithm's performance.

Figure 3. *Map representation of an indoor environment*

Table 1 provides visual representation which demonstrate the calculated paths from the starting to the goal destination for each of the algorithms of the paths produced by diverse path planning algorithms in the specified environment outlined in Figure 3.

The potential field algorithm encountered difficulties and didn't reach the goal point. It became imprisoned within the primary area located at the environment's upper left corner. This outcome is consistent with the method's susceptibility to local minima. The Voronoi diagram method achieved the objective but took longer and produced an inefficient route. The estimated path exceeds 170 metres in length. This behaviour is in line with the Voronoi approach's tendency to favour open space and may require post-processing for optimal path refinement. The A* algorithm exhibited its well-documented strength by generating the second shortest path. Specifically, it produced a path that was approximately 1.5 metres shorter than the path produced by the genetics algorithm approach. This finding aligns with A*'s ability to optimally balance exploration and exploitation while employing heuristic information, indicating the high degree of path optimisation achieved by the A* when navigating the intricate indoor environment. The approach employing genetic algorithm differentiated itself through accomplishing the third shortest path and finishing the task in under ten seconds - notably ai methods are quicker than conventional techniques. This confluence of a reduced path length and rapid computation confirms Genetics algorithm as an admirable selection in this setting. The Long Short-Term Memory (LSTM) model emerges as the dominant path planning algorithm among the assessed alternatives, also outpacing both A* and genetics algorithms. This neural network-based technique demonstrated outstanding results, excelling notably in its adaptability and efficiency in navigating complex indoor environments. This adaptability is particularly valuable where traditional algorithms may struggle due to their fixed rules and heuristics. The LSTM model, using its sequence-based data processing and memory retention, outperformed its counterparts in producing the shortest path and completing the task in a remarkably short amount of time, thereby ranking as the optimal choice for path planning in this specific setting. Notably, the model achieved a mean absolute error (MAE) of *0.0584* and a loss value of *0.0109* during training. Further exploration and refinement of the LSTM-based approach could provide even more promising results in intricate indoor environments.

Table 1. *Simulation results of the algorithms*

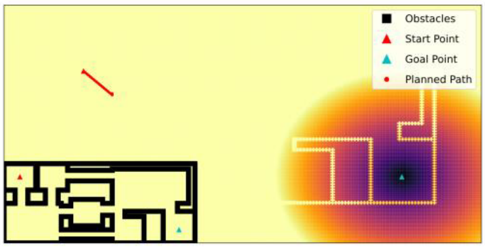
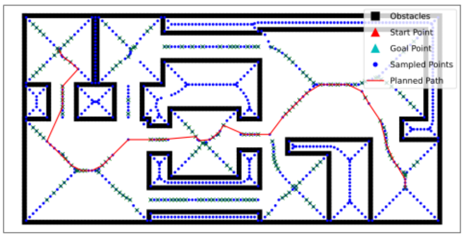
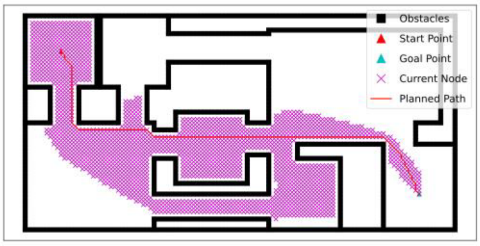
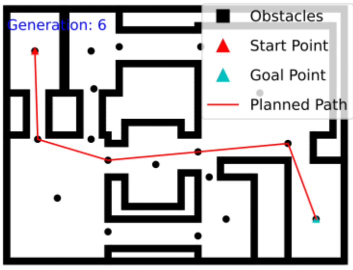
Algorithm	Planned Paths
Potential Field	
Voronoi Diagram	
A* Algorithm	
Genetics Algorithm	
Artificial Neural Networks	

Table 2 displays the path lengths calculated by all algorithms on the map and their respective computation times in comparison.

Table 2. *Path length (m) and computation time (sec) comparisons of algorithms.*

	Potential Field	Voronoi Diagram	A* Algorithm	Genetic Algorithm	LSTM
Path Length	Not Found	170.10	130.04	131.24	**128.33**
Computation Time	Not Found	28.58	2.74	8.54	**0.17**

In this study we also employed a simulation analysis aimed at evaluating the effectiveness of filtering techniques under different noise conditions. Two distinct scenarios were examined: one incorporating Gaussian noise and the other integrating exponential noise. These analyses serve as a valuable preliminary investigation into the efficacy of Kalman and particle filters. In both experiments, a robot navigated through a known environment from a specified starting point to a goal point covering a total length of approximately 130.04 metres using an A* path planning algorithm while facing the challenge of noisy sensor measurements and motion commands. The results of these experiments highlight critical insights into the filtering system's behaviour.

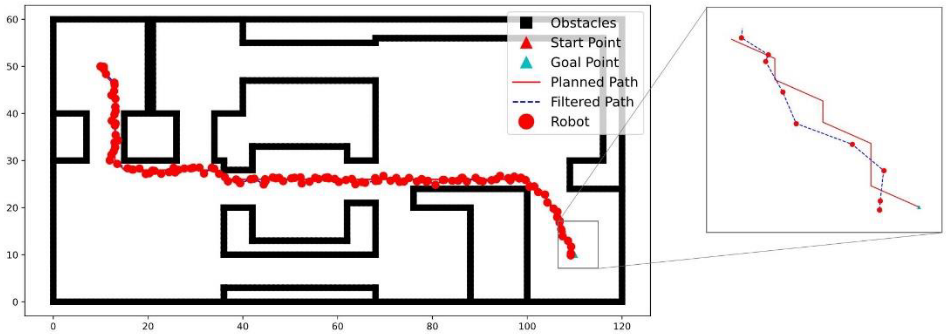


Figure 4. *Path tracking using the Kalman filter with added Gaussian noise*

Figure 5. *Path tracking using the Kalman filter with added exponential noise*

Figure 4 illustrates the path taken by the robot using the Kalman filter with added Gaussian noise scenario. The Kalman filter efficiently reduced the tracking error to approximately 136.54 metres, successfully guiding the robot to the goal. The filter's ability to achieve this high level of accuracy in the presence of Gaussian noise emphasizes the robustness and effectiveness of the Kalman filter for state estimation. The tracking error was used as a valuable metric for evaluating the filtering system's performance under this specific noise type, with an execution time of 9.19 seconds for this scenario. Conversely, during the experiment shown in Figure 5, featuring exponential noise, the Kalman filter exhibited a tracking error of around 140.97 metres, signifying a slightly higher level of inaccuracy compared to the Gaussian noise scenario. Although the robot ultimately reached its goal successfully in both scenarios, the results clearly indicate that Gaussian noise, with its symmetric and bell-shaped distribution, yields better tracking performance when compared to non-Gaussian exponential noise. This difference is expected, as Gaussian noise is more amenable to the linear, minimum mean square error estimation framework of the Kalman filter. The execution time for the Kalman filter in this scenario was 9.13 seconds.

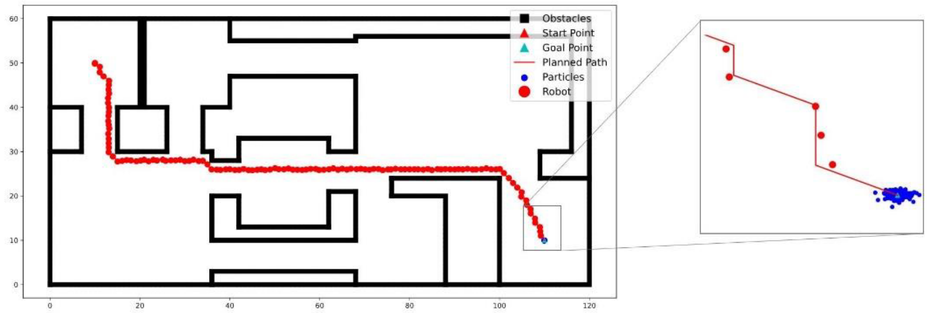


Figure 6. *Path tracking using the particle filter with added Gaussian noise*

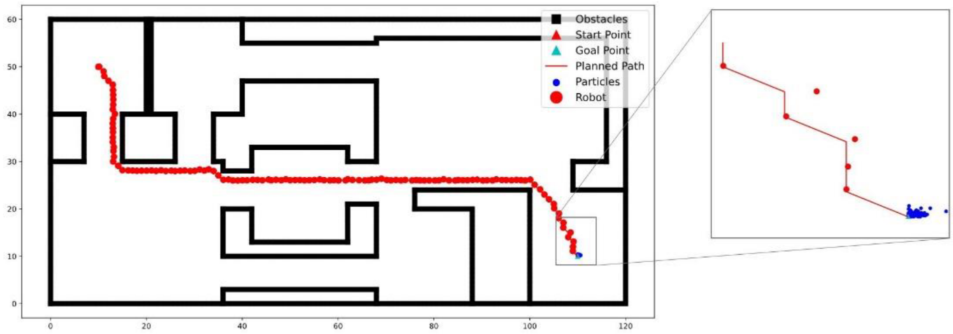


Figure 7. *Path tracking using the particle filter with added exponential noise*

Figure 6 exhibits the path taken by the robot after the application of the particle filter in a scenario with Gaussian noise. The tracking error, which measures the discrepancy between the estimated and true path taken by the robot, is found to be 132.08 metres. This indicates the average deviation between the Particle Filter's estimated path and the ground truth path. Since the filter estimates the robot's position based on noisy measurements and motion, minor deviations from the true path are expected. Additionally, the tracking error might be slightly affected by the resampling process, as particles with lower weights may not accurately represent the true path. The execution time for the particle filter with Gaussian noise was 10.44 seconds. On the other hand, in particle filter with added exponential distribution scenario as depicted in Figure 7 indicates that the robot successfully reached the goal with a tracking error of 124.82 meters. Notably, this value is lower than the tracking error in the Gaussian noise scenario. This suggests that the particle filter with an exponential distribution may perform better under these specific noise conditions, even though the execution time for this scenario was slightly longer at 13.33 seconds.

These findings offer valuable insights into the comparative performance of filtering techniques. It's interesting to note that the particle filter outperformed the Kalman filter in the presence of Gaussian noise, contrary to our initial expectations due to its adaptability and ability to handle a variety of noise distributions effectively. The particle filter's flexibility and resampling mechanism enable it to maintain accurate state estimates, even in cases where Gaussian noise is expected. This suggests that for certain scenarios or noise characteristics, the particle filter might be a more suitable choice for state estimation. The longer execution time of the particle filter compared to the Kalman filter's execution time is likely due to the particle filter's probabilistic nature and resampling process, which requires more computational resources.

7. Conclusion

In conclusion, the area of self-governing robot direction is swiftly progressing, with different conventional and artificial intelligence methods at hand to aid robots in navigating intricate environments. Conventional methods such as Voronoi diagram and potential field have limitations in handling indoor environments, whereas artificial intelligence methods such as genetics algorithm, and neural networks provide increased adaptability and strength in managing complex navigation tasks.

The A* algorithm has been demonstrated as a fundamental method for resolving pathfinding issues, and the integration of artificial intelligence approaches with conventional practices has the capability to construct navigation systems that are more durable and effective. The adeptness of robots to navigate unfamiliar surroundings without human intervention or guidance mechanisms has the potential to revolutionize numerous sectors.

Nevertheless, there remain significant obstacles to the advancement of fully autonomous robots, involving the necessity for precise and sturdy sensing, mapping, and localization, as well as the aptitude to operate instantaneously. Overcoming these issues will demand additional research and development in field like computer vision, machine learning, and sensor technology.

In addition, the simulation study emphasizes the significance of considering the noise characteristics when selecting a filtering technique. The choice between Kalman and particle filters should be based on the specific requirements of the application and the underlying noise distribution, as highlighted by the results obtained in our experiments. These findings provide valuable guidance for practitioners in the field of state estimation and robotics, enabling them to make informed decisions about filter selection based on the noise conditions they are likely to encounter in real-world applications.

The realm of autonomous robot navigation is an innovative and dynamically growing research domain with enormous capacity for transforming numerous aspects of our daily lives. With advancing technology and novel applications, a greater number of autonomous robots are projected to emerge in the future, substantially enhancing efficacy, productivity and safety in multiple industries.

Declaration of Interest

The authors declare that there is no conflict of interest.

Author Contributions

Alper Hüseyin DOĞAN: conceptualization, methodology, collecting the data, performed analysis, writing. Tarık Veli MUMCU: conceptualization, methodology, review, editing, supervision.

References

[1] G. Fragapane, R. De Koster, F. Sgarbossa, and J. O. J. E. J. o. O. R. Strandhagen, "Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda," vol. 294, no. 2, pp. 405-426, 2021.

[2] J. A. Oroko and G. Nyakoe, "Obstacle avoidance and path planning schemes for autonomous navigation of a mobile robot: a review," in *Proceedings of the Sustainable Research and Innovation Conference*, 2022, pp. 314-318.

[3] N. Adzhar, Y. Yusof, M. A. J. A. i. S. Ahmad, Technology, and E. S. Journal, "A review on autonomous mobile robot path planning algorithms," vol. 5, no. 3, pp. 236-240, 2020.

[4] J. R. Sanchez-Ibanez, C. J. Perez-del-Pulgar, and A. J. S. García-Cerezo, "Path planning for autonomous mobile robots: A review," vol. 21, no. 23, p. 7898, 2021.

[5] D. Chen, A. Xiao, M. Zou, W. Chi, J. Wang, and L. J. a. p. a. Sun, "GVD-Exploration: An Efficient Autonomous Robot Exploration Framework Based on Fast Generalized Voronoi Diagram Extraction," 2023.

[6] G. Klančar, A. Zdešar, and M. J. S. Krishnan, "Robot Navigation Based on Potential Field and Gradient Obtained by Bilinear Interpolation and a Grid-Based Search," vol. 22, no. 9, p. 3295, 2022.

[7] L. J. I. J. o. C. I. S. Magdalena, "What is soft computing? Revisiting possible answers," vol. 3, no. 2, pp. 148-159, 2010.

[8] Z. Zhang, R. Lu, M. Zhao, S. Luan, M. J. J. o. I. Bu, and F. Systems, "Robot path planning based on genetic algorithm with hybrid initialization method," vol. 42, no. 3, pp. 2041-2056, 2022.

[9] F. Cruz, R. Dazeley, P. Vamplew, I. J. N. C. Moreira, and Applications, "Explainable robotic systems: Understanding goal-driven actions in a reinforcement learning scenario," vol. 35, no. 25, pp. 18113-18130, 2023.

[10] D. Foead, A. Ghifari, M. B. Kusuma, N. Hanafiah, and E. J. P. C. S. Gunawan, "A systematic literature review of A* pathfinding," vol. 179, pp. 507-514, 2021.

[11] C. Urrea and R. J. J. o. S. Agramonte, "Kalman filter: historical overview and review of its use in robotics 60 years after its creation," vol. 2021, pp. 1-21, 2021.

[12] Y. Wang, X. J. C. i. Wang, and neuroscience, "Research on SLAM road sign observation based on particle filter," vol. 2022, 2022.

[13] R. C. J. T. I. j. o. r. r. Arkin, "Motor schema—based mobile robot navigation," vol. 8, no. 4, pp. 92-112, 1989.

[14] D. M. J. I. A. T. O. M. Auslander, "What is mechatronics?," vol. 1, no. 1, pp. 5-9, 1996.

[15] R. H. Bishop, *The mechatronics handbook-2 volume set*. CRC press, 2002.

[16] J. J. A. Mireles Jr and U. o. T. a. A. Robotics Research Institute, "Kinematic models of mobile robots," 2004.

[17] M. I. Ribeiro and P. J. I. d. S. e. R. Lima, "Kinematics models of mobile robots," vol. 1000, p. 1049, 2002.

[18] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to autonomous mobile robots*. MIT press, 2011.

[19] Z. Liang, X. Ma, and X. Dai, "Extended Monte Carlo algorithm to collaborate distributed sensors for mobile robot localization," in *2007 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2007, pp. 1647-1652: IEEE.

[20] M. Paskin and S. J. a. p. a. Thrun, "Robotic mapping with polygonal random fields," 2012.

[21] J. Fuentes-Pacheco, J. Ruiz-Ascencio, and J. M. J. A. i. r. Rendón-Mancha, "Visual simultaneous localization and mapping: a survey," vol. 43, pp. 55-81, 2015.

[22] T. Bailey, J. Nieto, J. Guivant, M. Stevens, and E. Nebot, "Consistency of the EKF-SLAM algorithm," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 3562-3568: IEEE.

[23] R. C. Smith and P. J. T. i. j. o. R. R. Cheeseman, "On the representation and estimation of spatial uncertainty," vol. 5, no. 4, pp. 56-68, 1986.

[24] G. Dissanayake, H. Durrant-Whyte, and T. Bailey, "A computationally efficient solution to the simultaneous localisation and map building (SLAM) problem," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, 2000, vol. 2, pp. 1009-1014: IEEE.

[25] R. S. Pol and M. Murugan, "A review on indoor human aware autonomous mobile robot navigation through a dynamic environment survey of different path planning algorithm and methods," in *2015 International conference on industrial instrumentation and control (ICIC)*, 2015, pp. 1339-1344: IEEE.

[26] P. Cheeseman, R. Smith, and M. Self, "A stochastic map for uncertain spatial relationships," in *4th international symposium on robotic research*, 1987, pp. 467-474: MIT Press Cambridge.

[27] S. Garrido, L. Moreno, M. Abderrahim, and F. Martin, "Path planning for mobile robot navigation using voronoi diagram and fast marching," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 2376-2381: IEEE.

[28] H. Choset and J. Burdick, "Sensor based planning. I. The generalized Voronoi graph," in *Proceedings of 1995 IEEE international conference on robotics and automation*, 1995, vol. 2, pp. 1649-1655: IEEE.

[29] P. Bhattacharya, M. L. J. I. R. Gavrilova, and A. Magazine, "Roadmap-based path planning-using the voronoi diagram for a clearance-based shortest path," vol. 15, no. 2, pp. 58-66, 2008.

[30] R. Mahkovic, T. J. I. T. o. S. Slivnik, Man,, C.-P. A. Systems, and Humans, "Constructing the generalized local Voronoi diagram from laser range scanner data," vol. 30, no. 6, pp. 710-719, 2000.

[31] R. Marie, H. B. Said, J. Stéphant, O. J. J. o. I. Labbani-Igbida, and R. Systems, "Visual servoing on the generalized voronoi diagram using an omnidirectional camera," vol. 94, pp. 793-804, 2019.

[32] A. Datta, S. J. I. T. O. S. Soundaralakshmi, Man,, C.-P. A. Systems, and Humans, "Fast parallel algorithm for distance transform," vol. 33, no. 4, pp. 429-434, 2003.

[33] F. Janabi-Sharifi and D. Vinke, "Integration of the artificial potential field approach with simulated annealing for robot path planning," in *Proceedings of 8th IEEE international symposium on intelligent control*, 1993, pp. 536-541: IEEE.

[34] H. Lyu and Y. J. A. S. Yin, "Fast path planning for autonomous ships in restricted waters," vol. 8, no. 12, p. 2592, 2018.

[35] T. Weerakoon, K. Ishii, A. A. F. J. J. o. A. I. Nassiraei, and S. C. Research, "An artificial potential field based mobile robot navigation method to prevent from deadlock," vol. 5, no. 3, pp. 189-203, 2015.

[36] J. Zhang, J. Yan, and P. J. I. A. Zhang, "Fixed-wing UAV formation control design with collision avoidance based on an improved artificial potential field," vol. 6, pp. 78342-78351, 2018.

[37] Z. Wu, G. Hu, L. Feng, J. Wu, and S. J. A. A. Liu, "Collision avoidance for mobile robots based on artificial potential field and obstacle envelope modelling," vol. 36, no. 3, pp. 318-332, 2016.

[38] F. A. Cosío, M. P. J. M. Castañeda, and c. modelling, "Autonomous robot navigation using adaptive potential fields," vol. 40, no. 9-10, pp. 1141-1156, 2004.

[39] M. G. Park and M. C. J. K. i. j. Lee, "A new technique to escape local minimum in artificial potential field based path planning," vol. 17, pp. 1876-1885, 2003.

[40] J. Agirrebeitia, R. Avilés, I. F. De Bustos, G. J. M. Ajuria, and M. Theory, "A new APF strategy for path planning in environments with obstacles," vol. 40, no. 6, pp. 645-658, 2005.

[41] T. Zhang, Y. Zhu, and J. J. I. R. A. I. J. Song, "Real-time motion planning for mobile robots by means of artificial potential field method in unknown environment," vol. 37, no. 4, pp. 384-400, 2010.
[42] T. Shibata and T. J. P. o. t. I. I. S. o. I. C. Fukuda, "Intelligent motion planning by genetic algorithm with fuzzy critic," pp. 565-570, 1993.
[43] P. J. Angeline, "Genetic programming: On the programming of computers by means of natural selection: John R. Koza, A Bradford Book, MIT Press, Cambridge MA, 1992, ISBN 0-262-11170-5, xiv+ 819pp., US $55.00," ed: Elsevier, 1994.
[44] Man-Tak Shing and G. B. Parker, "Genetic Algorithms for the Development of Real-Time Multi-Heuristic Search Strategies.," *ICGA*, pp. 565-572, 1993/7/17.
[45] P. Shi and Y. Cui, "Dynamic path planning for mobile robot based on genetic algorithm in unknown environment," in *2010 Chinese control and decision conference*, 2010, pp. 4325-4329: IEEE.
[46] R. L. Haupt and S. E. Haupt, *Practical genetic algorithms*. John Wiley & Sons, 2004.
[47] P. Beeson, N. K. Jong, and B. Kuipers, "Towards autonomous topological place detection using the extended voronoi graph," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005, pp. 4373-4379: IEEE.
[48] Bekey G. A. and G. K.Y., "Neural Networks in Robotics.," *Kluwer Academic Publishers*, 1993.
[49] I. Engedy and G. Horváth, "Artificial neural network based mobile robot navigation," in *2009 IEEE International Symposium on Intelligent Signal Processing*, 2009, pp. 241-246: IEEE.
[50] N. Leena, K. J. I. J. o. M. Saju, and C. Engineering, "A survey on path planning techniques for autonomous mobile robots," vol. 8, pp. 76-79, 2014.
[51] C. Li, J. Zhang, and Y. Li, "Application of artificial neural network based on q-learning for mobile robot path planning," in *2006 IEEE International Conference on Information Acquisition*, 2006, pp. 978-982: IEEE.
[52] P. E. Hart, N. J. Nilsson, B. J. I. t. o. S. S. Raphael, and Cybernetics, "A formal basis for the heuristic determination of minimum cost paths," vol. 4, no. 2, pp. 100-107, 1968.
[53] N. J. Nilsson, *Artificial Intelligence: A New Synthesis*. Morgan Kaufmann Publishers Inc., 1998.
[54] I. Pohl, "Bi-directional and heuristic search in path problems," 1969.
[55] P. E. Hart and N. J. Nilsson, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths.," *EEE Transactions Systems Science and Cybernetics*, vol. 4, pp. 100-107, 1968.
[56] A. Felner, R. E. Korf, and S. J. J. o. A. I. R. Hanan, "Additive pattern database heuristics," vol. 22, pp. 279-318, 2004.
[57] M. Goldenberg *et al.*, "Enhanced Partial Expansion A*," *The Journal of Artificial Intelligence Research (JAIR)*, vol. 50, 05/01 2014.
[58] S. Haykin, *Kalman filtering and neural networks*. John Wiley & Sons, 2004.
[59] K. Nagarajan, N. Gans, and R. Jafari, "Modeling human gait using a kalman filter to measure walking distance," in *Proceedings of the 2nd Conference on Wireless Health*, 2011, pp. 1-2.
[60] F. Ahmed and R. Mandal, "Survey on the Kalman Filter and Related Algorithms," in *Int. Res. J. of Eng. and Technol.*, vol. 5no. 5): International Research Journal of Engineering and Technology (IRJET), 2018, pp. 4219-4223.
[61] A. Doucet, N. De Freitas, and N. J. S. M. C. m. i. p. Gordon, "An introduction to sequential Monte Carlo methods," pp. 3-14, 2001.
[62] P. S. Maybeck, "Stochastic models, estimation and control," *Navtech Book & Software Store*, vol. volume 1, 1994.
[63] B. Øksendal and B. Øksendal, *Stochastic differential equations*. Springer, 2003.
[64] G. Kallianpur, *Stochastic filtering theory*. Springer Science & Business Media, 2013.
[65] S. Brahimi, R. Tiar, O. Azouaoui, M. Lakrouf, and M. Loudini, "Car-like mobile robot navigation in unknown urban areas," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 1727-1732: IEEE.