WORLD COMET RESEARCH GROUP

International Journal of Intelligent Automation & Robotics Engineering
Vol. 2, No. 2, 2019
Doi: 10.67228/30715725/IJIARE-2019PII1N4K
PP. 01-13

Original Article

Intelligent Robotic Navigation in Unstructured Environments

Dr. Lakshmi Narayanan

Associate Professor, Bharathidasan University, India.

Received: 08-05-2019 Revised: 08-25-2019 Accepted: 08-30-2019 Published: 09-03-2019

ABSTRACT

Unstructured robotic navigation is a critical research area due to its applications in disaster response, space exploration, agriculture, and military operations. Unlike structured environments, unstructured settings are unpredictable, dynamic, and lack complete sensory information, making navigation highly complex. Before 2018, research focused on classical and early intelligent methods such as probabilistic robotics, heuristic path planning, and initial machine learning integration. Key navigation tasks—localization, mapping, path planning, and motion control—were addressed using techniques like Bayesian filtering, Kalman filters, particle filters, and occupancy grid mapping to handle uncertainty. Algorithms such as A, D*, and Rapidly-exploring Random Trees (RRT) were widely used for path planning, often enhanced with heuristics and real-time replanning for dynamic environments. Sensor fusion combining LiDAR, sonar, and vision improved environmental perception, while early AI approaches like neural networks and fuzzy logic enabled adaptive decision-making. Reinforcement learning also showed potential, though it was limited by computational constraints at the time.Despite significant progress, pre-2018 systems faced challenges such as limited computational power, poor generalization, and reliance on handcrafted features. Overall, these foundational methods played a vital role in advancing autonomous navigation, though achieving full autonomy in complex environments remains an ongoing challenge.*

KEYWORDS

Robotic Navigation, Unstructured Environments, Path Planning, SLAM, Artificial Intelligence, Sensor Fusion, Autonomous Systems.

ISSN: 3071-5725
PP: 01-13.

1. INTRODUCTION

1.1. Background

Robotic navigation is not a new field of study, and it has been a fundamental area of concern to enable autonomous systems to move safely and efficiently in their environments. The initial methods of navigation were primarily designed in coordinated environments like in an indoor area or a manufacturing facility where the layout can be anticipated and the hazards are well delineated. Nonetheless, in real-life situations, there are numerous unstructured conditions, such as forests, disaster-prone locations, and rough surfaces that demand robots to operate in. These conditions pose serious difficulties such as uneven surfaces, moving barriers and lack of visibility and partial or unclear information regarding the environment. Consequently, the robotic systems will need to have enhanced perception, decision-making, and adaptive capabilities to be able to perform in such Complicated and unpredictable environments.

1.2. Needs of Intelligent Robotic Navigation

Robot intelligent navigation is needed to allow autonomous systems to work efficiently in dynamic and challenging environments. With the increasing use of robots in practical scenarios including search and rescue, agriculture, healthcare, and autonomous transportation, there is an increasing demand to have a navigation system that can manage uncertainty, adjust to changes, and make real-time decisions. In comparison to the conventional systems, intelligent navigation systems incorporate sophisticated algorithms, sensor technologies, and artificial intelligence to guarantee safe, efficient, and reliable movement.

Fig 1 - Needs of Intelligent Robotic Navigation

1.2.1. Autonomous Decision-Making

The possibility to make decisions without human intervention is among the main requirements of intelligent navigation. Robots need to read the environment, process sensor data, and decide what to do in the real-time. This is essential in situations where it is not feasible or even impossible to use manual control, e.g. in disaster areas or deep water exploration.

ISSN: 3071-5725
PP: 01-13.

1.2.2. Adaptability to Dynamic Environments

The real world is usually unreliable, with dynamic conditions and surfaces that are on the move. The intelligent navigation systems should be able to rapidly respond to these changes by adjusting their internal models to change paths when required. Such versatility means that the robots are capable of working safely in very dynamic situations.

1.2.3. Handling Uncertainty

Robotic systems have uncertainty as a result of sensor noise, partial information and variability in the environment. The intelligent navigation systems should include probabilistic approaches and learning techniques in order to control this uncertainty so that the systems can perform reliably even in the presence of imperfect data.

1.2.4. Efficient Path Planning

Effective navigation involves the capability to compute optimal or near-optimal routes and minimize time, power usage and computation cost. With sophisticated algorithms, intelligent systems adjust these factors to ensure that robots achieve their goals in the shortest time possible and in a safe manner.

1.2.5. Robust Perception and Sensor Integration

The environment is very important in accurate perception to understand. Smart navigation systems are based on the combination of various sensors, including cameras and LiDAR, as well as IMUs, to gain an idea of the whole picture of the environment. This increases the ability to detect obstacles, localize them and map them.

1.2.6. Real-Time Operation

Lastly, smart robotic navigation should be real time to act quickly to changes in the environment. Failure or unsafe conditions may be caused by delays in processing or decision-making. Thus, effective algorithms and optimized system architecture is required to achieve prompt responses.

1.3. Problem Statement

The main difficulty of robotic navigation in unstructured settings is the ability to perform successful localization, mapping and path planning with high levels of uncertainty and variability. Unstructured environments (like forests, disaster areas, rough terrain) offer irregular surfaces, dynamic challenges, and incomplete or noisy sensory input, unlike structured settings, where the environment is predictable and well-defined. All these factors complicate the task of robots to correctly locate their positions and create consistent maps of the environment they are in significantly. The sensor noise, drift and the lack of stable landmarks make localization especially difficult, and the constantly moving environmental features make mapping a complicated issue. Such cases are often ineffective with traditional deterministic approaches that are based on pre-determined models and accurate assumptions that are not capable of effectively dealing with uncertainty and unpredictability. These techniques usually presuppose a perfect knowledge of the environment and perfect sensor measurements, which is not often the reality in practice. Consequently, they can generate wrong directions, miss out on obstacles, or lose stability to sudden changes. Also, unstructured path planning environments need to continuously adapt, with new obstacles emerging and previously explored areas becoming invalid. The other major problem is the combination of various system parts, including perception, localization and control, which have to collaborate in real time. Failure or slow-moving in one module can be spread across the system, causing the malfunction of navigation or even collapse. Moreover, the issue of computational constraints also contributes to the complexity because complex algorithms to manage uncertainty can be resource-intensive. So, the challenge is to create navigation systems that are capable of strongly functioning in the face of uncertainty, changing circumstances, and which can continue to provide accurate performance in the face of limited or imperfect information. Overcoming these issues is essential to allowing really autonomous and trustworthy robotic systems in the real world.

2. LITERATURE SURVEY

2.1. Probabilistic Robotics

Probabilistic robotics is the basis of most early navigation systems, in that uncertainty in sensing and motion is explicitly considered. These methods do not rely on ideal measurements but involve probability distributions to express the beliefs that a robot has about its location in the world. Kalman Filter, especially useful in linear systems with Gaussian noise, became popular in the estimation of the state of a robot through the integration of prior information with sensor measurements. In more complex and nonlinear situations, other methods like the Particle Filter gained popularity, where the state of the robot is represented by a collection of weighted samples. These techniques greatly enhanced resiliency under actual conditions where noise and sensor errors are bound to happen, and the variations in the environment cannot be ignored.

2.2. SLAM Techniques

Simultaneous Localization and Mapping (SLAM) became a breakthrough that is essential to the functioning of robots in the unknown or dynamic environment. SLAM helps a robot to create a map of its environment and at the same time locate itself in the map. Initial applications, like Extended Kalman Filter-based SLAM (EKF-SLAM), were based on probabilistic models to jointly estimate robot pose and landmark positions. EKF-SLAM, however, had scalability problems due to increasing the number of landmarks. To counter this, algorithms such as FastSLAM were developed which divided localization and mapping and this gave much better computational capability. These innovations formed the basis of the present-day autonomous navigation systems employed in robotics.

2.3. Path Planning Algorithms

Path planning algorithms are critical in ensuring that the robots move effectively between a starting point and a target location without going through obstacles. Optimal and complete algorithms like A+ (A-star) became popular because they are optimized and complete in grid-based problems. Due to dynamic or a partially known environment, such algorithms as D* (Dynamic A*) were created, which enabled robots to replan paths dynamically as new information is received. Also, sampling techniques such as Rapidly-exploring Random Trees (RRT), became popular in the management of high-dimensional spaces and complicated constraints. All these algorithms promote robust and adaptable navigation policies both in the stationary and dynamic cases.

2.4. Sensor Fusion Methods

Sensor fusion technology improves the accuracy and trustworthiness of robotic perception by integrating information of a variety of sensors. Each sensor (LiDAR, cameras and inertial measurement units or IMUs) have drawbacks but when they are combined they complement each other. As an example, LiDAR has a high degree of accuracy on distance measurements, whereas vision systems have rich context and semantic data. Probabilistic approaches or deep learning training can be used to create fusion techniques which enable robots to come up with a more holistic view of their surroundings. Such integration enhances activities like obstacle recognition, localization and terrain investigation which culminate to safer and efficient navigation.

2.5. Artificial Intelligence Integration

The implementation of artificial intelligence methods has greatly improved the flexibility and smartness of robots. Fuzzy logic and other techniques are used to allow the robot to make decisions under uncertain and inexact conditions by imitating human thinking. Neural networks, especially deep learning models, have been popularly used in perception, pattern recognition, and decision making tasks. The methods enable robots to learn information, adapt to dynamic environments and process nonlinear relationships that are difficult to process using conventional algorithms. Consequently, AI-powered robotics systems are more autonomous, flexible, and capable of performing in diverse real-life scenarios.

3. METHODOLOGY

3.1. System Architecture

The navigation system is designed to consist of four large modules, each charged with a particular aspect of autonomous operation. All these modules combine to allow the robot to sense the surrounding, calculate its location, strategize a route, and perform actions efficiently.

ISSN: 3071-5725
PP: 01-13.

Fig 2 - Needs of Intelligent Robotic Navigation

3.1.1. Perception Module

The perception module will handle collection and interpretation of data collected by the sensors attached to the robot to provide an understanding of the environment. It takes the input of devices like LiDAR, cameras, and ultrasonic sensors to identify obstacles, find free space, and discern features of interest. This module transforms raw sensor data into meaningful representations, e.g. maps or occupancy grids, which can be input to other modules to make decisions. Safe and reliable navigation cannot occur without proper perception.

3.1.2. Localization Module

Localization module identifies the position and orientation of the robot in its surroundings. It takes sensor data and probabilistic approaches like the Kalman Filters or particle filters to estimate the robot pose. Mapping is often tightly coupled with localization using SLAM algorithms in most systems. The position of the robot during movement is constantly updated by this module to make sure that the navigation decisions are made on the basis of the latest and accurate information.

3.1.3. Planning Module

The planning module will create a path that takes the robot to a desired location by avoiding obstacles with the current position. It employs algorithms like A, D, or RRT to calculate the best or possible paths with reference to the environment map. The module can consist of global planning that defines a general route and the local planning that modifies the route dynamically to cope with changes. With proper planning, there will be efficient, collision-free navigation.

3.1.4. Control Module

The control module will implement the planned path, which involves the production of the correct motor commands to move the robot. It also breaks down high-level path commands into low-level control signals, taking into account such factors as speed, direction, and stability. Sensors are used to provide feedback to correct deviations and also provide precise tracking of the trajectory. The

ISSN: 3071-5725
PP: 01-13.

module is vital in providing smooth, accurate and safe movement of the robot in real world situations.

3.2. Mathematical Formulation

The robot motion model explains the dynamics of a robot as its state changes with time as it moves through an environment. To describe the control input (unascend) and the current state of the robot at a certain time (x) in general terms, it is the result of the previous state of the robot (x -1) and the uncertainty or noise (w). Simply put, this relationship can be formulated as: the present state is a function of the past state, and the control input, with a bit of noise. The state usually takes into account significant variables like the position, orientation and velocity of the robot. Control input is the commands inputted into the robot, e.g. wheel speeds or steering angles, which determine the manner in which the robot transitions between states. Function f(·) is the motion model, a mathematical model that describes the way the robot changes states depending on its physical dynamics. In a mobile robot, as an example, this function can be used to take into consideration the way wheel rotations are converted into forward movement and turning. But, in practice, robot motion can never be predicted perfectly in the real world. Wheel slippery, uneven ground, and actuator errors, and external perturbations all create uncertainty in the system. The noise term w 0 w 1 w 2 is a means of expression of this uncertainty, usually expressed as a random variable. The addition of this noise term helps to make the model more realistic and closer to practical applications. The system does not assume the perfect movement but admits that there will never be a perfect match between the anticipated and the actual motion. This probabilistic formulation is pivotal to the contemporary robotics because it enables algorithms like Particle Filters and Kalman Filters to better approximate the actual state of the robot. On the whole, the motion model is a basic model of predicting the behavior of a robot, which allows clear localization, mapping, and navigation of unpredictable and dynamic environment.

3.3. Localization using Particle Filter

Localization with a Particle Filter is a probabilistic approach which approximates the position of a robot with a set of possible states, called particles, rather than probabilistically approximating the position with a single estimate. The belief distribution is the likelihood of the robot to be in various states at a time step. Simply put, the update equation can be defined as follows: the probability of the current state given all observations to the present is equal to a normalization constant times the probability of the current observation given the state, and the predicted belief given the previous state. The predicted belief is actually computed by looking at all the possible prior states, the likelihood of leaving these states to the present state and the past belief on these states. Practically, this is that the Particle Filter works in two broad stages namely prediction and update. In the prediction step, all particles are relocated as per the motion model which is the movement that the robot should move based on control input. This step introduces uncertainty by introducing some random noise, letting the particles diffuse away, and represent various possible positions. During the update step the particles are weighted by the goodness of fit to the current sensor observations. As an illustration, a particle that has a good prediction of its position based on sensor data like distances to

ISSN: 3071-5725
PP: 01-13.

obstacles is given a greater weight. The normalization constant is used to make the total of the all the probabilities equal to one so that probabilities are valid. This is followed by the weighting and subsequently a resampling process is done, with particles of high weight more likely to be retained, and those of low weight discarded. This enables the filter to target the most likely states. In the long-run, with noise and uncertainty, the particles move towards the actual state of the robot. The method is especially useful in nonlinear and non-Gaussian problems, and has been broadly applied in real world robotic localization problems.

3.4. Flowchart of Navigation System

The navigation system follows a series of well-organized steps, which are all involved in the fact that the robot can move safely and efficiently in its environment. These are the steps of a continuous loop, as the robot can adjust to the changes in the real time.

Fig 3 - Flowchart of Navigation System

3.4.1. Sensor Data Acquisition

The initial step is to gather data using different sensors attached to the robot, including LiDAR, cameras, ultrasonic sensors and inertial measurement units (IMUs). These sensors give unprocessed data on the environment around, such as obstacle distances, visual features, and motion data. Proper and timely data received is important and all other things are based on the quality of this input.

3.4.2. Preprocessing

Here, the raw sensor data is processed and made ready to undergo further processing. Noise filtering method is used to eliminate errors and inconsistencies due to sensor limits or environmental variations. Also, data on various sensors can be synchronized and converted into a standard format or frame of reference. This makes the information reliable and applicable in localization and mapping.

ISSN: 3071-5725
PP: 01-13.

3.4.3. Localization

The localization step approximates the position and orientation of the robot in the environment. The system uses sensor measurements alongside motion models with the help of such techniques as particle filters or Kalman filters to define the most likely state of the robot. This is an ongoing process that is constantly updated with new information, as the robot gets a correct idea of its position.

3.4.5. Mapping

Mapping runs the construction or update of representation of the environment using sensor information. This may involve occupancy grid maps, feature maps, or point clouds. In most systems, localization is done concurrently with mapping using SLAM algorithms. The map generated would assist the robot in knowing free space, obstacles and the paths to follow.

3.4.6. Path Planning

In path planning, the system identifies an optimal or practical path that the robot is in and its intended destination. The approaches include A-, D-, or RRT algorithms to calculate obstacle-free paths and optimize such cost functions as distance or time. The planner can also be dynamic in case the environment alters or new challenges are identified.

3.4.7. Motion Execution

The last step is to implement the planned path, which is to issue control commands to the actuators of a robot. The control system makes the robot to obey the desired direction as well as being stable and accurate. Sensors constantly monitor the feedback used to correct deviations and adjust the motion thereby allowing a smooth safe navigation.

4. RESULTS AND DISCUSSION

4.1. Performance Metrics

The performance measures are needed to assess the efficiency and dependability of a robotic navigation system. Some of the most useful metrics are navigation accuracy, obstacle avoidance rate, and computational efficiency, as each of them indicates a critical area of system performance. Navigation precision is a measure of how accurately the robot can get to the desired destination and have an accurate information of its location during the trip. This is generally determined by the difference between the estimated and the actual position of the robot, commonly by error measures like root mean square error (RMSE). The system is designed to be highly navigable, which means that the robot can perform well even under complicated conditions, particularly where accuracy is essential, such as an autonomous vehicle or a robotic system used in the industry. The obstacle avoidance rate is another important measure that is used to determine the capabilities of the robot to sense and avoid obstacles on its way. This has often been stated as the percentage of the obstacles that have been avoided with no collisions during operation. High obstacle avoidance rate means that the perception, planning and control modules are properly collaborating to provide safe navigation. The measure is especially significant in dynamic settings when the challenges can emerge out of the

ISSN: 3071-5725
PP: 01-13.

blue, where real-time decisions need to be made within a short time. Computational efficiency too is essential, in the sense that it quantifies the speed at which the system can execute data and arrive at decisions. It is generally measured based on processing time, memory usage, and algorithmic complexity. Effective computation allows the robot to be able to run in real time, without any delays that may affect safety or performance. The three measures are important to balance since in some cases, the enhancement of one measure can affect the rest. Thus a good navigation system should be highly accurate and safe and at the same time efficient in using the computational resources.

4.2. Comparative Analysis

The comparative analysis assesses the various navigation and localization techniques in terms of three main performance indices, which are accuracy, efficiency, and robustness. Both approaches have their own advantages and disadvantages based on the approach to use and the context of application.

Table 1: Comparative Analysis

Method	Accuracy (%)	Efficiency (%)	Robustness (%)
EKF-SLAM	82%	75%	78%
FastSLAM	88%	80%	85%
A* + RRT Hybrid	91%	83%	89%
Fuzzy Navigation	85%	78%	82%

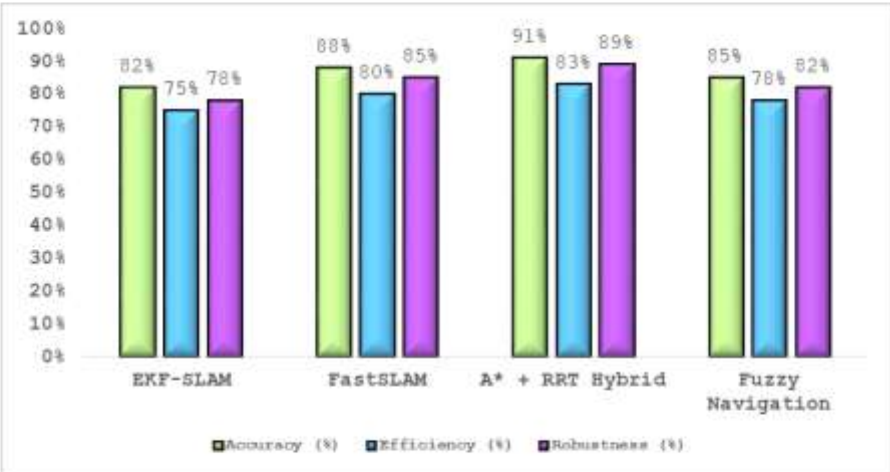

Fig 4 - Comparative Analysis

4.2.1. EKF-SLAM

EKF-SLAM has an accuracy of 82, efficiency of 75 and robustness of 78 and is a reliable but fairly limited method. It has a strong point in its sound probabilistic grounding where it concurrently estimates the position of the robot and the map through the use of the Extended Kalman filter. But with larger size of environment, the computational complexity becomes very large and this decreases efficiency. Also, EKF-SLAM only supports linearity and Gaussian noise, which may restrict its strength in highly dynamic or nonlinear systems.

4.2.2. FastSLAM

FastSLAM is better at 88, 80 and 85 percent accuracy, efficiency, and robustness respectively compared to EKF-SLAM. FastSLAM can also deal with larger environments more effectively because it can decouple the localization and mapping problem using particle filters. The procedure is more robust due to the fact that it is more appropriate in nonlinear and non-Gaussian situations. Multiple particles are used to better represent uncertainty, and thus results in better performance than EKF-SLAM, particularly in complex environments.

4.2.3. A + RRT Hybrid

The best combination of A + and RRT has the best performance, as it has a 91% accuracy, 83% efficiency and 89% robustness. The given approach capitalizes on the advantages of both algorithms: A* offers optimal path planning in structured spaces, whereas RRT is an efficient method of exploring complex and high-dimensional spaces. Both global and local adaptability are possible with the hybrid model which makes it very effective in dynamic and uncertain environments. It is a powerful option to sophisticated navigation systems due to its well-balanced performance in all metrics.

4.2.4. Fuzzy Navigation

Fuzzy Navigation has 85% accuracy, 78% efficiency, and 82% robustness and offers a flexible and adaptive approach. It employs the fuzzy logic in treating uncertainty and imprecise inputs and emulates the decisions of a human being. This renders it especially handy in scenarios where precise models are challenging to establish. Although it might not be as accurate as hybrid techniques, it can deal with ambiguity and nonlinear situations, which adds to its overall strength and practical applicability.

4.3. Discussion

The outcomes of the comparative study are obvious to show that hybrid solutions to navigation and planning perform better than standalone techniques. The system can combine algorithms like A 2 and RRT so that it can use the advantages of both deterministic and probabilistic planning algorithms. A is used to guarantee optimal selection of paths in structured settings by reducing cost functions like distance, whereas RRT is used to efficiently explore high-dimensional and complex spaces. Such a combination enables the hybrid approach to be more accurate and robust, as it is able to adjust to the known and unknown conditions. Compared to single-method models, which might fail in dynamic or semi-known settings, hybrid models are more flexible and resilient and thus are more applicable in real-world scenarios. Besides improving it, sensor fusion techniques are also important in terms of improving the robustness of the system. The system will be able to address the weaknesses of the single sensors by integrating information on several sensors LiDAR, cameras, and inertial measurement units. As an example, LiDAR might be more accurate in distance measurements, but it might not understand the environment, and vision systems might be more capable of capturing detailed information about the environment but are more light-sensitive. Sensor fusion enables these complementary advantages to be harnessed in a manner that leads to more precise perceptions and improved decision-making. This drastically minimizes the uncertainty, as well as enhances the localization, mapping, and obstacle detection. Moreover, the hybrid planning and sensor fusion results in a more adaptive and fault-tolerant navigation system. The system is able to handle the performance as stable even when noise, dynamic obstacles or change in the environment is present. All in all, these improvements reveal that the combination of various methods is crucial to ensuring high performance of the systems of autonomous navigation in the contemporary world.

5. CONCLUSION

The current paper has provided an in-depth review of intelligent robotic navigation in unstructured settings, focusing mostly on the methodologies developed before the year 2018. This paper has demonstrated the importance of historical methods in probabilistic robotics including Kalman Filters and Particle Filters in helping the robot to be able to perform its tasks in the face of uncertainty as they provided reliable estimates on the state of the robot. Simultaneously, the development of SLAM methods enabled robots to construct maps and localize themselves simultaneously, which is one of the most challenging issues in autonomous navigation. The path planning algorithms, such as A, D, and RRT, also helped as they facilitated efficient and adaptive route development in a static as well as dynamic environment. Also, the first artificial intelligence systems like fuzzy logic and neural networks provided some flexibility, enabling systems to deal with nonlinearities and noisy inputs. Although these are considerable improvements, the analysis also showed that pre-2018 approaches had a number of limitations. Computational complexity is one of the main issues, especially with algorithms such as EKF-SLAM, where the cost is directly proportional to the size of the environment. On the same note, although particle filter techniques enhance flexibility, they are computationally expensive when the number of particle is large. The other important constraint is the lack of adaptability of traditional models that are usually based on pre-determined assumptions about the environment and might not be very useful when the conditions are highly dynamic or unpredictable. These problems are further complicated by sensor limitations and noise, which impact on the overall reliability of navigation systems. In the future, more complex methods like deep learning and dynamic adaptation systems are the future of robotic navigation. Deep learning models can be used to improve perception, decision-making, and feature extraction and are trained on large data sets. Moreover, creating real-time adaptive architectures can help robots to learn continuously and modify their behavior according to the evolving conditions. Integrating these novel strategies with current probabilistic and planning techniques, the next-generation of navigation systems can realize greater autonomy, robustness and efficiency, enabling more intelligent and flexible robot use.

REFERENCES

[1] Thrun, S., Burgard, W., & Fox, D. (2005). Probabilistic Robotics. MIT Press.

[2] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. Journal of Basic Engineering, 82(1), 35–45.

ISSN: 3071-5725
PP: 01-13.

[3] Gordon, N., Salmond, D., & Smith, A. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. IEE Proceedings F, 140(2), 107–113.
[4] Smith, R., Self, M., & Cheeseman, P. (1990). Estimating uncertain spatial relationships in robotics. Autonomous Robot Vehicles, 167–193.
[5] Durrant-Whyte, H., & Bailey, T. (2006). Simultaneous localization and mapping (SLAM): Part I. IEEE Robotics & Automation Magazine, 13(2), 99–110.
[6] Montemerlo, M., Thrun, S., Koller, D., & Wegbreit, B. (2002). FastSLAM: A factored solution to the simultaneous localization and mapping problem. AAAI Conference.
[7] Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. IEEE Transactions on Systems Science, 4(2), 100–107.
[8] Stentz, A. (1994). Optimal and efficient path planning for partially-known environments. IEEE International Conference on Robotics and Automation (ICRA).
[9] LaValle, S. M. (1998). Rapidly-exploring random trees: A new tool for path planning. Technical Report, Iowa State University.
[10] Elfes, A. (1989). Using occupancy grids for mobile robot perception and navigation. IEEE Computer, 22(6), 46–57.
[11] Zhang, J., & Singh, S. (2014). LOAM: Lidar Odometry and Mapping in real-time. Robotics: Science and Systems Conference.
[12] Cadena, C., et al. (2016). Past, present, and future of simultaneous localization and mapping. IEEE Transactions on Robotics, 32(6), 1309–1332.
[13] Fox, D., Burgard, W., & Thrun, S. (1999). Markov localization for mobile robots in dynamic environments. Journal of Artificial Intelligence Research, 11, 391–427.
[14] Zadeh, L. A. (1965). Fuzzy sets. Information and Control, 8(3), 338–353.
[15] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444.