

Review

Not peer-reviewed version

Recent Developments in Path Planning for Unmanned Ground Vehicles in Underground Mining Environment

[Abdurauf Abdukodirov](#)^{*} and [Jörg Benndorf](#)

Posted Date: 9 April 2025

doi: 10.20944/preprints202504.0711.v1

Keywords: underground mine; navigation strategies; path planning algorithms; UGV; ROS



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Review

Recent Developments in Path Planning for Unmanned Ground Vehicles in Underground Mining Environment

Abdurauf Abdukodirov * and Jörg Benndorf

Department of Mine Surveying and Geodesy, TU Bergakademie Freiberg, 09599 Freiberg, Germany

* Correspondence: abdurauf.abdukodirov@doktorand.tu-freiberg.de

Abstract: This paper presents a thorough review of path-planning algorithms employed for the navigation of Unmanned Ground Vehicles (UGVs) in underground mining environments. It outlines the key components and requirements that are essential for an effective path planning framework, including sensors and the Robot Operating System (ROS). The review examines both global and local path-planning techniques, encompassing traditional graph-based methods, sampling-based approaches, nature-inspired algorithms and Reinforcement learning strategies. Through the analysis of the extant literatures on the subject, the review paper highlighted the strengths of the employed techniques, the application scenarios, the testing environments and the optimization strategies. The most favorable and relevant algorithms were identified. The paper acknowledges a significant limitation: the over-reliance on simulation testing for path-planning algorithms and the computational difficulties in implementing some of them in real mining condition. It concludes by emphasizing the necessity for full-scale research on path planning in real mining conditions.

Keywords: underground mine; navigation strategies; path planning algorithms; UGV; ROS

1. Introduction

In the age of technological advancement, automatization has permeated almost all industries, including underground mining [1]. The high risk and challenges inherent in traditional mining have precipitated the introduction of automation measures. The elevated probability of injury in such a harsh environment poses a significant threat to workers' lives [2]. In addition, survey accuracy is affected by worker fatigue, which impacts measurement accuracy and increases survey time.

Underground automation heavily depends on efficient and reliable navigation systems, serving as a key enabler for autonomous operations in complex and hazardous environments [3]. Autonomous navigation is applicable in various environments, including outdoor, indoor, underground mines, and even underwater [4]. Beyond improving precision and reliability, effective navigation reduces the risks associated with manual operations, making it a fundamental component of underground automation.

Compared to other environments, navigation in underground mine presents unique challenges, including the unpredictable nature of underground terrain, such as irregular surfaces, narrow passages and dynamic obstacles as listed in Table 1, requiring robust navigation to ensure safety and efficiency [5]. Additionally, absence of GPS signals, requiring advanced techniques like Simultaneous Localization and Mapping (SLAM) and Dead-reckoning methods [6,7].

Table 1. Unique factors of each environment and theirs impact for navigation.

Factors	Mine	Indoor	Outdoor
Surface	Irregular surfaces,	Smooth and structured	Varied surfaces
Conditions	narrow passages, mud	surfaces	
Obscurants	Dust, smoke, gas	Nearly clear	Rain, fog and snow
GPS Availability	GPS denied	Available with extended	Widely available and
		setups	effective
Obstacle Density	Relatively high	Low	Moderate

Navigation in mining is crucial for enabling autonomous vehicles and systems to perform various tasks efficiently and safely in challenging underground environments. For an example:

- For surveying and mapping. Autonomous vehicles equipped with LiDAR and SLAM systems navigate the mine to generate 3D maps for planning and operational purposes [8];
- Transportation. Autonomous haul trucks navigate predefined routes to transport ore, minerals, or waste material between loading and dumping points in underground mines [9];
- Drilling and Blasting Operations. Navigation systems guide autonomous drilling machines to precise coordinates within the mine for efficient drilling [10];
- Inspection and Maintenance. Autonomous robots navigate mine tunnels to detect structural integrity issues and gas leaks [11];
- Search and Rescue Operations (SAR). Navigation enables unmanned vehicles to explore hazardous or collapsed mine areas where human entry is unsafe [12].
- Underground navigation in autonomous vehicle applications involves several key steps to ensure safe and efficient operation. It begins with Localization, where the vehicle determines its position and orientation within its environment using techniques like SLAM, or Dead-reckoning. Next is Perception, which involves sensing and mapping the surroundings using Light Detection and Ranging (LiDAR), cameras, or radar to detect obstacles and terrain features. Then Path planning determines the optimal route to reach a destination while avoiding obstacles and adapting to environmental changes. Finally, Motion control executes the planned path by steering, accelerating or braking, ensuring precise and smooth movement. These steps are supported by sensor integration and real-time decision-making to handle dynamic conditions effectively [13].
- In the context of autonomous underground navigation, a variety of platforms assume significant roles in enhancing efficiency, safety and productivity. Unmanned Ground Vehicles (UGVs) and Unmanned Aerial Vehicles (UAVs) are among the key technologies driving automation in underground mining operations. Both of them are essential for surveying, exploration, inspection and SAR tasks, where human intervention is either constrained or infeasible. UGVs with high power and payload capacity have advantages in long-duration missions and transportation, but they are limited in mobility. UAVs, on the other hand, are good at quick inspections and can reach areas that are difficult to access [14]. Beyond unmanned vehicles, many mining machines, including LHDs, are being redesigned for full automation, further enhancing the efficiency and safety of underground operations [15].
- As one of the crucial components of navigation, path planning might be modified or enhanced with optimization strategies to achieve efficient and accurate navigation in the underground mining environment. While extensive research has been conducted on surface-level path planning [16], underground navigation remains a critical and evolving research area due to its unique environmental challenges and constraints. This paper provides a comprehensive review of recent advancements in underground UGV path planning, highlighting emerging trends, methodological improvements, and existing gaps in the field. The analysis aims to offer insights into current limitations and identify opportunities for future research and development in optimizing autonomous navigation for underground mining applications.

2. Requirement Analysis for Path Planning for UGV in Underground Mining Environment

The requirements outlined in this section serve as a key consideration for navigation and the evaluation of existing path planning research in underground mine using UGVs. General requirements for underground navigation:

- UGV must operate autonomously in GPS-denied environments;
- Performing effectiveness in low-light, high-dust and uneven terrain conditions with help of advanced sensors such as LiDAR, cameras, radars, IMUs and motor encoders for accurate environmental mapping and obstacle detection;
- UGV must be compatible with Robot Operating System (ROS) framework, enabling modular communication, multi-sensor integration and scalability;
- UGV must be reliably detect and avoid both static and dynamic obstacles, generating accurate and efficient paths in real-time while incorporating recovery behaviors to handle unexpected environmental changes;
- Adaptability to environmental complexity, such as dynamic obstacles and sudden changes in tunnel structures, is necessary for reliable underground navigation
- Energy-efficient operation is also essential to prolong the operational lifespan of UGVs in underground mine [17].

Path planning requirements are primarily determined by the algorithms employed. The selection of an appropriate algorithm is based on an analysis of environmental and vehicle parameters. To evaluate path planning algorithms and review existing research, the key requirements and evaluation criteria for planned paths are summarized in Table 2.

Table 2. Requirement and the evaluation criteria for planned path.

Requirements	Evaluation criteria
Optimal path	Minimal travel time and path length
Smoothness	Spatial and temporal smoothness coefficients [18]
High accuracy and safety	Avoids obstacles and ensures collision-free path
Success rate	Percentage of successful path completions
Computational cost	Processing time and resource consumption
Robustness	Ability to handle uncertainties
Handling narrow tunnels	Minimum passable width

After obtaining the results, the analysis focuses on aligning the evaluated criteria with the specific requirements of the desired mining tasks. These requirements provide a basis for evaluating UGV navigation systems and identifying research gaps.

3. Materials

3.1. Sensors to Perceive Environment

Sensors are critical components in underground mining applications, enabling UGVs to perceive their surroundings, detect obstacles, and generate optimal paths in challenging environments. Considering the different operating conditions in underground mines, the selection of suitable sensors depends on the specific navigation requirements of the UGV. The following sensors are frequently employed for navigation purposes in mining operations:

- LiDAR (Light Detection and Ranging);
- Depth Camera;
- Radar;
- IMU (Inertial Measurement Units);

- Motor encoders.
LiDAR is widely utilized technique for 3D mapping and obstacle detection, in which precise point clouds of the surroundings are generated. Laser scanners, on the other hand, operate similarly but offer high-resolution scans for detailed surface analysis. Depth cameras capture depth information to identify variations in the terrain and assist in real-time object detection. Radar is effective in low-visibility conditions such as dust and fog, using radio waves to detect obstacles beyond the limitations of optical sensors. IMUs enhance localization accuracy by tracking motion changes through accelerometers and gyroscopes. In addition, Motor encoders which measure wheel rotations, provide odometry data to estimate the vehicle’s displacement and support dead reckoning-based navigation [19]. Each sensor has its strengths and weaknesses in different applications, depending on the environment and accuracy required, as described in Table 3.

Table 3. Sensors Features.

Sensors	Strengths	Weaknesses	Range and frequency
LiDAR 3D [20]	Works well in low-light environments; High accuracy in 3D mapping and obstacle detection.	Performance can degrade in dust, smoke or reflective surfaces; High power consumption.	10 – 300 m; 10 – 100 Hz
Laser scanner 2D [21]	Works in low-light environments; Provides precise distance Measurements.	Performance may degrade in heavy particulate environments; Computationally Intensive.	0,5 - 25 m; 10 – 100 Hz
Depth Camera [22]	Provides both color (RGB) and depth (D) information;	Sensitive to reflective and transparent surfaces;	0.2 – 10 m; 30 – 90 Hz
	Compact and lightweight	Can be disrupted by dust or fog.	
Radar [23]	Works in harsh environments (Dust, smoke); Reliable for detecting dynamic Objects.	Difficult to interpret data without additional processing; Lower resolution.	0.1 – 250 m; 10 – 200 Hz
IMU [24]	Small, lightweight, and power-efficient; Provides high-frequency motion data.	Susceptible to external vibrations and sensor noise; Accumulates drift over time.	100 – 1000 Hz
Motor Encoder [25]	Simple integration with UGV control systems; High frequency data.	Accumulates drift over long distances (wheel slippage); Not reliable for rough terrain or slippery surfaces.	10 – 1000 Hz

For robust underground mine path planning, sensor fusion (combining multiple sensors) is often necessary. A typical UGV utilizes LiDAR or Scanners, along with IMUs and motor encoders. Additionally, Radar complements LiDAR in harsh conditions, while Depth cameras are used for object recognition and enhanced depth sensing [26].

In underground navigation, ROS (Robot Operating System) plays a key role as a middleware, facilitating communication between sensors, data acquisition, and processing in UGVs. It provides a structured framework for integrating multiple sensors, ensuring synchronized data flow for perception, localization and navigation.

3.2. ROS as Main Operation System

ROS is a flexible framework used primarily in robotics for developing robot software. Despite its name, ROS is not a traditional operating system but rather a middleware framework that provides tools, libraries, and conventions for creating complex and robust robot applications [27]. There are several advantages that ROS can provide:

- ROS is open source that can be augmented by developers;
- Modular and flexible framework allows researchers to customize and adapt it for a wide range of applications.
- Cross-platform compatibility and hardware-agnostic design supports diverse robotic systems from drones and robotic arms to autonomous vehicles.
- Robust built-in libraries for computer vision, motion planning and navigation;
- Simulation tools such as Gazebo and RViz, allowing researchers to efficiently prototype, test and refine robots.
- Integration with AI and machine learning technologies extends its usefulness in creating intelligent and autonomous systems [28].

ROS has evolved through two major generations. ROS 1 (2007) laid the foundation for robotics development with modularity and ease of use, widely adopted for research and non-critical applications. Key versions include ROS Kinetic (2016) and ROS Melodic (2018). However, it faced limitations in real-time performance and scalability. ROS 2 (2017) addressed these issues with parallel processing, secure communication and real-time support, making it ideal for industrial use. Built on Data Distribution Service (DDS), ROS 2 ensures robust communication and scalability, with key releases like Foxy (2020) and Humble (2022). Although ROS 1 is still in use, the focus is gradually shifting to ROS 2 as the emerging industry standard [29].

In ROS, a node is a fundamental building block that represents a single process performing a specific task, such as controlling a sensor or executing a robot's movement logic. Nodes communicate with each other using topics, which are named channels for message passing. A node can publish messages to a topic or subscribe to a topic to receive messages, enabling modular and scalable communication between different parts of a robot's system. For instance, a camera node might publish image data to a topic, while another node subscribes to process that data. The ROS file system provides a structured way to organize resources and facilitate development within the ROS framework. It is designed to manage the various components of a robotics project, including nodes, packages, messages and configurations.

Key Components of ROS File System:

- Packages - is the fundamental unit of software organization in ROS. It contains all the necessary files for a specific functionality, such as nodes, libraries, configurations and launch files. Each package is self-contained, enabling easy reuse and sharing;
- Metapackages - are collections of related packages grouped together under a common purpose, such as *ros_base* or *desktop_full*;
- Workspaces - are directories where ROS packages are developed and built.
- Each package has a standardized directory structure:
 - *src/*: Source code for nodes and other scripts;
 - *launch/*: Launch files to start nodes and set parameters;
 - *config/*: Configuration files of the robot descriptions to manage its behavior;
 - *msg/* and *srv/*: Definitions for custom message and service types;
 - *CMakeLists.txt* and *package.xml*: Build system and package metadata files.
- ROS defines specific file types for communication as Figure 1:
- Messages: Define data structures for node-to-node communication;
- Services: Specify request-response structures for synchronous interactions;
- Actions: Provide a framework for long-running tasks with feedback.

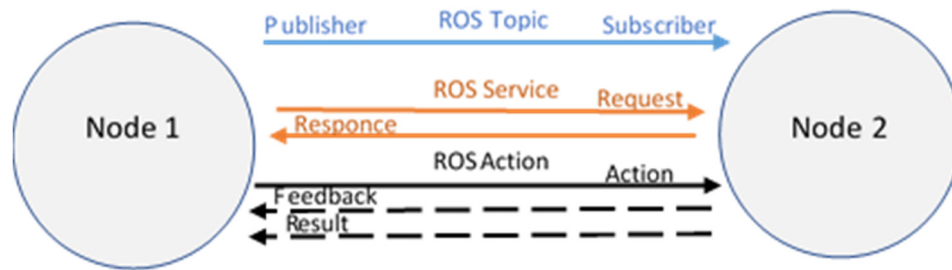


Figure 1. ROS communication types.

The ROS file system ensures consistency, modularity and ease of collaboration. By organizing code and resources into packages, it simplifies sharing, testing and scaling across projects [30].

The operating system requirements for ROS vary depending on the generation. ROS1 is primarily designed for Ubuntu Linux, as it leverages Ubuntu's package management system and development ecosystem. In contrast, ROS 2 offers broader compatibility and can be installed on multiple operating systems, including Ubuntu and Windows.

ROS is also ideal for underground and mining applications due to its modular and flexible architecture, enabling customization for navigation, mapping and inspection in challenging environments. It enables UGVs to perform essential tasks, including SLAM, path planning and visualization and supports multi-sensor integration such as LiDAR, radar and cameras for reliable operation in low-light and dusty conditions. ROS facilitates real-time data processing, remote operation and multi-robot coordination, enhancing safety and efficiency. Numerous completed and ongoing ROS-based projects focus on automating mining operations and conducting rescue missions in mining disasters. As an example, successful completion of the DARPA Subterranean Challenge in 2021 [31]. It was a robotics competition aimed at pushing the boundaries of autonomous navigation and exploration in complex underground environments. The challenge sought innovative approaches to quickly and effectively map, navigate and search through difficult subterranean environments such as caves, tunnels and urban underground structures.

4. Path Planning Framework

Path planning is a fundamental component of autonomous UGV navigation, responsible for determining an optimal or feasible path from a starting position to a destination while avoiding obstacles and considering environmental constraints. Developing a path-planning framework is inherently complex, as it encompasses environment representation, localization and perception, path-planning algorithms, obstacle avoidance and motion control. This framework ensures that UGVs can safely and efficiently navigate underground environments, where GPS is unavailable and traditional navigation methods are ineffective [32].

The environment representation plays a crucial role in path planning, as it defines how the UGV perceives and interacts with its surroundings. Various representation methods include grid-based maps, topological maps, point cloud maps and hybrid maps, each offering different advantages depending on the navigation requirements and computational constraints [33]. For localization in a known environment, sensor-based techniques such as Monte Carlo Localization (MCL) and feature-based localization are commonly employed [34]. However, in unknown environments, SLAM is required to build a map while simultaneously determining the UGV's position. Once the UGV has identified its start or current position, it can execute path planning algorithms to navigate efficiently toward the target location. After that, Motion control manages the movement of vehicles.

Path planning algorithms can be broadly classified into global and local planning approaches, each serving a specific role in navigation as described in Figure 2.

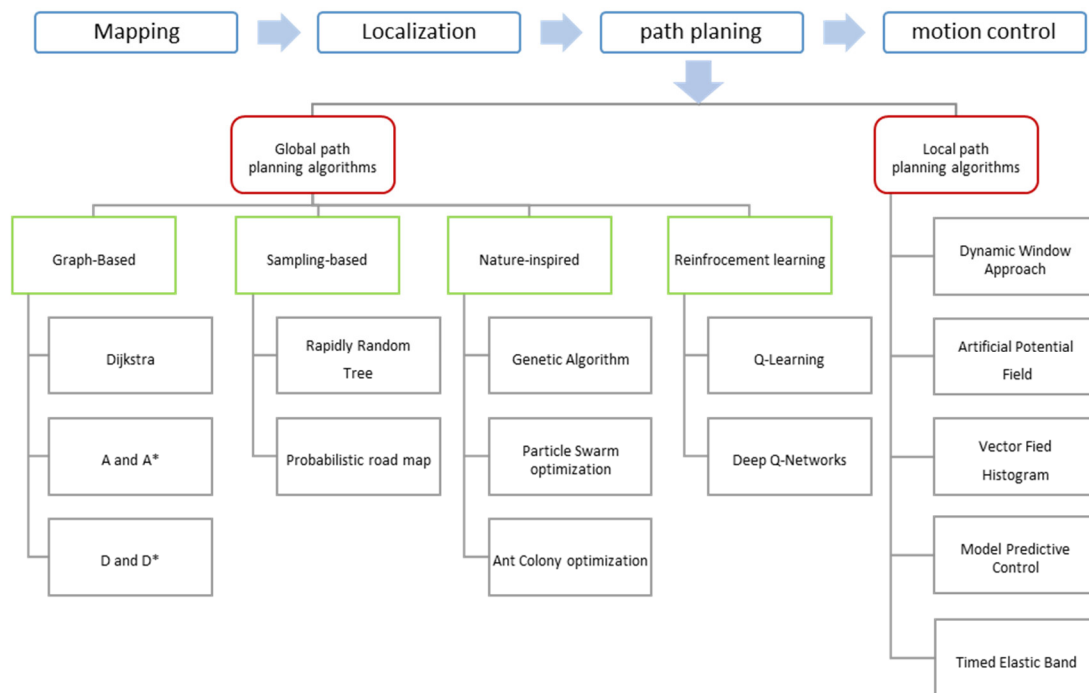


Figure 2. Workflow and types of path planning.

4.1. Global Path Planning Algorithms

Global path planning operates with a pre-existing map and computes a path before execution. These algorithms are suitable for structured environments where a detailed representation of the terrain is available. The global planner gives information about obstacles and environments contained in the map, the position of robot and targets in the world. It creates a global path to reach the target position. Based on how they model the environment and search strategies for an optimal path, global path planning algorithms divide into Graph-based, Sampling-based, Nature-Inspired Algorithms, Reinforcement learning (RL)-based algorithms and Hybrid [35].

1. Graph-Based algorithms. These algorithms model the environment as a graph and find the shortest path using various search techniques. Sample algorithms:
 - Dijkstra: Dijkstra's algorithm is designed to determine the minimum-cost path from a source vertex to all other vertices in a directed graph. The algorithm operates by iteratively selecting the closest unvisited node, updating the shortest known distances to its neighboring vertices and continuing this process until the destination node is reached or all reachable nodes have been explored. Since Dijkstra's algorithm follows a breadth-first search-like approach, it systematically expands the search from the starting node outward. However, this leads to relatively high time and space complexity, especially in large graphs, as it requires maintaining and updating distance information for multiple vertices throughout the execution [36];
 - A and A* algorithms: A algorithm, often linked to Dijkstra's approach, identifies the shortest path between nodes in a weighted graph by evaluating all possible routes based on cumulative cost. However, it lacks heuristic guidance, which can make it inefficient in larger environments. The A* algorithm improves upon this by integrating a heuristic function with cost-based search, allowing it to prioritize more promising paths. It evaluates the total estimated cost from the start node and estimates the remaining cost to the goal [37];
 - D* and D* lite: The D* algorithm is an incremental path-planning method designed for dynamic environments, where obstacles or terrain conditions may change over time. It initially computes an optimal path from the start position to the goal under the assumption

of a static environment. When the robot moves and encounters updated information, D* efficiently recalculates only the affected portions of the path instead of recomputing from scratch, making it suitable for real-time applications in navigation and robotics. D* Lite, a simplified version of D*, follows a similar approach but computes paths in reverse, from the goal to the start. This backward search allows for efficient path updates when obstacles appear or disappear. By maintaining a cost-efficient priority queue and selectively updating affected nodes, D* Lite reduces computational complexity while preserving optimality [38].

2. **Sampling-Based Algorithms.** These algorithms generate a path by randomly sampling points in the environment. Sample algorithms:
 - **Rapidly-exploring Random Tree (RRT).** It is designed to efficiently navigate complex, high-dimensional spaces. It incrementally builds a tree by randomly sampling points within the search space and connecting them to the nearest existing node, effectively exploring feasible paths in environments with numerous obstacles. While RRT is proficient at quickly finding a viable path, it doesn't guarantee optimality [39]. To address this limitation, RRT* was developed as an extension of the original algorithm. RRT* enhances the path quality by incorporating a process of iterative refinement, where it rewires the tree structure to explore shorter or more efficient paths as new samples are added. This approach guarantees that, under sufficient conditions and with an adequate number of samples, RRT* will reach an optimal solution. It achieves this by balancing the rapid exploration capabilities of RRT with optimality [40];
 - **Probabilistic Roadmap (PRM).** This method efficiently navigates high-dimensional spaces. It works in two phases: first, the construction phase generates random collision-free nodes and connects them to form a roadmap, and second, the query phase links the start and goal positions to the roadmap, using graph search algorithms like Dijkstra to determine the optimal path. PRM is particularly effective in static environments where multiple queries are needed, as the roadmap can be reused, reducing computational complexity [41].
3. **Nature-Inspired Algorithms.** These algorithms are modelled after biological processes and use mathematical optimization techniques to find the best path. Sample algorithms:
 - **Genetic Algorithm (GA)** is an evolutionary optimization technique inspired by the principles of natural selection. It begins with a randomly generated population of candidate solutions, which evolve over multiple iterations through selection, crossover and mutation. A fitness function evaluates each candidate, favoring the best solutions for reproduction. The algorithm applies crossover to combine parent solutions and mutation to introduce small variations, ensuring diversity in the search space. This iterative process continues until an optimal or near-optimal solution is found. GA is widely used in complex optimization problems where traditional methods struggle with large or nonlinear search spaces [42];
 - **Particle Swarm Optimization (PSO)** algorithm is a population-based optimization technique inspired by the collective movement of birds and fish. It initializes a group of candidate solutions, called particles, which explore the search space by updating their positions based on both their individual best-known solution and the globally best-known solution within the swarm. This dynamic adjustment allows particles to converge toward optimal solutions over multiple iterations. PSO is particularly useful for solving high-dimensional and nonlinear optimization problems due to its simplicity and efficiency in navigating complex search spaces [43];
 - **Ant Colony Optimization (ACO).** It mimics the foraging behavior of ants to find optimal paths in a graph. Artificial ants explore potential routes, depositing virtual pheromones that influence the decisions of subsequent ants. Over successive iterations, paths with stronger pheromone trails become more favorable, leading the swarm toward efficient solutions [44].

4. Reinforcement learning (RL)-based algorithms. RL-based methods learn optimal paths through trial-and-error interactions with the environment, refining their policies over time based on cumulative rewards and don't require a predefined model [45]. Sample algorithms:
 - Q-Learning algorithm. It builds a Q-table where each state-action pair is assigned a value, updated iteratively using the Bellman equation. By maximizing cumulative rewards, the algorithm ensures efficient obstacle avoidance and goal-reaching, making it particularly useful in dynamic or uncertain environments [46];
 - Deep Q-Networks (DQN) employs a Deep Neural Network (DNN) to approximate Q-values for state-action pairs, enabling the robot to learn an optimal policy through continuous interaction with the environment. This approach is particularly beneficial in dynamic, uncertain, or partially observable settings where traditional methods struggle. Additionally, experience replay and target networks stabilize training, mitigating the overestimation of Q-values and improving convergence [47].
5. Hybrid Algorithms. These algorithms combine two or more approaches to improve efficiency and robustness. Sample combinations: A-RRT*, D-lite with RRT* [48].

4.2. Local Path Planning Algorithms

Local path planning algorithms are crucial for UGVs, especially in dynamic environments. The global path planner determines the overall trajectory for vehicle motion but lacks efficient real-time obstacle avoidance. Direct execution of the global path often results in suboptimal performance. One limitation is the reduced robustness of global planners, particularly in handling dynamic obstacles and adapting to inaccuracies in environmental mapping. Additionally, as the search space expands, the computational time of global path planning increases significantly [49]. Local planning adapts planned path based on real-time data, ensuring that the robot can navigate effectively while considering its physical limitations and avoiding obstacles. They work within the framework established by the global planner but have a more dynamic role that involves constant adjustments rather than a strictly defined trajectory generation.

- Dynamic Window Approach (DWA) is a real-time motion planning algorithm that ensures both collision avoidance and adherence to dynamic constraints. It evaluates a range of possible velocities within a short time horizon and selects the trajectory that optimally balances safety, efficiency and goal direction. By considering the kinematic limitations of the robot and environmental obstacles, DWA enables smooth and reactive navigation in dynamic environments [50];
- Artificial Potential Field (APF). In this approach, the robot is influenced by an artificial force field composed of attractive forces pulling it toward the goal and repulsive forces pushing it away from obstacles. The robot navigates by following the resultant force vector, aiming for a collision-free path to the target. While APF is computationally efficient and straightforward to implement, it can encounter issues such as local minima, where the robot becomes trapped in a position that is not the goal. Various modifications have been proposed to address these limitations and enhance the effectiveness of this method [51];
- Vector Field Histogram (VFH) constructs a two-dimensional Cartesian histogram grid using range sensor data. This grid is continuously updated to reflect the environment of the robot. The algorithm reduces this grid to a one-dimensional polar histogram centered on the current position of the robot, where each sector represents the obstacle density in a specific direction. By analyzing these sectors, VFH identifies obstacle-free paths and determines the most suitable steering direction, allowing the robot to navigate toward its target while avoiding collisions. This method effectively balances reactive obstacle avoidance with goal-oriented navigation [52];
- Model Predictive Control (MPC) is an advanced control strategy that utilizes a dynamic model of a system to predict and optimize its future behavior over a specified time horizon. In the context of mobile robot local path planning, MPC involves formulating an optimization problem

that accounts for the kinematic constraints of the robot, environmental obstacles and a predefined cost function. At each time step, the controller solves this optimization problem to determine the optimal control inputs, resulting in a trajectory that guides the robot toward its target while avoiding collisions. This process is repeated in a receding horizon manner, allowing the robot to adapt its path in real-time to dynamic changes in the environment. MPC’s ability to handle multivariable control problems and incorporate constraints makes it particularly effective for complex path planning tasks in uncertain environments [53];

- Timed Elastic Band (TEB) algorithm optimizes the trajectory of the robot by considering both spatial and temporal constraints. Starting with an initial path, TEB refines it into a time-parametrized trajectory by adjusting the positions and velocities of intermediate points, ensuring adherence to the kinematic constraints of the robot and obstacle avoidance. This optimization process allows the robot to navigate efficiently in dynamic environments, responding adaptively to changes while maintaining smooth and feasible motion [54].

Several algorithms, such as ACO, RRT*, A* and RL, can be utilized for both global and local path planning after appropriate optimization [55].

However, despite the existence of a wide range of path-planning algorithms, their practical implementation often reveals specific limitations in meeting all operational requirements. These limitations are dependent on the testing environment and the characteristics of the UGV. As a result, many algorithms require optimization or fusion with other approaches to enhance efficiency, adaptability and robustness. Table 4 presents a detailed comparison of the advantages and limitations of each algorithm.

Table 4. Strengths and Weaknesses of algorithms.

Algorithms	Strength	Weakness
Dijkstra	Guarantees the shortest path; Works well in static environments.	Computationally intensive; Inefficiency in dynamic environment.
A*	Fast and Efficient; Guarantees optimal path.	Heuristic sensitivity; Slow in high-dimensional spaces.
D*	Efficient replanning; Handles dynamic obstacles well.	Higher memory usage; Slower with frequent changes.
RRT*	Efficient in high-dimensional spaces; Incremental path improvement.	Paths are not always smooth; Slower in narrow passages.
PRM	Reusable path for repeated queries; Works well in large, open spaces.	Paths are not always smooth; Struggles with dynamic obstacles.
GA	Efficient in high-dimensional spaces; Works well with noisy data.	Slow convergence; No guarantee of finding optimal path.
PSO	Simple to implement; Adaptive to dynamic environments.	Prone to local minima; Sensitive to parameter selection.
ACO	Good for multi-agent path planning; Works in dynamic environments.	Computationally intensive; Slower convergence.
Q-Learning	Works with Partial Information; Can Handle Complex Environments.	Higher memory usage; Slow convergence in large spaces.
DQN	Suitable for high-dimensional spaces; Works with Partial Observability.	Requires high computational power; Sensitive to parameter selection.
DWA	Smooth and feasible paths; Computationally lightweight.	Prone to local minima; Struggles with complex environment.
APF	Lightweight, fast obstacle avoidance; Simple, best for reactive navigation.	Prone to local minima; Struggles with dynamic obstacles.
VFH	Efficient real-time obstacle avoidance; Handles noisy sensor data well.	Prone to local minima; Oscillations in Narrow Spaces.
MPC	Optimizes performance;	Computationally intensive;

TEB	Generates smooth and efficient paths.	Limited real-time application.
	Generates smooth, time-optimal path;	Computationally intensive;
	Dynamic Obstacle Avoidance.	Sensitive to parameter selection.

Global path planning algorithms differ significantly in efficiency and adaptability based on the environment and constraints. Graph-based methods such as A* and Dijkstra guarantee optimal paths but are computationally expensive, while sample-based approaches like RRT* and PRM offer better scalability in high-dimensional spaces but may require additional smoothing. Nature-Inspired Algorithms like GA and ACO are useful for solving complex optimization problems but may suffer from slower convergence. RL-techniques, including DQN, provide adaptability in dynamic and unknown environments but require computational resources.

Local path planning algorithms offer also common and distinct advantages and drawbacks. DWA, APF and VFH are prone to local minima, however they are good at real time obstacle avoidance. Moreover, DWA, MPC and TEB ensures smooth and optimized paths, but MPC and TEB demand high computation power. TEB is excellent for dynamic trajectory adjustments but requires fine-tuned parameters. In the field of practical robotics, a universal algorithm that is best suited for all applications is not yet available. Consequently, there is a preference for hybrid methods that integrate multiple classical or RL-based approaches, particularly in the context of efficient path planning. Alternatively, the existing algorithms may be modified through the implementation of optimization strategies [56].

In terms of using path planning algorithms in underground mine, the evaluation criteria as mentioned in requirement section will be selected based on key operational and environmental requirements, including the optimality of the path, computational demand, robustness and obstacle avoidance. Additionally, the variability in excavation methods influence the requirements for these algorithms.

5. Related Works on Underground Mine Path Planning with UGVs

The automation of underground mine vehicles has been the focus of research and industry for several decades. However, path planning for autonomous navigation remains a critical challenge and an active area of investigation. Despite the several studies have been conducted on the review of path planning algorithms on the ground [57], [58], a limited number of studies have successfully addressed optimal path planning for mining vehicles and robotic applications in underground environments.

Effective path planning allows UGVs to traverse complex, unstructured terrains while ensuring efficiency, safety and adaptability. Researchers have explored various techniques, ranging from classical graph-based methods to modern metaheuristic and RL-based approaches, each aiming to optimize path efficiency, obstacle avoidance and computational feasibility. These approaches have been validated in real underground mine, controlled indoor environment and virtual simulations such as MATLAB, Python, Gazebo. Table 5 reviews key contributions by researchers in the field, categorizing existing studies based on algorithmic approaches, testing strategies and demonstrated advantages in underground mining applications.

Table 5. Analysis of various path planning techniques for UGVs in underground mine.

Author	Basic		Optimization Strategy	Test field	Advantages	Application
	Path type	algorithm				
[59]	Global	A*	Gaussian filtering method; Quadratic programming method.	MATLAB	Safe path; Smoother.	LHD in mining transportation
[60]	Global	A*	Expanding nodes by articulation angle;	Indoor; C++.	Enhanced search efficiency;	LHD in mining transportation

			Collision threat cost.		Collision free.	
			Exponential function weighting;		Smoother path;	
[61]	Global Local	A* + APF	Cubic spline interpolation; Repulsion potential field correction factor.	MATLAB ; Indoor.	More robust; Guaranteed safety.	Transportation in coal mine
[62]	Global Local	A*+ DWA	Key node selection strategy; Clamped-B spline; Fuzzy control.	MATLAB ; Indoor.	Safety; Smoother path; Optimal path.	Vehicle in mine transportation
[63]	Global Local	A*+DWA	-	Indoor	Shortest path; Smoothest path.	Mobile robot in inspection
[64]	Global Local	A*+DWA	Floyd Algorithm; B-Spline curves.	MATLAB ; Indoor.	Local optima solution	Inspection robot in coal mine
[65]	Global	VFH-A*	Bezier interpolation;	MATLAB ; Mine.	More efficiency search; Smooth path; Fast calculation speed.	LHD in mining operation
[66]	Global Local	Dijkstra - ACO	MAKLINK lines. Shifting locally; Symmetric polynomial curve.	Mine	Smooth path	Robot in SAR
[67]	Global	Dijkstra- PSO	PSO-based optimization	MATLAB	Shorter path; Safety.	Mine mapping and inspection
[68]	Global	Dijkstra	3D Environment-based adaptation	MATLAB	Feasible path	Survey robot in mine exploration
[69]	Global	D*	Manhattan distances	MATLAB	Safer path; Reduced planning time and cost.	Downhole robot in mine detection
[70]	Global	RRT*	Vectorized map; Optimal tree reconnection.	Python	Shorter path; Smoother.	LHD in mining operations
[71]	Global	RRT*	Line corner	MATLAB ; Outdoor.	Smooth path; Better in narrow passages with tight turn;	Robotic excavator loader in excavation
[72]	Global	RRT- PRM	Fan-shaped goal orientation; Adaptive step size expansion; Third-order Bessel curve.	MATLAB ; Indoor.	Higher success rate; Smoother path; Shorter path length.	Inspection robot in Mining
[73]	Global	RRG ¹	Ray tracing method; Next-Best View.	Gazebo	Shorter path; Fast calculation.	Mobile Robot in Mine rescue
[74]	Global	ACO	Retreat-punishment strategy; Serial number and Cartesian coordinate methods.	MATLAB	Consumes less costs	Coal mine robot
[75]	Local	ACO	Membrane computing	Simulation	Faster convergence; Better robustness.	Mobile robot
[76]	Global	ACO	16-directional 24- neighbourhood ant search approach; Ant retreat strategy.	MATLAB	Improved search efficiency; U-shaped trap solution; Shorter path.	Rescue robot in Mine
[77]	Global	ACO	Annealing algorithm; Entropy increase strategy.	MATLAB	Strong in robustness; High convergence speed.	Mining robot in SAR
[78]	Global Local	ACO+TE B	Pheromone updating model;	MATLAB ; Indoor.	Enhanced search capabilities; Faster convergence.	Drilling robot in mining rockburst

[79]	Global	BFA ² -PSA	Tested by traveling salesman problem (TSP)	MATLAB	Fast convergence speed; Better Robustness	Rescue robot in coal mine
[80]	Global Local	AFSA ³ + DWA	Improved genetic algorithm Adaptive trajectory evaluation function	MATLAB ; Gazebo; Indoor.	Shorter path; Smoother path.	Patrol robot in mine safety inspection
[81]	Global	QLearnin g	Even gray model; Multi-attribute intelligent.	MATLAB	Smoother convergence; Shorter path;	UGV in SAR
[82]	Local	Skeleton- Based	Thinning algorithm	Mine	High robustness; Stable;	LHD in mining operation
[83]	Local	APF	Velocity and acceleration fields; Global potential field line; Genetic Trust Region Algorithm	MATLAB ; Indoor	Dynamic collision avoidance; Escape local minima.	Mobile robot in coal mine rescue

1. Rapidly Random Graph; 2. Bacterial Foraging algorithm; 3. Artificial fish swam algorithm.

A thorough analysis of the data presented in the table indicates that several underground mining operations are being targeted by UGV navigation systems. The majority of research in path planning focused on mine transportation, which is primarily carried out using LHD. Additionally, mine exploration and inspection tasks with mobile robots equipped with specialized sensors emerged as significant area of research interest. SAR is another critical topic for underground and numerous researchers are exploring optimal path planning algorithms for successful rescue operations after disasters. It is notable that coal mines are considered to be high-risk environments due to their potential for accidents.

Despite the limited research in this area, existing studies have employed all types of environment representation and search strategy models. The most prevalent techniques employed in underground mine environments include classical such as graph-based and sampling-based algorithms as illustrated in Figure 3a based on an analysis current existed research.

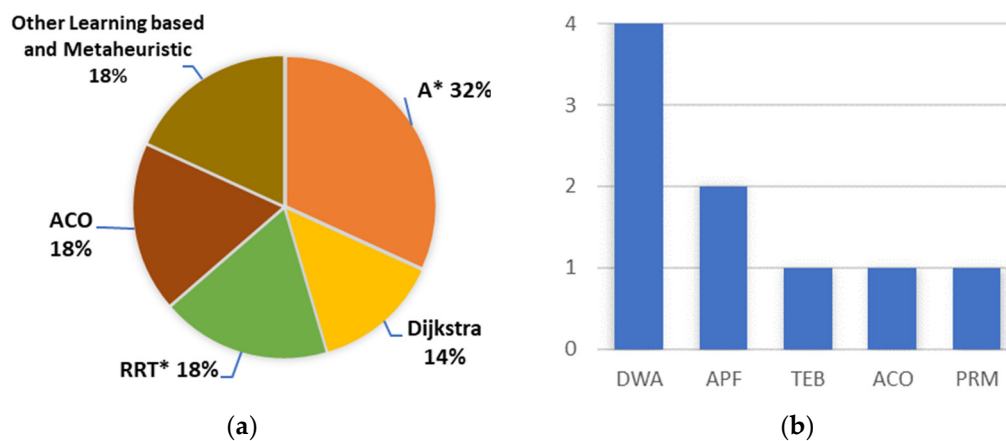


Figure 3. Percentage of existing path planning algorithms that are being investigated by researchers for the underground mining environment (a) Global path planning; (b) Local path planning.

It is illustrated that the A* algorithm is the most prevalent in the field, with approximately one in three studies utilizing this approach. This method incorporates a range of optimization strategies, including Gaussian filtering, quadratic programming, expanding and selection nodes, collision threat cost, exponential function weighting, spline and others. These techniques ensure safe, smooth and optimal paths for UGVs in mining transportation and inspection tasks [59–64]. Graph-based algorithms have also demonstrated a high level of efficiency in hybrid contexts. For example, the utilization of A* with VFH employing Bazier interpolation and Dijkstra with ACO and PSO

employing curve strategies have been shown to enhance search efficiency, smoothness and accelerate calculation in mining operations and SAR [65–67]. Dijkstra algorithm also demonstrated strong performance in path planning within 3D environments [68] and represents 14% of the analyzed studies.

Sampling-based methods such as RRT, has gained significant attention, accounting for 18% of all research in the field. Optimized versions incorporating vectorized map, optimal tree reconnection, and line corner for narrow passages have demonstrated superior path smoothness and shorter path lengths in operating mining transports [70,71]. A hybrid RRT* - PRM model integrating fan-shaped goal orientation, adaptive step-size expansion, and Bessel curve has been shown to enhance success rates and smoothness for mining inspection robots [72]. Similarly, the incorporation of ray-tracing and next-best-view methods into the RRG has been applied to mine rescue operations, achieving shorter path generation and faster computational speed [73].

Nature-inspired approach, such as ACO, has been explored at a similar rate to RRT*, approximately 18%, for underground mine navigation and rescue applications. Studies have employed retreat-punishment mechanisms, serial numbering and Cartesian coordinates, along with a 16-directional, 24-neighborhood ACO approach, the ant retreat strategy, annealing algorithms, pheromone updating models and entropy-increasing strategies to reduce consumption costs, overcome U-shaped obstacle traps and enhance convergence speed, efficiency and robustness [74,76–78].

The last 18% of the analyzed research is primarily divided between various types of nature-inspired and RL-based algorithms. For example, the hybrid approach BFA combined with PSA tested by traveling salesman problem and AFSA with improved genetic algorithm demonstrated fast convergence, better robustness and shorter path in SAR applications [79,80]. Additionally, RL algorithm, such as Q-learning combined with the Even Gray Model and Multi-Attribute Intelligent Decision-Making, demonstrated smoother convergence and high robustness in SAR operation [81].

Unlike global path planning, local path planning involves fewer algorithms. As illustrated in Figure 3b, the DWA is the most favorable method for adjusting the kinematics of UGVs. In the second place stands APF that could be modified with global potential field line and genetic Trust Region strategies to overcome local minima and dynamic collision in coal mine rescue tasks [83]. Moreover, membrane computing-based ACO enhances robustness and convergence speed [75], while skeleton-based thinning algorithms provide high stability and robustness [81]. Additionally, other applications of the TEB and PRM methods were utilized for local path planning.

The test environments used in the reviewed studies, highlight the progression from simulation-based validation to real-world implementation. Many algorithms have been tested in MATLAB and Python simulations, providing controlled conditions for evaluating path efficiency, convergence and obstacle avoidance. However, some studies extend their validation to indoor lab environments, where methods have been tested in structured, tunnel-like settings, enabling real-time sensor fusion and navigation performance analysis. The most reliable assessments come from real underground mine experiments. Only few algorithms have been deployed to evaluate practical feasibility, adaptability to real-world constraints and system robustness.

6. Discussion

The analysis of various path-planning algorithms for underground mine UGVs highlights significant trends and challenges in achieving efficient, safe and adaptive autonomous navigation. Most existing research is focused on transportation, inspection and SAR operations in mine. The comparison of graph-based, sampling-based, Nature-inspired, RL-based and Local path planning approaches demonstrates that each method has unique advantages and limitations depending on the operational environment and constraints.

Graph-based approaches such as A* and Dijkstra remain widely used due to their deterministic nature and ability to find optimal paths. However, these algorithms often struggle with real-time

adaptability, particularly in dynamic underground environments where sudden obstacles and unpredictable conditions are common. To overcome these drawbacks especially for A* Collision threat cost and Repulsion potential field correction factor strategies are used. Furthermore, A* typically produces paths that can be jagged or not smooth, especially in complex environments. Techniques like cubic spline interpolation and B-spline curves provide a way to refine paths.

The most commonly used sampling-based method, such as RRT, is effective in exploration tasks and offers better adaptability in large-scale environments. However, it can suffer from path irregularities and challenges in collision detection. To address these issues, several strategies, including vectorized maps, line corner handling and third-order Bessel curves proposed.

Nature-inspired approaches, particularly ACO, improve search efficiency, robustness and obstacle avoidance but require higher computational resources. To mitigate this problem, the 16-Directional 24-Neighbourhood Ant Search Approach and the Pheromone Updating Model were used.

RL-based approaches, such as grey QL, have demonstrated potential in adaptive decision-making, but their practical implementation is still limited by computational complexity and training data requirements.

Among local path planning algorithms, DWA has been identified as the most favorable. To address its limitations, the Adaptive Trajectory Evaluation Function and Fuzzy Control were employed.

Analyzing the current research, it is notable that some hybrid algorithms also demonstrated effective and optimal path planning, such as VFH-A*, Dijkstra-ACO, Dijkstra-PSO, RRT-PRM, BFA-PSA.

Several limitations persist in analyzing the existing research studies on the selected topic. (1) An Over-reliance on simulation environments which do not fully replicate real-world underground conditions. While indoor lab testing improves sensor integration and navigation accuracy, only a limited number of studies have been conducted in actual underground mines. Additionally, many path-planning algorithms do not account for real-time constraints, such as sensor delays, terrain variations and power limitations, which are critical for full-scale deployment in industrial mining operations. (2) Some algorithms may be overly complex, making them difficult to implement or adapt for real-world scenarios. Especially, hybrid, nature-inspired and RL algorithms can exhibit complexities. Simplicity and interpretability are often important for practical applications.

Future research should focus on bridging the gap between simulation and real-world implementation of several successful approaches modified with optimization strategies by conducting more large-scale real-mine experiments.

7. Conclusions

The automation of underground mining operations has advanced significantly in recent years, with UGVs playing a crucial role in improving efficiency, safety and productivity. Unlike outdoor and structured indoor environments, underground mines present unique challenges, including complex tunnel networks, dynamic obstacles, low visibility and unstable terrain. To navigate in these environment, autonomous mining vehicles rely on core navigation components such as mapping, SLAM and path planning, all of which must be tailored to the constraints of underground operations. UGVs are deployed in exploration, monitoring structural changes, production-related tasks, SAR tasks and inspection requiring robust navigation systems to ensure operational reliability.

Path planning in underground mines presents distinct challenges due to GPS-denied, unpredictable environmental changes and the need for efficient real-time decision-making. The requirement analysis in this study highlights the need for adaptive, efficient and safe path-planning strategies that can be implemented in UGVs. To address these challenges, ROS serves as a widely adopted implementation environment, offering flexibility, modular communication and support for multi-sensor integration. The integration of various sensors such as LiDAR, IMU and cameras, along

with ROS-based communication packages, enhances real-time perception and navigation capabilities.

The review and analysis of path-planning algorithms in this study involve several global and local path-planning techniques, including traditional graph-based and sampling-based approaches, as well as nature-inspired and reinforcement learning (RL)-based algorithms. For global path planning A*, Dijkstra, RRT* and ACO and for local path planning DWA remain widely used in underground mine application with UGVs. Several critical factors such as robustness, calculation speed, smoothness, length of the path is enhanced with optimization strategies.

The research limitation identified in the analysis of path planning techniques in underground mine with UGVs indicates the need for real-world deployment of several algorithms. Many algorithms have been validated in simulation environments only a limited number have been tested in real underground mines.

Author Contributions: Conceptualization, A.A. and J.B.; Methodology, A.A.; Formal Analysis, A.A.; Investigation, A.A.; Resources, J.B.; Data Curation, A.A.; Writing – Original Draft Preparation, A.A.; Writing – Review & Editing, J.B.; Visualization, A.A.; Supervision, J.B.; Project Administration, J.B.; Funding Acquisition, J.B.

Funding: This research was funded by the German Academic Scholarship Foundation, the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project number 422117092 using the research infrastructure INST 267/165-1 FUGG and by “El-yurt umidi” Foundation for training specialists abroad and the dialogue with compatriots – Certificate number DR-2022-0115.

References

1. A. Kokkinis: T. Frantzis, K. Skordis, G. Nikolakopoulos, and P. Koustoumpardis, “Review of Automated Operations in Drilling and Mining,” *Machines* 2024, Vol. 12, Page 845, vol. 12, no. 12, p. 845, Nov. 2024, doi: 10.3390/MACHINES12120845.
2. J. H. Saleh and A. M. Cummings, “Safety in the mining industry and the unfinished legacy of mining accidents: Safety levers and defense-in-depth for addressing mining hazards,” *Saf Sci*, vol. 49, no. 6, pp. 764–777, Jul. 2011, doi: 10.1016/j.ssci.2011.02.017.
3. J. Li and K. Zhan, “Intelligent Mining Technology for an Underground Metal Mine Based on Unmanned Equipment,” *Engineering*, vol. 4, no. 3, pp. 381–391, Jun. 2018, doi: 10.1016/j.eng.2018.05.013.
4. “Navigation: Principles of Positioning and Guidance - B. Hofmann-Wellenhof, K. Legat, M. Wieser - Google Books.” Accessed: Feb. 26, 2025. [Online]. Available: https://books.google.de/books?hl=en&lr=&id=dMXcBQAAQBAJ&oi=fnd&pg=PR21&dq=Autonomous+N+avigation+book&ots=uo1iWq71W5&sig=TTbM49kTej2P8pRFIHhZ9H-9v1w&redir_esc=y#v=onepage&q&f=false
5. J. M. Roberts, E. S. Duff, and P. Corke, “Reactive navigation and opportunistic localization for autonomous underground mining vehicles,” *Inf. Sci.*, vol. 145, pp. 127–146, 2002, [Online]. Available: <https://api.semanticscholar.org/CorpusID:1925516>
6. K. Ebadi et al., “Present and Future of SLAM in Extreme Underground Environments,” *ArXiv*, vol. abs/2208.01787, 2022, [Online]. Available: <https://api.semanticscholar.org/CorpusID:251280074>
7. L. Ojeda and J. Borenstein, “Personal Dead-reckoning System for GPS-denied Environments,” in 2007 IEEE International Workshop on Safety, Security and Rescue Robotics, IEEE, Sep. 2007, pp. 1–6. doi: 10.1109/SSRR.2007.4381271.
8. Li, J., Benndorf, J. & Trybała, P. Quantitative analysis of different SLAM algorithms for geo-monitoring in an underground test field. *Int J Coal Sci Technol* 12, 7 (2025). <https://doi.org/10.1007/s40789-025-00745-w>
9. H. Mäkelä, “Overview of LHD navigation without artificial beacons,” *Rob Auton Syst*, vol. 36, no. 1, pp. 21–35, Jul. 2001, doi: 10.1016/S0921-8890(01)00115-4.
10. Y. Li, P. Peng, H. Li, J. Xie, L. Liu, and J. Xiao, “Drilling Path Planning of Rock-Drilling Jumbo Using a Vehicle-Mounted 3D Scanner,” 2023, doi: 10.3390/su15129737.

11. H. Kim and Y. Choi, "Development of Autonomous Driving Patrol Robot for Improving Underground Mine Safety," *Applied Sciences*, vol. 13, no. 6, p. 3717, Mar. 2023, doi: 10.3390/app13063717.
12. W. Wang, W. Dong, Y. Su, D. Wu, and Z. Du, "Development of Search-and-rescue Robots for Underground Coal Mine Applications," *J Field Robot*, vol. 31, no. 3, pp. 386–407, May 2014, doi: 10.1002/rob.21501.
13. "Indoor Wayfinding and Navigation - Google Books." Accessed: Feb. 28, 2025. [Online]. Available: https://books.google.de/books?hl=en&lr=&id=I3oZBwAAQBAJ&oi=fnd&pg=PP1&dq=Underground+navigation+&ots=yAGTE7jg8&sig=5ZnjDg4hnXcrh6Z-qbov4UGJwyk&redir_esc=y#v=onepage&q=Underground%20navigation&f=false
14. D. Galar, U. Kumar, and D. Seneviratne, *Robots, Drones, UAVs and UGVs for Operation and Maintenance*. CRC Press, 2020. doi: 10.1201/9780429452260.
15. F. Inostroza, I. Parra-Tsunekawa, and J. Ruiz-del-Solar, "Robust Localization for Underground Mining Vehicles: An Application in a Room and Pillar Mine," *Sensors*, vol. 23, no. 19, p. 8059, Sep. 2023, doi: 10.3390/s23198059.
16. L. Liu, X. Wang, X. Yang, H. Liu, J. Li, and P. Wang, "Path planning techniques for mobile robots: Review and prospect," *Expert Syst Appl*, vol. 227, p. 120254, Oct. 2023, doi: 10.1016/j.eswa.2023.120254.
17. H. Biggie et al., "Flexible Supervised Autonomy for Exploration in Subterranean Environments," *Field Robotics*, vol. 3, pp. 125–189, Jan. 2023, doi: 10.55417/fr.2023004.
18. "Control Theoretic Splines: Optimal Control, Statistics, and Path Planning - Magnus Egerstedt, Clyde Martin - Google Books." Accessed: Mar. 06, 2025. [Online]. Available: <https://books.google.de/books?hl=en&lr=&id=crNiMAsPex8C&oi=fnd&pg=PP1&dq=Control+theoretic+splines+:+optimal+control,+statistics,+and+path+planning&ots=HPIQKvIHol&sig=zMPZKFWJX-OrCBIJwaO6WDE9yNs#v=onepage&q=Control%20theoretic%20splines%20%3A%20optimal%20control%2C%20statistics%2C%20and%20path%20planning&f=false>
19. P. Trybała et al., "MIN3D Dataset: Multi-sensor 3D Mapping with an Unmanned Ground Vehicle," *PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, vol. 91, no. 6, pp. 425–442, Dec. 2023, doi: 10.1007/s41064-023-00260-0.
20. P. Peng, J. Pan, Z. Zhao, M. Xi, and L. Chen, "A Novel Obstacle Detection Method in Underground Mines Based on 3D LiDAR," *IEEE Access*, vol. 12, pp. 106685–106694, 2024, doi: 10.1109/ACCESS.2024.3437784.
21. K. Nielsen, *Robust LIDAR-Based Localization in Underground Mines*, vol. 1906. Linköping: Linköping University Electronic Press, 2021. doi: 10.3384/lic.diva-175001.
22. N.-B. Jing, X. Ma, W. Guo, and M. Wang, "3D Reconstruction of Underground Tunnel Using Depth-camera-based Inspection Robot," *Sensors and Materials*, 2019, [Online]. Available: <https://api.semanticscholar.org/CorpusID:203068830>
23. F. Cunha and K. Youcef-Toumi, "Ultra-Wideband Radar for Robust Inspection Drone in Underground Coal Mines," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, May 2018, pp. 86–92. doi: 10.1109/ICRA.2018.8461191.
24. N. Ahmad, A. R. Ghazilla, N. M. Khairi, and V. Kasi, "Reviews on Various Inertial Measurement Unit (IMU) Sensor Applications", doi: 10.12720/ijsp.1.2.256-262.
25. C. M. Rodrigues, "Ground Mobile Vehicle Velocity Control using Encoders and Optical Flow Sensor Fusion," 2016. [Online]. Available: <https://api.semanticscholar.org/CorpusID:202662013>
26. K.-Q. Liu, S.-S. Zhong, K. Zhao, and Y. Song, "Motion control and positioning system of multi-sensor tunnel defect inspection robot: from methodology to application," *123AD*, doi: 10.1038/s41598-023-27402-z.
27. M. Quigley, "ROS: an open-source Robot Operating System," in *IEEE International Conference on Robotics and Automation*, 2009. [Online]. Available: <https://api.semanticscholar.org/CorpusID:6324125>
28. M. Aljamal, S. Patel, and A. Mahmood, "Comprehensive Review of Robotics Operating System-Based Reinforcement Learning in Robotics," *Applied Sciences*, vol. 15, no. 4, p. 1840, Feb. 2025, doi: 10.3390/app15041840.
29. A. S. Al-Batati, A. Koubaa, and M. Abdelkader, "ROS 2 in a Nutshell: A Survey," Oct. 16, 2024. doi: 10.20944/preprints202410.1204.v2.
30. "Mastering ROS for Robotics Programming: Design, build, and simulate complex ... - Lentin Joseph, Jonathan Cacace - Google Books." Accessed: Mar. 09, 2025. [Online]. Available:

[https://books.google.de/books?hl=en&lr=&id=MulODwAAQBAJ&oi=fnd&pg=PP1&dq=ROS+\(Robot+operation+system\)+evolution+and+versions&ots=Cmk4PZqYpN&sig=h0r8m5qs0X8cTNqax9UzVsi8nf8&redir_esc=y#v=onepage&q=ROS%20\(Robot%20operation%20system\)%20evolution%20and%20versions&f=false](https://books.google.de/books?hl=en&lr=&id=MulODwAAQBAJ&oi=fnd&pg=PP1&dq=ROS+(Robot+operation+system)+evolution+and+versions&ots=Cmk4PZqYpN&sig=h0r8m5qs0X8cTNqax9UzVsi8nf8&redir_esc=y#v=onepage&q=ROS%20(Robot%20operation%20system)%20evolution%20and%20versions&f=false)

31. A. Agha et al., "NeBula: Quest for Robotic Autonomy in Challenging Environments; TEAM CoSTAR at the DARPA Subterranean Challenge," Mar. 2021.
32. R. Losch, S. Grehl, M. Donner, C. Buhl, and B. Jung, "Design of an Autonomous Robot for Mapping, Navigation, and Manipulation in Underground Mines," in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, Oct. 2018, pp. 1407–1412. doi: 10.1109/IROS.2018.8594190.
33. S. Thrun, "Learning metric-topological maps for indoor mobile robot navigation," *Artif Intell*, vol. 99, no. 1, pp. 21–71, Feb. 1998, doi: 10.1016/S0004-3702(97)00078-7.
34. A. Gil, Ó. Reinoso, A. Vicente, C. Fernández, and L. Payá, "Monte Carlo Localization Using SIFT Features," 2005, pp. 623–630. doi: 10.1007/11492429_75.
35. L. Yang et al., "Path Planning Technique for Mobile Robots: A Review," *Machines*, vol. 11, no. 10, p. 980, Oct. 2023, doi: 10.3390/machines11100980.
36. M. A. Javaid, "Understanding Dijkstra Algorithm," *SSRN Electronic Journal*, 2013, doi: 10.2139/ssrn.2340905.
37. P. E. Hart, N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968, doi: 10.1109/TSSC.1968.300136.
38. S. Koenig and M. Likhachev, "D*lite," *AAAI/IAAI*, 2002.
39. S. M. LaValle, "Rapidly-exploring random trees: a new tool for path planning," *The annual research report*, 1998, [Online]. Available: <https://api.semanticscholar.org/CorpusID:14744621>
40. S. Karaman and E. Frazzoli, "Optimal kinodynamic motion planning using incremental sampling-based methods," in 49th IEEE Conference on Decision and Control (CDC), IEEE, Dec. 2010, pp. 7681–7687. doi: 10.1109/CDC.2010.5717430.
41. L. E. Kavraki, P. Švestka, J. C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robotics Autom.*, vol. 12, no. 4, pp. 566–580, 1996, doi: 10.1109/70.508439.
42. "Genetic algorithms in search, optimization, and machine learning: Goldberg, David E. (David Edward), 1953-: Free Download, Borrow, and Streaming: Internet Archive." Accessed: Feb. 02, 2025. [Online]. Available: <https://archive.org/details/geneticalgorithm0000gold>
43. J. Kennedy and R. Eberhart, "Particle Swarm Optimization-Neural Networks, 1995. Proceedings., IEEE International Conference on," 2004.
44. M. Dorigo and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53–66, 1997, doi: 10.1109/4235.585892.
45. R. S. Sutton and A. G. Barto, "Reinforcement Learning: An Introduction Second edition, in progress".
46. C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning* 1992 8:3, vol. 8, no. 3, pp. 279–292, May 1992, doi: 10.1007/BF00992698.
47. V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature* 2015 518:7540, vol. 518, no. 7540, pp. 529–533, Feb. 2015, doi: 10.1038/nature14236.
48. S. Al-Ansarry and S. Al-Darraj, "Hybrid RRT-A*: An Improved Path Planning Method for an Autonomous Mobile Robots," *Journal of Electrical and Electronic Engineering*, vol. 17, pp. 1–9, 2021, [Online]. Available: <https://api.semanticscholar.org/CorpusID:236371326>
49. D. V. Lu, D. Hershberger, and W. D. Smart, "Layered costmaps for context-sensitive navigation," *IEEE International Conference on Intelligent Robots and Systems*, pp. 709–715, Oct. 2014, doi: 10.1109/IROS.2014.6942636.
50. D. Fox, W. Burgard, and S. Thrun, "The Dynamic Window A pproach t o C o llision Avoidance".
51. O. Khatib, "REAL-TIME OBSTACLE AVOIDANCE FOR MANIPULATORS AND MOBILE ROBOTS.," *International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986, doi: 10.1177/027836498600500106.

52. J. Borenstein, Y. Koren, and S. Member, "THE VECTOR FIELD HISTOGRAM-FAST OBSTACLE AVOIDANCE FOR MOBILE ROBOTS," *IEEE Journal of Robotics and Automation*, vol. 7, no. 3, pp. 278–288, 1991.
53. J. Li, J. Sun, L. Liu, and J. Xu, "Model predictive control for the tracking of autonomous mobile robot combined with a local path planning," *Measurement and Control*, vol. 54, no. 9–10, pp. 1319–1325, Nov. 2021, doi: 10.1177/00202940211043070.
54. Z. Yongzhe, B. Ma, and C. K. Wai, "A Practical Study of Time-Elastic-Band Planning Method for Driverless Vehicle for Auto-parking," 2018 International Conference on Intelligent Autonomous Systems, ICoIAS 2018, pp. 196–200, Oct. 2018, doi: 10.1109/ICoIAS.2018.8494025.
55. Z. Chang, Y. Wang, Y. Cai, S. Li, and F. Gao, "Fusion of improved RRT and ant colony optimization for robot path planning," *Engineering Research Express*, vol. 6, no. 4, p. 045247, Dec. 2024, doi: 10.1088/2631-8695/AD9548.
56. V. D. Sharma, J. Lee, M. Andrews, and I. H. Hadžić, "Hybrid Classical/RL Local Planner for Ground Robot Navigation," Oct. 2024, Accessed: Feb. 02, 2025. [Online]. Available: <http://arxiv.org/abs/2410.03066>
57. L. Liu, X. Wang, X. Yang, H. Liu, J. Li, and P. Wang, "Path planning techniques for mobile robots: Review and prospect," *Expert Syst Appl*, vol. 227, p. 120254, Oct. 2023, doi: 10.1016/j.eswa.2023.120254.
58. L. Yang et al., "Path Planning Technique for Mobile Robots: A Review," *Machines*, vol. 11, no. 10, p. 980, Oct. 2023, doi: 10.3390/machines11100980.
59. T. Liu, Z. Huang, C. Wang, Z. An, X. Meng, and J. Zheng, "Path planning of underground mining area transportation scene based on improved A* algorithm," in 2023 7th International Conference on Transportation Information and Safety (ICTIS), IEEE, Aug. 2023, pp. 2265–2270. doi: 10.1109/ICTIS60134.2023.10243887.
60. Q. Yulong, M. Qingyong, and T. Xu, "Research on Navigation Path Planning for An Underground Load Haul Dump," *Journal of Engineering Science and Technology Review*, vol. 8, pp. 102–109, 2015, [Online]. Available: <https://api.semanticscholar.org/CorpusID:116378442>
61. B. A. O. Jiu-sheng et al., "Underground driverless path planning of trackless rubber tyred vehicle based on improved A* and artificial potential field algorithm," *Journal of China Coal Society*, vol. 47, no. 3, pp. 1347–1360, 2022, [Online]. Available: <https://www.mtxb.com.cn/en/article/id/d7bec14d-e1ed-4d3d-9990-eb127b3e130c>
62. C. Zhang, X. Yang, R. Zhou, and Z. Guo, "A Path Planning Method Based on Improved A* and Fuzzy Control DWA of Underground Mine Vehicles," *Applied Sciences* 2024, Vol. 14, Page 3103, vol. 14, no. 7, p. 3103, Apr. 2024, doi: 10.3390/AP14073103.
63. D. Zhu, Y. Zhang, J. Wang, K. Ren, and K. Yang, "Evaluation of Motion Planning Algorithms for Underground Mobile Robots," 2022, pp. 368–379. doi: 10.1007/978-981-19-8915-5_32.
64. C. Gu, S. Liu, H. Li, K. Yuan, and W. Bao, "Research on hybrid path planning of underground degraded environment inspection robot based on improved A* algorithm and DWA algorithm," *Robotica*, pp. 1–22, Jan. 2025, doi: 10.1017/S0263574725000037.
65. B. Zhang, Y. Zhang, C. Zhang, J. Cheng, G. Shen, and X. Wang, "Path planning for unmanned load-haul-dump machines based on a VHF_A* algorithm," *Proc Inst Mech Eng C J Mech Eng Sci*, vol. 238, no. 13, pp. 6584–6597, Jul. 2024, doi: 10.1177/09544062241228391.
66. X. Ma and R. Mao, "Path planning for coal mine robot to avoid obstacle in gas distribution area," *Int J Adv Robot Syst*, vol. 15, no. 1, Jan. 2018, doi: 10.1177/1729881417751505.
67. R. Xu and S. Yao, "Research on UGV Path Planning in Tunnel Based on the Dijkstra*-PSO* Algorithm," in 2022 6th Asian Conference on Artificial Intelligence Technology (ACAIT), IEEE, Dec. 2022, pp. 1–9. doi: 10.1109/ACAIT56212.2022.10137853.
68. Z. Chen, "Path Planning Method for Underground Survey Robot in Dangerous Scenarios," *Highlights in Science, Engineering and Technology*, vol. 43, pp. 207–214, Apr. 2023, doi: 10.54097/hset.v43i.7421.
69. T. Ma, J. Lv, and M. Guo, "Downhole robot path planning based on improved D* algorithm," in 2020 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), IEEE, Aug. 2020, pp. 1–5. doi: 10.1109/ICSPCC50002.2020.9259532.

70. H. Wang, G. Li, J. Hou, L. Chen, and N. Hu, "A Path Planning Method for Underground Intelligent Vehicles Based on an Improved RRT* Algorithm," *Electronics (Basel)*, vol. 11, no. 3, p. 294, Jan. 2022, doi: 10.3390/electronics11030294.
71. B. Hopfenblatt and N. Axel., "A combined RRT*-optimal control approach for kinodynamic motion planning for mobile robots," 2016. [Online]. Available: <https://api.semanticscholar.org/CorpusID:203702248>
72. J. Wu, L. Zhao, and R. Liu, "Research on Path Planning of a Mining Inspection Robot in an Unstructured Environment Based on an Improved Rapidly Exploring Random Tree Algorithm," *Applied Sciences* 2024, Vol. 14, Page 6389, vol. 14, no. 14, p. 6389, Jul. 2024, doi: 10.3390/APP14146389.
73. M. Steinbrink, P. Koch, B. Jung, and S. May, "Rapidly-Exploring Random Graph Next-Best View Exploration for Ground Vehicles," Aug. 2021, Accessed: Jan. 28, 2025. [Online]. Available: <http://arxiv.org/abs/2108.01012>
74. B. Song, H. Miao, and L. Xu, "Path planning for coal mine robot via improved ant colony optimization algorithm," *Systems Science & Control Engineering*, vol. 9, no. 1, pp. 283–289, Jan. 2021, doi: 10.1080/21642583.2021.1901158.
75. J. C. Xu, Y. R. Huang, and G. Y. Xu, "A path optimization algorithm for the mobile robot of coal mine based on ant colony membrane algorithm," *ACM International Conference Proceeding Series*, Dec. 2018, doi: 10.1145/3148453.3306267.
76. X. Shao, G. Wang, R. Zheng, B. Wang, T. Yang, and S. Liu, "Path Planning for Mine Rescue Robots Based on Improved Ant Colony Algorithm," in *2022 8th International Conference on Control, Automation and Robotics (ICCAR)*, IEEE, Apr. 2022, pp. 161–166. doi: 10.1109/ICCAR55106.2022.9782608.
77. K. Liu and M. Zhang, "Path planning based on simulated annealing ant colony algorithm," *Proceedings - 2016 9th International Symposium on Computational Intelligence and Design, ISCID 2016*, vol. 2, pp. 461–466, Jul. 2016, doi: 10.1109/ISCID.2016.2114.
78. C. Fang et al., "Real-time path planning of drilling robot using global and local sensor information fusion: mining-oriented validation," *The International Journal of Advanced Manufacturing Technology*, vol. 135, no. 11–12, pp. 5875–5891, Dec. 2024, doi: 10.1007/s00170-024-14873-1.
79. P. Li and H. Zhu, "Immune Optimization of Path Planning for Coal Mine Rescue Robot," 2016. [Online]. Available: <https://api.semanticscholar.org/CorpusID:13507361>
80. Y. Gao, Z. Dai, and J. Yuan, "A Multiobjective Hybrid Optimization Algorithm for Path Planning of Coal Mine Patrol Robot," *Comput Intell Neurosci*, vol. 2022, pp. 1–10, Jun. 2022, doi: 10.1155/2022/9094572.
81. S. Zhang and Q. Zeng, "Online Unmanned Ground Vehicle Path Planning Based on Multi-Attribute Intelligent Reinforcement Learning for Mine Search and Rescue," *Applied Sciences* 2024, Vol. 14, Page 9127, vol. 14, no. 19, p. 9127, Oct. 2024, doi: 10.3390/APP14199127.
82. [Y. Jiang, P. Peng, L. Wang, J. Wang, J. Wu, and Y. Liu, "LiDAR-Based Local Path Planning Method for Reactive Navigation in Underground Mines," *Remote Sens (Basel)*, vol. 15, no. 2, p. 309, Jan. 2023, doi: 10.3390/rs15020309.
83. T. Zi-jian, G. A. O. Xue-hao, and Z. Meng-xia, "Path planning based on the improved artificial potential field of coal mine dynamic target navigation," *Journal of China Coal Society*, vol. 41, no. S2, pp. 589–597, 2016, doi: 10.13225/j.cnki.jccs.2016.1165.

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.