

## Review

# Stability Analysis and Navigational Techniques of Wheeled Mobile Robot: A Review

Kailash Kumar Borkar <sup>1</sup>, Turki Aljrees <sup>2</sup> , Saroj Kumar Pandey <sup>3</sup> , Ankit Kumar <sup>3,\*</sup> , Mukesh Kumar Singh <sup>1</sup>, Anurag Sinha <sup>4</sup> , Kamred Udham Singh <sup>5</sup> and Vandana Sharma <sup>6</sup> 

<sup>1</sup> Department of Industrial & Production Engineering, School of Studies of Engineering & Technology, GGV, Bilaspur 495001, India; kailashborkar04@gmail.com (K.K.B.); mukeshetggv@gmail.com (M.K.S.)

<sup>2</sup> Department College of Computer Science and Engineering, University of Hafr Al-Batin, Hafar Al-Batin 39524, Saudi Arabia

<sup>3</sup> Department of Computer Engineering & Applications, GLA University, Mathura 281406, India; saroj.pandey@gla.ac.in

<sup>4</sup> Department of Computer Science and Information Technology, IGNOU, New Delhi 110001, India; anuragsinha257@gmail.com

<sup>5</sup> School of Computing, Graphic Era Hill University, Dehradun 248002, India; 11004033@gs.ncku.edu

<sup>6</sup> Computer Science Department, CHRIST (Deemed to be University), Delhi-NCR Campus, Ghaziabad 201204, India; vandana.juyal@gmail.com

\* Correspondence: kumar.ankit@gla.ac.in

**Abstract:** Wheeled mobile robots (WMRs) have been a focus of research for several decades, particularly concerning navigation strategies in static and dynamic environments. This review article carefully examines the extensive academic efforts spanning several decades addressing navigational complexities in the context of WMR route analysis. Several approaches have been explored by various researchers, with a notable emphasis on the inclusion of stability and intelligent capabilities in WMR controllers attracting the attention of the academic community. This study traces historical and contemporary WMR research, including the establishment of kinetic stability and the construction of intelligent WMR controllers. WMRs have gained prominence in various applications, with precise navigation and efficient control forming the basic prerequisites for their effective performance. The review presents a comprehensive overview of stability analysis and navigation techniques tailored for WMRs. Initially, the exposition covers the basic principles of WMR dynamics and kinematics, explaining the different wheel types and their associated constraints. Subsequently, various stability analysis approaches, such as Lyapunov stability analysis and passivation-based control, are discussed in depth in the context of WMRs. Starting an exploration of navigation techniques, the review highlights important aspects including path planning and obstacle avoidance, localization and mapping, and trajectory tracking. These techniques are carefully examined in both indoor and outdoor settings, revealing their benefits and limitations. Finally, the review ends with a comprehensive discussion of the current challenges and possible routes in the field of WMR. The discourse includes the fusion of advanced sensors and state-of-the-art control algorithms, the cultivation of more robust and reliable navigation strategies, and the continued exploration of novel WMR applications. This article also looks at the progress of mobile robotics during the previous three decades. Motion planning and path analysis techniques that work with single and multiple mobile robots have been discussed extensively. One common theme in this research is the use of soft computing methods to give mobile robot controllers cognitive behaviors, such as artificial neural networks (ANNs), fuzzy logic control (FLC), and genetic algorithms (GAs). Nevertheless, there is still a dearth of applications for mobile robot navigation that leverage nature-inspired algorithms, such as firefly and ant colony algorithms. Remarkably, most studies have focused on kinematics analysis, with a small number also addressing dynamics analysis.

**Keywords:** kinematic analysis; dynamic analysis; path planning; navigation; WMR



**Citation:** Borkar, K.K.; Aljrees, T.; Pandey, S.K.; Kumar, A.; Singh, M.K.; Sinha, A.; Singh, K.U.; Sharma, V. Stability Analysis and Navigational Techniques of Wheeled Mobile Robot: A Review. *Processes* **2023**, *11*, 3302. <https://doi.org/10.3390/pr11123302>

Academic Editor: Raul D. S. G. Campilho

Received: 6 September 2023

Revised: 31 October 2023

Accepted: 3 November 2023

Published: 26 November 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Autonomous WMRs are a tough research topic for a variety of reasons. The WMR should be capable of recognizing features, identifying obstructions and targets, learning by experience, and determining the best route and navigation [1]. For many years, mobile robot scientists and engineers have focused their efforts on developing alternative control algorithms for the path and motion planning problem [2]. According to reports, the researchers are working on the stability and navigation of WMR. The WMR capabilities necessitate the use of many WMR research areas at the same time [3]. The integration of several distinct forms of knowledge is required for WMR design. The WMR needs to comprehend mechanics, kinematics, dynamics, and control theory to handle locomotion concerns [4]. WMR navigation requires acquaintance with computer programming, artificial intelligence, information technology, and probability concepts [5].

Kinematic and dynamic analysis is also a difficult issue in WMR path and motion planning concerns since kinematics and dynamics play a vital part in obtaining mechanical behavior and developing control software for a special purpose WMR. When the WMR is cinematically stable, the next job is the development of a controller that must be capable of navigating intelligently in the real-world workspace [6]. Building intelligent controllers will need artificial intelligence (AI) techniques. This is because sophisticated methods such as fuzzy if-then rules, brain-intensive characteristics, genetic theories, and others are very good at imitating the fuzziness of human knowledge and reasoning processes [7]. Many attempts have been made to summarize the various techniques [8], considering the numerous research articles on the WMR path and motion planning challenges that have recently been published.

An autonomous WMR has difficulties with path and motion optimization. The robot must select a path and perform several robot environments to move from its starting point to its target point [9]. Path analysis and motion planning become more challenging when the environment is loaded with stationary and as well as moving obstructions. When this happens, we need to devise a fresh remedy. In addition to preventing collisions with obstruction, it needs to maintain more distance from the barrier and take shorter travel time. Performance metric optimization is required for motion planning and path analysis. Extensive WMR motion planning is required to guarantee that each WMR's route is optimum. The state of the art in robotic path planning and navigation is now concentrated on creating algorithms that let robots navigate complicated and dynamic situations in real time. Reinforcement learning, deep learning, and computer vision are just a few examples of artificial intelligence and machine learning techniques that are commonly used in these algorithms to help the robot comprehend its surroundings and make wise judgments about how to move through them [10].

### 1.1. Stability Analysis of Wheeled Robot

Different researchers have solved the problem of kinematic and dynamic analysis of WMR. In the present scenario, most of the research works published in the past three decades are summarized below for kinematic and dynamic analysis of WMR.

Stability analysis methods are used to evaluate the stability of a wheeled mobile robot (WMR) while it is moving. In general, these methods involve analyzing the dynamics of the robot and assessing its ability to maintain its position or follow a desired trajectory. Some of the main features of stability analysis methods are described below:

- I. Kinematic and dynamic models: Stability analysis approaches use mathematical models to describe the robot's dynamics and kinematics. The kinematic model is solely based on the robot's motion, disregarding any stresses or torques that may affect the robot's mobility. Contrarily, the dynamic model considers the pressures and torques that affect the robot's motion.
- II. The stability analysis criteria are used to evaluate the stability of the robot. The most common stability analysis criteria used in WMRs are Lyapunov stability, passivity-based stability, and control Lyapunov functions (CLFs). Lyapunov sta-

bility analysis uses Lyapunov functions to determine the stability of the robot's motion. Passivity-based stability analysis is based on the concept of passivity and uses energy functions to evaluate the stability of the robot. CLFs are used to design control laws that ensure the stability of the robot.

- III. Trajectory tracking: To assess the robot's ability to follow a desired trajectory, stability analysis techniques are also applied. The discrepancy between the robot's actual trajectory and its anticipated trajectory is known as the trajectory tracking error. The robot's stability is measured by its capacity to achieve a trajectory tracking error of zero.
- IV. Control design: Stability analysis methods are used to design control laws that ensure the stability of the robot. The control laws are designed based on the dynamics of the robot and the stability analysis criteria. The control laws are then implemented in the robot's control system to ensure stable motion.

Global and local navigation strategies for mobile robots may be generally divided into these two groups. Using global navigation algorithms, the robot's whole journey from its starting point to its destination is planned. The surroundings are mapped out beforehand using sensors like LiDAR or cameras, and this map serves as the basis for path planning. When the environment is well known and static and the robot must travel great distances, global navigation techniques are helpful.

On the other hand, local navigation techniques focus on the immediate surroundings of the robot and its ability to avoid obstacles in real time. These methods are especially helpful in surroundings that are dynamic and where impediments could arise or shift suddenly. Sonar or infrared sensors are frequently used in local navigation techniques so that the robot may identify impediments and change its course as necessary. The potential field approach, which models the robot as a point particle in a field of attracted and repulsive forces, is a well-liked local navigation methodology. The robot is drawn by the attractive force in the direction of its destination, while it is pushed away from obstacles by the repulsive force. The resultant route generated by the net force vector is then followed by the robot. Another technique is the occupancy grid mapping method, where the environment is represented as a grid of cells, and each cell is classified as either occupied or free. The robot's sensors are used to update the occupancy grid map, and the robot plans its path based on the current state of the grid [11].

- I. Simultaneous localization and mapping (SLAM): Popular mapping and localization methods include SLAM. It entails mapping the surroundings and determining the robot's position inside that map at the same time. When an existing map of the environment is not accessible, SLAM is very helpful.
- II. Monte Carlo localization (MCL): For determining the robot's location inside a known map, MCL is a particular kind of particle filter. It functions by converting the location of the robot into a probability distribution and updating it in response to sensor readings.
- III. Reactive navigation: Reactive navigation techniques focus on the robot's immediate environment and are designed to quickly and efficiently respond to changing situations. Examples of reactive techniques include potential fields and behavior-based control.
- IV. Model predictive control (MPC): In order to forecast the robot's future behavior and optimize control inputs, the MPC control approach employs a mathematical model of the robot's dynamics. Planning a course and avoiding obstacles are both possible with MPC.
- V. Reinforcement learning (RL): RL is a machine learning technique that enables the robot to learn how to navigate through trial and error. The robot receives rewards or punishments based on its actions and adjusts its behavior to maximize the rewards.

Overall, the field of robotic stability analysis and navigation is rapidly advancing, and many promising techniques are being developed and refined to enable robots to navigate in complex environments and perform increasingly complex tasks.

### 1.2. Major Objective

- a. Provide an extensive review that focus on navigational complexities in the context of wheeled mobile robots (WMRs).
- b. Trace the historical and contemporary developments in WMR research, including the establishment of kinetic stability and the construction of intelligent WMR controllers.
- c. Examine the importance of stability and intelligent capabilities in WMR controllers and their impact on WMR performance.
- d. Present a comprehensive overview of stability analysis techniques tailored for WMRs, including discussions on Lyapunov stability analysis and passivation-based control.
- e. Explore various navigation techniques for WMRs, covering aspects such as path planning, obstacle avoidance, localization and mapping, and trajectory tracking in both indoor and outdoor settings.

### 1.3. Paper Organization

The present paper is organized as follows: the introduction is presented in the first part, and stability analysis of WMR is discussed in the second part, which is subdivided into the kinematic and dynamic analysis of WMR. Path-planning and navigational techniques are discussed in the third and fourth parts, respectively. Part five presents the various navigational techniques. Finally, a comparison with the existing techniques and conclusions are drawn in the last part of the paper.

## 2. Literature Review

In this section, we review the various existing works and their related technology to identify the research gap.

### 2.1. Kinematic Analysis of WMR

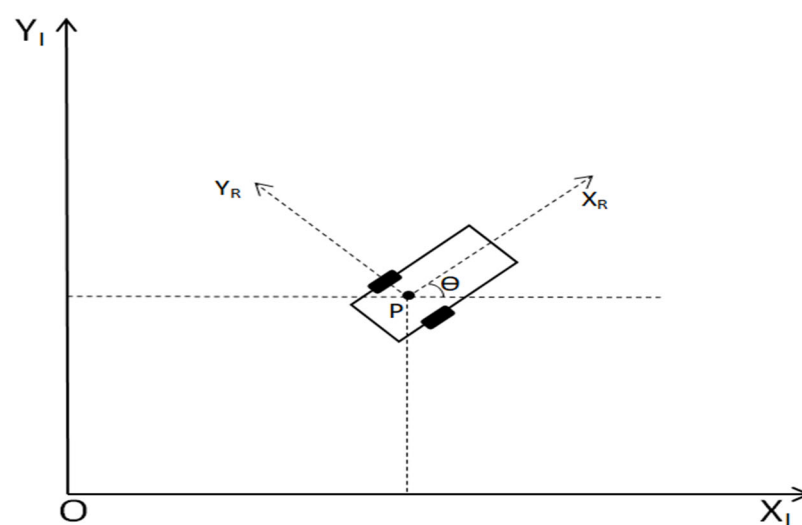
The foundational exploration of wheeled mobile robots (WMRs) to assess the mechanical system's dynamics involves a kinematic inquiry, which is imperative for ensuring the seamless execution of predefined trajectories [12]. This analytical process yields insights into devising control software for the WMR hardware, as well as for the design of robots tailored to tasks demanding a comprehensive grasp of the mechanical conduct of the robot [13]. Given that WMRs necessitate lower exertion compared to legged or tracked robots, they hold significant potential for widespread adoption within contemporary industrial scenarios [14,15]. Lin et al. presented various configurations for mobility [16]. Among these, differential and synchronous mechanisms are the prevailing choices for enabling tricycle- or car-like robotic locomotion [17]. The intricate theoretical underpinnings related to autonomous motion planning and control in the context of WMRs captivate academic interest, despite their tangible practical applications [18]. Recent focus has been directed towards WMR motion control, with a special focus on systems exhibiting non-homonymic behaviors, as they suggest that a reduced set of robot actuators can effectively govern WMR actions. Such systems epitomize non-homonymic mechanics due to the stringent constraints imposed on wheel movement by the rolling dynamics [19]. A multitude of controllers cater to wheeled mobile robots (WMRs) while grappling with non-homonymic constraints [20]. The dominant strategies for governing WMR behavior encompass trajectory tracking and posture stabilization mechanisms [20]. The application of smooth time-invariant control fails to ensure feedback stabilization at specific points [21]. This intrinsic nonlinearity underscores the fundamental nature of the issue, rendering conventional design methods, even within local contexts, inadequate. Dynamic feedback linearization, an invaluable design technique, provides a solution that remains valid for trajectory tracking and set point control concerns, utilizing the illustrative framework of

unicycle kinematics [22]. Ultimately, a nonlinear feedback system was adopted to globally resolve the trajectory tracking challenge [23], following preliminary attempts to devise localized controllers. The concept of backtracking is amenable to developing a recursive chained trajectory tracking methodology for non-homonymic systems. Posture stabilization is attainable through the deployment of discontinuous feedback controllers [24]. Varied approaches to discontinuous control, including dynamic feedback linearization, have been harnessed to address posture stabilization issues [25]. Pioneering the realm of smooth time-varying stabilization, Samson's contributions are noteworthy [26]. To achieve optimal control efficacy, a robust autonomous driving control framework must seamlessly adapt to fluctuations in slip conditions.

The examination of how wheel motion influences the mobility and positioning of wheeled mobile robots (WMRs) is encapsulated by WMR kinematic analysis. This analytical pursuit is pivotal for formulating control strategies that unravel the robot's behavior and performance intricacies. Scrutinizing the robot's position, velocity, acceleration, and wheel movement constitutes the bedrock of this study [27]. The outcomes of such an analysis can be employed to formulate equations encapsulating the robot's motion and position, thereby furnishing the foundation for the development of navigation algorithms and control methodologies. In the realm of WMR kinematic analysis, the robot is commonly conceptualized as an assembly of interconnected rigid bodies linked by joints.

According to authors [28,29], Figure 1 visually elucidates the interplay between the global and local reference frames within WMRs. These reference frames are imperative for determining the robot's spatial coordinates on the plane. In the context of our illustration, the global reference frame pertains to the inertial frame denoted as O with coordinates  $X_i$ ,  $Y_i$  on the robot's work plane. Conversely, the local reference frame is represented by  $X_R$ ,  $Y_R$  for a WMR. By leveraging a reference point P situated on the WMR chassis, the angular differential between the global and local frames, as delineated by [30], serves as the foundation for calculating the WMR's position. Consequently, the broader academic community has universally embraced the subsequent mathematical definition governing WMR positioning.

$$\xi_I = \begin{bmatrix} X \\ Y \\ \theta \end{bmatrix} \quad (1)$$



**Figure 1.** WMR global reference frame and the robot local reference frame.



Now, the mapping between these frames is accomplished using the given orthogonal rotation matrix:

$$R(\theta) = \begin{bmatrix} C\theta & -S\theta & 0 \\ S\theta & C\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

where  $S\theta$  is  $\sin\theta$  and  $C\theta$  is  $\cos\theta$ .

Equation (2) computes the WMR motion in the global reference frame from motion in a local frame as follows:

$$\xi_R = R(\theta)\xi_I \text{ and } \dot{\xi}_R = R(\theta)\dot{\xi}_I \quad (3)$$

Therefore,

$$\xi_R = \begin{bmatrix} C\theta & -S\theta & 0 \\ S\theta & C\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\theta} \end{bmatrix} \quad (4)$$

and

$$\dot{\xi}_I = R(\theta)^{-1}\dot{\xi}_R \quad (5)$$

Equation (4) satisfies the entire research community to pose estimation for WMRs.

The challenge concerning the kinematic analysis of wheeled mobile robots (WMRs) within irregular workspaces was addressed by [30]. They formulated a mathematical kinematic model for WMR, conceptualizing the wheels as tori, and introduced the concept of a passive joint to facilitate lateral degrees of freedom. Parham and Deepak's investigation of WMR kinematic behavior resulted in the classification of WMRs into five distinct categories: fixed standard WMR, steered standard WMR, Swedish WMR, spherical WMR, and castor WMR. Leveraging highly nonlinear two-wheeled inverted pendulum system, authors proposed a proportional integral sliding mode control (PISMIC) framework for two-wheeled inverted pendulum WMRs, thus enabling robust stabilization and disturbance rejection via sliding mode control techniques [31].

In [32] introduced a novel differentiable, time-varying controller to address the kinematic control intricacies of WMRs. This innovative control structure emphasized the differentiable nature of the problem and demonstrated the generation of a global exponential controller for precise dynamic models using conventional backstepping methodologies. It is proposed a distinctive omnidirectional and quasi-omnidirectional drive mechanism for dynamic models of WMRs. They introduced an inventive epicyclic cam transmission that conferred several advantages over conventional gear-trains, including reduced friction, heightened rigidity, and enhanced precision [33]. These drives were substantiated through virtual prototypes [34]. It is suggested the viability of a reliable adaptive controller for faultless velocity tracking, while, showcased the capability of temerity structures to execute prescribed trajectory routes through the manipulation of their elements [35]. Meanwhile, it derived fundamental track control algorithms for differentially steered WMRs by exploring the robot's intrinsic kinematics, providing globally convergent tracking control methods for differentiable reference paths [36]. While the kinematic model of WMRs yields satisfactory accuracy for scenarios involving reduced speed, light weights, and diminished acceleration, it faces limitations in representing the complexities of WMR tasks necessitating high speeds and substantial loads. Novel drives have been designed to cater to unidirectional and quasi-unidirectional robots, employing an innovative epicyclic cam transmission with beneficial attributes [37]. It is developed an interactive interface to elucidate various strategies used for addressing mobile robot motion planning issues [38]. Numerous control solutions have been proposed in the literature to tackle kinematic stability issues in WMRs [39]. In [40], semiglobal practical stabilizing control methods address non-homonymic WMRs with saturated inputs, offering a twofold advantage in terms of stabilizer types and ensuring simulations adhere to appropriate upper bounds by employing original system inputs [41]. Vivo robot wheel mechanisms have been explored for camera and sensory equipment transport, supporting laparoscopy [42]. In [43], combined finite element modeling, kine-

matic analysis, and animal research to develop wheeled in vivo mobile robots, ultimately redefining their work based on experimental findings [44,45].

The simplest but most effective techniques for kinematic stabilization are nonlinear controllers based on Lyapunov theory. The following is the positive Lyapunov function:

$$V = \frac{1}{2}(x_e^2 + y_e^2) + (1 - \cos \theta) \quad (6)$$

The time derivative of Equation (6) becomes

$$\dot{V} = v_r x_e \cos \theta - v_d x_e + v_r \sin \theta_e y_e + w_r \sin \theta_e - w_d \sin \theta_e$$

or

$$\dot{V} = (v_r \cos \theta - v_d) x_e + \sin \theta_e (v_r y_e + w_r - w_d) \quad (7)$$

where  $v_d$  and  $w_d$  are

$$v_d = v_r \cos \theta_e - k_e x_e$$

or

$$w_d = w_r + v_r y_e + k_\theta \sin \theta_e \quad (8)$$

Substituting Equation (8) in Equation (7) gives

$$\dot{V} = -k_e x_e^2 - v_r k_\theta \sin^2 \theta_e \quad (9)$$

The above equation shows the negative semidefinite.

The framework devised by [46] introduces a comprehensive approach to incorporating geometry and kinematics in the obstacle avoidance process, distinctly segregating these constraints from the avoidance technique's implementation. Conventionally, obstacle avoidance algorithms tend to overlook kinematic restrictions. Minguez and Montano's methodology caters to a wide range of nonholonomic vehicles and aims to construct a two-dimensional manifold within the configuration space, characterized by differentiability along simple circular paths. This manifold encompasses configurations reachable at each step of the obstacle avoidance process, serving as a foundation for various approaches. Their work significantly advances through accurate obstacle depiction calculations specific to each robot's shape within this manifold.

In [47], present an allocated effective controller tailored for compliant framed wheeled mobile robots (WMRs). Notably, this study achieves a novel accomplishment for single-axle unicycle-type robots by employing backstopping techniques to develop a distributed nonlinear damping controller. Furthermore, the controller's robustness was enhanced to encompass multi-axle controller systems. It now demonstrates improved resilience against disruptions arising from inaccurate modeling, particularly nonlinear frame forces attributed to axle contact. The controller was evaluated utilizing a two-axle scout configuration, and their findings were validated through real robot experimentation. In a distinct vein, in [48] explored a moving platform equipped with offset motorization and a caster wheel to assess its singularity and motion potential. Their analysis delved into motion equations for kinematic analysis, culminating in the identification of necessary and sufficient conditions for actuation. This led to the creation of a wheeled mobile robot (WMR) workspace devoid of singular configurations, amplifying its utility for practical applications.

## 2.2. Dynamic Analysis of Mobile Robot

The dynamic behavior of the wheeled robot provides a relationship between the torques and motions of the wheeled robot for the simulation and design of the control algorithm. The problems of dynamic analysis of WMR have been solved by many researchers, summarized below. Dynamic analysis of mobile robots involves studying the forces and torques that act on the robot and how they affect the robot's motion and stability. This analysis is important in understanding the robot's behavior and performance in dynamic

environments, as well as in designing control strategies for the robot that can ensure its stability and safety. The dynamic analysis of mobile robots involves modeling the robot as a mechanical system and using the laws of physics to describe its motion, such as those due to gravity, friction, and external forces, and asking how these forces and torques affect the robot's motion.

Accounting for the nonlinearities and uncertainties that result from the complex dynamics and interactions of mobile robots with their environment is one of the main issues in the dynamic analysis of these systems. This issue can be resolved by using numerical approaches like finite element analysis or multimode simulation, which can simulate the robot's movement and behavior in a variety of situations. The dynamic analysis of the robot path examines how the robot's center of mass moves and alters over time as a result of the pressures and torques operating on it. Understanding the robot's performance and stability while in motion requires this examination. It is important to think about the pressures and torques acting on a robot and how they interact to analyze its dynamic route. This may be accomplished by using the equations of motion, which explain the connection between the forces and torques acting on the robot and the center of mass motion as a result. Newton's principles of motion and the idea of conservation of angular momentum may be used to develop the equations of motion. In terms of the robot's location, velocity, and acceleration, as well as the forces and torques acting on it, these equations may be utilized to explain the robot's motion. The robot's dynamic route may be examined, and its stability and performance during mobility can be determined, using the equations of motion that we have derived. This entails taking into account the control, mass, geometry, and external forces of the robot.

The WMR dynamic model [49], represents a heavy, differentially steered WMR in both loaded and empty states. These findings were applied to support a dynamic model that was recently built and contains a complicated depiction of a tire to appropriately account for the interaction between the tire and the ground. The inadequacies of the kinematic models under conditions of high load and/or high speed were then illustrated using the dynamic model. Using the active caster, [50] enabled distributed actuation of objects on the planar workspace. A mobile robot with a dual-wheel active caster was studied and tested. We instead looked at the dynamics of the WMR using the force transmission ratio and the kinematic isotropic index. A simple WMR's exponential stabilization was studied and solved using [51] time-invariant, discontinuous, pursued state feedback control approach. Its effectiveness was evaluated in the presence of noisy data and incorrect models. The closed-loop system's operation was thoroughly examined. In [52,53] developed a dynamic model for a mobile robot with wheels that can operate while carrying high weights.

A generalized set of nonlinear dynamic models for various non-homonymic WMRs operating in two-dimensional Cartesian spaces was developed by [54] using the Lagrange formalism, which offers a more organized modeling approach. For two differentially steered, non-homonymic WMRs linked to a common payload [55], offered a nonlinear controller based on the Lagrange idea. Stability analysis and control design are challenging for WMRs due to their non-homonymic dynamics [56]. The ability to characterize the dynamics of output-tracking control laws in terms of full-state tracking errors has several advantages, including a better comprehension of the internal dynamics and zero dynamics of the tracking-error system and an enhanced comprehension of trajectory tracking. By using selective state feedback to block-decouple the resulting large-order multivariable system model into five separate subsystems, we can accurately represent the dynamics of each robot's departure from a predetermined path, as well as the dynamics of their forward motion in two other subsystems and the dynamics of the payload in the fifth subsystem [57]. There is a pole-placement method-based intelligent controller available for each robot subsystem, including self-tuning adaptive controllers for the nonlinear deviation dynamics subsystems. The effectiveness of the system is evaluated afterward through simulation for the case in which each robot is traveling in a curvilinear motion. In [58], the issues of dynamic stabilizing and dynamic modeling of a non-homonymic WMR were



both overcome. The dynamic model, which also has nonholonomic restrictions, was built on the kinematic model. Linear controllers can employ the recommended control strategy, which just needs the robot's localization coordinates, to circumvent the control problem. The dynamic model of a non-homonymic wheeled robot was stabilized by [59] with the right orientation. The first technique makes use of physical modeling, whereas the second depends on determining the experimental dynamics characteristics of mobile robots. Then, a WMR dynamics model might be developed [60].

The following set of dynamic equations, which Gholipour and Yazdanpanah constructed based on the Euler–Lagrange formulation [61], characterize a sizable class of mechanical non-homonymic systems.

$$M(\xi_I)\ddot{\xi}_I + P(\xi_I, \dot{\xi}_I)\dot{\xi}_I + G(\xi_I) = B(\xi_I)\tau + J^T(\xi_I) \quad (10)$$

where  $M(\xi_I)$  are a symmetric positive definite  $n \times n$  matrix, representing the mass and inertia of the system. The vector  $P(\xi_I, \dot{\xi}_I)$  contains the centripetal and Coriolis torques, which arise due to the motion of the system. The vector  $G(\xi_I)$  represents the gravitational torques that act on the system due to its orientation relative to a gravitational field. In the area of robotics and mechanics, two mathematical expressions involving vectors and matrices are combined in the phrase  $B(\xi_I)\tau + J^T(\xi_I)$ . Let us dissect the parts:

- $B(\xi_I)$  denotes a matrix or function that is dependent on a *vector*  $\xi_I$ . You would need to specify B's precise nature and how it depends on  $\xi_I$  in your particular circumstance.  $B(\xi_I)$  may stand for a transformation matrix or a function associated with a specific component of the system under study, for example.
- $\tau$ : This vector represents the force or torque that is being applied. When employed in the context of dynamics,  $\tau$  usually denotes outside forces or control inputs operating on a mechanical system.
- $J^T(\xi_I)$ : In this case,  $J$  is a Jacobian matrix.

One can write the dynamic equation of a wheeled mobile robot according to Equation (6), using the fact that  $G(\xi_I)$  and  $P(\xi_I, \dot{\xi}_I)$  are zero.

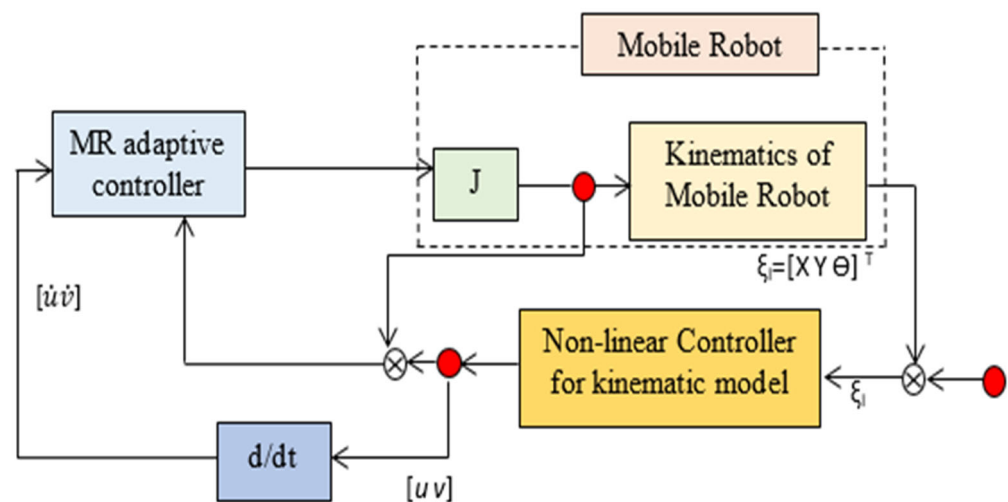
$$\begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix} = \frac{1}{R} \begin{bmatrix} \cos \theta & \cos \theta \\ \sin \theta & \sin \theta \\ L & -L \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} + \begin{bmatrix} \sin \theta \\ -\cos \theta \\ 0 \end{bmatrix} \lambda \quad (11)$$

It is a 3 by 3 matrix. The off-diagonal components are all zeros, whereas the diagonal elements are "m" and "I". With "m" standing for mass and "I" for moment of inertia, this matrix most likely depicts the inertia characteristics of a rigid body L. In addition, they presented a control scheme for a dynamic model of a mobile robot, as illustrated in the block diagram in Figure 2, which outputs the appropriate linear and angular velocities for kinematic stabilization from a nonlinear controller.

### 2.3. Eliminate Modeling Uncertainties and Environmental Disturbances

Precise dynamic modeling of the robot and its surroundings is necessary if we want to improve wheeled mobile robot stability analysis and navigational strategies [61,62], as well as lower modeling uncertainties and environmental disturbances in a study. It is important to develop strong obstacle avoidance techniques through obstacle detection and path planning; use sophisticated feedback and adaptive control strategies [63–65] for stable navigation; and enhance perception accuracy through sensor fusion and calibration. The accuracy and dependability of a wheeled mobile robot study should be ensured by performing both simulated and real-world testing to verify the algorithms' performance and robustness across a range of environmental situations. To improve wheeled mobile robot stability analysis and navigational methods while lowering modeling uncertainties and environmental disturbances in a study, one may use a multicentric strategy [66,67]. This can be achieved by making the dynamic models more accurate by taking sensor behaviors,

wheel–ground interactions, and environmental elements into consideration, enhancing perception reliability by implementing sensor fusion techniques and maintaining sensor calibration, and creating sophisticated control and feedback schemes to ensure steady and accurate robot navigation [68,69]. Utilizing advanced obstacle detection algorithms and dynamic path-planning techniques can enhance the focus on robust obstacle avoidance. To guarantee the accuracy and robustness of the wheeled mobile robot's performance, the study can be validated using thorough simulations and rigorous real-world testing in a variety of environmental situations. The focus should be on developing accurate dynamic models that take wheel–ground interactions and environmental elements into account to improve stability analysis and navigational strategies of wheeled mobile robots in the study. The use of sensor fusion and calibration will increase the precision of perception, and for steady navigation, sophisticated control techniques like PID [70], LQR [71], or MPC are used. Reliable algorithms should be created to detect obstacles and dynamic path-planning techniques should be used to ensure efficient obstacle avoidance. To assure the accuracy and dependability of a wheeled mobile robot's performance, the study should be validated using comprehensive simulations and rigorous real-world testing in a variety of environmental circumstances.



**Figure 2.** Block diagram of control structure for dynamic model of mobile robot.

### 3. Path and Motion Planning

The issue of mobile robot route and motion planning challenges has been addressed in several academic articles. The time it takes the robot to arrive at the goal has been optimized along with several paths. Following the course taken by the mobile robot without colliding was each author's aim. The following describes the various authors' works.

Traditional maps may have issues with length, sharp curves, or interactions with barriers. Examples are visibility graphs, probability maps, and skeleton maps. Several conventional solutions to path-planning challenges were presented by [72]. By creating roadmaps with polygons surrounding the crossing locations, their recommended solution addressed the problems with conventional techniques. In [73] offered interactive tools to assist in the study of several well-known techniques and tactics for tackling WMR motion and path-planning problems. They focused on clearly outlining obstacles to be useful for instructional purposes in a variety of professions. A robot exploring complex environments must find a compromise between the requirement for efficient and ideal courses and the need to react to unanticipated circumstances. In [74] developed the technique and design for a spherical mobile robot that utilizes two internal rotors to roll on a plane and relies on the conservation of angular momentum. They suggested that because the spherical mobile robot does not fall into any of these categories, the path-planning techniques now in use are inappropriate for this system.

In [75] studied the problem of motion planning and control in the presence of uncertainty for a wheeled mobile robot subject to state and actuator constraints that follow a predetermined path inside an environment with moving obstacles, as shown in Figure 3. The difficulty of motion planning with moving objects in an uncertain environment was overcome. The robot's mobility must be as close to a nominal velocity profile as is practical in time. The control law parameters must impose as minimal restriction as possible on the ranges of motion. The recommended method may be applied online and is efficient from a computing standpoint. A tough simulation instance was used to demonstrate the method's potential. In [76] first presented sampling-based robot motion planning for WMRs. It was emphasized that the creation of WMR motion planning algorithms will make them more useful in real-world settings and a variety of circumstances. They ultimately offered concepts that can be executed almost exactly as-is on a platform for actual mobile robots. For a WMR that is described using second-order dynamics, [77] took up the problem of temporal logic motion planning. These temporal logic specifications can capture more complex specifications like sequencing and obstacle avoidance in addition to more common control specifications like reachability and invariance. Three fundamental steps were engaged in their investigation. Due to a control rule they initially developed, the dynamic model may adhere to a more straightforward kinematic model with a globally bounded inaccuracy. Second, it offers a robust definition of temporal logic that takes into consideration the tracking errors from the preceding stage. Finally, they developed a robust temporal logic path-planning problem for the kinematic model using automata theory and simple local vector fields. For WMR route planning, [78] applied unique repulsive potential functions and potential field techniques. In doing so, they addressed the problem of goals being unreachable with close obstacles. Based on multilayered cellular automata, in [79] developed the reactive path-planning method for a non-homonymic mobile robot. They came up with the algorithm that offers the shortest steering radius, the best course, and more fluid trajectories that do not stop and turn regularly. Also, they developed a technique based on the directed (anisotropic) propagation of attractive and repulsive potential. Their method finds all the ideal, collision-free pathways that follow the shortest valley of a hypothetical, embedded hypersurface in 4D space, built under the given constraints.

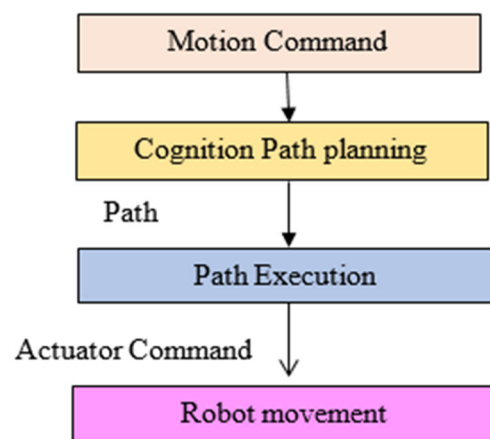


Figure 3. Motion control for WMR.

The technique proposed by [80] focused on planning the trajectories and motions of WMR information. This aimed to produce a wheeled robot's appropriate mobility; particular geometrical constraints were imposed on the relative positions and orientations of the robots throughout their motion. Their research introduced particular motion planning techniques for WMR formations with non-homonymic constraints. The kinematic equations were developed to accommodate various formation classes that had to be kept while the group as a whole was moving. For members of a team of differentially-driven (DD) WMR traveling in formation for final deployment in cooperative payload transport

operations, in [81] proposed the ideal relative configuration. Synchronized distributed data collecting and cooperative payload transfer are two growing applications for robot collectives where precise formation-based activities are required. To create the vertices of a virtual structure, they solved the formation issues and provided the motion plans for the individual DD-WMRs.

A technique for motion planning of a WMR traveling in a cluttered environment, formerly known as forbidden zones with an arbitrary shape, size, and placement, was devised by [82]. They proposed a method that can mathematically characterize the whole robot environment and is based on the cutting-edge concept of B-surfaces. The motion-planning solution was sought on a higher-dimension B-surface such that its inverse was guaranteed. Their study offered a calculated solution for a WMR that connects the starting and finishing sites in a straight line with no self-loops in the shortest time. For the sake of clarity, they described a technique that can resolve path-planning issues on a two-dimensional (2D) flat terrain with static impediments, as well as a generalization to motion-planning issues on curved terrains. To show the usefulness and efficiency of the proposed planning, they lastly implemented the recommended technique in a range of difficult situations.

In [83], to give a dependable mobility plan that works at the sensor frequency, merged map-based and sensor-based planning operations. A workable strategy for coordinated motion planning of many robots without conflict was developed by Chiddarwar and Babu [84]. Their suggested technique's two-phase decoupled mechanism can offer the necessary coordination amongst the participating robots in offline mode. A route modification approach is utilized in the second phase to resolve conflicts and enable coordination across multiple robots. In the first phase, an algorithm is employed to identify each robot's collision-free path about stationary objects. A route modification approach is utilized in the second phase to resolve conflicts and enable coordination across multiple robots. An algorithm is used in the first stage to determine each robot's collision-free route relative to stationary objects. The coordination of the robots is also accomplished using several approaches to ensure the efficacy of the recommended plan. The timing and the route were both enhanced.

#### 4. Navigation of Mobile Robot

Mobile robot navigation is a study area with several potential applications, both military and nonmilitary. Nonetheless, a great deal of problems remain, and many of them will probably need to be addressed using either a novel theory or another approach. The autonomous navigation of mobile robots has been studied by several researchers. According to [84] mobile robot navigation is a significant and challenging problem. Robots may encounter static interior environments or dynamic environments that change swiftly. The mobile robot's goals may include reaching a particular place, following a predetermined course, or mapping out a space for later use. Many studies have been conducted throughout the world regarding this. Different methodologies were proposed on this topic, each with its advantages and disadvantages based on the recursive research. Navigation of a mobile robot involves determining the robot's trajectory through an environment to reach a desired goal location while avoiding obstacles [85]. To achieve this, we can use a control system that takes sensor measurements and generates control signals to steer the robot. The employment of a feedback control system, which generates control signals to move the robot in the direction of the destination location using the robot's present orientation and position, is a typical method [86]. The robot's motion's kinematic equations may be used to model this quantitatively. The following equations can be used, for instance, to describe a differential-drive robot:

$$w = \left( \frac{r}{wb} \right) (w_r - w_l) \quad (12)$$

where  $r$  is the diameter of the wheels,  $w_b$  is the wheelbase,  $v$  is the robot's linear velocity,  $w$  is its angular velocity, and so on. By the variables  $w_l$  and  $w_r$ , the left and right wheel velocities are represented. With the help of these equations, we can identify the control signals required to steer the robot safely away from obstacles and in the direction of the

objective location. This can be performed using a variety of algorithms, such as potential fields, artificial potential functions, or behavior-based control. In addition to kinematic equations, dynamic models can also be used to control the motion of mobile robots. These models take into account the robot's mass, moment of inertia, and other physical properties to generate control signals that ensure stable and accurate motion. This is important when the robot is moving at high speeds or encountering rough terrain.

#### 4.1. Indoor Navigation

It is conducted groundbreaking research on robotic vehicle navigation, which highlighted the importance of incorporating knowledge of the device's internal environment to navigate effectively. Early vision systems developed for mobile robot navigation relied heavily on the geometry of space and other metric data to drive the vision processes and enable self-localization. To model the internal environment, extremely complex algorithms were used to create CAD models. It is believed that these programmers can be grouped into three general categories [86].

##### 4.1.1. Map-Based Navigation

There are navigation systems that rely on topological or geometrical maps of the environment, known as model-based or map-based navigation systems. These systems typically use a geometric model or a topological representation of the surroundings to plan the movement of the robot. In a geometric model-based system, the environment is represented as a 3D geometric model using point clouds or meshes, which are built using the robot's sensors, such as cameras or LiDAR. The robot can then plan a collision-free path through the model using algorithms like A\*, RRT, or PRM. In a topological-map-based system, the environment is represented as a graph where nodes represent significant sites and edges represent routes between them. The robot's starting point and goal are connected using a series of nodes and edges that avoid hazards. The Dijkstra's algorithm or A graph search algorithm can be used for this purpose. Model-based and map-based navigation systems have several advantages, including the ability to handle complex environments with many obstacles, plan efficient paths that avoid obstacles, and adapt to changes in the environment. They can also be used with other robot capabilities, such as object recognition or manipulation, to perform complex tasks.

However, these systems also have some limitations. They require accurate models of the environment, which can be time-consuming and difficult to create. They can also be sensitive to errors in the model, such as occlusions or inaccuracies in the sensor data. Finally, they may not be suitable for environments that are constantly changing or where the robot must navigate in real time.

##### 4.1.2. Map-Building-Based Navigation

These systems use sensors to create their own topological or geometric representations of the world, which they subsequently use to aid with navigation, i.e., systems that use sensors as navigational aids. These systems frequently combine several sensors, such as cameras, LiDAR, and sonar, to provide a real-time map of the area. The robot plans its path and moves around the region using this map. The sensors aboard the robot are utilized in a sensor-based navigation system to gather information about the surroundings. A topological or geometric representation of the environment that is utilized for navigation may be made using this data. For example, a topological map might consist of nodes that represent significant locations in the environment, and edges that represent the paths between these locations. A geometric map might consist of a 3D point cloud that represents the environment's geometry. Once the map is created, the robot can use it to plan its path through the environment. This is typically carried out using algorithms such as A\*, RRT, or PRM, which find a collision-free trajectory through the map. The robot's sensors are used to update the map in real time as the robot navigates through the environment, allowing it to respond to changes in the environment and avoid obstacles. A control method called



fuzzy logic control (FLC) employs fuzzy sets and rules to make choices. Given that it permits imprecise or ambiguous inputs and outputs, it is very helpful when working with complicated or uncertain systems. A fuzzifier, a rule basis, and a fuzzified are the three primary parts of FLC. The rule base determines the output using a collection of IF–THEN rules, the fuzzifier transforms the fuzzy output into a crisp value, and the fuzzifier maps the input variables to fuzzy sets. Navigation is based on a representation of the environment in C. The firefly method (FA) is a metaheuristic optimization method that draws inspiration from firefly behavior. Finding a function’s global optimum is one of its main applications in problem-solving. In FA, each firefly stands in for a potential remedy to the issue, and their behavior is determined by their allure and closeness to other fireflies. A firefly’s attractiveness is defined by the objective function that is being optimized, and its proximity is determined by its location in the solution space. FLC and FA work well together to solve optimization and control issues. FLC, for instance, may be used to regulate a mobile robot’s speed or steering angle, whereas FA can be used to tailor the FLC settings for a particular job or environment [87].

#### 4.1.3. Mapless Navigation

These navigation systems merely distinguish items in the environment or follow those objects by creating movements based on visual observations, as opposed to using any explicit representation of the space in which navigation is to take place. The issue of a mobile robot traveling in an interior environment that is unknown or just partially understood was raised by [88]. Based on the integration of fundamental behaviors, a navigation strategy for uncharted territory was devised. The bulk of these qualities are achieved via fuzzy inference methods. Two separate methods of avoiding obstacles are included in the recommended navigator: one for convex obstacles and the other for concave ones. To develop basic behaviors like “reaching the middle of the collision-free space” and “wall-following”, zero-order Takagi–Sugeno fuzzy inference systems are used. This method is fairly straightforward and intuitive, but one can never eliminate the possibility that the rules derived from straightforward human expertise are somewhat inadequate. We attempted to obtain these criteria automatically for this reason. It is possible to optimize the parameters of a fuzzy inference system by minimizing a cost function using a method based on a backpropagation-like algorithm. It derive set of rules from the experimental data without using any empirical methods, and it is especially crucial to keep in mind this last aspect. In the case of a partially known environment, a hybrid technique is used to benefit from both global and local navigation algorithms. A fuzzy inference method that compares the actual situation to one that has been previously remembered online is used to coordinate these tactics. To design the journey, an algorithm and visibility graph are employed. Fuzzy controllers are made available for both the virtual robot’s planned path following in the hypothetical environment and the actual robot’s navigation when the real environment is local compared to the recalled one. Both methods are employed with the help of the tiny mobile robot Khepera, which possesses ultrasonic sensors. To correctly maneuver a tiny, four-wheeled, indoor mobile robot in real time without prior knowledge of the surroundings, Jazayeriet al. [68] described the implementation of an intelligent system. Finding the robot’s optimum route to its objective was achieved using a recurrent neural network. Using a laser range finder scenario and a modified dead reckoning method, a precise grid-based map was produced. The presentation of a motion control technique came last. The mobile robot Resquake used these strategies, and they were tested. Intelligent autonomous mobile robots must be able to perceive and recognize 3D inside spaces where they reside or operate, according to a study by [89]. The presentation by [90] stated that intelligent autonomous mobile robots must be able to observe and comprehend the 3D interior environment where they live or work. Robotic systems are increasingly being used in crowded environments, where accurately differentiating between multiple objects can be a significant challenge. While monocular and binocular vision sensors are commonly used in mobile robotics, they face issues related to visual intensity variations, a lack of

feature information, and correspondence problems. To address these challenges, a group of researchers developed a unique 3D sensing system using laser-structured lighting. This approach is preferred due to its robustness in navigating complex environments and its simplicity in extracting feature information. The proposed active trinocular vision system is composed of a flexible multistripe laser projector and two cameras. The system uses trinocular epipolar constraints to represent the laser projector as a virtual camera and establish matching pairs of line characteristics between the two actual camera images. This allows for the extraction of 3D information from a single image of the patterned scene. To ensure robust feature matching, the researchers proposed a novel correspondence-matching approach based on line grouping and probabilistic voting. The proposed sensor system was tested in several experiments to demonstrate its simplicity, efficacy, and precision in 3D environment sensing and recognition. Overall, the new 3D sensing system provides a promising solution to the challenges posed by crowded environments and can potentially improve the performance of robotic systems in these scenarios.

#### 4.2. Outdoor Navigation

Outdoor navigation frequently includes features like obstacle avoidance, landmark recognition, map construction and updating, and location estimation. But, at least in the research on outdoor navigation that has so far been published, seldom is the complete map of an environment known in advance; thus, the system must deal with the objects as they appear in the image without knowing where they belong. Based on how organized the environment is, the two types of outdoor navigation—outdoor navigation in structured surroundings and outdoor navigation in unstructured situations—can still be separated. Mobile robots' outdoor navigation generally involves course planning, global localization, and obstacle avoidance. The objective is to locate the robot in a global coordinate system and plot a path to a destination while avoiding environmental hazards. The method of pinpointing the location of a robot using sensor measurements is known as global localization. Numerous methods, including GPS, visual odometry, and laser-based localization, can be used to accomplish this. Once the robot's location is established, a path to the desired place may be planned using path-planning algorithms.

Path-planning algorithms take into account the robot's kinematic and dynamic constraints, as well as the geometry of the environment, to generate a path that is both feasible and optimal. The A\* search, Dijkstra's algorithm, and rapidly-exploring random trees (RRT) are a few popular path-planning techniques. Once a path has been created, the robot is guided around environmental obstacles using obstacle avoidance algorithms. This can be achieved using sensor-based techniques, such as sonar, LiDAR, or vision, to detect obstacles and generate control signals that avoid collisions. Outdoor navigation also involves dealing with uncertainties in the environment, such as changes in lighting conditions or moving obstacles. To address these uncertainties, some outdoor navigation systems use adaptive algorithms that adjust the robot's behavior in real-time based on sensor feedback.

##### 4.2.1. Outdoor Navigation in Structured Environments

One of the first outdoor navigation systems for a car that could drive itself at 30 km/h was reported [91] in a somewhat constrained set of circumstances. This system used two vertically positioned stereo cameras to detect potential impediments. The main navigational tactic was avoiding obstacles. In organized environments, road-following is typically required for outdoor navigation. Being able to discern between the lines that separate the lanes of the road from the berm, the texture of the road surface, and the surrounding surfaces, among other things, is necessary for being able to follow a road. Systems that follow roads often utilize simple environment models that just include information on vanishing points, lane lengths, and road widths.

#### 4.2.2. Outdoor Navigation in Unstructured Environments

An outdoor area without any predictable characteristics that may be observed and used for navigation is referred to as an unstructured environment. The vision system can only use a broad description of the probable environmental obstructions under these conditions [92]. Outdoor navigation in unstructured environments can be a challenging task for mobile robots. It involves navigating through an environment that is unknown, unpredictable, and contains various obstacles such as rocks, bushes, and trees. To accomplish this task, the robot needs to perform simultaneous localization and mapping (SLAM), plan a path to a goal location, and avoid obstacles along the way.

Building an environment map and simultaneously locating the robot inside the map are both steps in the SLAM process. This can be accomplished by creating a 3D or 2D map of the surroundings using sensor data from devices like LiDAR, sonar, or cameras. Using methods like the extended Kalman filter (EKF) or the particle filter, the position of the robot on the map is approximated. Path-planning algorithms can be used to plot a path to a destination after an environment map has been created and the robot's position has been calculated. These algorithms provide a viable and ideal path by taking into consideration the environment's barriers as well as the robot's kinematic and dynamic restrictions.

The A\* search method, Dijkstra's algorithm, and rapidly-exploring random trees (RRT) are a few examples of path-planning algorithms.

Obstacle avoidance is an essential component of outdoor navigation in unstructured environments. This involves detecting and avoiding obstacles along the planned path using sensors such as LiDAR or sonar. The robot's control system generates steering commands that enable it to avoid obstacles while following the planned path. The control system can be based on a variety of algorithms, such as potential fields, dynamic window approach, or artificial potential fields. To handle uncertainties in the environment, adaptive algorithms can be used to adjust the robot's behavior in real-time based on sensor feedback. This enables the robot to respond to changes in the environment such as moving obstacles or changes in lighting conditions.

### 5. Various Navigational Approach of WMR

In recent years, the development of intelligent controllers for the route and motion planning of WMRs has increasingly used artificial intelligence (AI) approaches. Due to their capacity to learn from data and generalize, artificial neural networks (ANNs) have become one of the approaches that have gained the most popularity. Biological neural networks in the human brain serve as the basis for the construction and operation of ANNs, which are computer models. To improve their performance on a particular job, they may be taught using a variety of learning techniques, including backpropagation. The ANN typically receives sensor data as input and outputs the control signal for the actuators of the robot.

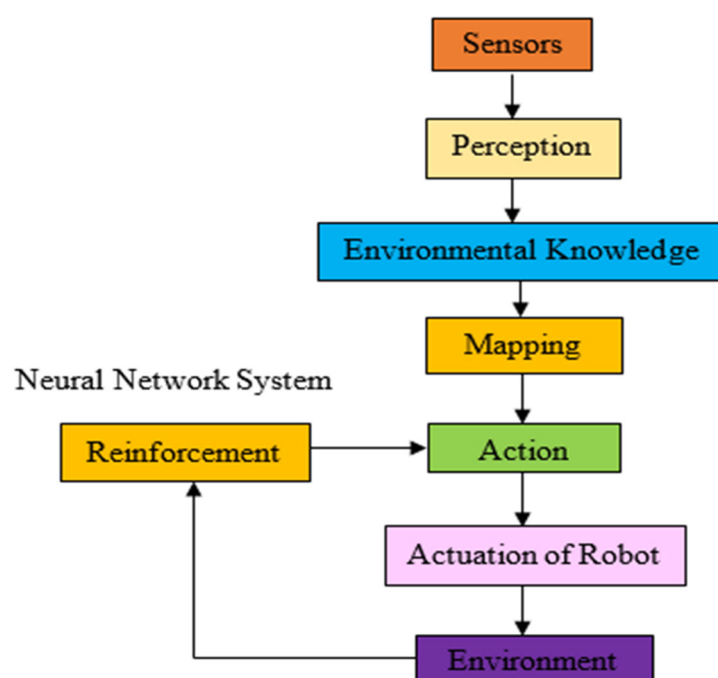
Genetic algorithms (GAs) are a type of optimization algorithm inspired by the process of natural selection. Global route planning and obstacle avoidance in mobile robot research may be accomplished using GAs. A population of potential solutions is created in a GA, and each one is assessed according to its fitness function. An acceptable solution is eventually identified by selecting the options with the best fitness for reproduction. GAs are effective in handling complex, nonlinear problems that cannot be solved analytically.

Fuzzy logic is a method of reasoning that allows for imprecise or uncertain inputs. In mobile robot research, fuzzy logic has been used for obstacle avoidance, localization, and mapping. Fuzzy logic controllers (FLCs) are used to transform sensor data into control signals for the robot's actuators. FLCs use fuzzy sets to represent uncertainty and fuzzy rules to make decisions based on these sets. Fuzzy logic is particularly useful in environments where there is a high degree of uncertainty or where traditional control methods are difficult to implement.

### 5.1. Artificial Neural Network (NN) Technique

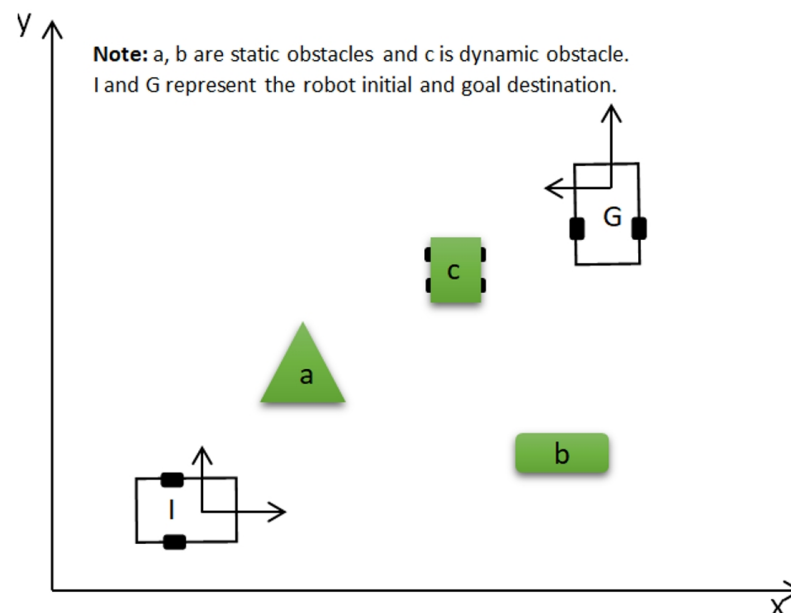
The brain is a highly developed parallel, nonlinear computer. The billions of neurons that make up the brain are connected by trillions of connections. Interest in neural networks is fueled by the quest to understand ideas that, in some way, reveal the basic principles of human brain function as well as the need to build robots that can perform challenging tasks. The fundamental tenet of neural network theory is that significant properties of real neurons may be extracted and used to simulate a genuine brain. Autonomous mobile robots, often known as machines with strong autonomous navigation capabilities, are now the focus of a lot of research. Artificial neural networks (ANNs) are a relatively recent advancement in computer technology that has been effectively used to solve a variety of challenging real-world problems [85,86]. Researchers have enthusiastically embraced recent developments in neural network technology for the route and motion planning of WMR. Below is a description these developments.

For robots to effectively navigate their environment, they must possess the ability to understand and interpret the geography of their surroundings. Neural networks gained attention in 1943 when simpler neurons were discovered by McCulloch and Pitts, and Hebbian learning was introduced. As digital computers were developed during this time, neural network research progressed and became more practical. Rosenblatt's construction of neuron models for hardware in 1957 led to the perceptron concept, which is still widely used today. However, funding for neural network research decreased when Minsky and Papert's book *Perceptron*, which exposed the shortcomings of perceptron models and encouraged researchers to leave the area, was released [87]. The controller for the dynamic model of a nonholonomic wheeled robot was developed by [88], and they also suggested an adaptive extension of the controller for fusion of the kinematic controller and a torque controller [89,90]. Even with unknown parameters, an adaptive tracking controller was used for both the kinematic and dynamic models to improve the mobility of mobile robots. For robots to effectively navigate their surroundings and select optimal routes, they must possess the ability to learn [91]. The NN navigational path-planning approach for WMR involves the robot first learning to navigate, then creating a map based on data gathered from its sensors, updating the map, and utilizing it to plan and control its movements [92]. In [93] presented a general structure for mobile robot navigation using NN, as depicted in Figure 4.



**Figure 4.** Example of path definition with static as well as moving obstacle.

Figure 5 shows a multilayer perceptron neural network for the study and planning of WMR pathways. This network makes use of input, hidden, and output layers as well as nodes with feedforward connections. A novel method of route tracking for moving robots that mimic vehicles was presented [94]. Based on neural predictive control, this plan was implemented. To address the issue of nonlinear kinematics in robots, a multilayer backpropagation neural network was used to adjust it for a wide operating range. Route tracking, a type of model-based predictive control, was used to generate data on the robot's kinematics and intended trajectory. THEY used an adaptive hop-field neural network to find real-time collision-free robot motion planning between two arbitrary source and target configurations through the roadmap to solve the problem of motion planning for a wheeled mobile robot with two degrees of freedom through polygonal obstacles [95]. They represented the robot as a point using the Minkowski sum and employed conventional strategies to tackle the issue while keeping additional clearance, including visibility graphs, basic and generalized Voronoi diagrams, and decomposition methods. Simulation tests verified the benefits of the suggested system [96]. In a static context, Kala et al. introduced a unique approach for path planning, utilizing the multi neuron heuristic search technique they had previously created. In [97] proposed a neural-network-based navigation method for mobile intelligent robots. Robots must be able to train to be able to travel in environments like their fictitious maze while creating an NN route-planning method.



**Figure 5.** The general structure of NN navigational path planning.

Modern mobile robots usually have ultrasonic sensors mounted to them to monitor their present condition. The value of the findings has been confirmed by a variety of simulated studies using fuzzy [98] for the navigation of mobile robots. In [99] presented a concept for a neuro-fuzzy system that enables behavior-based control of mobile robots in an unknown environment. They used ultrasonic sensors to obtain data on the range. They combined several sorts of behavior using fuzzy logic to manage the speed of the robot's two rear wheels based on a reference motion direction and the distances between the robot and objects. Additionally, they displayed the simulation results for the suggested tactic. Figure 5 shows "a" and "b" as static obstacles. In this context, the term probably refers to an immovable obstruction that stays in one place. Walls, structures, or any other item whose location remains constant over time might be considered static impediments. Because the robot must avoid hitting them or find a way around them, they are crucial to robotics and navigation. An impediment in the environment that is in motion or has the potential to



change location over time is denoted by the letter “c” as a dynamic obstacle. Moving cars, pedestrians, or any other nonstationary item that might endanger the robot’s navigation are examples of dynamic impediments. Dynamic barriers are more difficult for the robot to navigate than static ones because it has to continuously determine their locations.

### 5.2. Genetic Algorithm Technique

The genetic algorithm (GA) has been used by many researchers for the path and motion planning of wheeled mobile robots. Most of the work carried out by them using GA in recent years for path and motion planning of mobile robots is described below.

The ideal route must be found by the WMR to minimize the number of steps required to move from the starting place to the desired finishing position in [100] solved the problem of multiple robot motion and path planning by proposing a method that takes into account the diverse sources and goals of each robot. Each robot creates a perfect path using genetic programming based on grammar. They enable the multiple robots’ evolutionary algorithms to work together, assuring the creation of all-around optimal pathways. The tests made use of several robots and used the wall to examine robot behavior. Using a genetic algorithm, in [101] developed an energy-efficient path-planning program for wheeled robot systems (GA). Wheeled robots may move smoothly over effective collision-free paths in known static situations, as per path planning algorithms [102]. In [103] were the first to present the idea of using genetic algorithms to help a steerable wheeled robot choose the optimal path between two spots in a grid environment with moving obstacles. Their study has given direction for creating simulation software that would provide the optimal WMR path by utilizing GA techniques. By allowing four-neighbor movements, the suggested controlling method makes it simple for path planning to adapt to challenging search areas. The genetic algorithm (GA) was used to construct a simulation program for route analysis and planning by [104], who offered a method for mobile robot path planning in the face of dynamic obstacles. They achieved this by using the mutation operator for the genetic algorithm. The GA optimization capabilities were used to produce an ideal path and memory-based lookup was employed for local optimization to update the most efficient paths between different map intersections. In [105] presented a modified ant system (AS) algorithm as a genetic algorithm methodology for real-time global optimal path planning of wheeled mobile robots. The proposed approach involves using the MAKLINK graph theory to create the free space model of the robot, the Dijkstra algorithm to find a suboptimal collision-free path, and the modified AS algorithm to optimize the location of the suboptimal path to generate the globally optimal path. In [106] discussed a design strategy and structure for hardening soft computing, based on a qualitative computer model of a mechanical robotic system’s dynamics using physics and a genetic algorithm search for potential solutions (GA). They also developed a direct human–robot communication method using cognitive graphics and human language (NL). In mobile robotics, active adaptation building blocks like fuzzy neural networks, fuzzy control, and genetic algorithms are crucial to enable robots to acquire new behaviors and scripts.

### 5.3. Fuzzy Logic

The development of an intelligent controller for a mobile robot relies heavily on fuzzy logic. This method might be used to plan the movements and course of mobile robots. For describing and controlling uncertainty in terms of ambiguities, imprecision, informational gaps, and partial truth, fuzzy set theory offers a mathematical framework. Fuzzy control systems use approximate reasoning to make decisions, much like people do. The list below compiles the route and motion planning problems with mobile robots that have been handled by various authors utilizing this approach.

The route-planning issue in robotic research was addressed by [107] by proposing a hierarchical fuzzy control system using the fuzzy logic approach. Each robot in the system has to be autonomous and have network connectivity for communication. Acceleration, speed, and degree of danger are the three primary judgmental elements that affect path

choice. The wheeled robot's movement is controlled by a hierarchical fuzzy control system. The effectiveness and viability of their technology for multirobot route planning were demonstrated in simulation experiments. In [108] study on intelligent path planning for various mobile robots explored this topic. Rule-based and rule-based neuro-fuzzy algorithms were studied for the navigation of several mobile robots in an environment that is unknown or only partially understood. They also developed a route-planning and motion control system for wheeled robots. An original ant colony optimization meta-heuristic (SACO-MH) was proposed by [109] to address the path-planning issue for wheeled robots. An autonomous mobile robot's controllers frequently employ fuzzy logic to address issues like path and motion planning, navigation, and many more. To plan and steer the path of an autonomous mobile robot, in [110] offered an automated controller that makes use of fuzzy logic and simulated annealing. It was used to simulate the annealing process. C-space was used to represent the working space, while b-spline curves were used to represent the trajectories. The main objective of the project was to create a control algorithm that would allow the vehicle to follow the specified course and avoid colliding with moving objects. A fuzzy logic technique was utilized to address the route and motion planning problems in mobile robotics. The simulation was tested on 200 mobile robots by [111], and simulation and implementation results were shown.

In [112] adopted the fuzzy logic approach of an autonomous mobile manipulator capable of scanning natural terrain using a detector, such as a landmine detector, to tackle the problem of real-time mobility and path planning. A topographical map was made using information from laser and ultrasonic rangefinders, and the map was then utilized to design the detector's course without any obstacles in its way. They proposed a fuzzy adaptive Kalman-filter that adjusts the gain following a fuzzy model of the terrain technique, and experiments on actual robots have validated their claims. Because they offer a framework for developing and implementing a wheeled robot control programmer that is easily expandable and transferable to other robotic platforms, the methods of [113] for designing an autonomous vehicle as the test bed for the upcoming development of an intelligent wheelchair are very helpful. Using a hierarchical structure and a variety of intelligence-related skills, in [114] created a method for a fuzzy-based intelligent WMR system. Figure 6 shows the whole fuzzy control structure for the WMR's route and motion planning, which includes all elements necessary to maintain the proper relative robot posture ( $u_1$ ,  $u_2$ , and  $d_e$ ).

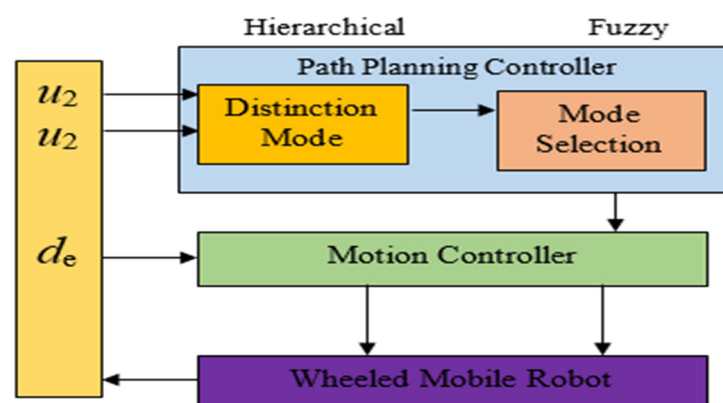


Figure 6. Fuzzy control structure.

Existing research makes significant advances in path planning and autonomous robots. A thorough analysis of autonomous path-planning algorithms for mobile robots was provided by who also provided information on several strategies, their benefits, and their drawbacks [115]. A context-aware attention-based network for informative path planning (CATNIPP), which includes contextual information to improve the effectiveness and efficiency of robotic path planning, was introduced by Cao et al. in 2023. The particle

swarm optimization approach was suggested as a trajectory planning technique for a six-axis robotic arm. With the help of differential evolution and fast marching square algorithms, they concentrated on geometrically constrained path planning for robotic grasping. An enhanced simulated annealing approach for dynamic path planning of mobile devices was presented by [116]. A summary of path-planning methods for numerous mobile robots was given. To plan robotic paths, GUL et al. (2022) suggested using a hybrid bioinspired algorithm that includes a mutation operator. In 3D arbitrary path planning for a spherical underwater robot, it can be applied to the avoidance of uncertain moving obstacles. Using an updated RRT (rapidly-exploring random tree) algorithm, described a path-planning method for robotic manipulators in complicated environments with multiple obstacles [117]. Together, the aforementioned articles advance the topic of path planning in robotics by proposing novel ideas, investigating fresh possibilities, and tackling difficulties posed by various robotic settings and systems [118].

#### 5.4. Probabilistic Roadmap (PRM) Planning

- I Sample configurations: The PRM algorithm starts by randomly sampling a set of configurations from the robot's configuration space. Let  $Q$  be the configuration space, and  $q_{\text{sample}}$  be a randomly generated sample from  $Q$ , where  $q_{\text{sample}} \in Q$ .
- II Validate configurations: Each sample is then checked to see if it is a valid configuration that the robot can reach without colliding with obstacles. This is achieved by performing collision detection using the robot's sensor data or environment maps. Let  $C(q_{\text{sample}})$  be the collision detection function that checks if  $q_{\text{sample}}$  is in the collision, where  $C(q_{\text{sample}}) = \text{True}$  if  $q_{\text{sample}}$  is in a collision and  $\text{False}$  otherwise.
- III Connect valid configurations: Valid samples are then connected with edges in the roadmap to create a graph. Two configurations  $q_1$  and  $q_2$  are connected if there exists a valid path between them. The edge weight between  $q_1$  and  $q_2$  is typically set to the Euclidean distance between the two configurations. Let  $E$  be the set of edges between connected configurations, where  $e = (q_1, q_2) \in E$  if  $C(q_1) = \text{False}$ ,  $C(q_2) = \text{False}$ , and there exists a valid path between  $q_1$  and  $q_2$ .
- IV Find a path: The next step is to utilize a search technique, such as Dijkstra's algorithm, to identify a path connecting the start and goal configurations in the graph. Assume that the graph created in step 3 is  $G = (V, E)$ , with  $V$  being the collection of configurations. Let the start and target configurations be denoted by  $s$  and  $g$ , respectively. The Dijkstra algorithm determines the shortest route in  $G$  between  $s$  and  $g$ .
- V Smooth the path: The final path can be smoothed to remove unnecessary turns and improve efficiency. This is typically achieved using path smoothing algorithms, such as the spline-based or polynomial-based methods.
- VI The PRM algorithm is probabilistic because it relies on random sampling to generate the roadmap and does not guarantee the optimal path. However, it is often effective and scalable, particularly in high-dimensional configuration spaces where other algorithms may struggle.

Consider a scenario where we need to design a path for a mobile robot to travel along in a 2D environment that has obstacles. To determine the shortest route between the start and target sites, we may utilize the A\* search method. The A\* search algorithm is a heuristic search method that directs the search toward the desired location using an evaluation function. The definition of the evaluation function  $f(n)$  is

$$f(n) = g(n) + h(n) \quad (13)$$

where  $g(n)$  is the cost of the path from the start location to node  $n$ , and  $h(n)$  is the estimated cost of the path from node  $n$  to the goal location [119].

To apply the A\* search algorithm to our problem, we can define the following steps:

- Create a graph: Create a graph where each node represents a location in the 2D environment. The edges between nodes represent possible movements from one location to another.
- Initialize start and goal nodes: Set the start location as the starting node and the goal location as the goal node.
- Calculate heuristic function: Calculate the heuristic function  $h(n)$  for each node  $n$ . This can be achieved using a distance metric, such as Euclidean distance, to estimate the distance between the node and the goal location.
- Initialize costs: Set the cost of the start node  $g(\text{start})$  to 0, and the cost of all other nodes to infinity.
- Add start node to open set: Add the start node to the open set, which is the set of nodes to be evaluated.

While the open set is not empty: Repeat the following steps until the goal node is reached or the open set is empty:

- Select the node with the lowest  $f(n)$  value from the open set.
- Stop the search and retrace the path if the chosen node is the goal node.
- Otherwise, for each neighbor of the selected node:
  - Calculate the tentative cost of the path from the start node to the neighbor node,  $g'(\text{neighbor}) = g(\text{selected}) + \text{cost}(\text{selected}, \text{neighbor})$ .
  - If  $g'(\text{neighbor})$  is less than the current cost of the neighbor node, update the cost of the neighbor node to  $g'(\text{neighbor})$  and set its parent to the selected node.
  - If the neighbor node is not already in the open set, include it and determine the  $f(n)$  value. If the goal node is not reached, return failure. Once the A\* search algorithm has found a path, it can be represented as a sequence of nodes that correspond to locations in the 2D environment. The robot can then move along this path by following the sequence of locations. If obstacles are detected during movement, the algorithm can be rerun to find a new path around the obstacle.

##### 5.5. Rapidly-Exploring Random Tree (RRT) Planning

The rapidly-exploring random tree (RRT) is a randomized motion planning algorithm that has become increasingly popular for solving motion planning problems for robots in complex and high-dimensional spaces. The algorithm incrementally builds a tree of feasible paths in the configuration space of the robot, which can be seen as a high-dimensional space representing all possible configurations of the robot. To solve a specific motion planning problem, such as moving a mobile robot in a cluttered environment from an initial configuration to a goal configuration without colliding with any obstacles, the RRT algorithm samples random configurations in the configuration space and connects them to the existing tree, growing it in a random and exploratory manner. By repeating this process, the algorithm explores the space to find a path from the start to the goal configuration. The RRT algorithm has several advantages over other motion planning algorithms, such as its ability to handle high-dimensional spaces, its probabilistic completeness, and its relatively fast convergence to a feasible solution. Additionally, the algorithm can handle various types of constraints and objectives, making it a versatile tool for motion planning problems. The RRT algorithm is a powerful and flexible tool for solving motion planning problems for robots in complex and cluttered environments. Its random and exploratory nature allows it to efficiently search the configuration space for feasible paths, making it a valuable tool for robotics researchers and practitioners [119].

The Algorithm 1 works as follows:

**Algorithm 1:** Rapidly-exploring random tree (RRT) planning.

**Step 1:** Initialize the tree with the start configuration as the root node.

**Step 2:** Repeat until the goal configuration is reached or a maximum number of nodes are added:

- a. Sample a random configuration in the space.
- b. Find the nearest node in the tree to the sampled configuration.
- c. Generate a new node by extending the nearest node towards the sampled configuration, while ensuring that the robot does not collide with any obstacles.
- d. Add the new node to the tree and connect it to its nearest neighbor.

**Step 3:** If the goal configuration is reached, return the path from the start to the goal configuration by backtracking through the tree from the goal node to the root node.

**Step 4:** The RRT algorithm is simple and efficient and can handle high-dimensional and complex spaces. It can also deal with moving obstacles by updating the tree in realtime.

Algorithm 1's probabilistic nature also allows it to explore the space efficiently and avoid becoming stuck in local minima.

Mathematically, the RRT algorithm can be represented in Algorithm 2:

**Algorithm 2:** Pseudocode for navigation planning.

S: a set of all valid configurations in the space.

q\_start: the start configuration of the robot.

q\_goal: the goal configuration of the robot.

K: maximum number of nodes to add to the tree.

T: tree of configurations, initially containing only q\_start.

r: maximum distance to extend the tree towards a new configuration.

$T \leftarrow \{q_{start}\}$

for k = 1 to K do

- a.  $q_{rand} \leftarrow \text{RandomConfiguration}()$
- b.  $q_{near} \leftarrow \text{NearestNeighbor}(q_{rand}, T)$
- c.  $q_{new} \leftarrow \text{Extend}(q_{near}, q_{rand}, r)$
- d. if  $\text{CollisionFree}(q_{near}, q_{new})$  then  
 $T \leftarrow T \cup \{q_{new}, (q_{near}, q_{new})\}$
- e. if  $\text{Distance}(q_{new}, q_{goal}) \leq \varepsilon$  then

return Path(T, q\_start, q\_new)

return Failure

Where

- RandomConfiguration() returns a random configuration in S.
- NearestNeighbor( $q_{rand}, T$ ) returns the configuration in T that is closest to  $q_{rand}$ .
- Extend( $q_{near}, q_{rand}, r$ ) returns a new configuration that is a distance r away from  $q_{near}$  towards  $q_{rand}$ .
- CollisionFree( $q_{near}, q_{new}$ ) returns true if the path from  $q_{near}$  to  $q_{new}$  is collision-free.
- Distance( $q_{new}, q_{goal}$ ) returns the distance between  $q_{new}$  and  $q_{goal}$ .
- $\varepsilon$  is a small value that determines the proximity of  $q_{new}$  to  $q_{goal}$ .
- Path(T, q\_start, q\_new) returns the path from  $q_{start}$  to  $q_{new}$  by backtracking through the tree T.

**Model Predictive Control (MPC)**

Model predictive control (MPC) is a control strategy that utilizes a mathematical model of the system to predict the system's future behavior over a finite horizon. MPC is widely used in many applications, including robotics, autonomous vehicles, and process control.



Here, we explain the MPC algorithm in general and then provide an example problem with relevant assumptions and equations [120].

**a. Assumptions:**

The model of the system is a linear time-invariant (LTI), discrete-time system that may be explained using state-space equations:

$$x_{\{k+1\}} = Ax_k + Bu_k + w_k \quad (14)$$

$$y_k = Cx_k + v_k \quad (15)$$

where  $x_k$  is the state vector at time step  $k$ ;  $u_k$  is the control input at time step  $k$ ;  $y_k$  is the output at time step  $k$ ;  $A$ ,  $B$ , and  $C$  are system matrices;  $w_k$  is the process noise; and  $v_k$  is the measurement noise [121]. The objective is to minimize a quadratic cost function over a finite horizon  $N$ :

$$J(u_{\{0:N-1\}}) = \sum_{\{k=0\}}^{\{N-1\}} (x_k^T Q x_k + u_k^T R u_k + 2x_k^T S u_k) \quad (16)$$

where  $Q$ ,  $R$ , and  $S$  are cost matrices.

The constraints on the system inputs and states are linear and can be written as

$$u_{\{min\}} \leq u_k \leq u_{\{max\}} \text{ in } x_k \leq x_{\{max\}} \quad (17)$$

**b. Algorithm:**

At each time step  $k$ , the current state  $x_k$  is measured, and a prediction of the future states  $x_{\{k+1:N\}}$  is made using the system model and the current control input  $u_k$ . A cost function  $J(u_{\{k:N-1\}})$  is formulated based on the predicted states and the desired objective.

- The control input  $u_k$  that minimizes the cost function subject to the system constraints is computed using an optimization solver.
- The first element of the optimal control input sequence  $u_{\{k:k+N-1\}}$  is applied to the system as the control input, and the process is repeated at the next time step.

Consider a simple inverted pendulum system that is controlled by applying torque to the pivot point. The objective is to stabilize the pendulum at the upright position while minimizing the control effort. The system is modeled as a second-order LTI system, and the MPC algorithm [122] is used to generate the control input sequence.

**c. Assumptions:**

The system is modeled as a second-order LTI system, which can be described by the state-space equations

$$x_{\{x+1\}} = Ax_x + Bu_x + w_x \quad (18)$$

$$y_x = Cx_y + v_y \quad (19)$$

where  $x_k = [\theta_k, \dot{\theta}_k]^T$  is the state vector reciprocal to it at time step  $k$ ; the  $\theta_k$  is the angle of the pendulum from the upright position;  $\dot{\theta}_k$  is the angular velocity of the pendulum;  $u_x$  is the control input at time step  $k$ ;  $y_x$  is the output at time step  $x$ ;  $A$ ,  $B$ , and  $C$  are system matrices;  $w_k$  is the process noise; and  $v_y$  is the measurement noise. The objective is to stabilize the pendulum at the upright position while minimizing the control effort, which can be achieved by minimizing the cost function over a finite horizon  $N$  [123].

### 5.6. SLAM (Simultaneous Localization and Mapping)

A robotics approach known as simultaneous localization and mapping (SLAM) is used to map an uncharted area while also localizing the robot within it. To create a map of the environment and estimate the location and orientation of the robot inside it, data

from sensors like cameras, LiDAR, and wheel encoders must be integrated. SLAM has several practical uses, including autonomous vehicles, drones, and robot navigation. It entails minimizing the error between the expected and real measurements collected from the sensors by solving a challenging optimization problem [124].

The problem can be formulated mathematically as follows: Given a set of sensor measurements  $z_1 : T$  and control inputs  $u_1 : T$ , estimate the joint probability distribution over the robot's path  $x_1 : T$  and the map of the environment  $m$ .

$$P(x_1 : T, m | z_1 : T, u_1 : T) \quad (20)$$

SLAM can be solved using various algorithms, including EKF (extended Kalman filter) [125], FastSLAM [126], and GraphSLAM [127], each with its strengths and weaknesses. SLAM can be approached using two main paradigms: filter-based and graph-based. In the filter-based approach, a recursive Bayesian filter, such as the EKF, is used to estimate the state of the robot and the map [128]. The filter estimates the posterior probability distribution over the state of the robot and the map given the measurements and controls. However, filter-based methods have limitations when it comes to large-scale environments and nonlinear sensor models. In contrast, graph-based methods use optimization techniques to find the most likely estimate of the robot's path and the map of the environment. This approach is more suitable for large-scale environments and nonlinear sensor models [129,130]. Graph-based methods [131,132] represent the problem as a factor graph, where the nodes represent the robot poses and map landmarks and the edges represent the constraints between them.

### 5.7. Comparative Analysis of Existing Work

The constraints are typically derived from the sensor measurements and the robot motion model [133]. The optimization problem is solved by minimizing the error between the predicted and actual measurements [134]. A summary of wheeled mobile robot control techniques shows the benefits and drawbacks of each technique. Although they have moderate to high processing needs, artificial neural networks provide excellent trajectory tracking and uncertainty adaptation [135]. Genetic algorithms offer resilience and global optimization, but they can also be computationally demanding and not naturally able to tolerate perturbations. Although it may encounter difficulties in complicated situations, fuzzy logic control uses linguistic variables and rules for interpretability. Although the firefly algorithm may be used for both Lyapunov stability analysis and algorithm optimization, its efficacy varies depending on the application. By integrating AI techniques, hybrid approaches offer adaptability, but careful analysis of constituent methodologies is necessary [136]. Real-time optimization and trajectory tracking are two areas where model predictive control shines, but it can be computationally demanding. Though computationally demanding at times, model predictive control is excellent at real-time optimization and trajectory monitoring [137]. Though it may not be able to manage disturbances by nature, particle swarm optimization includes global optimization and Lyapunov stability analysis. Although it requires a moderate to high amount of computer power, reinforcement learning allows for adaptive learning and disruption adaptability. Although they need little to moderate computing power, robust control, and sliding mode control are particularly made to deal with uncertainties and disruptions. According to particular application needs, these summaries help determine which methods are most appropriate [138].

Table 1 shows different methods used for the stability and navigation of mobile robots, along with their corresponding kinematic and dynamic models, stability analysis criteria, trajectory tracking ability, control design, and success rate in percentage. The first three methods listed are artificial neural networks, genetic algorithms, and fuzzy logic control. All of them use kinematic and dynamic models and stability analysis criteria such as Lyapunov stability and control Lyapunov functions (CLFs) for stability analysis. They also have high trajectory tracking ability, with success rates ranging from 73% to 89%. The fourth method listed is the firefly algorithm, which also uses kinematic and dynamic models and

stability analysis criteria such as Lyapunov stability and CLFs. It has a high trajectory tracking ability with a success rate of 76%. The control design for the firefly algorithm is based on algorithm optimization [139]. The fifth method listed is hybrid approaches, which use a combination of AI techniques for the stability and navigation of mobile robots. It has a high trajectory tracking ability with a success rate of 92%. The control design for hybrid approaches is based on the combination of different AI techniques in Figure 7. Control and optimization techniques differ in how long they take to calculate. Inference and training of artificial neural networks (ANNs) take time, especially for big datasets and sophisticated structures. Due to variables like population size and issue complexity, genetic algorithms (GAs) are typically computationally costly. Because it uses rule-based reasoning, fuzzy logic control is often quick. Computation times for the firefly algorithm are influenced by the convergence rates of individual problems. Depending on the mix of techniques and the particular issue, hybrid approaches cover a broad range of computing durations [140]. Especially for real-time applications, model predictive control (MPC) is frequently somewhat time-consuming. GAs take longer than PSO, which is dependent on the amount of particles and iterations. The learning method and the complexity of the algorithm have an impact on the computing time for reinforcement learning. Because they use control laws rather than iterative optimization procedures, robust control and sliding mode control often have quicker calculation times; nonetheless, the exact implementation and system dynamics affect the computation time. Comparative analysis is used to assess the features and efficacy of several control and optimization techniques in Table 1 [141]. All the methods are evaluated according to their capacity to simulate kinematics and dynamics, the stability analysis standards they use, their ability to follow trajectories, their control design approaches, their computing time, how they handle modeling uncertainties, and how well they adjust to environmental perturbations [142]. How well each strategy has been used in real-world situations is indicated by the success rate %. Researchers and practitioners may select the best technique based on their unique application needs by using this table, which offers a succinct summary of the advantages and disadvantages of each option [143].

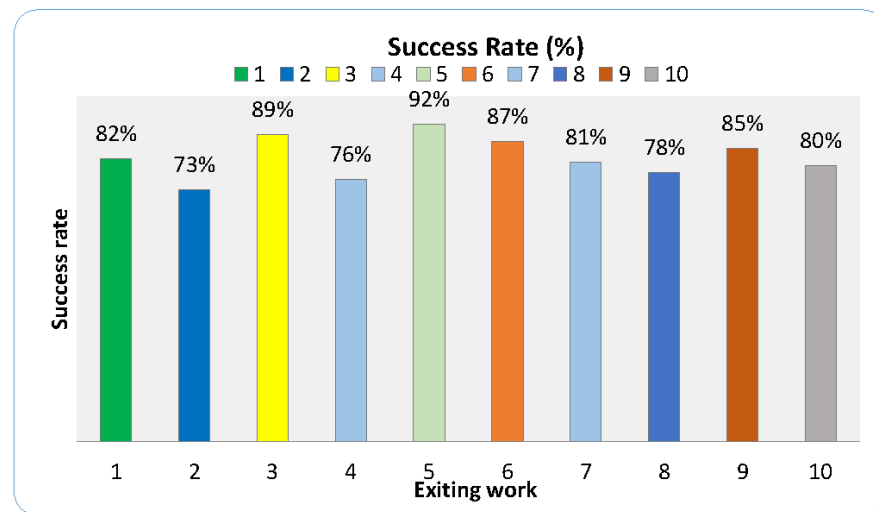
**Table 1.** Comparative analysis of method used for stability analysis and navigation.

Method	Kinematic and Dynamic Models	Stability Analysis Criteria	Trajectory Tracking	Control Design	Computational Time	Modeling Uncertainties	Environmental Disturbances	Success Rate (%)
Artificial neural networks	Yes	Lyapunov stability, CLFs	High	Design based on neural network training	Moderate to high	Can handle some uncertainties, depending on training	Can adapt to some disturbances if in training data	82%
Genetic algorithms	Yes	Lyapunov stability, passivity-based stability	Moderate	Design based on genetic algorithm optimization	High	Can address uncertainties through the optimization process	May not inherently handle disturbances	73%
Fuzzy logic control	Yes	Lyapunov stability, CLFs	High	Design based on fuzzy rules and membership functions	Low to moderate	Designed to handle uncertainties with linguistic variables and rules	Can adapt to some disturbances with appropriate rule sets	89%
Firefly algorithm	Yes	Lyapunov stability, CLFs	High	Design based on algorithm optimization	Varies (moderately high)	Not inherently designed to handle uncertainties	Effectiveness may vary depending on the problem and adaptation mechanisms	76%

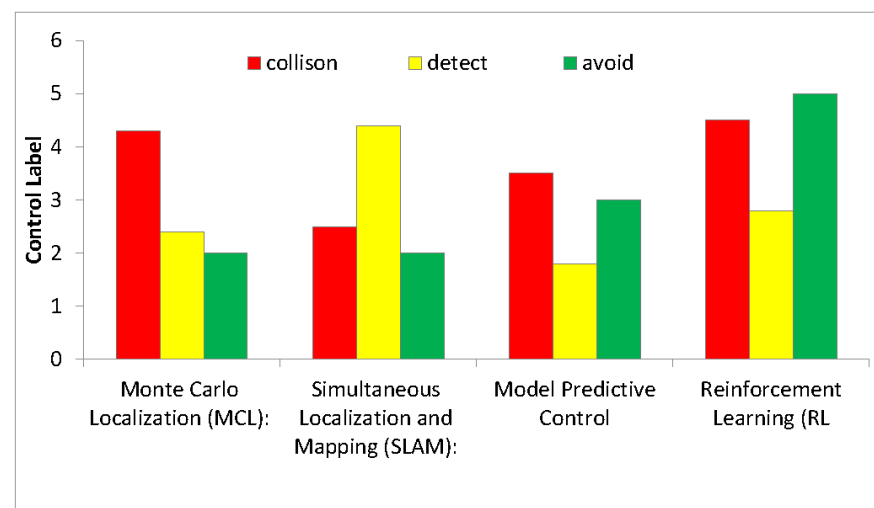
Table 1. Cont.

Method	Kinematic and Dynamic Models	Stability Analysis Criteria	Trajectory Tracking	Control Design	Computational Time	Modeling Uncertainties	Environmental Disturbances	Success Rate (%)
Hybrid approaches	Yes	Lyapunov stability, CLFs	High	Design based on a combination of AI techniques	Varies (Depends on constituent methods)	Depends on the individual methods in the hybrid approach	Depends on the individual methods in the hybrid approach	92%
Model predictive control	Yes	Lyapunov stability, CLFs	High	Design based on optimal control strategies	Moderate to high	Can address uncertainties through the optimization process	Can adapt to disturbances through real-time optimization	87%
Particle swarm optimization	Yes	Lyapunov stability, CLFs	High	Design based on particle swarm optimization	Moderate to high	Can address uncertainties through an optimization process	May not inherently handle disturbances	81%
Reinforcement learning	Yes	Lyapunov stability, CLFs	High	Design based on reward function optimization	Moderate to high	Can adapt to some uncertainties with robust training	Can adapt to disturbances through exploration and learning	78%
Robust control	Yes	Lyapunov stability, CLFs	High	Design based on robust control theory	Low to moderate	Specifically designed for handling uncertainties with robust control techniques	Can handle disturbances with designed robustness	85%
Sliding mode control	Yes	Lyapunov stability, CLFs	High	Design based on sliding mode control theory	Low to moderate	Specifically designed for handling uncertainties with sliding mode control techniques	Can handle disturbances with designed robustness	80%

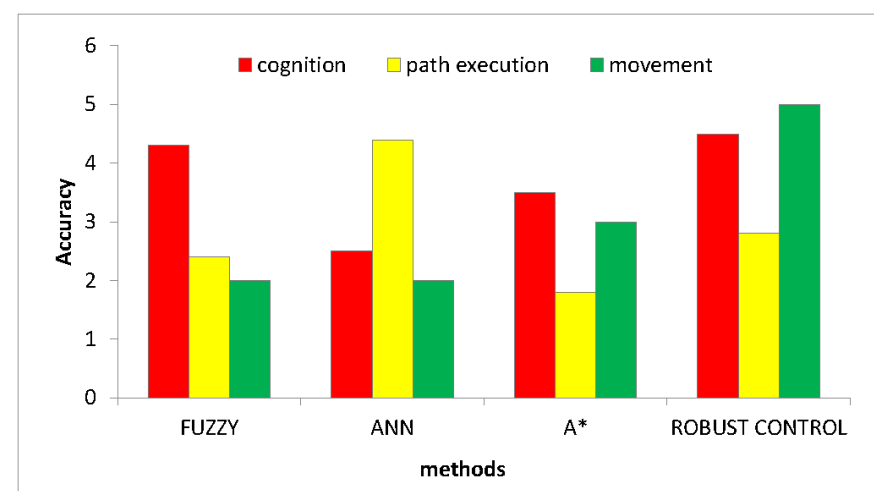
The selection of control technique has an impact on the positioning accuracy of mobile robots with wheels. While genetic algorithms (GAs) optimize pathways to increase accuracy, artificial neural networks (ANNs) may attain high accuracy when trained properly. Model predictive control (MPC) is excellent at real-time optimization, whereas fuzzy logic control uses linguistic variables and rules to achieve consistent correctness. Over time, reinforcement learning adjusts to attain accuracy, while particle swarm optimization (PSO) indirectly affects accuracy through route optimization. Accurate placement is indirectly aided by robust control and sliding mode control, which work to preserve stability. Accurate placement ultimately depends on a mix of ambient factors, sensor quality, and management strategies [143].



(a)



(b)



(c)

**Figure 7.** Comparative analysis of methods used in navigation and stability analysis. (a) Comparative analysis of success rate of existing work. (b) Comparative analysis of the control method. (c) Comparative analysis of methods.



The comparative analysis of methods employed in navigation and stability analysis within the field of robotics is a critical endeavor that underpins the development and deployment of mobile robotic systems in a variety of contexts. This analytical exploration involves a meticulous assessment of diverse approaches, algorithms, and techniques aimed at achieving proficient navigation, efficient obstacle avoidance, and robust stability for these robotic platforms. In the realm of navigation methods, two prominent categories, filter-based and graph-based approaches, come to the forefront. Filter-based methods, which often rely on recursive Bayesian filtering, empower robots to iteratively estimate their state using sensor measurements and probabilistic models. A quintessential example of this is the simultaneous localization and mapping (SLAM) technique, wherein robots continuously refine their perceived pose and environment map. Filter-based methodologies offer real-time adaptability, proving particularly effective for tasks like localization and mapping. Conversely, graph-based methods operate by constructing graph representations of the environment, wherein nodes denote robot poses and edges signify interconnections between these poses. Optimization techniques are harnessed to determine the optimal trajectory through this graph, facilitating efficient path planning. The A\* algorithm, an exemplar of this approach, exploits heuristic methods to navigate the graph and discern the shortest path. Within stability analysis, comprehensive evaluations of both kinematic and dynamic aspects are pivotal. Kinematic stability revolves around a robot's geometric properties and constraints, aiming to ensure balance solely based on these factors. This entails scrutinizing elements like wheel configurations, footprints, and tipping thresholds. Analytical methods and simulations are often applied to assess kinematic stability, identifying potential precarious zones. In contrast, dynamic stability concerns the robot's interactions with its surroundings during motion, encompassing external forces and torques. It involves an evaluation of factors such as the center of mass, mass distribution, and the robot's responsiveness to external disturbances. Techniques from control theory and simulation platforms facilitate the prediction and management of dynamic stability challenges. The core of the comparative analysis lies in evaluating the merits and drawbacks of each approach in the context of specific scenarios and robot types. Parameters considered encompass computational efficiency, accuracy, adaptability to diverse environments, resilience to uncertainties, and the adeptness to negotiate dynamic obstacles. Simulation experiments and real-world tests often serve as quantitative measures to compare performance metrics like navigation accuracy, obstacle avoidance success rate, and stability margin. Ultimately, the goal of the comparative analysis is to make informed decisions regarding the choice of navigation and stability methods that best align with the requirements of a given robotic application. Balancing accurate navigation, effective obstacle evasion, and steadfast stability is the crux of this pursuit, ultimately enhancing the capabilities and dependability of mobile robotic systems across a wide array of practical scenarios. An analysis of control strategies shows that H-infinity control provides robustness against both structured and unstructured uncertainties, while H2 control minimizes the H2 norm of the transfer function, making it appropriate for systems with disturbances modeled as stochastic noise. Because it is so good at managing parametric uncertainty, H2 control is recommended for situations where noise is regularly distributed. H-infinity control, on the other hand, offers versatility and resilience against a wider variety of uncertainty models, such as complicated uncertainty structures and non-Gaussian disturbances. H2 control is superior in some situations, but H-infinity control provides a more complete robustness framework. The decision between H2 and H-infinity control is based on the particulars of the system as well as the kind of uncertainties involved.

## 6. Conclusions

The review encompasses an extensive exploration of both kinematic and dynamic models for robots, alongside their stability analysis and control methodologies. The significance of path planning and obstacle avoidance in achieving successful navigation across diverse environments is prominently underscored. A focal point of this work is the distinc-

tion between filter-based and graph-based navigation approaches. The former relies on recursive Bayesian filtering, while the latter employs optimization techniques to estimate the robot's path and environment map. Additionally, this article examines the evolutionary trajectory of mobile robotics over the preceding three decades. Path analysis and motion planning methods, applicable to both singular and multiple mobile robots, have been subjects of significant discourse. A trend among these studies involves the application of soft computing techniques, such as artificial neural networks (ANNs), fuzzy logic control (FLC), and genetic algorithms (GAs), to confer intelligent behaviors upon mobile robot controllers. However, the utilization of nature-inspired algorithms like ant colony and firefly algorithms remains relatively limited for mobile robot navigation. Notably, the majority of research has primarily centered on kinematics analysis, with a minority delving into dynamics analysis. Most investigations have primarily addressed static obstacles, while only a handful have considered dynamic obstacles, in the context of mobile robot navigation. The theoretical frameworks advanced by researchers predominantly cater to single robots, with a smaller subset focusing on theories for multiple mobile robots. These theoretical constructs are frequently validated through simulation results, while a limited number also offer experimental validation. Recent assessments indicate the potential to enhance mobile robot path planning, particularly for intricate and unstructured scenarios, by leveraging the aforementioned techniques. The investigation into route and motion planning for mobile robots benefits from the insights presented in this review, with potential advancements envisaged in the realms of path optimization and temporal considerations, especially concerning wheeled mobile robots in unstructured environments. Convolutional neural networks (CNNs) are utilized for obstacle identification and terrain perception, and the integration of deep learning and vision-based techniques has become more popular. Particularly for uses such as autonomous driving, these techniques might completely change how wheeled mobile robots view and interact with their surroundings. With the further development of robots and artificial intelligence, the efficiency of navigation techniques will rely on the particular needs of the application and their ability to adjust to changing and unpredictable surroundings. With a focus on robustness, flexibility, and real-time decision making, the approach used should be specific to the job at hand. Subsequent investigations in this domain ought to additionally examine the amalgamation of contemporary safety methodologies and mixed-method approaches that use various ways to guarantee the dependable and effective maneuvering of wheeled mobile robots in constantly changing and demanding settings.

**Author Contributions:** Validation, K.K.B.; formal analysis, T.A.; investigation, S.K.P.; resources, T.A. and A.K.; data curation, M.K.S. and V.S.; writing—original draft, A.S. and K.U.S.; funding acquisition, T.A. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** No new data were created or analyzed in this study. Data sharing is not applicable to this article.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Wang, B.; Liu, Z.; Li, Q.; Prorok, A. Mobile robot path planning in dynamic environments through globally guided reinforcement learning. *IEEE Robot. Autom. Lett.* **2020**, *5*, 6932–6939. [\[CrossRef\]](#)
2. Qi, J.; Yang, H.; Sun, H. MOD-RRT\*: A sampling-based algorithm for robot path planning in dynamic environment. *IEEE Trans. Ind. Electron.* **2020**, *68*, 7244–7251. [\[CrossRef\]](#)
3. Poornima, E.; Muthu, B.; Agrawal, R.; Kumar, S.P.; Dhingra, M.; Asaad, R.R. Fog robotics-based intelligence transportation system using line-of-sight intelligent transportation. *Multimed. Tools Appl.* **2023**. [\[CrossRef\]](#)
4. Nair, R.S.; Supriya, P. Robotic path planning using recurrent neural networks. In Proceedings of the 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kharagpur, India, 1–3 July 2020; pp. 1–5.
5. Ma, J.; Liu, Y.; Zang, S.; Wang, L. Robot path planning based on genetic algorithm fused with continuous Bezier optimization. *Comput. Intell. Neurosci.* **2020**, *2020*, 9813040. [\[CrossRef\]](#) [\[PubMed\]](#)

6. Tang, B.; Xiang, K.; Pang, M.; Zhanxia, Z. Multi-robot path planning using an improved self-adaptive particle swarm optimization. *Int. J. Adv. Robot. Syst.* **2020**, *17*, 1729881420936154. [\[CrossRef\]](#)
7. Santos, L.C.; Santos, F.N.; Pires, E.S.; Valente, A.; Costa, P.; Magalhães, S. Path planning for ground robots in agriculture: A short review. In Proceedings of the 2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), Ponta Delgada, Portugal, 15–17 April 2020; pp. 61–66.
8. Han, S.D.; Yu, J. Ddm: Fast near-optimal multi-robot path planning using diversified-path and optimal sub-problem solution database heuristics. *IEEE Robot. Autom. Lett.* **2020**, *5*, 1350–1357. [\[CrossRef\]](#)
9. Zhang, S.; Pu, J.; Si, Y. An adaptive improved ant colony system based on population information entropy for path planning of mobile robot. *IEEE Access* **2021**, *9*, 24933–24945. [\[CrossRef\]](#)
10. Zhang, S.; Pu, J.; Si, Y.; Sun, L. Path planning for mobile robot using an enhanced ant colony optimization and path geometric optimization. *Int. J. Adv. Robot. Syst.* **2021**, *18*, 17298814211019222. [\[CrossRef\]](#)
11. Wang, H.; Chen, W. Multi-robot path planning with due times. *IEEE Robot. Autom. Lett.* **2022**, *7*, 4829–4836. [\[CrossRef\]](#)
12. Yang, X.; Xiong, N.; Xiang, Y.; Du, M.; Zhou, X.; Liu, Y. Path planning of mobile robot based on adaptive ant colony optimization. In Proceedings of the IECON 2021–47th Annual Conference of the IEEE Industrial Electronics Society, Toronto, ON, Canada, 13–16 October 2021; pp. 1–4.
13. Muhammad, A.; Ali, M.A.; Turaev, S.; Shanono, I.H.; Hujainah, F.; Zubir, M.N.M.; Faiz, M.K.; Faizal, E.R.M.; Abdulghafor, R. Novel algorithm for mobile robot path planning in constrained environment. *Comput. Mater Contin.* **2022**, *71*, 2697–2719. [\[CrossRef\]](#)
14. Zan, J.; Ku, P.; Jin, S. Research on robot path planning based on whale optimization algorithm. In Proceedings of the 2021 5th Asian Conference on Artificial Intelligence Technology (ACAIT) 2021, Haikou, China, 29–31 October 2021; pp. 500–504.
15. Shao, J. Robot Path Planning Method Based on Genetic Algorithm. *J. Phys. Conf. Ser.* **2021**, *1881*, 022046. [\[CrossRef\]](#)
16. Lin, G.; Zhu, L.; Li, J.; Zou, X.; Tang, Y. Collision-free path planning for a guava-harvesting robot based on recurrent deep reinforcement learning. *Comput. Electron. Agric.* **2021**, *188*, 106350. [\[CrossRef\]](#)
17. Li, S.; Zhao, D.; Sun, Y.; Yang, J.; Wang, S. Path planning algorithm based on the improved RRT-connect for home service robot arms. In Proceedings of the 2021 IEEE International Conference on Intelligence and Safety for Robotics (ISR), Tokoname, Japan, 4–6 March 2021; pp. 403–407.
18. Chakraborty, N.; Ghosal, A. Kinematics of wheeled mobile robots on uneven terrain. *Mech. Mach. Theory* **2004**, *39*, 1273–1287. [\[CrossRef\]](#)
19. Parhi, D.R.; Deepak, B.B.V.L. Kinematic model of three wheeled mobile robots. *J. Mech. Eng. Res.* **2011**, *3*, 307–318.
20. Deepak, B.B.V.L.; Parhi, D.R. Kinematic analysis of wheeled mobile robot. *J. Autom. Syst. Eng.* **2011**, *5*, 96–111.
21. Nawawi, S.W.; Ahmad, M.N.; Osman, J.H.S.; Husain, A.R. *Controller Design for Two-Wheels Inverted Pendulum Vehicle Using PISMIC*; Faculty of Electrical Engineering, University Technology Malaysia: Johor Bahru, Malaysia, 2008.
22. Janchiv, A.; Batsaikhan, D.; hwan Kim, G.; Lee, S.G. Complete coverage path planning for multi-robots based on. In Proceedings of the 2011 11th International Conference on Control, Automation and Systems, Gyeonggi-do, Republic of Korea, 26–29 October 2011; pp. 824–827.
23. Kim, M.-S.; Shin, J.-H.; Hong, S.-G.; Lee, J.-J. Designing a robust adaptive dynamic controller for non-holonomic mobile robots under modeling uncertainty and disturbances. *Mechatronics* **2003**, *13*, 507–519. [\[CrossRef\]](#)
24. Rovira, A.G.; Mirats Tur, J.M. Control and simulation of a tensegrity-based mobile robot. *Robot. Auton. Syst.* **2009**, *57*, 526–535. [\[CrossRef\]](#)
25. Yuan, J.; Huang, Y.; Tao, T.; Sun, F. A cooperative approach for multi-robot area exploration. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 1390–1395.
26. Liu, S.; Li, S.; Pang, L.; Hu, J.; Chen, H.; Zhang, X. Autonomous exploration and map construction of a mobile robot based on the TGHM algorithm. *Sensors* **2020**, *20*, 490. [\[CrossRef\]](#)
27. Renzaglia, A.; Doitsidis, L.; Martinelli, A.; Kosmatopoulos, E.B. Multi-robot three-dimensional coverage of unknown areas. *Int. J. Robot. Res.* **2012**, *31*, 738–752. [\[CrossRef\]](#)
28. Wang, Q.; Li, J.; Yang, L.; Yang, Z.; Li, P.; Xia, G. Distributed Multi-Mobile Robot Path Planning and Obstacle Avoidance Based on ACO–DWA in Unknown Complex Terrain. *Electronics* **2022**, *11*, 2144. [\[CrossRef\]](#)
29. Burgard, W.; Moors, M.; Stachniss, C.; Schneider, F.E. Coordinated multi-robot exploration. *IEEE Trans. Robot.* **2005**, *21*, 376–386. [\[CrossRef\]](#)
30. Bhattacharya, S.; Ghrist, R.; Kumar, V. Multi-robot coverage and exploration in non-Euclidean metric spaces. In *Algorithmic Foundations of Robotics X: Proceedings of the Tenth Workshop on the Algorithmic Foundations of Robotics*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 245–262.
31. Minguez, J.; Montano, L. Abstracting vehicle shape and kinematic constraints from obstacle avoidance methods. *Springer Sci. Auton. Robot.* **2006**, *20*, 43–59. [\[CrossRef\]](#)
32. Zhu, A.; Yang, S.X. Neuro-fuzzy based approach to mobile robot navigation in unknown environments. *IEEE Trans. Syst. Man Cybern. Part C* **2007**, *37*, 610–621. [\[CrossRef\]](#)
33. Ang, M.H., Jr.; Oetomo, E. Singularity-free joint actuation in omnidirectional mobile platforms with powered offset caster wheels. *ASME J. Mech. Des.* **2008**, *130*, 054501.

34. Puig, D.; García, M.A.; Wu, L. A new global optimization strategy for coordinated multi-robot exploration: Development and comparative evaluation. *Robot. Auton. Syst.* **2011**, *59*, 635–653. [\[CrossRef\]](#)
35. Astolfi, A. Exponential stabilization of a wheeled mobile robot via discontinuous control. *ASME* **1999**, *121*, 121–126.
36. Lei, T.; Luo, C.; Jan, G.E.; Bi, Z. Deep learning-based complete coverage path planning with re-joint and obstacle fusion paradigm. *Front. Robot. AI* **2022**, *9*, 843816. [\[CrossRef\]](#)
37. Wu, L.; PuigValls, D.; García García, M.Á. Balanced multi-robot exploration through a global optimization strategy. *J. Phys. Agents* **2010**, *4*, 35–44. [\[CrossRef\]](#)
38. Lin, W.S.; Chang, L.H.; Yang, P.C. Adaptive critic anti-slip control of wheeled autonomous robot. *IET Control Theory Appl.* **2007**, *1*, 51–57. [\[CrossRef\]](#)
39. Siegwart, R.; Nourbakhsh, I.R. *Introduction to Autonomous Mobile Robots*; A Bradford books; The MIT Press: Cambridge, MA, USA; London, UK, 2004.
40. Bar, A.; Dale, S. Dynamic modeling and stabilization of wheeled mobile robot. In Proceedings of the International Conference on Dynamical Systems and Control, La Laguna, Spain, 1–3 July 2009.
41. Aguiar, A.P.; Atassi, A.N.; Pascoal, A. Stabilization of a non-holonomic dynamic wheeled mobile robot with parametric modeling uncertainty using Lyapunov functions. In Proceedings of the 4th Portuguese Conference on Automatic Control, Guimares, Portugal, 4–6 October 2000.
42. Azizi, M.R.; Rastegarpanah, A.; Stolkin, R. Motion Planning and Control of an Omnidirectional Mobile Robot in Dynamic Environments. *Robotics* **2021**, *10*, 48. [\[CrossRef\]](#)
43. Gholipour, A.; Yazdanpanah, M.J. Dynamic tracking control of non-holonomic mobile robot with model reference adaptation for uncertain parameters. In Proceedings of the 2003 European Control Conference (ECC), Cambridge, UK, 1–4 September 2003.
44. Ibrahim, M.Y.; Fernandes, A. Study on mobile robot navigation techniques Industrial Technology. In Proceedings of the 2004 IEEE International Conference on Industrial Technology, Hammamet, Tunisia, 8–10 December 2004; Volume 1, pp. 230–236.
45. Miah, M.S.; Knoll, J. Area coverage optimization using heterogeneous robots: Algorithm and implementation. *IEEE Trans. Instrum. Meas.* **2018**, *67*, 1380–1388. [\[CrossRef\]](#)
46. Luo, C.; Yang, S.X.; Mo, H.; Li, X. Safety aware robot coverage motion planning with virtual-obstacle-based navigation. In Proceedings of the 2015 IEEE International Conference on Information and Automation, Lijiang, China, 8–10 August 2015; pp. 2110–2115.
47. Araki, B.; Strang, J.; Pohorecky, S.; Qiu, C.; Naegeli, T.; Rus, D. Multi-robot path planning for a swarm of robots that can both fly and drive. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 5575–5582.
48. Maaref, H.; Barret, C. Sensor-based navigation of a mobile robot in an indoor environment. *Robot. Auton. Syst.* **2002**, *38*, 1–18. [\[CrossRef\]](#)
49. Jazayeri, A.; Fatehi, A.; Taghirad, H. Mobile robot navigation in an unknown environment. In Proceedings of the 9th IEEE International Workshop on Advanced Motion Control, Istanbul, Turkey, 27–29 March 2006; pp. 295–299.
50. Kim, M.Y.; Cho, H. An active trinocular vision system of sensing indoor navigation environment for mobile robots. *Sens. Actuators* **2006**, *125*, 192–209. [\[CrossRef\]](#)
51. Huang, H.; Savkin, A.V. Viable path planning for data collection robots in a sensing field with obstacles. *Comput. Commun.* **2017**, *111*, 84–96. [\[CrossRef\]](#)
52. Zhang, Y.; Li, S. Distributed biased min-consensus with applications to shortest path planning. *IEEE Trans. Autom. Control* **2017**, *62*, 5429–5436. [\[CrossRef\]](#)
53. Yang, D.-H.; Hong, S.-K. A roadmap construction algorithm for mobile robot path planning using skeleton maps. *Adv. Robot.* **2007**, *21*, 51–63. [\[CrossRef\]](#)
54. Guzman, J.L.; Berenguel, M.; Rodriguez, F.; Dormido, S. An interactive tool for mobile robot motion planning. *Robot. Auton. Syst.* **2008**, *56*, 396–409. [\[CrossRef\]](#)
55. Joshi, V.A.; Banavar, R.N.; Hippalgaonkar, R. Design and analysis of a spherical mobile robot. *Mech. Mach. Theory* **2009**, *45*, 130–136. [\[CrossRef\]](#)
56. Tzafestas, S.G.; Tzamtzi, M.P.; Rigatos, G.G. Robust motion planning and control of mobile robots for collision avoidance in terrains with moving objects. *Math. Comput. Simul.* **2002**, *59*, 279–292. [\[CrossRef\]](#)
57. Tsianos, K.I.; Scan, I.A.; Kavraki, L.E. Sampling-based robot motion planning: Towards realistic applications. *Comput. Sci. Rev.* **2007**, *1*, 2–11. [\[CrossRef\]](#)
58. Fainekos, G.E.; Girard, A.; Kress-Gazit, H. George Journal Pappas, Temporal logic motion planning for dynamic robots. *Automatica* **2009**, *45*, 343–352. [\[CrossRef\]](#)
59. Akbaripour, H.; Masehian, E. Semi-lazy probabilistic roadmap: A parameter-tuned, resilient and robust path planning method for manipulator robots. *Int. J. Adv. Manuf. Technol.* **2017**, *89*, 1401–1430. [\[CrossRef\]](#)
60. Barfoot, T.D.; Clark, C.M. Motion planning for formations of mobile robots. *Robot. Auton. Syst.* **2004**, *46*, 65–78. [\[CrossRef\]](#)
61. Bhatt, R.M.; Tang, C.P.; Krovi, V.N. Formation optimization for a fleet of wheeled mobile robots: A geometric approach. *Robot. Auton. Syst.* **2009**, *57*, 102–120. [\[CrossRef\]](#)

62. Gao, J.; Ye, W.; Guo, J.; Li, Z. Deep Reinforcement Learning for Indoor Mobile Robot Path Planning. *Sensors* **2020**, *20*, 5493. [[CrossRef](#)] [[PubMed](#)]
63. Garrido, S.; Moreno, L.; Blanco, D.; Jurewicz, P. Path planning for mobile robot navigation using voronoi diagram and fast marching. *Int. J. Robot. Autom.* **2011**, *2*, 42–64.
64. Chiddarwar, S.S.; Babu, N.R. Conflict free coordinated path planning for multiple robots using a dynamic path modification sequence. *Robot. Auton. Syst.* **2011**, *59*, 508–518. [[CrossRef](#)]
65. Parhi, D.R.; Singh, M.K. Real time navigational control of mobile robots using an artificial neural network. *J. Mech. Eng. Sci. Part C* **2009**, *223*, 1713–1725. [[CrossRef](#)]
66. Liu, J.; Chen, Z.; Zhang, Y.; Li, W. Path planning of mobile robots based on improved Genetic algorithm. In Proceedings of the 2020 2nd International Conference on Robotics, Intelligent Control and Artificial Intelligence, Suzhou, China, 1–3 May 2020; pp. 49–53.
67. Suzuki, M.; Floreano, D.; Paolo, E.A.D. The contribution of active body movement to visual development in evolutionary robots. *Neural Netw.* **2005**, *18*, 656–665. [[CrossRef](#)] [[PubMed](#)]
68. Maurović, I.; Seder, M.; Lenac, K.; Petrović, I. Path planning for active SLAM based on the D\* algorithm with negative edge weights. *IEEE Trans. Syst. Man Cybern. Syst.* **2017**, *48*, 1321–1331. [[CrossRef](#)]
69. Luo, M.; Hou, X.; Yang, J. Surface optimal path planning using an extended Dijkstra algorithm. *IEEE Access* **2020**, *8*, 147827–147838. [[CrossRef](#)]
70. Fukao, T.; Nakagawa, H.; Adachi, N. Adaptive Tracking Control of a Non-holonomic Mobile Robot. *IEEE Trans. Robot. Autom.* **2000**, *16*, 609–615. [[CrossRef](#)]
71. Gu, D.; Hu, H. Neural predictive control for a car-like mobile robot. *Robot. Auton. Syst.* **2002**, *39*, 73–86. [[CrossRef](#)]
72. Ghatee, M.; Mohades, A. Motion planning in order to optimize the length and clearance applying a Hopfield neural network. *Expert Syst. Appl.* **2009**, *36*, 4688–4695. [[CrossRef](#)]
73. Kala, R.; Shukla, A.; Tiwari, R. Robotic path planning in static environment using hierarchical multi-neuron heuristic search and probability based fitness. *Neurocomputing* **2011**, *74*, 2314–2335. [[CrossRef](#)]
74. Ouarda, H. A neural network-based navigation for intelligent autonomous mobile robots. *Int. J. Math. Models Methods Appl. Sci.* **2010**, *4*, 177–186.
75. Wang, M.; Liu, J.N.K. Fuzzy logic-based robot path planning in unknown environment. *Mach. Learn. Cybern.* **2005**, *2*, 813–818.
76. Panov, A.I.; Yakovlev, K. Behavior and path planning for the coalition of cognitive robots in smart relocation tasks. In *Robot Intelligence Technology and Applications 4: Results from the 4th International Conference on Robot Intelligence Technology and Applications*; Springer International Publishing: Berlin/Heidelberg, Germany, 2017; pp. 3–20.
77. Zhang, H.Y.; Lin, W.M.; Chen, A.X. Path planning for the mobile robot: A review. *Symmetry* **2018**, *10*, 450. [[CrossRef](#)]
78. Kim, M.; Han, D.K.; Park, J.H.; Kim, J.S. Motion planning of robot manipulators for a smoother path using a twin delayed deep deterministic policy gradient with hindsight experience replay. *Appl. Sci.* **2020**, *10*, 575. [[CrossRef](#)]
79. Tuncer, A.; Yildirim, M. Dynamic path planning of mobile robots with improved genetic algorithm. *Comput. Electr. Eng.* **2012**, *38*, 1564–1572. [[CrossRef](#)]
80. Wang, X.; Luo, X.; Han, B.; Chen, Y.; Liang, G.; Zheng, K. Collision-free path planning method for robots based on an improved rapidly-exploring random tree algorithm. *Appl. Sci.* **2020**, *10*, 1381. [[CrossRef](#)]
81. Xu, R.; Wang, Q.; Shi, L.; Chen, L. Design of multi-robot path planning system based on hierarchical fuzzy control. *Procedia Eng.* **2011**, *15*, 235–239.
82. Wu, M.; Dai, S.L.; Yang, C. Mixed reality enhanced user interactive path planning for omnidirectional mobile robot. *Appl. Sci.* **2020**, *10*, 1135. [[CrossRef](#)]
83. Wang, X.; Liu, Z.; Liu, J. Mobile robot path planning based on an improved A\* algorithm. In Proceedings of the International Conference on Computer Graphics, Artificial Intelligence, and Data Processing (ICCAID 2022), Guangzhou, China, 23–25 December 2022; Volume 12604, pp. 1093–1098.
84. Lin, S.; Liu, A.; Wang, J.; Kong, X. An intelligence-based hybrid PSO-SA for mobile robot path planning in warehouse. *J. Comput. Sci.* **2023**, *67*, 101938. [[CrossRef](#)]
85. Najjaran, H.; Goldenberg, A. Real-time motion planning of an autonomous mobile manipulator using a fuzzy adaptive Kalman filter. *Robot. Auton. Syst.* **2007**, *55*, 96–106. [[CrossRef](#)]
86. Yukihiro, O.; Uchiyama, H.; Potter, W.D. A mobile robot for corridor navigation: A multi-agent approach. In Proceedings of the 42nd Annual Southeast Regional Conference, ACM-SE 42, New York, NY, USA, 2–3 April 2004.
87. Hung, L.C.; Chung, H.Y. Design of hierarchical fuzzy logic control for mobile robot system, Robotics and Automation. *IEEE Trans.* **2006**, *18*, 235–239.
88. Qin, H.; Shao, S.; Wang, T.; Yu, X.; Jiang, Y.; Cao, Z. Review of Autonomous Path Planning Algorithms for Mobile Robots. *Drones* **2023**, *7*, 211. [[CrossRef](#)]
89. Cao, Y.; Wang, Y.; Vashisth, A.; Fan, H.; Sartoretti, G.A. CATNIPP: Context-aware attention-based network for informative path planning. *PMLR* **2023**, *205*, 1928–1937.
90. Ekrem, Ö.; Aksoy, B. Trajectory planning for a 6-axis robotic arm with particle swarm optimization algorithm. *Eng. Appl. Artif. Intell.* **2023**, *122*, 106099. [[CrossRef](#)]



91. Muñoz, J.; López, B.; Quevedo, F.; Barber, R.; Garrido, S.; Moreno, L. Geometrically constrained path planning for robotic grasping with Differential Evolution and Fast Marching Square. *Robotica* **2023**, *41*, 414–432. [\[CrossRef\]](#)
92. Shi, K.; Wu, Z.; Jiang, B.; Karimi, H.R. Dynamic path planning of mobile robot based on improved simulated annealing algorithm. *J. Frankl. Inst.* **2023**, *360*, 4378–4398. [\[CrossRef\]](#)
93. Lin, S.; Liu, A.; Wang, J.; Kong, X. A Review of Path-Planning Approaches for Multiple Mobile Robots. *Machines* **2022**, *10*, 773. [\[CrossRef\]](#)
94. Gul, F.; Mir, I.; Alarabiat, D.; Alabool, H.M.; Abualigah, L.; Mir, S. Implementation of bio-inspired hybrid algorithm with mutation operator for robotic path planning. *J. Parallel Distrib. Comput.* **2022**, *169*, 171–184. [\[CrossRef\]](#)
95. An, R.; Guo, S.; Zheng, L.; Hirata, H.; Gu, S. Uncertain moving obstacles avoiding method in 3D arbitrary path planning for a spherical underwater robot. *Robot. Auton. Syst.* **2022**, *151*, 104011. [\[CrossRef\]](#)
96. Jiang, L.; Liu, S.; Cui, Y.; Jiang, H. Path planning for robotic manipulator in complex multi-obstacle environment based on improved\_RRT. *IEEE/ASME Trans. Mechatron.* **2022**, *27*, 4774–4785. [\[CrossRef\]](#)
97. Kyaw, P.T.; Le, A.V.; Veerajagadheswar, P.; Elara, M.R.; Thu, T.T.; Nhan, N.H.K.; Duc, P.V.; Vu, M.B. Energy-efficient path planning of reconfigurable robots in complex environments. *IEEE Trans. Robot.* **2022**, *38*, 2481–2494. [\[CrossRef\]](#)
98. Kumar, S.; Dadas, S.S.; Parhi, D.R. Path planning of mobile robot using modified DAYKUN-BIP virtual target displacement method in static environments. *Wirel. Pers. Commun.* **2023**, *128*, 2287–2305. [\[CrossRef\]](#)
99. Ntakolia, C.; Moustakidis, S.; Siouras, A. Autonomous path planning with obstacle avoidance for smart assistive systems. *Expert Syst. Appl.* **2023**, *213*, 119049. [\[CrossRef\]](#)
100. Chen, L.; Wang, Y.; Miao, Z.; Mo, Y.; Feng, M.; Zhou, Z.; Wang, H. Transformer-based Imitative Reinforcement Learning for Multi-Robot Path Planning. *IEEE Trans. Ind. Inform.* **2023**, *19*, 10233–10243. [\[CrossRef\]](#)
101. Aslan, M.F.; Durdu, A.; Sabanci, K. Goal distance-based UAV path planning approach, path optimization and learning-based path estimation: GDRRT\*, PSO-GDRRT\* and BiLSTM-PSO-GDRRT. *Appl. Soft Comput.* **2023**, *137*, 110156. [\[CrossRef\]](#)
102. An, D.; Mu, Y.; Wang, Y.; Li, B.; Wei, Y. Intelligent Path Planning Technologies of Underwater Vehicles: A Review. *J. Intell. Robot. Syst.* **2023**, *107*, 22. [\[CrossRef\]](#)
103. Kumar, S.; Sikander, A. An intelligent optimize path planner for efficient mobile robot path planning in a complex terrain. *Microsyst. Technol.* **2023**, *29*, 469–487. [\[CrossRef\]](#)
104. Cai, J.; Du, A.; Liang, X.; Li, S. Prediction-Based Path Planning for Safe and Efficient Human–Robot Collaboration in Construction via Deep Reinforcement Learning. *J. Comput. Civ. Eng.* **2023**, *37*, 04022046. [\[CrossRef\]](#)
105. Weingartshofer, T.; Bischof, B.; Meiringer, M.; Hartl-Nesic, C.; Kugi, A. Optimization-based path planning framework for industrial manufacturing processes with complex continuous paths. *Robot. Comput.-Integr. Manuf.* **2023**, *82*, 102516. [\[CrossRef\]](#)
106. Liu, L.; Wang, X.; Yang, X.; Liu, H.; Li, J.; Wang, P. Path Planning Techniques for Mobile Robots: Review and Prospect. *Expert Syst. Appl.* **2023**, *227*, 120254. [\[CrossRef\]](#)
107. Bilal, H.; Yin, B.; Kumar, A.; Ali, M.; Zhang, J.; Yao, J. Jerk-bounded trajectory planning for rotary flexible joint manipulator: An experimental approach. *Soft Comput.* **2023**, *27*, 4029–4039. [\[CrossRef\]](#)
108. Luan, P.G.; Thinh, N.T. Hybrid genetic algorithm based smooth global-path planning for a mobile robot. *Mech. Based Des. Struct. Mach.* **2023**, *51*, 1758–1774. [\[CrossRef\]](#)
109. Jin, J.; Zhang, Y.; Zhou, Z.; Jin, M.; Yang, X.; Hu, F. Conflict-based search with D\* lite algorithm for robot path planning in unknown dynamic environments. *Comput. Electr. Eng.* **2023**, *105*, 108473. [\[CrossRef\]](#)
110. Singh, I.; Blukis, V.; Mousavian, A.; Goyal, A.; Xu, D.; Tremblay, J.; Fox, D.; Thomason, J.; Garg, A. Progprompt: Generating situated robot task plans using large language models. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023; pp. 11523–11530.
111. Shridhar, M.; Manuelli, L.; Fox, D. Perceiver-actor: A multi-task transformer for robotic manipulation. *PMLR* **2023**, *205*, 785–799.
112. Liu, Y.; Ping, Y.; Zhang, L.; Wang, L.; Xu, X. Scheduling of decentralized robot services in cloud manufacturing with deep reinforcement learning. *Robot. Comput.-Integr. Manuf.* **2023**, *80*, 102454. [\[CrossRef\]](#)
113. Mees, O.; Zeng, A.; Burgard, W. Visual language maps for robot navigation. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023; pp. 10608–10615.
114. Wang, W.; Guo, Q.; Yang, Z.; Jiang, Y.; Xu, J. A state-of-the-art review on robotic milling of complex parts with high efficiency and precision. *Robot. Comput.-Integr. Manuf.* **2023**, *79*, 102436. [\[CrossRef\]](#)
115. Klenert, D.; Fernandez-Macias, E.; Anton, J.I. Do robots really destroy jobs? Evidence from Europe. *Econ. Ind. Democr.* **2023**, *44*, 280–316. [\[CrossRef\]](#)
116. Tokgoz, E.; Levitt, S.; Sosa, D.; Carola, N.A.; Patel, V. Robotics applications in total knee arthroplasty. In *Total Knee Arthroplasty: A Review of Medical and Biomedical Engineering and Science Concepts*; Springer: Berlin/Heidelberg, Germany, 2023; pp. 155–174.
117. Zarei, M.; Lee, G.; Lee, S.G.; Cho, K. Advances in biodegradable electronic skin: Material progress and recent applications in sensing, robotics, and human–machine interfaces. *Adv. Mater.* **2023**, *35*, 2203193. [\[CrossRef\]](#)
118. Zhang, Y.; Li, P.; Quan, J.; Li, L.; Zhang, G.; Zhou, D. Progress, challenges, and prospects of soft robotics for space applications. *Adv. Intell. Syst.* **2023**, *5*, 2200071. [\[CrossRef\]](#)
119. Karamcheti, S.; Nair, S.; Chen, A.S.; Kollar, T.; Finn, C.; Sadigh, D.; Liang, P. Language-driven representation learning for robotics. *arXiv* **2023**, arXiv:2302.12766.



120. Hirose, N.; Shah, D.; Sridhar, A.; Levine, S. ExAug: Robot-conditioned navigation policies via geometric experience augmentation. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023; pp. 4077–4084.
121. Li, C.; Zheng, P.; Yin, Y.; Pang, Y.M.; Huo, S. An AR-assisted Deep Reinforcement Learning-based approach towards mutual-cognitive safe human-robot interaction. *Robot. Comput.-Integr. Manuf.* **2023**, *80*, 102471. [\[CrossRef\]](#)
122. Lopez, I.B.; Benzakour, A.; Mavrogenis, A.; Benzakour, T.; Ahmad, A.; Lemée, J.M. Robotics in spine surgery: Systematic review of literature. *Int. Orthop.* **2023**, *47*, 447–456. [\[CrossRef\]](#)
123. Stella, F.; Hughes, J. The science of soft robot design: A review of motivations, methods and enabling technologies. *Front. Robot. AI* **2023**, *9*, 1059026. [\[CrossRef\]](#)
124. Kent, M.S.; Hartwig, M.G.; Vallières, E.; Abbas, A.E.; Cerfolio, R.J.; Dylewski, M.R.; Fabian, T.; Herrera, L.J.; Jett, K.G.; Lazzaro, R.S.; et al. Pulmonary Open, Robotic and Thoracoscopic Lobectomy (PORTaL) Study: Survival Analysis of 6,646 Cases. *Ann. Surg.* **2023**, *277*, 1002–1009. [\[CrossRef\]](#)
125. Wu, L.; Huang, X.; Cui, J.; Liu, C.; Xiao, W. Modified adaptive ant colony optimization algorithm and its application for solving path planning of mobile robot. *Expert Syst. Appl.* **2023**, *215*, 119410. [\[CrossRef\]](#)
126. Madariaga, L.; Allendes, C.; Nussbaum, M.; Barrios, G.; Acevedo, N. Offline and online user experience of gamified robotics for introducing computational thinking: Comparing engagement, game mechanics and coding motivation. *Comput. Educ.* **2023**, *193*, 104664. [\[CrossRef\]](#)
127. Li, J.; Zou, L.; Luo, G.; Wang, W.; Lv, C. Enhancement and evaluation in path accuracy of industrial robot for complex surface grinding. *Robot. Comput.-Integr. Manuf.* **2023**, *81*, 102521. [\[CrossRef\]](#)
128. Sinha, A.; Mishra, P.; Ramish, M.; Mahmood, H.R.; Upadhyay, K.K. Employing Unsupervised Learning Algorithm for Stock Market Analysis and Prediction. In Proceedings of the 2021 First International Conference on Advances in Computing and Future Communication Technologies (ICACFCT), Meerut, India, 16–17 December 2021; pp. 75–79. [\[CrossRef\]](#)
129. Ramish, M.; Sinha, A.; Desai, J.; Raj, A.; Rajawat, Y.S.; Punia, P. IT Attack Detection and Classification using Users Event Log Feature And Behavior Analytics through Fourier EEG Signal. In Proceedings of the 2022 IEEE 11th International Conference on Communication Systems and Network Technologies (CSNT), Indore, India, 23–24 April 2022; pp. 577–582. [\[CrossRef\]](#)
130. Sinha, A.; Ramish, M.; Kumari, S.; Jha, P.; Tiwari, M.K. ANN-ANT-LION-MLP Ensemble Transfer Learning Based Classifier for Detection and Classification of Oral Disease Severity. In Proceedings of the 2022 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 27–28 January 2022; pp. 530–535. [\[CrossRef\]](#)
131. Sinha, A.; Kumar, B.; Banerjee, P.; Ramish, M. HSCAD: Heart Sound Classification for Accurate Diagnosis using Machine Learning and MATLAB. In Proceedings of the 2021 International Conference on Computational Performance Evaluation (ComPE), Shillong, India, 1–3 December 2021; pp. 115–120. [\[CrossRef\]](#)
132. Raj, A.; Jadon, S.; Kulshrestha, H.; Rai, V.; Arvindhan, M.; Sinha, A. Cloud Infrastructure Fault Monitoring and Prediction System using LSTM based predictive maintenance. In Proceedings of the 2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 13–14 October 2022; pp. 1–6. [\[CrossRef\]](#)
133. Bhargavi, M.; Sinha, A.; Desai, J.; Garg, N.; Bhatnagar, Y.; Mishra, P. Comparative Study of Consumer Purchasing and Decision Pattern Analysis using Pincer Search Based Data Mining Method. In Proceedings of the 2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, 3–5 October 2022; pp. 1–7. [\[CrossRef\]](#)
134. Bhargavi, M.; Sinha, A.; Rao, G.M.; Bhatnagar, Y.; Kumar, S.; Pawar, S.R. Application of IoT for Proximity Analysis and Alert Generation for Maintaining Social Distancing. In *Proceedings of the ISMS 2022: Key Digital Trends Shaping the Future of Information and Management Science*; Lecture Notes in Networks and Systems; Springer: Cham, Switzerland, 2023; Volume 671. [\[CrossRef\]](#)
135. Kumar, B.; Roy, S.; Singh, K.U.; Pandey, S.K.; Kumar, A.; Sinha, A.; Shukla, S.; Shah, M.A.; Rasool, A. A Static Machine Learning Based Evaluation Method for Usability and Security Analysis in E-Commerce Website. *IEEE Access* **2023**, *11*, 40488–40510. [\[CrossRef\]](#)
136. Hai, T.; Zhou, J.; Lu, Y.; Jawawi, D.N.; Sinha, A.; Bhatnagar, Y.; Anumbe, N. Posterior probability and collaborative filtering based Heterogeneous Recommendations model for user/item Application in use case of IoVT. *Comput. Electr. Eng.* **2023**, *105*, 108532. [\[CrossRef\]](#)
137. Wewerka, J.; Reichert, M. Robotic process automation-a systematic mapping study and classification framework. *Enterp. Inf. Syst.* **2023**, *17*, 1986862. [\[CrossRef\]](#)
138. Abeyruwan, S.W.; Graesser, L.; D'Ambrosio, D.B.; Singh, A.; Shankar, A.; Bewley, A.; Jain, D.; Choromanski, K.; Sanketi, P.R. i-sim2real: Reinforcement learning of robotic policies in tight human-robot interaction loops. *PMLR* **2023**, *205*, 212–224.
139. Wurm, K.M.; Hornung, A.; Bennewitz, M.; Stachniss, C.; Burgard, W. OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems. In Proceedings of the ICRA 2010 Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation, Anchorage, AK, USA, 3–7 May 2010; Volume 2, p. 3.
140. Thrun, S.; Burgard, W.; Fox, D. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping. In Proceedings of the 2000 ICRA Millennium Conference, IEEE International Conference on Robotics and Automation, Cat. No. 00CH37065. San Francisco, CA, USA, 24–28 April 2000; Volume 1, pp. 321–328.
141. Borrmann, D.; Nüchter, A.; Đakulović, M.; Maurović, I.; Petrović, I.; Osmanković, D.; Velagić, J. A mobile robot based system for fully automated thermal 3D mapping. *Adv. Eng. Inform.* **2014**, *28*, 425–440. [\[CrossRef\]](#)

142. Hirose, S.; Ootsukasa, N.; Shirasu, T.; Kuwahara, H.; Yoneda, K. Fundamental considerations for the design of a planetary rover. In Proceedings of the 1995 IEEE International Conference on Robotics and Automation, Nagoya, Japan, 21–27 May 1995; Volume 2, pp. 1939–1944.
143. Ding, L.; Deng, Z.; Gao, H.; Nagatani, K.; Yoshida, K. Planetary rovers' wheel–soil interaction mechanics: New challenges and applications for wheeled mobile robots. *Intell. Serv. Robot.* **2011**, *4*, 17–38. [[CrossRef](#)]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.