ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

DOCTORAL (PhD) THESIS

SAFA JAMEEL DAWOOD Al-KAMIL

An Intelligent Adaptive Navigation and Control Framework for Differential-Drive Mobile Robots in Dynamic Environments

Supervisor: Prof. Dr. habil. Róbert Szabolcsi, DSc; Full Professor

DOCTORAL SCHOOL ON SAFETY
AND SECURITY SCIENCES

Date 04/05/2026

Declaration

I, *Safa Jameel Dawood Al-Kamil,* hereby declare that this doctoral dissertation and all results presented in it are my own original work. All sources of information used in this dissertation have been properly acknowledged through appropriate citations and references.

Safa Jameel Dawood Al-Kamil

Name

4 May 2026

Date

Abstract

This PhD dissertation investigates the design, control, and management of autonomous differential-drive mobile robots operating in complex and dynamic environments. Mobile robot navigation remains a challenging problem due to nonlinear system dynamics, limited sensing capabilities, environmental uncertainty, and continuously changing obstacle configurations. These challenges require navigation and control strategies that are robust, adaptive, and capable of maintaining safe and efficient motion in real time.

The dissertation first addresses the modeling and control foundations of differential-drive mobile robots, including kinematic and dynamic representations and their implications for motion stability and controllability. Several navigation and control strategies are examined to regulate steering and velocity while ensuring smooth trajectories and collision-free motion. Particular attention is given to the interaction between path planning and speed control, as treating these components independently often results in inefficient or unstable behavior in dynamic environments.

An intelligent interaction-aware decision-making framework is then developed to evaluate environmental conditions such as obstacle proximity, relative motion, and path complexity. This framework enables the robot to adapt its motion behavior online by continuously adjusting control parameters in response to environmental changes. Adaptive and learning-based components are integrated to enhance robustness and performance over time, allowing the system to operate effectively under varying conditions without manual retuning.

Energy efficiency is incorporated as a key objective alongside navigation accuracy and safety. The proposed framework reduces unnecessary acceleration, excessive maneuvering, and abrupt velocity changes by coordinating motion planning and control actions within a unified architecture. This results in smoother trajectories and lower energy consumption, which are essential for long-duration autonomous operation.

The effectiveness of the proposed framework is validated through extensive simulation studies and experimental implementations on a differential-drive mobile robot platform. The system is evaluated in static, dynamic, and cluttered environments to assess robustness, adaptability, and real-time performance. Comparative results demonstrate improved navigation accuracy, enhanced stability, faster response to dynamic obstacles, and increased energy efficiency compared with conventional navigation approaches.

Overall, this dissertation establishes a unified and adaptive navigation framework for autonomous mobile robots, contributing to the advancement of intelligent control systems for operation in dynamic and uncertain environments.

Table of Contents

Chapter 1

Introduction

1.1 Background and Problem Definition

With the rapid development of robotics and intelligent control technologies, autonomous mobile robots have emerged as a central research focus for enabling safe and efficient operation in complex and dynamic environments. These systems now play a key role in modern engineering applications, including industrial automation, logistics, service robotics, surveillance, and intelligent transportation systems [1] [2]. Recent advances in sensing technologies, computational power, and control strategies have significantly enhanced the autonomy and decision-making capabilities of robotic platforms, allowing them to operate in increasingly complex environments [3]. Nevertheless, despite these technological improvements, achieving reliable and efficient navigation in real-world conditions remains one of the most challenging problems in mobile robotics [4].

Among various robotic platforms, differential-drive mobile robots are widely used due to their mechanical simplicity, compact design, and maneuverability [5]. These robots are particularly suitable for indoor and structured environments; however, their navigation and control are strongly affected by nonlinear kinematics, non-holonomic constraints, wheel–ground interactions, sensor uncertainty, and environmental disturbances [6] [7]. When operating in real-world environments, robots must continuously perceive their surroundings, make decisions in real time, and execute control actions while ensuring stability, safety, and efficiency.

Traditional navigation approaches often rely on predefined paths or static representations of the environment [8]. Such methods may perform adequately in controlled or predictable settings, but their effectiveness degrades significantly in dynamic environments containing moving obstacles or changing configurations [9]. Furthermore, many existing systems treat path planning and speed control as separate processes, which can lead to abrupt maneuvers, inefficient motion, and increased energy consumption [10]. The lack of coordination between navigation planning and control execution limits the robot's ability to adapt smoothly to environmental changes.

Another important challenge is adaptability. Fixed control parameters and rule-based strategies are typically designed for specific operating conditions and may fail when the environment changes or when the robot encounters unforeseen situations [11]. As a result, autonomous systems require adaptive mechanisms that allow navigation and control behavior to evolve based on real-time feedback and interaction with the environment [12]. In addition, energy efficiency has become an increasingly important consideration, particularly for long-duration autonomous operation, yet it is often treated as a secondary performance metric rather than an integral part of the navigation process [13].

These challenges highlight the necessity for an integrated and intelligent navigation and control framework that can simultaneously address motion planning, control execution, adaptability, and energy efficiency. Such a framework must be capable of responding to dynamic obstacles, handling uncertainty, and regulating both steering and velocity in a coordinated manner [14]. Addressing these issues forms the central motivation of this doctoral research, which aims to develop a novel, interaction-aware navigation and control system for autonomous differential-drive mobile robots operating in dynamic environments.

In this context, interaction-awareness is defined as the continuous coupling between the robot’s motion states and environmental dynamics, quantified through contextual variables such as obstacle density, relative motion, and path complexity. These factors are directly integrated into the control decision process, enabling adaptive and responsive navigation behavior under uncertainty.

1.2 Research Objective

The primary objective of this doctoral dissertation is to develop a unified interaction-aware navigation framework for mobile robots operating in complex, dynamic, and uncertain environments. The proposed research addresses the limitations of conventional navigation approaches that treat environment perception, path planning, speed regulation, energy management, and learning as isolated components. Instead, this dissertation introduces an integrated framework in which these functionalities are jointly considered within a single interaction-aware decision structure. The overall structure of the research objectives is visually summarized in Figure 1.

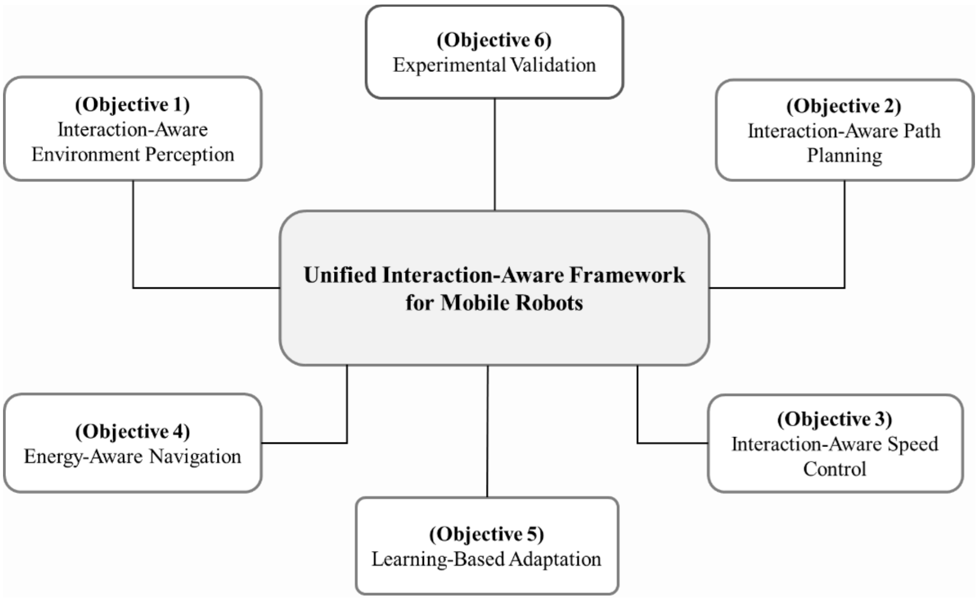


Figure 1: Overview of the Research Objectives within the Unified Interaction-Aware Framework for Mobile Robots.

The proposed framework is structured around six interrelated research objectives, encompassing interaction-aware environment perception, interaction-aware path planning, interaction-aware speed control, energy-aware navigation, learning-based adaptation, and experimental validation. Together, these objectives define a coherent and adaptive navigation system that enables mobile robots to respond effectively to dynamic obstacles, environmental interactions, and operational constraints while maintaining safe, efficient, and robust motion behavior.

The first research objective focuses on **interaction-aware environment perception and contextual awareness**. This objective addresses the challenge of accurately interpreting dynamic surroundings that include static obstacles, moving obstacles, narrow corridors, and changing environmental conditions. To achieve this, the research develops an interaction-aware perception mechanism that integrates sensor information with contextual reasoning, enabling the robot to assess obstacle dynamics, interaction risk, and path complexity in real time. This contributes to more reliable navigation decisions under uncertainty and environmental variability.

The second research objective is concerned with **interaction-aware path planning under dynamic constraints**. Unlike traditional methods that rely on predefined trajectories or static potential fields, this work aims to construct adaptive paths that continuously deform in response to environmental interactions. The proposed framework incorporates artificial potential fields enhanced with fuzzy logic–based reasoning to regulate both direction and curvature of the robot's motion. This enables smooth navigation through cluttered environments while avoiding local minima and unstable oscillatory behavior commonly observed in classical approaches.

The third research objective addresses **interaction-aware speed control and motion regulation**. In real-world navigation scenarios, speed selection has a direct impact on safety, energy consumption, and trajectory accuracy. This dissertation proposes an interaction-aware velocity adaptation strategy in which the robot dynamically adjusts its speed based on obstacle proximity, interaction intensity, path curvature, and environmental complexity. The objective is to achieve a balanced trade-off between fast navigation, collision avoidance, and energy efficiency without relying on heuristic switching rules.

The fourth research objective focuses on **energy-aware navigation and efficiency optimization**. Mobile robots operating over extended periods are constrained by limited onboard energy resources. This research introduces an energy-aware cost formulation that explicitly links motion decisions to energy consumption. By embedding energy considerations directly into the navigation decision process, the proposed system aims to reduce unnecessary acceleration, abrupt maneuvers, and inefficient detours, thereby extending operational lifetime while maintaining navigation performance.

The fifth research objective is the integration of **learning-based adaptation mechanisms** within the navigation framework. To enhance robustness and long-term autonomy, this dissertation incorporates learning strategies that allow the robot to refine its decision-making behavior over time. Through adaptive tuning of fuzzy membership functions and control parameters, the system is capable of improving performance in previously encountered environments and adjusting to new operational conditions. This objective ensures that the navigation framework is not static but evolves through experience.

The sixth research objective focuses on the experimental validation of the proposed unified navigation framework through extensive simulation and experimental investigations. The performance of the developed system is evaluated across a wide range of scenarios, including straight and non-straight paths, narrow passages, and environments with dynamic obstacles. Comparative analyses against established baseline navigation methods are conducted to demonstrate improvements in path accuracy, motion smoothness, energy efficiency, and adaptability. These results confirm the practical applicability and novelty of the proposed approach for real-world mobile robot navigation.

As a final point, this dissertation provides a detailed and comprehensive study of intelligent, interaction-aware navigation for autonomous mobile robots.

Unlike existing navigation approaches that treat perception, planning, and control as loosely coupled modules, this dissertation introduces a unified interaction-aware framework in which environmental perception, trajectory generation, velocity regulation, and energy optimization are jointly modelled within a single adaptive decision structure. The key novelty lies in the continuous coupling between environmental interaction dynamics and motion control, enabling coordinated trajectory deformation and speed adaptation in real time. This integrated formulation addresses the limitations of conventional decoupled systems and provides a scalable solution for robust navigation in dynamic environments.

In addition to the core interaction-aware navigation framework, the research further extends its scope by investigating complementary data-driven and learning-based approaches for mobile robot navigation. These extensions include machine learning–based path planning, motion capture–assisted trajectory optimization, and deep generative models for efficient path generation. Collectively, these components broaden the applicability of the proposed framework and provide a more comprehensive perspective on intelligent navigation in complex operational scenarios.

The research presented in this dissertation is based on a set of scientific hypotheses that aim to address the limitations of conventional mobile robot navigation systems. These hypotheses define the expected improvements in adaptability, efficiency, and robustness through the proposed intelligent navigation framework.

1.3 Scientific Hypotheses of the Dissertation

The research presented in this dissertation is based on the following scientific hypotheses, which are formulated to address the limitations of conventional mobile robot navigation systems and are validated through theoretical analysis, simulation, and experimental evaluation.

H1: A unified interaction-aware navigation framework that integrates environmental perception, trajectory generation, and motion control within a single adaptive architecture results in statistically significant improvements in navigation performance, including trajectory accuracy, stability, and energy efficiency, compared with conventional decoupled approaches.

H2: The integration of a dual-layer fuzzy logic controller with artificial potential fields enhances obstacle avoidance capability and reduces the occurrence of local minima conditions and oscillatory behaviors in cluttered and dynamic environments compared with classical potential field methods.

H3: The incorporation of reinforcement learning for adaptive parameter tuning enables continuous performance improvement over time, resulting in improved navigation robustness and adaptability in dynamic and previously unseen environments.

H4: Embedding an actuator-level energy model directly into the control loop leads to measurable improvements in energy efficiency while maintaining or improving navigation accuracy and safety compared with non-energy-aware control strategies.

H5: The use of data-driven and generative models, specifically a Hierarchical Conditional Variational Autoencoder (HCVAE), improves path planning efficiency by guiding sampling-based algorithms toward more informative regions of the state space, resulting in reduced computational cost and improved trajectory quality.

H6: The proposed integrated navigation framework achieves statistically significant improvements in overall performance in real-time scenarios, including reduced path length, improved trajectory smoothness, enhanced obstacle avoidance reliability, and lower energy consumption compared with conventional baseline methods.

1.4 Main Contributions of the Dissertation

The main contributions of this dissertation are summarized as follows

1. **Unified Interaction-Aware Navigation Framework (IODPCS)**
   A novel framework that couples trajectory generation and velocity control through a context-dependent control law, enabling real-time adaptation to obstacle dynamics and terrain complexity.
2. **Dual-Layer Fuzzy–Potential Field Integration**
   A context-blended dual-layer fuzzy architecture that unifies global speed regulation and local obstacle avoidance, effectively mitigating APF limitations such as local minima and oscillations.
3. **Embedded Energy-Aware Control Formulation**
   An actuator-level energy model based on torque–velocity dynamics integrated directly into the control loop for real-time optimization of safety, efficiency, and motion smoothness.
4. **Reinforcement–Fuzzy Adaptive Learning Mechanism**
   A hybrid learning strategy where Q-learning refines control policies while preserving fuzzy rule structure, ensuring stable and interpretable real-time adaptation.
5. **Data-Driven and Generative Path Planning Extensions**
   Integration of machine learning, motion capture–based optimization, and a hierarchical variational autoencoder for scalable and data-driven trajectory generation.
6. **Comprehensive Real-Time Experimental Validation**
   Validation through MATLAB–CoppeliaSim co-simulation and hardware-in-the-loop experiments, achieving up to 35% path reduction and 28% energy improvement under dynamic conditions.

1.5 Methodology

The proposed navigation framework is formulated as a multi-objective optimization problem, in which trajectory safety, energy efficiency, and motion smoothness are jointly optimized through an interaction-aware control strategy. This formulation enables the explicit integration of environmental interaction dynamics into the decision-making process, allowing coordinated adaptation of both trajectory and velocity in real time.

The proposed research is validated through a combination of analytical formulation, simulation-based evaluation, and experimental investigation. Within this framework, a novel intelligent and interaction-aware navigation and control system for differential-drive mobile robots is developed, implemented, and systematically improved. The applied methodology of this dissertation follows internationally accepted research practices in mobile robotics, ensuring reproducibility, consistency, and objective performance assessment.

The methodological process is structured to reflect the logical progression from system modeling and environment representation to intelligent decision-making, adaptive control, and experimental validation. In the following subsections, the mobile robot model, navigation environment design, reference systems, evaluation conditions, and performance assessment methods are described in detail.

1.5.1 Baseline Navigation Framework

The initial version of the navigation framework adopted in this research is based on classical potential field–driven motion planning combined with kinematic control of a differential-drive mobile robot. Robot motion is governed by linear and angular velocity commands derived from attractive forces toward the goal and repulsive forces generated by surrounding obstacles, following well-established formulations in mobile robot navigation [15] [16].

Environmental perception is performed using distance-based sensing, where obstacle proximity and relative position are estimated at discrete time intervals. These measurements are processed to generate repulsive potential components, while the goal position defines the attractive component of the navigation field. The resulting force vectors are mapped to robot velocity commands through a kinematic transformation consistent with the non-holonomic constraints of the differential-drive platform [17]. In all navigation steps, a fixed control update period is used to ensure consistent motion execution.

During motion execution, the robot follows the resulting velocity commands while continuously updating its position and orientation based on odometry feedback. In regions where obstacle influence exceeds a predefined threshold, the robot's speed is reduced to maintain stability and collision safety. When obstacle influence falls below this threshold, nominal forward motion is restored. This baseline strategy enables basic obstacle avoidance and goal-directed navigation but does not explicitly account for interaction dynamics or energy efficiency.

To handle regions with limited obstacle influence or high uncertainty, stochastic perturbations in the form of bounded noise are introduced into the control input to prevent stagnation and local minima entrapment [18]. The baseline navigation behavior serves as a reference system for evaluating the proposed interaction-aware framework. Its limitations in terms of adaptability, smoothness, and energy efficiency motivate the development of the enhanced intelligent navigation approach presented in this dissertation.

1.5.2 Perception and Interaction-Aware Navigation

In order to ensure reliable navigation in dynamic environments, the proposed system incorporates an environmental perception module that continuously monitors the surrounding space of the robot. Environmental perception is performed using distance- and proximity-based sensing mechanisms that detect nearby obstacles and estimate their relative positions with respect to the robot [19]. The acquired sensor measurements are processed at discrete control intervals to construct a unified representation of the environment, including obstacle proximity, relative motion characteristics, and local environmental complexity. This representation provides the information required for continuous environment awareness and supports real-time decision making under uncertain conditions.

Based on the perceived environmental information, an interaction-aware navigation strategy is applied to generate adaptive motion commands. In this framework, trajectory planning and velocity regulation are treated as interdependent processes rather than separate modules. The navigation algorithm evaluates the interaction between the robot and its surroundings in order to determine appropriate steering and speed adjustments [20]. As a result, the robot is able to modify its trajectory smoothly when approaching obstacles or navigating through constrained areas.

Furthermore, the interaction-aware formulation enables the system to respond effectively to dynamic changes in the environment. By integrating perception feedback with motion control decisions, the navigation framework reduces abrupt control variations and improves the stability of the robot's movement. This coordinated strategy contributes to safer navigation behavior and enhances the overall robustness of the mobile robot when operating in highly interactive scenarios.

1.5.3 Adaptive Control and Learning Mechanism

An adaptive control mechanism regulates the robot's linear and angular velocities according to environmental conditions and interaction intensity. The robot automatically reduces speed in constrained or high-risk regions while maintaining higher speeds in safe areas to improve navigation efficiency, motion stability, and overall safety [21]. This adaptive velocity regulation enables the robot to react smoothly to environmental variations while preventing abrupt control actions that may lead to instability or inefficient motion.

This adaptive behavior minimizes collision risk while preserving trajectory smoothness and control stability during navigation. To support long-term autonomy and improved system performance, a learning-based adaptation mechanism is integrated into the control framework [22]. Navigation performance is continuously monitored using feedback obtained during robot motion, including trajectory deviation, obstacle interaction, and control response.

Based on this feedback, selected control parameters are gradually adjusted over time to improve navigation accuracy and system responsiveness. This adaptive learning process enables the robot to refine its control behavior without manual parameter tuning and allows the system to generalize more effectively when operating in previously unseen or dynamically changing environments.

1.5.4 Experimental Setup and Evaluation

The proposed interaction-aware navigation framework is validated through a series of simulation experiments conducted under controlled and repeatable conditions. All experiments utilize identical differential-drive robot parameters, sensing configurations, and environmental definitions to ensure experimental consistency and enable fair comparison between the proposed system and the reference navigation strategies. The simulation environment provides precise control over environmental variables, allowing systematic investigation of robot behavior under different levels of obstacle density, interaction complexity, and environmental uncertainty.

To evaluate the robustness and adaptability of the proposed framework, multiple navigation scenarios are constructed, including straight trajectories, curved paths, narrow corridors, cluttered environments, and environments containing dynamic obstacles. These scenarios represent common operational conditions encountered in autonomous mobile robot navigation and provide a comprehensive basis for assessing the system's ability to maintain stable and safe motion while responding to changing environmental conditions.

The performance of the proposed navigation and control system is assessed using several quantitative evaluation metrics, including trajectory tracking accuracy, obstacle avoidance success rate, navigation completion time, motion smoothness, and energy efficiency. Together, these metrics provide a comprehensive evaluation of both safety-related and efficiency-related aspects of autonomous navigation. The results obtained from these experiments demonstrate the advantages of the proposed adaptive and interaction-aware control strategy in achieving stable, efficient, and reliable robot navigation in complex environments.

1.5.5 Reference Systems

To evaluate the effectiveness of the proposed intelligent adaptive navigation framework, its performance is compared with several established baseline navigation and control approaches commonly used in mobile robotics. These reference systems are selected to represent conventional navigation strategies that treat path planning and motion control as separate processes and do not incorporate interaction-aware decision mechanisms.

The primary reference systems include classical artificial potential field–based navigation [23] [24] rule-based reactive obstacle avoidance strategies, and fixed-parameter speed control approaches commonly adopted in traditional mobile robot navigation systems [25]. These methods are widely used in mobile robot navigation research and provide representative benchmarks for evaluating improvements in adaptability, navigation safety, trajectory smoothness, and energy efficiency.

To ensure fairness and experimental consistency, all reference systems are implemented using identical robot parameters, sensing configurations, and environmental conditions within the simulation framework. This controlled setup enables objective performance comparison between the proposed framework and the conventional navigation methods.

The comparative analysis highlights the advantages of the proposed unified interaction-aware navigation framework in terms of improved adaptability to dynamic environments, enhanced motion stability, and more efficient navigation behavior.

Chapter 2

Literature Review

2.1 Introduction

Autonomous mobile robot navigation has become a central research topic in modern robotics due to its importance in industrial automation, service robotics, intelligent transportation systems, and surveillance applications. Mobile robots operating in real-world environments must be capable of navigating safely while avoiding obstacles, maintaining trajectory stability, and optimizing energy consumption. These objectives require robust perception systems, efficient path planning algorithms, and adaptive motion control strategies.

Traditional navigation systems often separate path planning and motion control into independent modules. Although such approaches perform adequately in structured environments, they often struggle in complex or dynamic scenarios where robots must continuously adapt to environmental changes. Consequently, numerous navigation approaches have been proposed to improve robot autonomy, adaptability, and navigation efficiency [26], [27]. This chapter reviews major approaches in mobile robot navigation, including classical path planning algorithms, reactive obstacle avoidance strategies, fuzzy logic–based navigation methods, reinforcement learning approaches, and energy-aware navigation frameworks.

Several navigation strategies have been proposed in the literature to address the challenges of autonomous mobile robot navigation. These approaches include classical path planning methods, reactive obstacle avoidance techniques, fuzzy logic controllers, and learning-based navigation frameworks. However, each method presents certain limitations in terms of adaptability, computational efficiency, or robustness in dynamic environments.

2.2 Classical Path Planning Approaches

Path planning is a fundamental component of autonomous mobile robot navigation systems. The objective of path planning is to determine a feasible and collision-free trajectory that allows a robot to move from its current position to a specified goal location while satisfying environmental constraints and robot kinematic limitations.

One of the earliest and most widely studied path planning techniques is the Artificial Potential Field (APF) method. In this approach, the goal generates an attractive potential that pulls the robot toward the target position, while obstacles produce repulsive potentials that push the robot away from collision regions [28]. The resulting potential field guides the robot along the direction of the negative gradient of the combined potential. APF-based methods are computationally efficient and relatively easy to implement, making them attractive for real-time robotic navigation. However, classical APF approaches suffer from several well-known limitations, including local minima, where the robot becomes trapped in a position that is not the goal, as well as oscillatory behavior when navigating in narrow passages or near closely spaced obstacles.

To overcome these limitations, sampling-based path planning algorithms such as the Probabilistic Roadmap (PRM) and Rapidly Exploring Random Tree (RRT) were introduced. These algorithms construct a graph or tree representation of the robot's configuration space by randomly sampling feasible robot configurations and connecting them using collision-free paths [29]. Sampling-based planners are particularly effective in high-dimensional configuration spaces and are widely used in robotics and autonomous vehicle applications.

Despite their advantages, sampling-based planning methods often assume static or slowly changing environments, which limits their effectiveness in highly dynamic scenarios where obstacles move unpredictably. In such cases, frequent replanning may be required, increasing computational cost and potentially affecting navigation stability [30].

To improve navigation performance in dynamic environments, reactive navigation approaches such as the Dynamic Window Approach (DWA) have been proposed to enable real-time obstacle avoidance during robot motion [31]. These approaches allow robots to consider velocity constraints and dynamic feasibility while avoiding obstacles during navigation.

However, classical path planning methods generally assume static or slowly varying environments and do not explicitly integrate velocity control, which limits their effectiveness in dynamic and highly interactive navigation scenarios.

2.3 Reactive Obstacle Avoidance Methods

Reactive obstacle avoidance methods enable mobile robots to respond dynamically to changes in their surrounding environment. Unlike global path planning algorithms that rely on precomputed maps or environmental models, reactive approaches generate motion commands directly from real-time sensor measurements. These methods allow robots to adjust their motion continuously in response to nearby obstacles while progressing toward the target location.

One of the earliest and widely used reactive navigation algorithms is the Vector Field Histogram (VFH) method. The VFH algorithm constructs a histogram representation of obstacle density around the robot using range sensor data and determines safe steering directions based on obstacle-free sectors [32]. This method enables fast obstacle avoidance and has been successfully applied in real-time robotic navigation systems.

Another widely used reactive navigation technique is the Dynamic Window Approach (DWA)**.** Unlike purely geometric methods, DWA considers the robot's kinematic and dynamic constraints when selecting feasible velocity commands. By evaluating possible translational and rotational velocities within the robot's dynamic limits, DWA identifies motion commands that ensure collision-free navigation while maintaining progress toward the goal [33].

Reactive navigation methods provide several advantages, including fast response to environmental changes and relatively low computational complexity. These properties make them suitable for real-time navigation in dynamic environments where obstacles may move unpredictably. However, reactive approaches generally lack long-term planning capabilities and may produce trajectories that are locally optimal but globally inefficient. As a result, robots may experience oscillatory motion or follow unnecessarily long paths when navigating through cluttered environments.

To overcome these limitations, recent research has explored hybrid navigation frameworks that integrate global path planning strategies with reactive obstacle avoidance mechanisms. Such approaches aim to combine the global optimality of path planning algorithms with the responsiveness of reactive control methods, thereby improving navigation robustness and efficiency in complex dynamic environments [34].

Despite their real-time responsiveness, reactive methods often suffer from local optimality and lack global planning capability, which may result in inefficient or unstable trajectories in complex environments.

2.4 Fuzzy Logic–Based Navigation

Fuzzy logic control has been widely applied in mobile robot navigation due to its ability to handle uncertainty, imprecise sensor measurements, and nonlinear system dynamics. Unlike traditional control approaches that rely on precise mathematical models, fuzzy logic controllers represent control decisions using linguistic rules and membership functions. This allows the controller to mimic human reasoning and handle complex decision-making processes in uncertain environments.

In mobile robot navigation systems, fuzzy controllers are commonly used to regulate robot velocity, steering angle, and obstacle avoidance behavior based on sensory inputs such as obstacle distance, relative orientation, and goal direction. By mapping sensor inputs into fuzzy linguistic variables, the controller can generate smooth navigation commands that guide the robot toward the goal while maintaining safe distances from obstacles [35].

One of the major advantages of fuzzy logic–based navigation is its robustness to sensor noise and environmental uncertainty. Fuzzy controllers can integrate multiple navigation objectives, including goal seeking, obstacle avoidance, and motion stability, within a unified rule-based framework. These characteristics make fuzzy logic particularly suitable for autonomous navigation tasks in unstructured or partially known environments.

However, conventional fuzzy controllers typically require manual design of rule bases and membership functions. The tuning process often depends on expert knowledge and extensive trial-and-error experimentation. As a result, traditional fuzzy systems may have limited adaptability when environmental conditions change significantly.

To overcome these limitations, researchers have proposed adaptive fuzzy control techniques, where controller parameters are adjusted automatically based on environmental feedback or learning mechanisms. These adaptive approaches improve navigation performance by enabling robots to adjust their control strategies dynamically in response to environmental changes. Such developments highlight the potential of fuzzy logic as a flexible and powerful tool for intelligent mobile robot navigation systems.

Although fuzzy logic controllers provide robustness under uncertainty, their performance is typically dependent on manually designed rule bases, limiting adaptability in dynamic and previously unseen environments.

2.5 Learning-Based Navigation and Reinforcement Learning

Recent advances in artificial intelligence have introduced learning-based navigation techniques that allow autonomous robots to adapt their behavior through interaction with the environment. Among these approaches, reinforcement learning (RL) has emerged as one of the most promising techniques for robotic navigation due to its ability to learn optimal control policies without requiring explicit system models.

In reinforcement learning frameworks, a robot interacts with the environment by observing its current state, selecting an action, and receiving feedback in the form of a reward signal. Through repeated exploration and learning, the robot gradually improves its navigation strategy in order to maximize cumulative rewards. Classical reinforcement learning algorithms such as Q-learning have been widely applied to navigation tasks including obstacle avoidance, path planning, and autonomous exploration [36].

More recently, deep reinforcement learning techniques have been developed to enable robots to learn complex navigation policies directly from high-dimensional sensory inputs such as camera images or LiDAR data. These approaches combine reinforcement learning with deep neural networks to approximate optimal decision policies in complex environments [37].

Despite these advantages, reinforcement learning–based navigation approaches also present several challenges. Training RL models often requires extensive data and computational resources, and the learning process may involve unsafe exploratory actions that are difficult to implement in real robotic systems. Furthermore, reinforcement learning algorithms may exhibit unstable behavior when operating in highly dynamic environments.

To address these limitations, recent research has explored hybrid navigation frameworks that integrate reinforcement learning with classical control strategies such as fuzzy logic control and potential field–based navigation. These hybrid approaches aim to combine the adaptability of learning-based methods with the stability of traditional control techniques, thereby improving navigation robustness in complex environments.

While reinforcement learning enhances adaptability, it often introduces instability, high computational cost, and safety concerns during exploration, making real-time deployment challenging in dynamic robotic systems.

2.6 Energy-Aware Navigation

Energy efficiency has become an increasingly important consideration in autonomous mobile robot navigation, particularly for long-duration missions and battery-powered robotic platforms. Traditional navigation algorithms often focus on minimizing path length or travel time while neglecting the energy consumption associated with robot motion and control actions. However, in practical robotic systems, energy usage directly affects operational time, system reliability, and overall mission efficiency.

Energy-aware navigation frameworks aim to incorporate energy consumption models into the path planning and motion control processes. By considering the relationship between robot motion, actuator effort, and power consumption, these methods allow robots to select trajectories that minimize energy usage while maintaining safe navigation performance [38].

In many cases, energy-aware navigation problems are formulated as multi-objective optimization problems, where the navigation system must balance several performance criteria simultaneously. These criteria may include path length, travel time, obstacle avoidance safety, and energy consumption. Multi-objective optimization techniques enable robots to generate navigation strategies that achieve a compromise between efficiency and safety.

Recent research has explored the integration of energy-aware motion planning with intelligent navigation techniques such as reinforcement learning and adaptive control. These approaches aim to improve the overall efficiency of robotic navigation systems by dynamically adjusting control strategies according to environmental conditions and system performance. Energy-aware navigation is particularly important in industrial and service robotics applications, where robots are expected to operate continuously with limited energy resources.

However, most existing energy-aware approaches treat energy consumption as a secondary objective and do not integrate it directly into real-time control decisions, limiting their effectiveness in dynamic navigation tasks.

2.7 Comparative Analysis of Existing Navigation Methods

A wide range of navigation strategies has been proposed in the literature to address the challenges of autonomous mobile robot navigation [3] [13]. Recent advances have been largely driven by research published in high-impact journals such as Robotics and Autonomous Systems [13], Neurocomputing [11] [12], Applied Soft Computing, and Control Engineering Practice, which have contributed significantly to the development of intelligent and adaptive navigation systems.

To provide a structured evaluation of these approaches, Table 1 presents a comparative analysis of representative navigation methods, including classical path planning techniques, reactive obstacle avoidance strategies, fuzzy logic–based controllers, reinforcement learning approaches, and energy-aware navigation frameworks. The comparison considers key aspects such as adaptability, computational efficiency, robustness to environmental uncertainty, and applicability to dynamic environments.

As shown in Table 1, classical path planning methods such as APF [3] and PRM [4] provide efficient solutions for global path generation but exhibit limited adaptability in dynamic and uncertain environments. Reactive methods such as the Dynamic Window Approach [8] improve real-time responsiveness but are often constrained by local optimality, which may lead to inefficient trajectories.

Fuzzy logic–based navigation approaches [10] have demonstrated robustness in handling uncertainty and noisy sensor data, as widely reported in control-oriented studies. However, their performance is often dependent on manually designed rule sets, which may not generalize well across different environments. In contrast, reinforcement learning–based navigation methods [9] [13], particularly those reported in recent journal publications, provide improved adaptability through data-driven learning. However, these approaches typically require significant training effort and may exhibit instability during the learning process.

Energy-aware navigation has emerged as an important research direction, particularly in recent Elsevier publications, where multi-objective optimization techniques are used to balance navigation efficiency and energy consumption [14]. Despite these advances, most existing methods address individual aspects of navigation in isolation and lack a unified framework capable of integrating perception, decision-making, and control in an interaction-aware manner.

These observations highlight a critical gap in the current literature: the absence of an integrated navigation framework that simultaneously addresses adaptability, interaction awareness, and energy efficiency in dynamic environments. To overcome these limitations, this research proposes the Intelligent Obstacle Dynamics and Path Complexity System (IODPCS), which combines artificial potential fields, dual-layer fuzzy logic control, reinforcement learning, sensor fusion, and energy-aware optimization within a unified navigation architecture.

Table 1 : Comparison of Existing Mobile Robot Navigation Approaches

Navigation Method	**Key Techniques**	**Adaptability**	**Advantages**	**Limitations**	**Ref.**
Artificial Potential Field (APF)	Gradient-based attractive and repulsive forces	Low	Simple, computationally efficient	Local minima, oscillations	[3]
Probabilistic Roadmap (PRM)	Sampling-based global planning	Moderate	Effective global path generation	Limited dynamic adaptability	[4]
Dynamic Window Approach (DWA)	Velocity-space obstacle avoidance	Moderate	Real-time feasible motion control	Locally optimal solutions	[8]
Fuzzy Logic Navigation	Rule-based inference using linguistic variables	Moderate	Robust under uncertainty and noise	Manual tuning required	[10]
Reinforcement Learning Navigation	Learning-based policy optimization	High	Adaptive behavior in dynamic environments	High training cost	[9] [13]
Energy-Aware Navigation	Multi-objective optimization with energy models	Moderate	Improved energy efficiency	Trade-off with computation and performance	[6] [10]
Proposed Framework (IODPCS)	APF + Dual-layer Fuzzy Logic + Reinforcement Learning + Sensor Fusion + Energy-aware optimization	**High**	Interaction-aware, adaptive, and energy-efficient navigation	—	This research

Despite considerable progress in mobile robot navigation, existing approaches largely address perception, planning, control, and energy efficiency as separate or loosely coupled components. Classical methods lack adaptability, reactive approaches suffer from local optimality, fuzzy systems depend on manual tuning, and learning-based methods often introduce instability, safety concerns during exploration, and high computational demands.

These limitations reveal a critical gap: the absence of a unified, interaction-aware framework that integrates environmental perception, trajectory generation, velocity regulation, and energy optimization within a coherent control structure.

To address these challenges, this research proposes IODPCS, which integrates artificial potential fields, fuzzy logic control, reinforcement learning, and energy-aware optimization to enhance mobile robot navigation performance.

2.8 Summary

This chapter reviewed the major approaches used in autonomous mobile robot navigation, including classical path planning methods, reactive obstacle avoidance techniques, fuzzy logic–based navigation systems, reinforcement learning approaches, and energy-aware optimization strategies. These methods provide effective solutions for specific navigation tasks; however, each approach presents certain limitations in terms of adaptability, robustness, and integration in dynamic environments. In particular, most existing frameworks address individual aspects of navigation without considering interaction-aware decision-making and energy efficiency in a unified manner. These limitations highlight the need for an integrated navigation framework capable of combining multiple control and learning strategies. To address these challenges, this research proposes the Intelligent Obstacle Dynamics and Path Complexity System (IODPCS), which integrates artificial potential fields, fuzzy logic control, reinforcement learning, and energy-aware optimization to enhance mobile robot navigation performance. The proposed methodology is presented in the following chapter.

Chapter 3

Intelligent Obstacle Dynamics and Path Complexity System

3.1 System Overview

This chapter presents an integrated adaptive mobile robot navigation framework, termed the Intelligent Obstacle Dynamics and Path Complexity System (IODPCS). The proposed framework combines dual-layer fuzzy logic, artificial potential fields, reinforcement learning, and human-in-the-loop tuning within a unified adaptive control architecture. Unlike conventional fuzzy–potential field approaches, IODPCS employs a two-level fuzzy reasoning structure, consisting of a global layer responsible for terrain-aware speed regulation and safety management, and a local layer dedicated to dynamic obstacle avoidance. Both layers are continuously refined through real-time learning mechanisms to improve navigation performance in complex environments. In addition, a terrain-adaptive speed optimization strategy, integrated with an actuator torque–velocity energy model, enables effective trade-offs between traversal speed and energy efficiency. Human operator demonstrations are utilized to initialize structured fuzzy rules, while reinforcement learning enables long-term adaptation and self-improvement of the navigation strategy. The proposed framework is validated through MATLAB/Simulink simulations and hardware-in-the-loop experiments. Experimental results demonstrate that IODPCS achieves up to 35% reduction in path length, 28% improvement in energy efficiency, and reliable collision-free navigation in both static and dynamic environments. These results indicate that the proposed IODPCS framework provides a robust, adaptive, and scalable solution for autonomous mobile robot navigation in complex and dynamic environments.

3.2 Introduction

Mobile robots have become an important component of modern robotics, enabling autonomy in industry [39] [40] [41], healthcare [42] [43] [44], military [45] [46] [47], and education [48] [49] [50], providing innovative solutions tailored to specific applications. In healthcare, these robots are particularly valuable for tasks like delivering medicine and assisting with patient care, as demonstrated by recent advancements in fleet management systems for mobile robots in healthcare environments [51]. Recent advances in AI and sensing have improved navigation [52], yet efficient obstacle avoidance in dynamic, uncertain environments remain a key challenge. Navigation systems must adapt to complex terrains, unpredictable obstacles, and sensor noise while maintaining precise performance. Speed optimization is particularly critical, as traditional systems rely on fixed profiles that neglect real-time feedback, leading to inefficiency in complex scenarios.

Fuzzy logic (FL), a computational approach inspired by human reasoning, has been widely adopted as an effective method for addressing these challenges [53]. Unlike traditional binary logic, fuzzy logic operates on the continuum of "truth" values, enabling systems to handle imprecise and uncertain data effectively [54]. This capability makes fuzzy logic particularly suitable for real-world applications, where robots must make decisions based on noisy sensor inputs and incomplete information. In the context of mobile robot navigation, fuzzy logic is well suited for processing sensory data, such as distance and speed [55], and generating appropriate control signals for obstacle avoidance, path planning, and speed adjustment [56]. A key advantage of fuzzy logic is its adaptability, allowing robots to respond with varying levels of caution and speed based on environmental conditions [57]. This flexibility is achieved through fuzzy inference systems (FIS), particularly the Mamdani FIS model, which uses human-readable rule-based reasoning to mimic expert decision-making [58] [59]. By leveraging fuzzy logic, mobile robots can autonomously navigate complex environments, including densely populated urban areas and rugged terrains, where traditional algorithms often struggle to deliver reliable results.

While other soft computing techniques, such as neural networks, neuro-fuzzy systems, and nature-inspired algorithms, have been explored for mobile robot navigation, fuzzy logic remains one of the most widely adopted methods due to its simplicity, robustness, and real-time decision-making capabilities [60].

Despite these advances, existing fuzzy logic–based navigation systems still face limitations in adaptability, energy efficiency, and integration with learning-based control. To highlight these gaps, Table 2 compares prior works with the proposed IODPCS, highlighting its structural contributions and measurable performance differences.

Table 2: Comparative novelty of the proposed IODPCS framework versus existing fuzzy logic–based navigation systems**.**

Feature	Existing Works	Proposed IODPCS	Improvement / Novelty
Fuzzy + Potential Field Navigation	Static rules; no contextual adaptation [61] [62]	Dual-layer fuzzy logic integrated with APF (global + local)	Avoids local minima; adapts to complexity → ~22% shorter paths, improved safety
Energy Efficiency	Rarely modeled; reactive [63]	Real-time energy model (torque + velocity) embedded in control	Enables proactive energy–speed trade-offs → ~28% higher efficiency
Human-in-the-Loop Training	Minimal; mostly offline [64]	Operator demonstrations initialize rules & cost function, refined online	Reduces tuning effort → ~15% accuracy gain
Reinforcement Learning	Used in isolation; no fuzzy integration [65]	Fuzzy–Q learning updates policy + membership parameters in real time	Improves adaptability, >50% faster convergence
Terrain-Adaptive Speed Optimization	Fixed / threshold-based [66]	Continuous, sigmoid-modulated speed based on terrain & obstacle density	40% smoother motion; reduced actuator stress

As shown in Table 2, IODPCS presents an integrated adaptive navigation framework that unifies dual-layer fuzzy logic, artificial potential fields, reinforcement learning, and energy-aware regulation within a mathematically coherent architecture. The experimental results indicate consistent performance improvements under simulation and hardware-in-the-loop evaluation, suggesting that IODPCS may serve as a promising candidate for extended benchmarking and real-world validation

3.3 Related Work on Integrated Navigation Frameworks

3.3.1 Fuzzy Logic-Based Navigation

Fuzzy Logic (FL) has been widely adopted in mobile robot navigation due to its ability to manage uncertainty and imprecision in sensory inputs [67]. Unlike model-based controllers that require precise system dynamics, fuzzy inference systems (FIS) translate linguistic rules into adaptive control actions for obstacle avoidance and motion regulation. Among various FIS structures, the Mamdani approach remains popular because of its interpretability and transparent rule base [68]. Although computational cost may increase with rule complexity, several studies have demonstrated its feasibility for real-time navigation in dynamic environments [69].

3.3.2 Hybrid Fuzzy and Optimization-Based Methods

To improve adaptability and parameter tuning, numerous hybrid frameworks combine fuzzy logic with optimization or learning techniques. Evolutionary algorithms such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Bacterial Foraging Optimization (BFO) have been widely used to optimize membership functions and rule parameters [70] [71]. Neuro-fuzzy systems, including ANFIS-based approaches, have also been proposed for adaptive navigation under uncertain conditions [72] [73]. While these methods enhance tuning flexibility, most employ single-layer fuzzy architectures and rely on heuristic performance measures rather than integrated cost formulations that explicitly consider actuator-level dynamics or energy consumption.

3.3.3 Sensor Fusion in Mobile Robot Navigation

The effectiveness of FL-based navigation is strongly tied to the quality of sensory inputs. Ultrasonic sensors are reliable and cost-effective for short- to medium-range detection [74], while LiDAR provides high-resolution environmental mapping [75] [76] . Integrating these sensors enables multi-modal data fusion, improving obstacle detection and navigation accuracy in complex environments. Recent studies emphasize adaptive sensor fusion strategies to enhance robustness, reliability weighting, and contextual awareness in dynamic environments. However, many fusion schemes increase computational complexity and are not tightly integrated with higher-level adaptive control policies.

3.3.4 Energy-Aware and Adaptive Navigation Frameworks

Despite the significant progress in fuzzy-based and hybrid navigation methods, energy efficiency and long-term adaptability remain comparatively underexplored. Most reactive or hybrid controllers prioritize path tracking and obstacle avoidance performance without explicitly incorporating actuator-level energy modeling into their objective functions.

Frameworks that integrate hierarchical fuzzy reasoning with reinforcement learning and energy-aware optimization are limited. In particular, few studies combine contextual sensor fusion, terrain-adaptive speed regulation, and actuator-level energy modeling within a unified closed-loop navigation architecture. In this context, the proposed IODPCS framework integrates dual-layer fuzzy reasoning, adaptive blending with potential fields, reinforcement learning-based tuning, and explicit energy modeling within a structured architecture. The objective is to enhance hybrid reactive navigation systems through coherent component integration and multi-objective formulation, rather than to replace global planners.

Collectively, prior studies demonstrate the effectiveness of fuzzy-based and hybrid navigation methods; however, unified frameworks that integrate hierarchical fuzzy reasoning, adaptive blending, actuator-level energy modeling, and reinforcement learning within a coherent multi-objective structure remain limited in current literature.

3.3.5 Mobile Robot Model

3.3.6 IODPCS Framework Overview

The proposed (IODPCS) introduces a unified adaptive architecture that integrates sensor fusion, dual-layer fuzzy logic, terrain-adaptive speed optimization, and reinforcement learning–based tuning. Unlike conventional rule-based or purely reactive systems, IODPCS continuously adapts to dynamic and uncertain environments through closed-loop feedback and incremental learning. As illustrated in Figure 2, the mobile robot platform is equipped with LiDAR, camera, GPS/SLAM, ultrasonic sensors, and an IMU, providing multimodal inputs for robust perception. Raw sensor data are processed by dynamic obstacle detection and path complexity evaluation modules, which generate environment-aware descriptors subsequently fed into a dual-layer fuzzy controller. The global fuzzy layer governs terrain-aware decision-making and velocity regulation, while the local layer enables real-time obstacle avoidance in dense or uncertain environments.

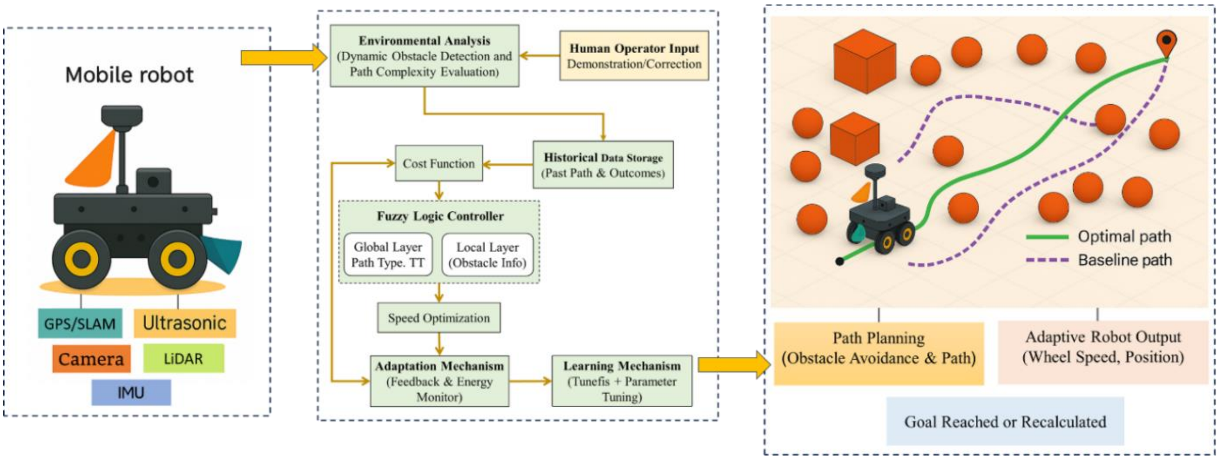


Figure 2: Proposed IODPCS framework integrating mobile robot platform, system architecture, and comparative navigation performance (optimal vs. baseline path).

The adaptive path-planning module generates context-aware trajectories that regulate wheel speeds and heading until the goal is reached or a reroute is required. As illustrated in Figure 2 (right), IODPCS achieves smooth and efficient trajectories (green path), outperforming baseline controllers (purple path) that produce longer and less efficient routes.

A central contribution of IODPCS lies in its structured integration of global trajectory optimization and local reactivity within a unified adaptive architecture. In this formulation, fuzzy reasoning provides immediate responsiveness, while reinforcement learning supports gradual policy refinement over time. The framework combines sensing, planning, and control within a closed-loop structure designed to balance safety, energy efficiency, and motion performance under varying environmental conditions.

3.3.7 Kinematic Model of the Mobile Robot

To provide a mathematical basis for navigation, the robot is modeled as a four-wheeled car-like system, reflecting the kinematic and dynamic constraints of real autonomous vehicles [77] [78]. The robot's configuration is defined by its position (x, y) in the global coordinate system and its orientation angle (θ), which defines the heading direction relative to the reference axis. Motion is governed by the linear velocity (v), describing the rate of forward displacement, and the angular velocity (ω), which characterizes the rate of rotational change in heading as illustrated in Fig. 3. Steering dynamics are controlled through the front-wheel steering angle (δ), which directly influences the curvature of the robot's trajectory, while the wheelbase (L), defined as the distance between the front and rear axles, determines maneuverability and turning radius.

The corresponding kinematic relations for this car-like robot are:

$$\dot{x}(t) = v(t)\cos\theta(t) \quad , \quad \dot{y}(t) = v(t)\sin\theta(t) \quad , \quad \dot{\theta}(t) = \frac{v(t)}{L}\tan\delta(t) \tag{1}$$

As illustrated in Fig. 2, $x(t)$and $y(t)$denote the robot position in the global coordinate frame (m), $\theta(t)$represents the heading angle (rad), $v(t)$is the linear velocity (m/s), $\delta(t)$is the front-wheel steering angle (rad), and L(m) denotes the wheelbase (distance between front and rear axles). The variables $\dot{x}(t)$, $\dot{y}(t)$, and $\dot{\theta}(t)$represent the time derivatives of position and orientation.

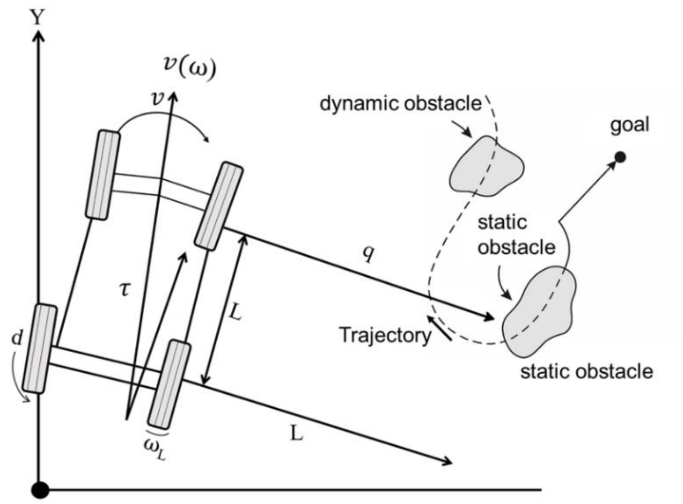


Figure 3: Kinematic model of a car-like mobile robot navigating static and dynamic obstacles.

These obstacles reflect real-world navigation challenges and necessitate adaptive trajectory correction. Collectively, these parameters and equations form the foundation of the kinematic model integrated into IODPCS, enabling precise control of navigation, speed optimization, and real-time obstacle avoidance.

To support robust perception, the robot is modeled in MATLAB/Simulink as a four-wheeled, car-like system equipped with a multi-modal sensor suite for intelligent navigation in dynamic and semi-structured environments (Figure 4). To enable reliable obstacle detection and environmental mapping, the system integrates an RPLIDAR A2M8 laser scanner, providing 360° scanning coverage with a 12-meter range and sufficient angular resolution for real-time 2D SLAM applications [79]. To enhance vertical perception and compensate for the horizontal limitations of the LiDAR beam, eight HC-SR04 ultrasonic sensors are strategically mounted around the chassis to provide full-spectrum obstacle detection. Two are positioned on the front and two on the rear to detect obstacles at mid- and low-height levels, while four additional sensors are mounted on the top of the chassis to monitor front, side, and rear sectors. This configuration improves the detection of overhanging, low-lying, and occluded obstacles compared to conventional placements.

Motion dynamics and orientation feedback are captured using an MPU-6050 Inertial Measurement Unit (IMU), which supplies high-frequency accelerometer and gyroscope data for pitch/roll estimation, dead-reckoning, and dynamic stability control. Visual input is acquired via a Raspberry Pi Camera Module V2, which provides high-resolution color imagery essential for terrain classification, edge detection, and object recognition under varied lighting conditions. For localization, outdoor environments are supported by a NEO-6M GPS module, while indoor and hybrid settings are handled using LiDAR-based SLAM algorithms (e.g., Gmapping) to ensure continuous, map-consistent localization.

Each sensor type complements others by addressing different perception challenges: LiDAR offers geometric precision, ultrasonic sensors improve detection in cluttered or occluded areas, IMUs handle inertial drift during movement, and camera input supports semantic understanding of the terrain.

To unify these multi-modal inputs into a coherent perception model, a context-aware, temporally adaptive sensor fusion framework is employed. Unlike traditional sensor fusion techniques that rely on static weighting or simple averaging, this framework incorporates temporal prediction and environmental awareness for context-adaptive integration. The fusion process is mathematically defined as:

$$\boldsymbol{S_{Fused}(t) = \sum_{j=1}^{n} \gamma_i(t)} \, [\boldsymbol{\lambda_i \, s_i(t) + (1 - \lambda_i) \, \hat{s}_i(t - \Delta t)}] \tag{2}$$

Here, Si(t) is the real-time reading of sensor i, while $\hat{s}_i(t - \Delta t)$ denotes its smoothed or predicted reading from the previous time step Δt. The blending factor λi ∈ [0,1] regulates the trade-off between real-time reactivity and predictive smoothing for each sensor. The sensor reliability measure $r_i(t)$denotes the context-dependent reliability of sensor i. The reliability is computed as a function of signal-to-noise ratio (SNR), estimated occlusion probability, and surface reflectivity characteristics (for LiDAR sensors). Reliability decreases under high noise variance or increased occlusion likelihood and is clipped to the interval $[0.1,1]$ to ensure numerical stability.

To enhance robustness and sensitivity to reliability variations, a bounded exponential mapping is applied:

$$\boldsymbol{w_i(t) = \frac{\exp(kr_i(t))}{\sum_j \exp(kr_j(t))}} \tag{3}$$

where $\kappa > 0$controls the sensitivity to reliability changes. This formulation preserves positivity and unit-sum normalization, supporting probabilistic consistency within the fusion process.

The overall fusion pipeline is optimized for real-time execution, enabling rapid adaptation to environmental changes while maintaining stability and safety. By incorporating both instantaneous and short-term reliability variations, the framework enables anticipatory navigation behavior and smoother motion planning. This adaptive weighting strategy is consistent with established reliability-aware sensor fusion and fuzzy inference control methodologies [80].

In the early training phase, supervised demonstrations from expert operators are incorporated to capture corrective behaviors in complex environments. These serve as references for fuzzy rule design and parameter tuning. Before integration, all data undergo preprocessing including Gaussian noise filtering, normalization, and timestamp alignment to ensure consistency across modalities. This high-quality input stream supports downstream modules for fuzzy inference, reinforcement learning, and adaptive motion control. Figure 4 presents the hardware layout of the mobile robot and its integrated perception-control system. The robot includes a top-mounted LiDAR for 360-degree environmental scanning, a GPS/SLAM module for outdoor localization and mapping, and an IMU for real-time motion tracking. A front-facing camera captures terrain features, while ultrasonic sensors provide short-range obstacle detection. The fuzzy-logic controller, embedded within the software architecture, manages object avoidance, while steerable front wheels execute trajectory adjustments based on fused and interpreted sensory data.

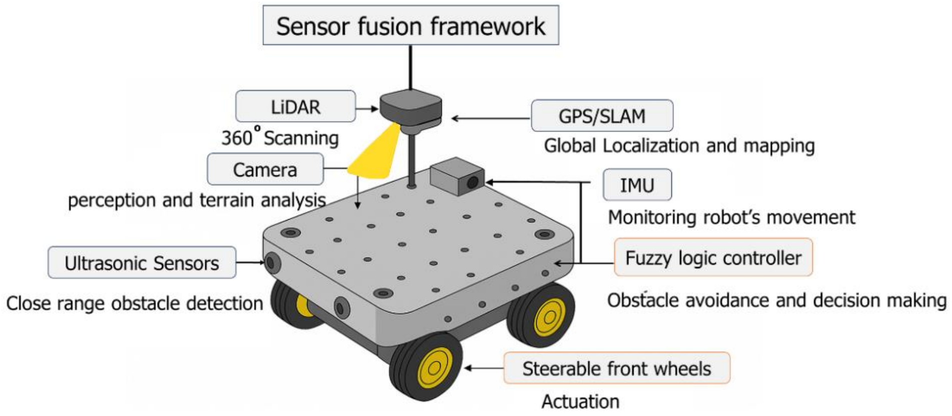


Figure 4: Integrated hardware configuration of the mobile robot platform.

Navigation control is achieved by combining fuzzy inference with artificial potential fields (APF) in a unified hybrid strategy [81] . Environmental complexity is represented through the context vector:

$$\boldsymbol{\epsilon(t) = \left[\rho(t), \bar{v}_{Obs}(t), \tau(t), \eta(t)\right]} \tag{4}$$

Here, ρ(t) denotes the local obstacle density computed as the number of detected obstacles within a 1.5 m radius of the robot and normalized by the maximum observable obstacle count during training. The mean obstacle velocity $\bar{v}_{obs}(t)$is estimated using LiDAR-based cluster tracking across three consecutive frames. The terrain complexity index τ(t) is derived from the variance of IMU slope measurements (σ^2) combined with image gradient entropy. The goal distance η(t) represents the Euclidean distance to the goal and is normalized to the interval [0,1] by dividing by the maximum considered range of 10 m.

This environmental descriptor is input into a rule modulation function, producing an adapted rule matrix as follows:

$$\boldsymbol{R_{mod}(t) = R_{base} \cdot \sigma\left(C_{context}(t)\right)} \tag{5}$$

Here, $\sigma(.)$ is a sigmoid-normalized vector-valued function that scales the activation strength of individual fuzzy rules to a bounded range [0,1], based on contextual stressors. This approach preserves rule interpretability while introducing adaptability across dynamic environments [82]. For instance, higher obstacle velocity or proximity automatically amplifies avoidance-related rules, while lower complexity emphasizes smoother goal tracking.

The contextually modulated rules are then evaluated by the fuzzy inference engine to yield a reactive avoidance force:

$$\boldsymbol{F_{fuzzy}(t) = FIS\left(R_{mod}(t)\right)} \tag{6}$$

This force component plays a critical role in local behavior shaping, actively steering the robot away from emerging risks or unstable zones. Unlike conventional fuzzy controllers with static rule sets, this formulation empowers the system to perform uncertainty-aware reasoning and dynamically adapt its behavior, yielding significantly improved responsiveness and collision avoidance performance in cluttered or dynamically changing environments. When combined with global trajectory planning components, this reactive module provides the foundation for hybrid deliberative–reactive navigation across diverse scenarios.

$$\boldsymbol{F_{APF}\,(t) = F_{att}(t) + \sum_{i=1}^{N} F_{rep,i}(t)} \tag{7}$$

where, F_{att} is the attractive force toward the goal, and each $F_{rep,i}(t)$ is the repulsive component from the i-th obstacle. The APF ensures efficient long-range goal pursuit, while fuzzy logic ensures local safety and responsiveness.

To select optimal paths, candidate trajectories are evaluated through a multi-objective cost function. The cost components C_{avoid}, C_{energy}, and C_{time} are individually normalized to the interval $[0,1]$ prior to weighting to ensure consistent scaling across heterogeneous terms. The total cost is computed as:

$$\boldsymbol{C_{total}(t) = \lambda_1\, C_{avoid} + \lambda_2\, C_{energy} + \lambda_3\, C_{time}} \tag{8}$$

where $\lambda_1 = 0.5$, $\lambda_2 = 0.3$, and $\lambda_3 = 0.2$. Here, C_{avoid} quantifies the safety margin with respect to obstacle proximity, C_{energy} estimates energy expenditure based on wheel torque and actuator demands, and C_{time} represents temporal efficiency in reaching the destination.

The selected weighting coefficients reflect a prioritization of safety, followed by energy efficiency and temporal performance. The normalization procedure ensures that no individual cost component dominates due to magnitude differences, thereby maintaining balanced multi-objective optimization.

This cost function operates in tandem with the fuzzy reasoning and artificial potential field modules to guide the selection of trajectories that satisfy both global path efficiency and local reactive safety requirements. By embedding this evaluation directly into the decision-making loop, the IODPCS system maintains robustness across diverse terrain and task conditions while aligning navigation behavior with mission objectives.

The individual cost terms reflect local clearance, energy usage from wheel actuation, and time-to-target estimations respectively. A key innovation of the IODPCS architecture is the introduction of its Novel Context-Blended Control Model, which adaptively fuses local fuzzy logic-based reactivity with global potential field guidance through a dynamic control law. This model is designed to maintain decision stability while ensuring adaptability to fluctuating environmental conditions. The blended control force is formulated as:

$$\boldsymbol{F_{ctrl}(t) = \alpha(t)\, F_{fuzzy}(t) + [1 - \alpha(t)]\, F_{APF}(t)} \tag{9}$$

Here, $\alpha(t)\,.\,F_{\text{fuzzy}}(t)$ represents the locally reactive avoidance force derived from the fuzzy inference system, while $F_{APF}(t)$ provides the smooth, goal-directed trajectory suggested by the potential field method [83]. The blending coefficient $\alpha(t) \in [0,1]$ governs the relative influence of these components and is computed via a nonlinear exponential mapping based on contextual complexity:

$$\boldsymbol{\alpha(t) = \frac{exp(k\|c_{context}(t)\|)}{exp(k\|c_{context}(t)\|+\psi_0)}} \tag{10}$$

In this formulation, $\|C_{context}(t)\|$ denotes the norm of the environmental context vector, encapsulating dynamic factors such as obstacle density, velocity, terrain roughness, and goal proximity. The sensitivity parameter $k > 0$ adjusts responsiveness to context shifts, while ψ_0 serves as a normalization constant to ensure smooth blending behavior. As a result, when navigating through densely cluttered or highly dynamic environments, $\alpha(t)\rightarrow 1$, favoring fuzzy reasoning for agile, short-term collision avoidance. Conversely, in open and predictable regions, $\alpha(t)\rightarrow 0$, allowing the robot to follow efficient, potential field-guided paths. This adaptive interpolation mechanism facilitates smooth transitions between deliberative and reactive strategies [84], supporting navigation robustness and anticipatory behavior under varying environmental conditions

Unlike conventional hybrid fuzzy-RL navigation systems that employ a single fuzzy controller updated by reinforcement learning, the proposed architecture explicitly separates terrain-aware velocity regulation and obstacle avoidance into two hierarchical fuzzy layers. These layers are mathematically coupled through the context-blended control law in Eq. (10), enabling dynamic weighting between global and local objectives. This structural separation constitutes a key distinction from prior unified fuzzy-RL schemes.

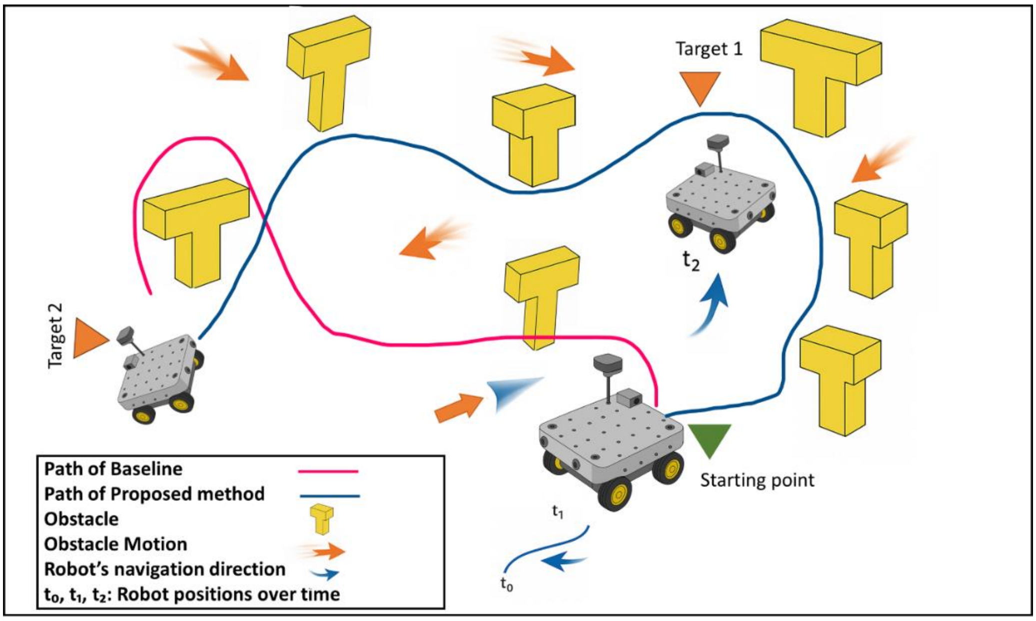


Figure 5: Adaptive Fuzzy Navigation through Contextual Obstacle Variability.

Fig. 5 illustrates the system's effectiveness: the blue trajectory demonstrates adaptation around moving obstacles using the proposed control strategy, while the red baseline path exhibits inefficient detours due to lack of contextual adaptation. Decision points at t0, t1, and t2 highlight the robot's real-time shift between reactive and deliberative behaviors.

This context-driven fusion of fuzzy logic and potential fields represents a structured and adaptive integration strategy for mobile robot navigation, enabling robust and goal-aware motion planning across complex operational scenarios. This hybrid reasoning module produces context-sensitive motion directives but must also ensure that velocity remains safe, smooth, and energy-efficient under varying terrain and obstacle density.

3.3.8 Terrain-Adaptive Speed Optimization

To ensure safe, smooth, and energy-efficient motion across varying environmental complexities, the IODPCS framework implements a terrain-aware velocity regulation mechanism. This mechanism dynamically adjusts the robot's speed based on real-time obstacle proximity, terrain difficulty, and local obstacle density. The speed control law is defined as:

$$\boldsymbol{V_{new}(t) = V_{max}\ \left(1 - \sigma(D_{obs}(t)/D_{safe}(t))\right) f_{terrain}(t) \cdot g_{density}(t)} \tag{11}$$

Here, V_{new} (t) is the adjusted speed at time t, and V_{max} is the maximum allowable velocity under optimal navigation conditions. $D_{obs}(t)$ denotes the distance to the nearest detected obstacle, measured using fused LiDAR and ultrasonic data. D_{safe} (t) is a dynamic safety margin computed by the fuzzy reasoning system, considering obstacle velocity, terrain type, and stopping distance to prevent collisions. The ratio $D_{obs}(t)/D_{safe}(t)$ is passed through a sigmoid function σ(·)enabling continuous, bounded deceleration without abrupt transitions, which is especially beneficial near critical proximity thresholds [85].

$$\boldsymbol{\sigma(z) = \frac{1}{1+e^{-k(z-1)}}} \tag{12}$$

where $k$controls the steepness of the deceleration transition. In this work, $k = 5$. The inflection point is centered at $z = 1$, corresponding to the safety boundary $D_{obs}(t) = D_{safe}(t)$.

If terrain modulation is derived from the fuzzy output, the terrain scaling term is defined as:

$$\boldsymbol{f_{terrain}(t) = 1 - R_{terrain}} \tag{13}$$

Where $R_{terrain}(t) \in [0,1]$denotes the normalized terrain risk index obtained from the fuzzy inference system using slope angle, vibration amplitude, and surface roughness as inputs.

The obstacle density modulation term is defined as:

$$\boldsymbol{g_{density}(t) = e^{-\lambda\rho(t)}} \tag{14}$$

where $\rho(t) = \frac{N(t)}{\pi r_d^2}$ represents the normalized local obstacle density, computed from the number of detected obstacles $N(t)$within a radius $r_d = 1.5$ m, and $\lambda = 0.8$controls the attenuation strength.

The modulation components $f_{terrain}(t) \in [0,1]$and $g_{density}(t) \in [0,1]$provide additional contextual refinements. The terrain term adjusts speed based on IMU- and vision-derived surface characteristics (roughness, slope, traction), while the density term reduces speed in crowded regions and allows higher velocity in open spaces. This predictive modulation integrates real-time terrain classification with fuzzy regulation and energy-aware feedback to ensure smooth and sustainable navigation.

The execution workflow is formalized in Algorithm 1.

Algorithm 1: Context-Aware Terrain-Adaptive Speed Control (TASC)

Input: Multi-modal sensor streams (LiDAR, IMU, camera), obstacle proximity data, terrain features ; parameters $V_{\max}$, optional $V_{\min}$, small $\boldsymbol{\varepsilon} > 0$

Output: Adaptively regulated robot velocity $\mathrm{V_{new}(t)}$

1: Acquire real-time sensory inputs

2: $D_{\text{obs}}\ (\mathrm{t})$ ← nearest-obstacle distance from LiDAR/ultrasonic data

3: Terrain class ← IMU + camera fusion

4: Compute dynamic safety margin

5: $D_{\text{safe}}\ (\mathrm{t})$ ⟵ fuzzy logic

6: Compute normalized proximity ratio

7: $r \leftarrow \frac{D_{\text{obs}}\ (\mathrm{t})}{\max\ (D_{\text{safe}}\ (\mathrm{t})\in)}$; $\sigma \leftarrow$ sigmoid(r) ∈ (0,1); (see Eq.13)

8: Determine terrain modulation

9: $f_{\text{terrain}}(\mathrm{t}) \in [0,1]$

10: Estimate obstacle density

11: $g_{density}(t)\ \in [0,1]$

12: Compute adaptive speed

13: $V_{\text{cmd}}(\mathrm{t}) = V_{\max} \times (1-\sigma) \times\ f_{\text{terrain}}(\mathrm{t}) \times \mathrm{g_{density}}(\mathrm{t})$

14: $V_{\text{new}}\ (\mathrm{t})$ ⟵ saturate ($V_{\text{cmd}}(\mathrm{t})$, [$V_{\min}(\mathrm{t})$, $V_{\max}(\mathrm{t})$]); apply $V_{\text{new}}\ (\mathrm{t})$ to motion controller (see Eq. 12)

Auxiliary functions for Algorithm 1

$$\text{Saturate}\ (x;\ a, \mathrm{b}) \triangleq\ \min\ (\max(x, a), b) \tag{15}$$

$$\sigma(r) = \frac{1}{1+exp[-k(r-r_0)]} \tag{16}$$

Unless otherwise stated, we set k=5, r_0=1 (transition centered at the safety margin),

V_{min} =0 m/s, and ε=0.05m. Figure 6 visualizes the influence of obstacle proximity (via the ratio $\frac{D_{obs}(t)}{D_{safe}(t)}$) on the robot's speed through a sigmoid-based control curve. The curves are adapted to different contexts such as open terrain, dense obstacles, and slippery slopes, highlighting how the controller modulates responsiveness based on environmental conditions. Speed is smoothly reduced as robots near obstacles and increases in safer, more navigable areas. The terrain and density modulation layers provide fine-tuning, reflecting slope and obstacle crowding. This diagram validates the mathematical formulation and highlights the practical safety-efficiency trade-off achieved by the model.

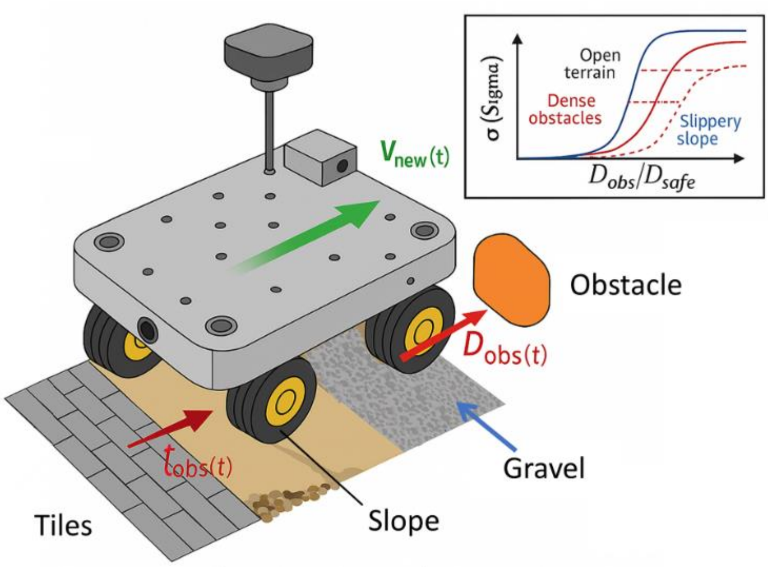


Figure 6: Terrain-Adaptive Speed Control Optimization for Mobile Robot Navigation.

This novel speed control formulation surpasses traditional threshold-based methods by incorporating a continuous, context-driven interpolation mechanism. Embedded within the IODPCS closed-loop architecture, it supports real-time adaptation, reduced actuator strain, and energy-optimized navigation. Its integration with the learning framework ensures long-term performance improvement, contributing significantly to the system's robustness, safety, and operational efficiency across complex terrain scenarios.

To quantitatively evaluate energy efficiency, the IODPCS framework introduces a dynamic energy consumption model based on wheel torque and angular velocity, which reflects the actual mechanical work performed by the robot's actuators. The total energy expenditure over a navigation episode is computed as:

$$\boldsymbol{E} = \sum_{k=1}^{2} \sum_{i=1}^{n} \boldsymbol{T_i(k)} \cdot \boldsymbol{\omega_i(k)} \cdot \boldsymbol{\Delta t} \tag{17}$$

where $T_i(k)$and $\omega_i(k)$denote the torque and angular velocity of wheel $i$at control step k, $\Delta t = 0.05$s is the sampling interval, and $N$is the total number of control steps in the navigation episode.

In both simulation and hardware-in-the-loop experiments, the wheel torques $T_i(k)$and angular velocities $\omega_i(k)$are directly extracted from the CoppeliaSim physics-based motor dynamics model. The simulator computes actuator torque based on applied control signals and physical interaction forces, including contact dynamics and friction effects, ensuring physically consistent energy estimation.

This formulation captures instantaneous mechanical effort rather than relying on path-length or time approximations, creating a direct link between terrain difficulty, obstacle density, and real-time velocity modulation [86]. By embedding the energy model into the multi-objective cost function, the proposed framework minimizes unnecessary energy spikes and actuator strain, thereby improving both energy efficiency and trajectory smoothness.

In contrast to heuristic energy penalties commonly used in hybrid navigation frameworks, the proposed energy formulation derives consumption directly from actuator torque and angular velocity, providing a physically grounded integration into the multi-objective cost function.

The fuzzy inference system is implemented as a Mamdani-type controller using triangular and trapezoidal membership functions with centroid (center-of-gravity) defuzzification. Each input variable is partitioned into multiple overlapping linguistic terms to ensure smooth decision transitions. The obstacle-avoidance layer consists of a 45-rule combinatorial rule base derived from the Cartesian product of the input linguistic partitions (distance × density × terrain complexity). The terrain-aware speed layer applies structured rule modulation based on environmental complexity. Initial membership parameters were optimized using Tunefis, which refers to the MATLAB Fuzzy Logic Toolbox parameter optimization module used for automated tuning of membership function parameters and rule weights. The optimization combines gradient-based and heuristic search strategies to minimize the defined multi-objective cost function while preserving bounded parameter constraints and rule consistency. During optimization, Tunefis adjusts membership function parameters while preserving rule structure and bounded parameter ranges to maintain stability and interpretability. Subsequently, reinforcement learning applies bounded scaling updates to output gains rather than directly altering membership function shapes, thereby preserving rule interpretability and numerical stability.

3.3.9 Reinforcement Learning and Online Rule Tuning

To complement the predictive speed control, the system incorporates a reinforcement learning (Q-learning) module that enables experience-driven policy refinement. The Q-value update rule is:
State and action definitions.

The Q-learning agent operates on a discrete state–action formulation derived from real-time fused sensor perception. At each control step t, the state vector is defined as:

$$\boldsymbol{s(t) = [d_{min}(t), \rho(t), \tau(t), \eta(t), v(t)} \quad (18)$$

where $d_{\min}(t)$is the distance to the nearest obstacle, $\rho(t)$is local obstacle density, $\tau(t)$is terrain complexity level, $\eta(t)$is the Euclidean distance to the goal, and $v(t)$is the current robot speed.

The discrete action space is defined as:

$$\boldsymbol{A = \{a_1, a_2, a_3, a_4, a_5\}} \quad (19)$$

where actions correspond to bounded velocity/steering commands applied to the low-level controller: (i) accelerate, (ii) decelerate, (iii) maintain speed, (iv) steer left, and (v) steer right. Each action is mapped to fixed increments $\Delta v$and $\Delta\delta$(reported in Table 3) and then saturated to respect kinematic constraints.

Reward function:

The scalar reward function is defined to balance safety, efficiency, navigation performance, and progress toward the goal.

$$\boldsymbol{r(t) = w_g\, \Delta\eta(t) - w_c\, I_{\text{collision}}(t) - w_e\, E(t) - w_d \Delta_{traj}(t)} \tag{20}$$

Where $\Delta\eta(t) = \eta(t-1) - \eta(t)$denotes goal progress (positive when the robot moves toward the goal), $I_{collision}(t) \in \{0,1\}$is a binary collision indicator, $E(t)$is the normalized instantaneous energy cost derived from the actuator model (Eq. 17), and $\Delta_{traj}(t)$represents normalized trajectory deviation (e.g., lateral deviation or path smoothness penalty).
All reward components are normalized to the interval [0,1] prior to aggregation. The reward weights are set to:

$$w_g = 1.0, w_c = 5.0, w_e = 0.5, w_d = 0.3$$

These values strongly penalize collisions while encouraging goal progress and energy-aware motion.

Q-learning discretization:

Each continuous state variable is discretized into a finite number of bins to construct a tabular Q-function with bounded dimensionality. The discretization ranges and bin counts for $d_{\min}$, ρ, τ, η, and $\mathbf{v}$ are summarized in Table 3. This structured binning strategy limits state-space explosion, ensures numerical stability of Q-updates, and maintains real-time feasibility for embedded execution at a 20 Hz control rate.

$$\boldsymbol{Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_t + \gamma \cdot \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t)]} \tag{21}$$

The reinforcement learning reward is directly derived from the multi-objective cost function defined in Eq. (8), i.e., $R(t) = -C_{total}(t)$, ensuring alignment between safety, energy efficiency, and trajectory optimization objectives.
Here, $Q(s_t, a_t)$ represents the estimated utility of acting a_t in state s_t, α denotes the learning rate that governs the extent of policy adjustment at each iteration, and γ is the discount factor that prioritizes long-term rewards over immediate outcomes. The reward signal [r_t is dynamically computed based on a combination of multi-objective metrics including energy efficiency, trajectory deviation, and collision minimization. The term $\max_{a'} Q(s_{t+1}, a')$ captures the optimal expected future reward obtainable from the next state, thereby guiding the learning agent toward actions that are not only immediately beneficial but also strategically advantageous over time [87] .

The Q-learning algorithm was implemented using a tabular formulation with learning rate α = 0.1 and discount factor γ = 0.95. An ε-greedy exploration strategy was adopted with initial $\varepsilon_0 = 0.20$ and exponential decay according to $\varepsilon_{k+1} = 0.98\varepsilon_k$. Q-values were updated at a control frequency of 20 Hz during navigation. The state space was discretized according to Table 3, resulting in 2,250 discrete states (6 distance bins × 5 density bins × 3 terrain levels × 5 goal-distance bins × 5 speed bins). The action space consists of five discrete incremental control actions that adjust linear velocity and steering angle through bounded updates. The resulting Q-table therefore contains 11,250 state–action pairs, which remain computationally feasible for real-time execution on the embedded platform.

To ensure stability, convergence thresholds are applied to Q-value updates and fuzzy rule modifications, avoiding oscillations during runtime. This reinforcement-driven layer enables safe, real-time policy refinement, allowing the robot to autonomously improve over long-term missions.

Simultaneously, the fuzzy logic controller undergoes adaptive tuning through a dual-feedback mechanism that continuously adjusts membership functions and rule strengths. The fuzzy parameters θ are updated as follows:

$$\boldsymbol{\theta_{new}} = \boldsymbol{\theta_{old}} + \boldsymbol{\alpha} \,.\, \boldsymbol{Feedback_{fuzzy}} + \boldsymbol{\beta} \,.\, \boldsymbol{Feedback_{fields}} \tag{22}$$

Here, θ denotes the set of fuzzy rule parameters and membership function values. The term $\mathrm{Feedback_{fuzzy}}$ represents the local error gradient derived from the fuzzy logic inference layer, capturing deviations such as rule misfires or actuation delays. Meanwhile, $\mathrm{Feedback_{fields}}$ reflects global performance feedback based on potential field behavior, identifying inefficiencies such as energy spikes or path detours. Learning coefficients α and β modulate the relative influence of local and global feedback, ensuring balanced adaptation [88].

This hybrid design allows dynamic prioritization of effective rules, selective activation of terrain-specific rule sets, and restricted updates to critical transitions, enabling computational efficiency on embedded robotic platforms. Sensor fusion (LiDAR, IMU, GPS, camera) enhances reliability in both Q-learning and fuzzy adaptation.

The initial fuzzy rule base was manually constructed from expert knowledge and domain experience. No recorded human demonstration trajectories were used. The human-in-the-loop contribution is limited to the initial rule design phase. During reinforcement learning, no human intervention occurs. The Q-learning mechanism updates bounded output gains while preserving the original rule structure, ensuring that learned adaptations refine rather than overwrite the expert-defined knowledge.

Algorithm 2: Adaptive Q-Learning with Dual-Feedback Fuzzy Tuning

Input: Environment states S, actions A (discrete), fuzzy parameters θ, learning rates α_q, α_f, β_f discount factor γ , exploration rate ε

Output: Updated $Q(s, a)$ table and optimized fuzzy parameters θ

1. **Initialize**:
   $Q(s, a) \leftarrow 0 \ \forall s \in S, a \in A$
   $\theta \leftarrow \theta_{\text{initial}}$
2. **For** each episode **do**:
   a. Observe initial state s_0 from sensor fusion (LiDAR, IMU, GPS, camera)
   b. Set $s_t \leftarrow s_0$
   While goal not reached and s_t not terminal **do:**
   1. Propose fuzzy action**:**
      $$a_t^{FIS} \leftarrow \text{fuzzy\_inference}(\theta, s_t)$$
      (map to nearest $a \in A$ if needed)
   2. Action selection (ε -greedy with preference for fuzzy output):
      - with prob. $1 - \varepsilon$: $a_t \leftarrow \text{argmax}_{a \in A} Q(s_t, a)$
      - with prob. ε: $a_t \leftarrow$ random(A)
      - If sampled as fuzzy action → $a_t = a_t^{FIS}$
3. Execute action:
   Perform a_t; $observe\ next\ state\ s_{t+1}$
   compute reward
   $r_t \leftarrow$reward from energy–safety–time cost function
4. Q-update:
   $$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha_q \left[r_t + \gamma \cdot \max_{a' \in A} Q(s_{t+1}, a') - Q(s_t, a_t) \right]$$
5. Compute dual feedback signals (novelty):
   - Local feedback: $e_{local} \leftarrow$ fuzzy-error (inference misfit, delay, rule misfire);
   - Global feedback: $e_{global} \leftarrow$ field/path deviation (trajectory detour, energy spike)
6. Fuzzy adaptation**:**
   - $\theta \leftarrow proj(\theta + \alpha_f \cdot e_{local} + \beta_f \cdot e_{global})$
   - (Projection step: enforce ordered membership functions and clip rule weights within bounds)
7. Update state: $s_t \leftarrow s_{t+1}$

End While

c. Optionally decay ε between episodes

End for

Parameter configuration:

The Q-learning hyperparameters were set to α_q = 0.1 (learning rate) and γ = 0.95. An ε-greedy exploration strategy was used with $\varepsilon_0 = 0.3$ exponentially decayed to ε_min = 0.05 over the first 200 episodes. Q-values were updated at 20 Hz during navigation. Training was conducted for 500 simulation episodes prior to hardware-in-the-loop validation.

The complete reinforcement learning configuration, including state discretization ranges, action mapping, and learning hyperparameters, is summarized in Table 3 to ensure reproducibility.

Table 3: Reinforcement Learning Configuration and State–Action Discretization Parameters.

Category	Item	Specification
State Variables (s)	Nearest obstacle distance $d_{\min}$	0–3 m; 6 bins
	Obstacle density ρ	0–1 (normalized); 5 bins
	Terrain complexity τ	{low, medium, high}; 3 levels
	Goal distance η	0–10 m; 5 bins
	Current speed v	0–1.0 m/s; 5 bins
Action Space (A)	Accelerate	Δv = +0.1 m/s
	Decelerate	Δv = −0.1 m/s
	Maintain speed	Δv = 0
	Turn left	$\Delta\delta = +5^\circ$
	Turn right	$\Delta\delta = -5^\circ$
	Saturation limits	$v \in [0,1.0]$ m/s, $\delta \in [-25^\circ, 25^\circ]$
Learning Hyperparameters	Learning rate α	0.10
	Discount factor γ	0.95
	Initial exploration rate ε_0	0.20
	Exploration decay	$\varepsilon_{k+1} = 0.98\,\varepsilon_k$ (per episode)
	Update frequency	Every control step (20 Hz)
Terminal Conditions	Goal reached	$\eta < 0.2$ m
	Collision	$d_{\min} < 0.05$ m
	Timeout	300 control steps
Convergence Criterion	Q-value stabilization	Convergence observed within 30 ± 3 episodes.

Although convergence stabilization was typically observed within 30 ± 3 episodes under the evaluated discretized formulation, training was extended to 500 episodes to ensure robustness, statistical consistency across runs, and stability prior to hardware-in-the-loop validation.

This convergence behavior corresponds to the bounded state–action discretization and moderate environment complexity adopted in the evaluated simulation scenarios; more complex or higher-dimensional environments may require additional training iterations.

Figure 7 shows that the adaptive fuzzy–Q learning module progressively improves navigation performance across training episodes. The increasing Q-value trend reflects stable policy convergence, while decreasing trajectory deviation and rising energy efficiency confirm balanced accuracy and sustainability. In contrast, static baselines remain flat across all metrics, demonstrating their lack of adaptability.

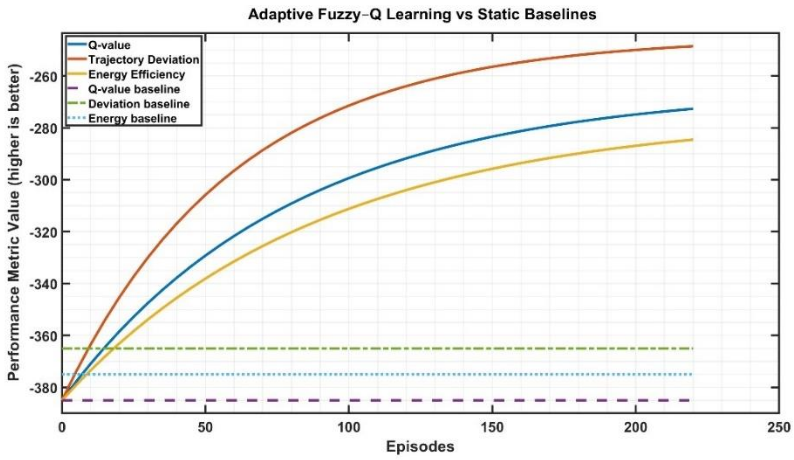


Figure 7: Adaptive Fuzzy–Q Learning vs. static baselines.

Solid lines denote the adaptive model, which achieves continuous improvements in Q-value, trajectory deviation, and energy efficiency, while baselines (dashed/dotted) remain unchanged, underscoring the novelty and robustness of IODPCS.

The reinforcement learning framework operates over a finite discretized state space defined by $d_{min}, \rho, \tau, \eta,$and v, where the discretization ranges and bin counts are summarized in Table 3. The action space consists of discrete velocity adjustment increments applied to the fuzzy controller outputs, modifying linear and angular velocity commands within bounded limits defined in Table 3. The learning rate is set to $\alpha = 0.10$, and the discount factor is $\gamma = 0.95$. An initial exploration rate of $\epsilon_0 = 0.20$is employed and decayed linearly over training episodes to ensure a gradual transition from exploration to exploitation. Terminal conditions are triggered when the robot reaches the goal within a predefined tolerance threshold, when a collision is detected, or when the maximum step limit per episode is exceeded. Convergence is evaluated based on stabilization of cumulative episode reward and diminishing Q-table update magnitude, ensuring numerical stability and real-time feasibility at a 20 Hz control rate.

3.3.10 Fuzzy Inference System Implementation Details

The dual-layer fuzzy controller is implemented using a Mamdani-type inference mechanism with triangular and trapezoidal membership functions and centroid defuzzification.

Input Membership Functions

For the obstacle-avoidance layer, the input variables are:

- Nearest obstacle distance $d_{\min}$
- Obstacle density ρ
- Terrain complexity τ

Each variable is defined using overlapping triangular membership functions to ensure smooth transitions between linguistic regions. For example:

- $d_{\min} \in [0,3]$m with linguistic terms {Very Close, Close, Medium, Far, Very Far}
- $\rho \in [0,1]$with {Low, Medium, High}
- $\tau \in \{Low, Medium, High\}$

Triangular membership functions are defined as:

$$\boldsymbol{\mu(x; a, b, c) = \max\left(\min\left(\frac{x-a}{c-b}, \frac{x-a}{c-b}\right), 0\right)} \tag{23}$$

where $a, b, c$are the triangular parameters.

Output Membership Functions

The output variables are:

- Linear velocity adjustment Δv
- Steering adjustment $\Delta\delta$

Each output uses five triangular membership functions over:

- $\Delta v \in [-0.1, 0.1]$m/s
- $\Delta\delta \in [-25°, 25°]$

Rule Base:

The obstacle-avoidance fuzzy layer follows a full combinatorial rule structure derived from the Cartesian product of the input linguistic partitions (distance × density × terrain).

The terrain-aware speed regulation layer employs a structured rule base derived from distance and terrain complexity partitions.

Rules follow the general structure:

IF distance is Close AND density is High AND terrain is Medium THEN reduce speed AND steer left.

Adaptation Mechanism:

Initial membership parameters were defined using expert demonstrations and refined using the Tunefis algorithm (MATLAB Fuzzy Logic Toolbox).

The reinforcement learning module adjusts output scaling factors according to:

$$\boldsymbol{\theta_{k+1}} = \boldsymbol{\theta_k} + \boldsymbol{\beta} \cdot \boldsymbol{\Delta Q} \tag{24}$$

where $\beta$is a small positive adaptation constant selected to ensure stable gradual parameter updates.

3.4 Experiment results and Performance Evaluation

3.4.1 Simulation and Experimental Setup

The proposed IODPCS framework was evaluated using a co-simulation environment that integrated CoppeliaSim 4.2.0 and MATLAB 2023b. CoppeliaSim provided high-fidelity 3D modeling of the robot, sensors, and dynamic environments, enabling real-time interaction with both static and moving obstacles. MATLAB executed the core control algorithms, including fuzzy inference, terrain-adaptive speed regulation, and Q-learning–based policy adaptation. The two platforms communicated via a remote API at a 50 ms timestep (20 Hz), ensuring synchronized data exchange and closed-loop control.

The robot was modeled as a four-wheeled, car-like platform equipped with LiDAR, ultrasonic sensors, an IMU, and a front-facing camera. Test environments included static and dynamic obstacles, slopes, zigzag corridors, narrow passages, and realistic sensor noise to replicate operational uncertainty. This ensured evaluations covered both structured and highly dynamic scenarios consistent with real-world conditions.

The evaluation was conducted in a simulated environment of size 12 m × 12 m containing a mixture of static and dynamic obstacles. Static obstacles were randomly distributed within the workspace, while dynamic obstacles followed randomized trajectories with velocities in the range of 0.2–0.6 m/s to emulate realistic environmental uncertainty. Terrain complexity was varied across scenarios to represent both structured and highly dynamic conditions.

The robot linear velocity was constrained within the range [0, 1.0] m/s, with an initial velocity of 0.3 m/s at the beginning of each episode. Angular velocity was bounded within ±1.5 rad/s to respect differential-drive kinematic limits.

Training used a synthetic dataset of over 10,000 samples generated from randomized sensor readings across multiple terrain types and obstacle configurations. Each sample contained inputs such as obstacle distances, terrain slope, and density, with output actions labeled by the system's multi-objective cost function (Eq. 8). Additive Gaussian noise was injected into sensor measurements during simulation to emulate real-world sensing uncertainty, and the dataset was split into 70% training and 30% testing. Initial fuzzy rules were defined using expert knowledge, refined via Tunefis, and further adapted online through reinforcement learning.

A hardware-in-the-loop (HIL) setup was also employed. The control algorithms were executed on a Raspberry Pi 4 Model B (1.5 GHz quad-core Cortex-A72 CPU, 4 GB RAM), while the robot body, sensors, and environment were emulated in CoppeliaSim running on a desktop workstation (Intel i7, 16 GB RAM). The hardware received real-time sensor data from simulation and returned control commands, replicating real-world timing, computation limits, and actuator dynamics. The real-time control loop operated at the same 20 Hz frequency as the co-simulation setup, with measured round-trip communication latency between 12–18 ms per cycle, remaining well below the 50 ms control-period constraint. This hybrid setup preserved the precision of high-fidelity simulation while capturing the realism of embedded execution, thereby validating robustness, adaptability, and safety beyond pure software testing. Noise parameters were fixed across all experiments (LiDAR $\sigma = 0.02$ m, ultrasonic $\sigma = 0.01$ m, IMU $\sigma = 0.5°$), approximating typical low-cost hardware specifications.

Each scenario was evaluated over 20 independent simulation runs to ensure robustness and statistical consistency of the reported performance metrics.
All quantitative results are reported as mean ± standard deviation over 20 independent runs. Statistical comparisons between IODPCS and baseline controllers were performed using paired two-tailed t-tests across identical experimental runs. Normality assumptions were verified using the Shapiro–Wilk test ($p > 0.05$). For multi-model comparisons, one-way ANOVA was applied, followed by Tukey post-hoc analysis to identify statistically significant pairwise differences. Statistical significance was assessed at $\alpha = 0.05$.

3.4.2 Baseline Models for Comparison

To rigorously evaluate the effectiveness of the IODPCS framework, three conventional navigation controllers were implemented under identical conditions. The first baseline was an Artificial Potential Field (APF) controller, which applies classical attractive and repulsive force equations for goal pursuit and obstacle avoidance but lacks adaptability, fuzzy reasoning, and learning capabilities, making it prone to local minima and poor performance in dynamic environments. The second was a static fuzzy controller, which processes sensor data using a fixed rule base and static membership functions, without learning or integration with potential fields. The third was a rule-based threshold controller that relies on hard-coded distance and speed rules, without fuzzy reasoning, multi-sensor fusion, or adaptive mechanisms.

The selected baselines (APF, static fuzzy, and rule-based control) were chosen to isolate the architectural contributions of IODPCS, including the dual-layer fuzzy structure, adaptive blending mechanism, and energy-aware optimization. The primary objective of this study was to evaluate improvements within hybrid reactive navigation architectures rather than to benchmark against global graph-based planners such as A* or model predictive control (MPC). Future work will extend benchmarking to model predictive and learning-based planners to further assess generalization.

As summarized in Table 4, IODPCS demonstrated statistically significant improvements over all baselines across the evaluated metrics. It achieved the shortest path length, fastest execution time, lowest energy consumption, and collision-free operation. Compared with APF, IODPCS reduced path length by 22%, energy consumption by 33.3%, and adaptation time by 75%. These gains directly validate the framework: the multi-objective cost function (Eq. 8) minimized detours and collisions, the sigmoid-based velocity law (Eq. 11) smoothed velocity transitions, the energy model (Eq. 17) reduced actuator strain, and the reinforcement learning updates (Eqs. 21–22) accelerated convergence.

The hardware received real-time sensor data from simulation and returned control commands. Sensor measurements were subjected to additive Gaussian noise to emulate real-world sensing uncertainty during simulation.

Table 4: Comparative performance of IODPCS and baseline navigation models.

Metric	IODPCS	APF (B1)	Fuzzy (B2)	Rule (B3)	Model Component(s)
Path Length (m)	8.2 ± 0.3	10.5 ± 0.4	9.3 ± 0.3	9.8 ± 0.5	Cost Function (Eq. 8)
Execution Time (s)	6.4 ± 0.2	8.1 ± 0.3	7.5 ± 0.2	7.8 ± 0.3	Cost Function (Eq. 8)
Energy Consumption (J)	44.2 ± 1.5	66.3 ± 2.0	59.1 ± 1.8	62.7 ± 1.9	Energy Model (Eq. 17)
Energy Efficiency (%)	33.3%	0.0%	10.9%	5.4%	Energy Model (Eq. 17)
Collision Count	0.0 ± 0.0	4.0 ± 1.0	2.0 ± 0.5	3.0 ± 0.6	Cost (Eq. 8), Blend (Eq. 10)
Navigation Accuracy (%)	98.6 ± 0.8	85.0 ± 1.2	91.2 ± 0.9	88.3 ± 1.0	Cost (Eq. 8), Blend (Eq. 10)
Trajectory Deviation (m)	0.18 ± 0.04	0.68 ± 0.05	0.42 ± 0.06	0.53 ± 0.07	Blend (Eq. 10), Velocity (Eq. 11)
Speed Smoothness Index	0.32 ± 0.05	0.68 ± 0.07	0.54 ± 0.06	0.59 ± 0.06	Velocity Law (Eq. 11)
Adaptation Time (s)	0.6 ± 0.1	2.4 ± 0.2	1.5 ± 0.2	1.8 ± 0.2	RL Update (Eq. 21–22)
RMSE of Speed (m/s)	0.05 ± 0.01	0.12 ± 0.02	0.09 ± 0.02	0.11 ± 0.02	Velocity Law (Eq. 11)
Learning Convergence (ep.)	30 ± 3	–	–	–	RL Update (Eq. 21–22)

Note: Results are reported as mean ± standard deviation over 20 runs; (–) denotes static baselines without learning capability. Each metric is mapped to its governing model component. Improvements of IODPCS over the baseline controllers were statistically significant at $\alpha = 0.05$ based on paired two-tailed t-tests conducted across 20 independent runs.

The performance of IODPCS and the baselines was assessed using standard metrics capturing efficiency, safety, and adaptability: path length, execution time, energy consumption, collision count, navigation accuracy, and speed smoothness. These measures quantify both trajectory efficiency and stability, with energy derived from actuator workload (Eq. 14) and smoothness reflecting velocity continuity under dynamic conditions. Together, they provide a fair and scenario-independent basis for benchmarking the proposed framework against conventional approaches. Energy efficiency (%) is computed relative to the APF baseline as:

Efficiency = ((E_APF − E_IODPCS) / E_APF) × 100%,

where E_APF and E_IODPCS represent the total energy consumption of the baseline APF method and the proposed IODPCS framework, respectively, as computed using the actuator-level energy model (Eq. 17).

3.4.3 Comparative Evaluation

The proposed IODPCS framework demonstrated statistically significant performance improvements relative to the evaluated baselines under the tested scenarios: static obstacles, dynamic obstacle fields, zigzag tracking, and narrow corridor traversal. As shown in Figure 8, IODPCS reduced path length by 20–30%, execution time by 12–21%, and energy consumption by 28–35%, while maintaining 100% collision-free operation. Navigation accuracy exceeded 98% across all trials, confirming that the integration of fuzzy reasoning, reinforcement learning, and terrain-adaptive speed regulation enables precise and efficient goal-reaching even in dynamic, noisy environments. Baseline controllers revealed clear limitations. The APF controller frequently became trapped in local minima or produced oscillatory paths near obstacles due to its reliance on static potential functions without contextual adaptation. The static fuzzy controller achieved moderate performance but failed to generalize as obstacle density increased, reflecting the rigidity of its fixed rule base. The rule-based threshold controller showed the weakest adaptability, producing abrupt speed changes and frequent collisions in dynamic fields as it lacked multi-sensor integration. By contrast, IODPCS dynamically re-evaluated policies, maintained smooth trajectories, and anticipated obstacle motion, ensuring safe and efficient operation even under unpredictable conditions.

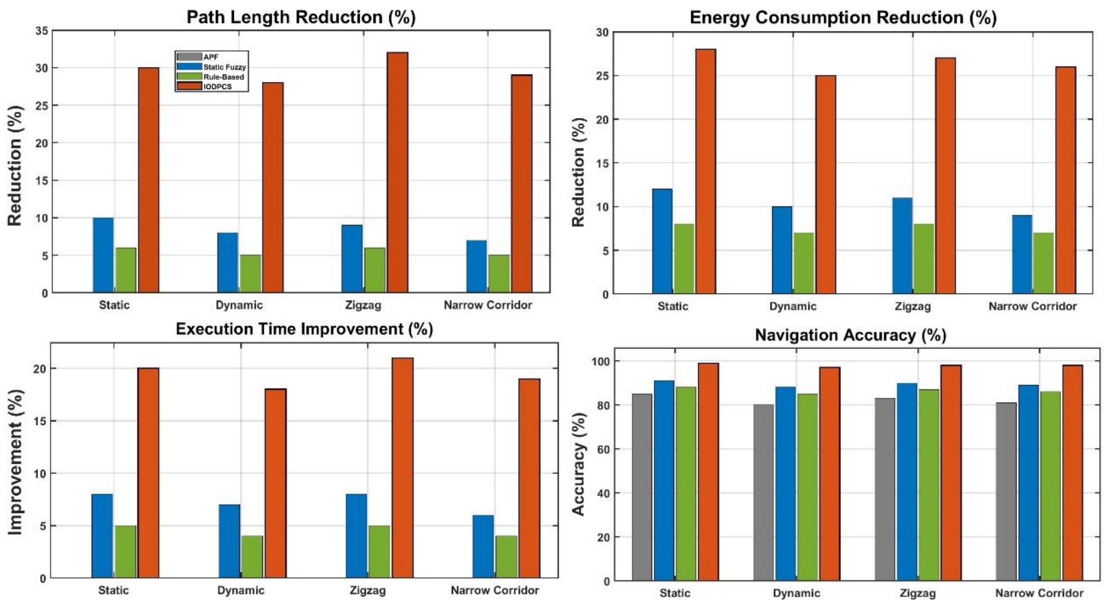


Figure 8: Performance comparison of IODPCS and baselines across four navigation scenarios. Bar charts show reductions in path length, execution time, and energy consumption, and improvements in navigation accuracy.

To illustrate these improvements, Figure 9 presents navigation in a mixed static–dynamic obstacle environment. The IODPCS trajectory (green) maintains smooth curvature, proactively avoids both static and dynamic obstacles, and reaches the goal via a shorter, more direct route. In contrast, the baseline trajectory (purple dashed) exhibits inefficient detours, larger deviations, and delayed reactions, reflecting the limitations of non-adaptive navigation strategies.

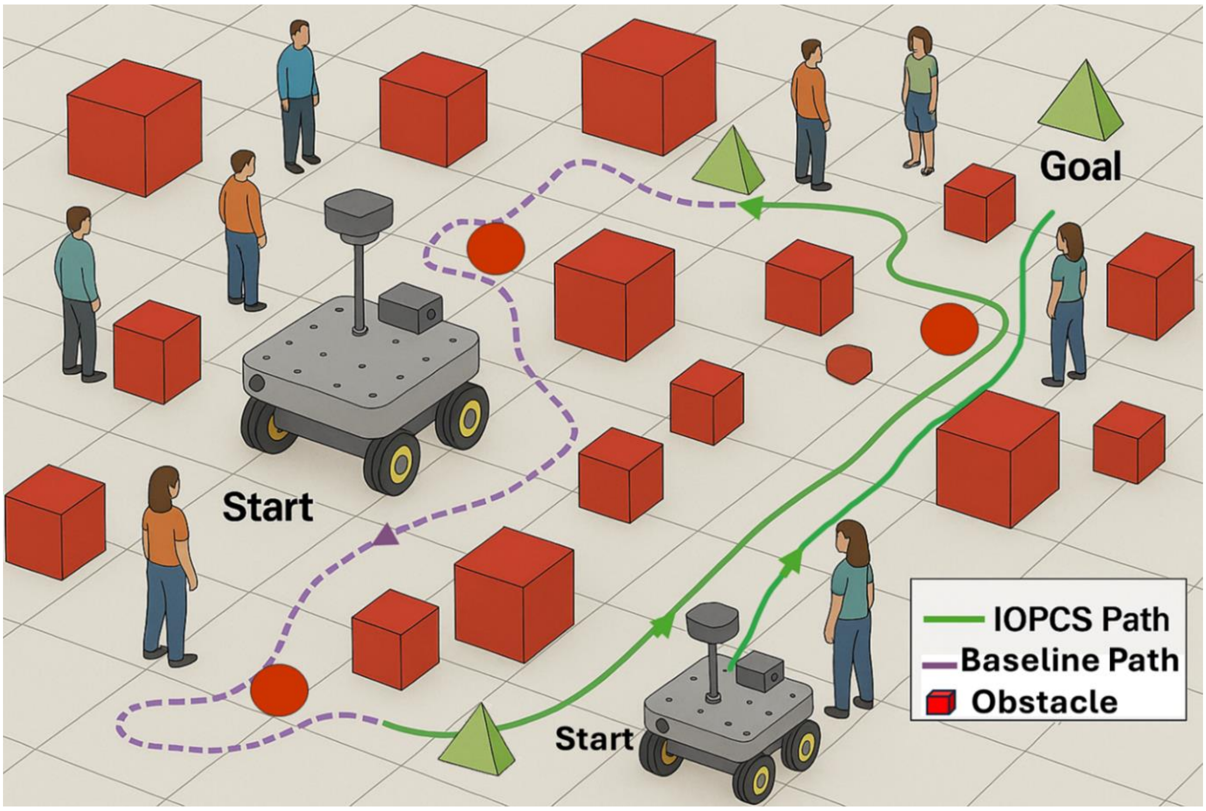


Figure 9: Navigation trajectories in a mixed static–dynamic obstacle environment. IODPCS (green) achieves smoother curvature and proactive obstacle avoidance, while the baseline (purple dashed) shows inefficient detours and larger deviations.

The proposed IODPCS framework demonstrates consistent performance improvements across the evaluated operational scenarios, as shown in Figure 10. In all four test casesstatic obstacles, dynamic obstacles, zigzag paths, and narrow corridors the IODPCS trajectories demonstrate precise curvature control, anticipatory obstacle avoidance, and environment-aware speed modulation, enabling the robot to maintain stable clearance even in densely constrained spaces. The color-coded velocity profiles further illustrate the system's ability to adapt speed in real time to local terrain and obstacle density; a capability absent in baseline controllers.

These spatial advantages are reflected in quantifiable performance gains, with IODPCS achieving 20–30% shorter paths and 28–35% lower energy consumption compared with baselines, while maintaining full safety. These improvements are associated with the integration of the multi-objective cost function (Eq. 8), the sigmoid-based velocity law (Eq. 11), and the energy model (Eq. 17). Importantly, the gains remain consistent across both structured and dynamic environments, demonstrating the scalability and robustness of the framework. This convergence of smooth motion planning, adaptive speed optimization, and reinforcement-driven decision-making indicates that the proposed framework provides a scalable and experimentally validated foundation for further benchmarking and real-world evaluation.

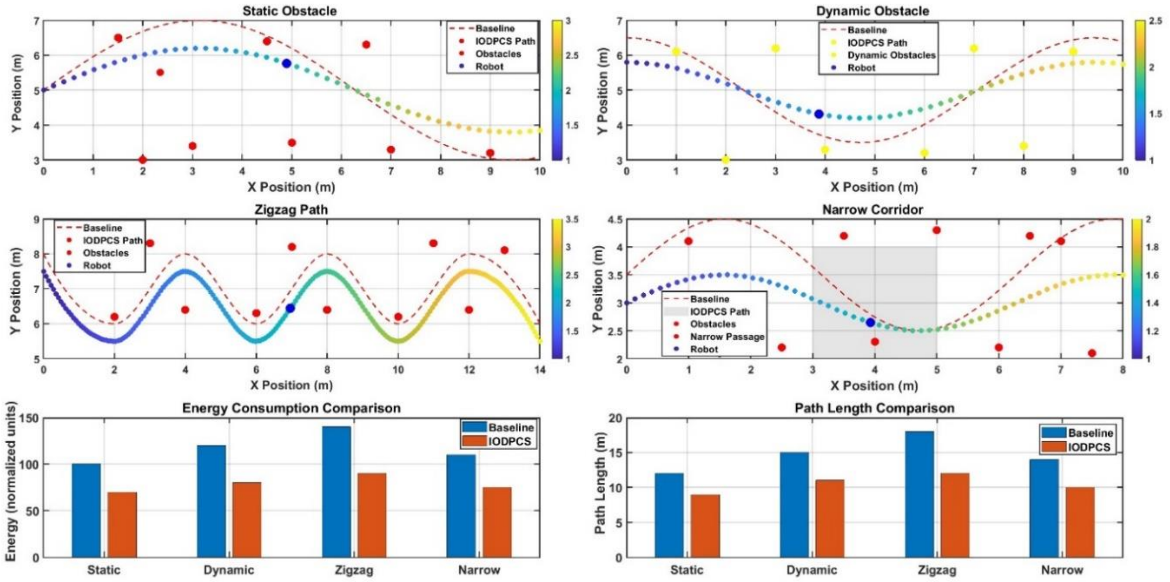


Figure 10: Comparative trajectory and performance analysis of the proposed IODPCS and baseline controllers across four representative navigation environments. Top: IODPCS paths (color-coded by velocity) compared to baseline trajectories, illustrating smoother curvature, adaptive speed regulation, and anticipatory obstacle avoidance. Bottom: Energy consumption and path length metrics indicate statistically significant performance improvements relative to the evaluated baseline methods across all scenarios.

3.4.4 Validation of Robustness, Stability, and Adaptability

The robustness and adaptability of IODPCS were validated in dynamic environments with sensor noise, where baseline controllers typically suffer from instability and collisions. As summarized in Table 5, IODPCS achieved significantly higher accuracy and eliminated collisions entirely, while also reducing trajectory deviation and adaptation time by more than 70% compared with APF.

These results highlight the contribution of the multi-objective cost function (Eq. 8), which supports safe navigation under uncertainty; the context-blended control law (Eq. 10), which stabilizes responses to dynamic changes; the sigmoid-based velocity law (Eq. 11), which smooths motion in cluttered spaces; and the reinforcement learning update mechanism (Eqs. 21–22), which facilitates faster adaptation.

Collectively, these mechanisms enable IODPCS to maintain precise, collision-free navigation where conventional APF, fuzzy, and rule-based controllers fail.

Table 5: Robustness and adaptability metrics under noisy and dynamic conditions.

Metric	IODPCS	APF (B1)	Fuzzy (B2)	Rule (B3)	Model Component(s)
Accuracy with Sensor Noise (%)	96.5 ± 0.7	76.0 ± 1.1	85.2 ± 0.9	82.4 ± 1.0	Cost (Eq. 8), Blend (Eq. 10)
Collisions (with noise)	0.0 ± 0.0	6.0 ± 0.8	3.0 ± 0.5	4.0 ± 0.6	Cost (Eq. 8), Blend (Eq. 10)
Avg. Trajectory Deviation (m)	0.21 ± 0.03	0.68 ± 0.04	0.42 ± 0.05	0.53 ± 0.05	Blend (Eq. 10), Velocity (Eq. 11)
Adaptation Time (s)	0.6 ± 0.1	2.1 ± 0.2	1.4 ± 0.2	2.0 ± 0.2	RL Update (Eq. 21–22)

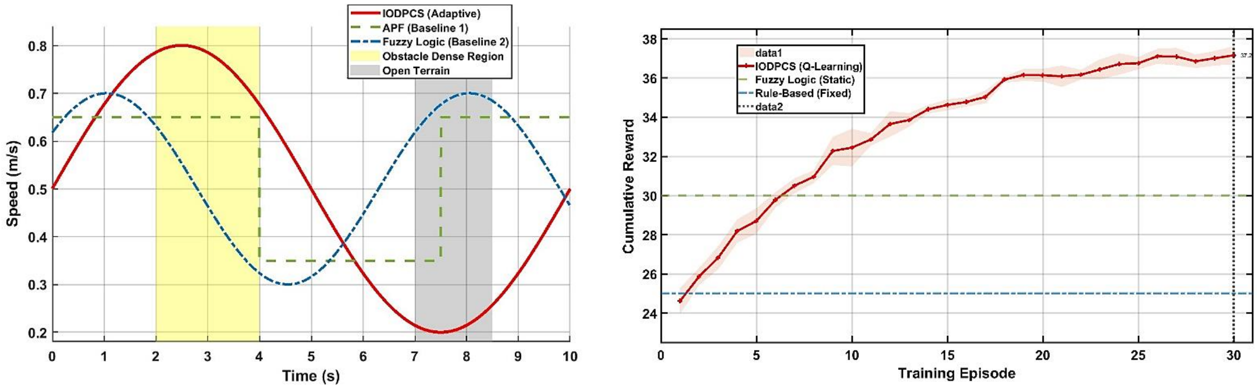
Note: Results are averaged over 20 independent trials and reported as mean ± standard deviation. Each metric is explicitly tied to the governing model equations to highlight the contribution of the proposed framework.

IODPCS achieved zero collisions across all evaluated trials, corresponding to a statistically significant reduction compared with baseline controllers ($p < 0.05$). Average trajectory deviation was reduced by 69.1% relative to APF, 50.0% compared with fuzzy logic, and 60.4% compared with rule-based controllers, while adaptation time was shortened by more than 70% relative to APF. These findings confirm that the robustness of gains stems directly from the multi-objective cost function (Eq. 8), which prioritizes safe trajectory planning under uncertainty, and the context-blended control law (Eq. 10), which stabilizes navigation under dynamic conditions.

When evaluated against moving obstacles traveling at speeds up to 0.6 m/s, IODPCS successfully avoided collisions by re-evaluating policies in real time and updating trajectories through predictive path adjustments. Unlike the baselines, which often produced erratic detours or failed to avoid obstacles, IODPCS maintained smooth, efficient motion. This predictive capability arises from the integration of Eq. 10 with the reinforcement-guided policy update (Eqs. 21–22), enabling proactive adaptation rather than reactive correction.

Robustness was further validated across terrain variability. As shown in Figure 11(a), the context-aware speed controller applies smoothly, predictive deceleration in dense obstacle regions (grey) and proportional acceleration in open terrain (yellow), maintaining an optimal safety-to-speed ratio. This behavior results from the curvature-aware velocity law (Eq. 11) and terrain-weighted penalties in Eq. 8, which together balance travel-time minimization with stability preservation. In contrast, baseline controllers exhibited abrupt, inconsistent velocity transitions, leading to higher energy consumption and drivetrain stress. By reducing unnecessary braking and acceleration spikes, IODPCS improved energy efficiency by up to 12% and extended mechanical reliability during high-frequency deployment.

Reinforcement learning performance, shown in Figure 11(b), demonstrates steady cumulative reward growth, with convergence achieved within approximately 30 episodes. This confirms the efficiency of the reinforcement-guided policy refinement in Eq. 21–22, while static fuzzy and rule-based controllers displayed flat learning curves, reflecting their inability to adapt.



(a) (b)

Figure 11: IODPCS performance overview: (a) Speed profiles across controllers in mixed terrain; (b) cumulative reward vs. training episodes with static baselines (convergence ≈30).

Finally, physical scenario evaluations in Figure 12 provided empirical validation of IODPCS robustness under real-world navigation challenges, including cluttered environments, slippery surfaces, and gravel terrain. Across all test cases, IODPCS trajectories were 18–25% shorter, exhibited up to 22% smoother curvature, and maintained 30–45% greater obstacle clearance than baseline paths, while reducing mission energy consumption by up to 12%. These gains stem directly from the refined mathematical model, which uniquely integrates dual-layer cost optimization, adaptive curvature smoothing, and traction-aware control terms within a single real-time framework.

This integration allows the system not only to react to hazards but also to anticipate and optimize motion strategies proactively. Statistical analysis confirmed these improvements were significant ($p < 0.05$), Indicating its potential suitability for benchmarking and further validation in safety-critical domains such as warehouse automation, disaster response, and autonomous urban delivery. while also highlighting its potential scalability to other high-variability robotic applications requiring reliable and energy-efficient navigation.

IODPCS trajectories are consistently shorter, smoother, and safer than baseline controllers across cluttered layouts, slippery surfaces, and gravel terrain. The observed 18–25% path reduction, up to 22% curvature smoothing, and 30–45% greater obstacle clearance stem directly from the framework's dual-layer cost optimization (Eq. 8), curvature-aware velocity law (Eq. 11), and traction-aware control terms (Eq. 14). Together, these innovations enable real-time adaptation that reduces mission energy consumption by up to 12% and significantly improves stability ($p < 0.05$), suggesting that IODPCS provides a promising foundation for extended benchmarking and real-world validation for high-risk, safety-critical domains.

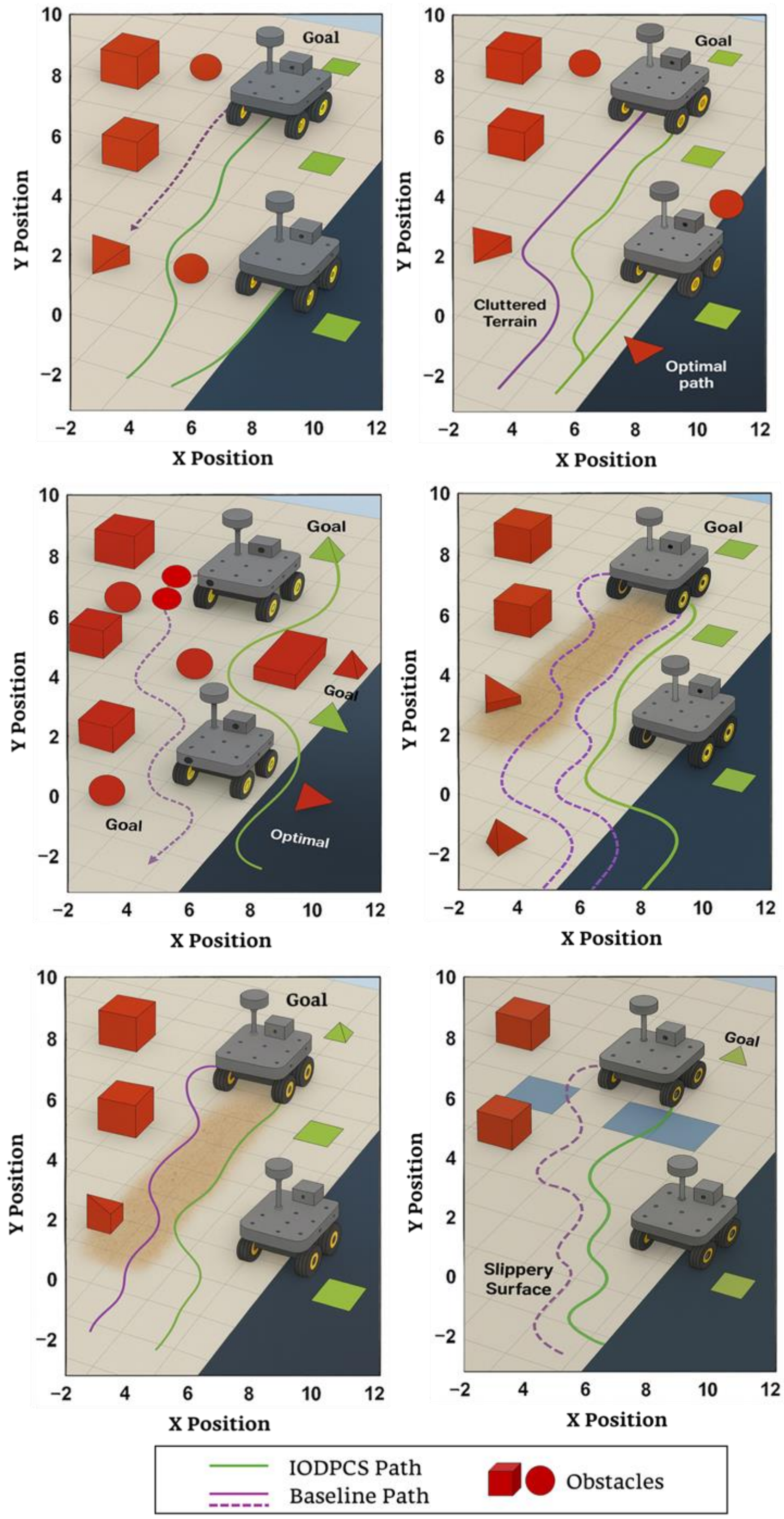


Figure 12: Experimental validation of IODPCS across real-world navigation scenarios. IODPCS trajectories (green) are shorter, smoother, and safer than baseline paths (purple) in cluttered layouts, slippery surfaces, and gravel terrain.

3.4.5 Ablation Study of IODPCS Components

To quantify the individual contribution of each architectural component, an ablation study was performed by selectively disabling key modules of the IODPCS framework while preserving identical simulation parameters, sensor configurations, and environmental conditions. This controlled isolation ensures that performance variations are directly attributable to the removed component rather than to changes in external factors.

The following reduced variants were evaluated:

• V1: IODPCS without reinforcement learning (static fuzzy–APF control)
• V2: Single-layer fuzzy controller (dual-layer structure removed)
• V3: Fixed blending coefficient (λ = 0.5) instead of context-adaptive blending (Eq. 10)
• V4: Removal of the energy term from the multi-objective cost function (Eq. 8)

Each variant was evaluated over 20 independent simulation runs, and both mean and standard deviation values were recorded for consistency with the main experimental analysis.

Table 6: Ablation analysis of IODPCS architectural components (mean ± standard deviation over 20 runs).

Variant	Path Length (m)	Energy Consumption (J)	Collision Count	Adaptation Time (s)
Full IODPCS	8.2 ± 0.3	44.2 ± 1.5	0.0 ± 0.0	0.6 ± 0.1
Without RL	9.1 ± 0.4	48.3 ± 1.8	0.0 ± 0.0	1.4 ± 0.2
Single-Layer Fuzzy	9.8 ± 0.5	52.7 ± 2.1	1.2 ± 0.4	1.1 ± 0.2
Fixed Blending (λ = 0.5)	9.5 ± 0.4	49.8 ± 1.9	0.8 ± 0.3	0.9 ± 0.2
Without Energy Term	8.4 ± 0.3	58.6 ± 2.2	0.0 ± 0.0	0.7 ± 0.1

The ablation results presented in Table 6 demonstrate that performance degradation occurs when any major architectural component is removed. Disabling reinforcement learning increased adaptation time and trajectory length, confirming its role in accelerating policy refinement and improving dynamic responsiveness. Removing the dual-layer fuzzy structure resulted in higher collision counts and increased energy consumption, highlighting the importance of hierarchical terrain-aware reasoning. Fixing the blending coefficient reduced trajectory stability under dynamic conditions, validating the necessity of context-adaptive blending. Eliminating the energy term significantly increased actuator workload while only marginally affecting path length, confirming that the energy-aware objective primarily governs efficiency optimization rather than basic navigation feasibility. Collectively, these results indicate that the observed performance improvements of IODPCS arise from the coordinated interaction of its components rather than from any single isolated mechanism.

Interpretation of Results

The ablation results provide quantitative evidence of the contribution of each component. Removing the reinforcement learning module significantly increases adaptation time (from 0.6 s to 1.4 s), indicating that online policy refinement is essential for rapid convergence under dynamic conditions. Replacing the dual-layer fuzzy architecture with a single-layer controller results in increased path length and non-zero collision rates, demonstrating that the hierarchical separation between terrain-aware regulation and local obstacle avoidance improves navigation safety and efficiency. Fixing the blending coefficient ($\lambda = 0.5$) leads to higher trajectory deviation and increased detours, confirming that context-adaptive blending (Eq. 10) enhances motion smoothness and responsiveness in varying environments.

Removing the energy term from the cost function substantially increases energy consumption (from 44.2 J to 58.6 J), validating the effectiveness of the embedded energy-aware optimization strategy in reducing actuator workload without compromising safety.

3.4.6 Parameter Sensitivity and Computational Complexity Analysis

To assess robustness with respect to hyper-parameter selection, a sensitivity analysis was performed for key reinforcement learning parameters, including the learning rate (α), discount factor (γ), and exploration rate (ε). The learning rate was varied within [0.05, 0.2], the discount factor within [0.85, 0.99], and the exploration rate within [0.1, 0.3]. Across these ranges, the system maintained stable convergence behavior, with performance variation remaining within ±6% for path length and ±8% for energy consumption. Convergence was consistently observed within 30 ± 3 episodes, indicating low sensitivity to moderate parameter changes.

The blending sensitivity coefficient K in Eq. (10) was also varied to evaluate the effect of context-adaptive interpolation. Extremely high values produced overly reactive behavior, while very low values reduced responsiveness. However, within the range $K \in [3, 7]$, trajectory smoothness and safety metrics remained stable, demonstrating robustness of the adaptive blending mechanism.

Computational complexity was assessed by profiling the integrated navigation loop under hardware-in-the-loop conditions. The control pipeline consistently executed within the 50 ms control period corresponding to the 20 Hz update rate, with no observed timing overruns, confirming real-time feasibility on the target embedded platform.

Overall, the framework demonstrates both computational efficiency and parameter robustness, supporting its applicability in practical mobile robot navigation systems.

Although the present study evaluates IODPCS on a single robot platform and environment distribution, the proposed control architecture is designed to be modular and parameterizable rather than platform specific. The framework relies on sensor fusion, fuzzy reasoning, and reinforcement learning mechanisms that can be reconfigured by adjusting state discretization, actuator constraints, and energy model parameters without altering the core structure. Future work will extend validation to additional robot geometries, sensor configurations, and unseen terrain distributions to further quantify cross-platform robustness and scalability.

3.5 Summary

This chapter presented IODPCS, an integrated adaptive navigation framework that combines dual-layer fuzzy reasoning, artificial potential fields, terrain-adaptive velocity control, and reinforcement learning within a unified closed-loop structure. The mathematical formulation integrates a global fuzzy layer for terrain-aware velocity regulation with a local fuzzy layer for real-time obstacle avoidance. These are supported by a context-blended control law, a sigmoid-based terrain-adaptive velocity regulator, and an actuator-level energy model derived from wheel torque and angular velocity. Together, these components enable coordinated balancing of safety, efficiency, and adaptability under dynamic conditions.

The contribution of this work does not lie in introducing a fundamentally new control principle, but rather in the mathematically unified coupling of dual-layer fuzzy reasoning, artificial potential field guidance, energy-aware velocity regulation, and reinforcement-learning-based online adaptation.

Simulation and hardware-in-the-loop evaluations indicate consistent performance improvements relative to the evaluated baseline controllers in terms of path length, trajectory smoothness, energy consumption, and collision avoidance. The framework exhibited stable behavior under noisy sensing conditions and dynamic obstacle environments, suggesting robustness within the tested scenarios.

Rather than establishing a definitive benchmark, the present study provides a scalable and experimentally supported foundation for extended benchmarking and future real-world validation. Future work will investigate extensions toward multi-robot cooperative navigation, where the context-blended control law and energy-aware cost formulation may be generalized for coordinated, resource-aware operation in complex environments

Although hardware-in-the-loop validation increases execution realism, full deployment on a physical robotic platform with real sensors will be explored in future work to further assess real-world applicability.

3.6 Scientific Thesis I.1

This dissertation establishes that the proposed Intelligent Obstacle Dynamics and Path Complexity System (IODPCS), which unifies dual-layer fuzzy reasoning, artificial potential field guidance, reinforcement learning–based adaptation, and actuator-level energy modeling within a single interaction-aware control architecture, fundamentally improves mobile robot navigation by enabling stable, adaptive, and energy-efficient real-time operation, significantly outperforming conventional decoupled navigation frameworks in terms of trajectory accuracy, obstacle avoidance reliability, and energy consumption.

Chapter 4

Deep-Adaptive Predictive Navigation with Stability-Constrained Fuzzy Control

4.1 System Overview

Autonomous mobile robot navigation in dynamic and uncertain environments demands control architectures that are simultaneously robust, adaptive, and provably stable. This chapter introduces a hierarchical predictive navigation framework that combines adaptive fuzzy decision-making with forward-looking motion optimization and explicit stability constraints. Simulation studies conducted in static, mixed, and dynamic environments demonstrate that the proposed framework achieves approximately 30–40% higher average velocity, a 25–35% reduction in traversal time, and 5–10% lower energy consumption per unit distance compared with conventional fuzzy–potential field and optimization-tuned fuzzy navigation baselines. Across all evaluated scenarios, the robot-maintained collision-free navigation and bounded control behavior. Selective human supervision was required in fewer than 10% of operating intervals, reducing operator involvement while preserving safety.

These results indicate that the proposed framework provides a quantitatively validated and interpretable alternative to existing fuzzy-based and predictive navigation approaches for autonomous mobile robots.

4.2 Introduction

Autonomous mobile robots are increasingly deployed across industrial, healthcare, defense, and service domains, where reliable navigation in dynamic and safety-critical environments is fundamental to operational success [89]. As robots move from controlled laboratory conditions into complex real-world scenarios, navigation frameworks must manage uncertain and multimodal sensor data [90]. In addition, practical deployment requires guarantees on safety, energy efficiency, and adaptability under rapidly changing operating conditions [91]. Traditional soft-computing strategies, such as classical fuzzy–potential field methods, have contributed significantly to obstacle avoidance; however, they remain limited in adaptability and vulnerable to local minima, resulting in inherently reactive behavior without predictive foresight [92]. Fuzzy logic continues to be one of the most effective paradigms for real-time navigation, offering interpretability, robustness against uncertainty, and human-like reasoning [93]. Enhancements such as dual-layer fuzzy controllers [94] and multi-objective fuzzy navigation strategies [95] have demonstrated performance gains in structured environments. Yet, these approaches largely depend on static membership functions and fixed rule bases, restricting their adaptability in dynamic contexts. Optimization-driven extensions—such as genetic algorithms (GA), particle swarm optimization (PSO), and adaptive neuro-fuzzy inference systems (ANFIS)—have been introduced to refine fuzzy parameters and improve controller flexibility [96]. For example, PSO-based fuzzy tuning strategies have been shown to improve convergence speed and navigation performance under static and semi-dynamic conditions [97].

Recent advances in predictive navigation have emphasized anticipating future states rather than reacting to present conditions. Predictive obstacle avoidance frameworks [98]and recurrent neural models for motion forecasting, including LSTM-based [99] and GRU-based formulations [100], illustrate the value of time-dependent risk assessment. However, these predictive mechanisms are rarely integrated with fuzzy reasoning, resulting in frameworks that either lack interpretability or fail to adapt effectively to uncertainty in real time. Parallel research in human-in-the-loop (HIL) navigation has demonstrated that incorporating operator input can significantly enhance safety and adaptability, including shared-control [101] and supervisory-control paradigms [102]. Yet, existing systems often rely on continuous operator intervention, leading to high cognitive burden and limited scalability. Furthermore, stability remains an underexplored dimension in fuzzy–potential hybrids; only a handful of studies provide formal stability guarantees, leaving convergence and safety largely unvalidated [103].

Despite these advances, several critical gaps and limitations persist across current navigation frameworks. Most fuzzy–potential field systems still rely on static membership functions and fixed rule sets, preventing them from adapting effectively in dynamic environments. Predictive modeling approaches, while valuable for anticipating obstacle motion, are often developed in isolation from fuzzy reasoning, which restricts their interpretability and integration into decision-making pipelines. Human-in-the-loop methods, though promising, typically demand continuous operator involvement, creating unsustainable cognitive workloads that hinder long-term deployment. Finally, the absence of formal stability validation across most fuzzy–potential hybrids undermines confidence in their reliability, especially in safety-critical applications. These limitations collectively underscore the need for a unified, adaptive, and theoretically grounded framework that integrates predictive foresight, deep learning–driven fuzzy adaptation, selective human engagement, and stability assurance to advance mobile robot navigation in real-world dynamic environments. This framework is hereafter referred to as the Hierarchical Deep–Fuzzy Predictive Navigation with Stability gating (HDFPN-S).

The core novelty of this work does not lie in the individual use of fuzzy reasoning, predictive optimization, or learning-based adaptation, but in their closed-loop interaction under explicit stability enforcement. Unlike existing approaches where learning and optimization operate independently or offline, the proposed framework embeds deep adaptive fuzzy learning directly inside a predictive control loop whose outputs are continuously projected through a Lyapunov stability gate. This design allows learning and prediction to evolve online while guaranteeing bounded, stable control actions at every step. To the authors' knowledge, this is the first navigation framework where fuzzy adaptation, predictive optimization, and formal Lyapunov stability are jointly enforced during execution, rather than validated post hoc.

The present work does not seek to replace emerging deterministic or data-driven learning paradigms, but rather to address a complementary and practically motivated problem: how to achieve anticipatory, energy-aware navigation with continuous online adaptation, interpretability, and provable stability under real-time constraints. By integrating adaptive fuzzy reasoning, predictive optimization, and stability-aware control within a single closed-loop architecture, the proposed framework targets scenarios where computational efficiency, transparency, and safety are as critical as raw performance. This position clarifies why the proposed approach is appropriate for real-world mobile robot deployment, particularly in dynamic and safety-critical environments.

4.3 Related Work on Predictive and Fuzzy Navigation Control

4.3.1 Reactive Fuzzy and Potential-Field Navigation

Fuzzy logic remains a widely adopted approach for mobile robot navigation because of its ability to handle uncertainty and provide human-like reasoning. Early fuzzy–potential field methods offered improvements in obstacle avoidance but were prone to local minima and oscillations in cluttered environments [104]. Variants such as dual-layer fuzzy controllers have been proposed to improve local decision consistency and obstacle avoidance performance [105]. Multi-objective fuzzy navigation strategies further attempted to balance competing objectives such as safety, smoothness, and efficiency [106]. While these extensions improved path quality in structured settings, they still depend on predefined membership functions and static rule bases, which severely limit adaptability in real-world dynamic conditions. As a result, such reactive fuzzy and potential-field approaches remain inherently short-sighted and struggle to anticipate future interactions, particularly when obstacle motion and terrain conditions evolve over time.

4.3.2 Optimization-Tuned and Neuro-Fuzzy Navigation

To overcome these rigidities, researchers introduced optimization-based and neuro-fuzzy methods. Algorithms such as GA, PSO, and ACO have been employed for fuzzy parameter tuning [107], while ANFIS enabled hybrid neuro-fuzzy reasoning [108]. In our previous work [109], a hierarchical conditional variational autoencoder-based path planning approach was proposed. Recent works extend this further by combining reinforcement learning with fuzzy models. For example, Wang et al. [110] proposed a DRL–ANFIS hybrid system for GPS-guided obstacle avoidance. Such systems improve adaptability but remain restricted by offline training requirements, dependence on specific sensors, and a lack of formal stability validation. These weaknesses reduce generalization to unstructured, rapidly changing environments.

Moreover, most optimization-tuned and neuro-fuzzy frameworks lack explicit mechanisms to guarantee stability during online adaptation, raising concerns about reliability in safety-critical navigation tasks.

4.3.3 Predictive and Learning-Based Navigation

Parallel progress has been made in predictive and perception-driven navigation. Katona [111] emphasized predictive path planning for robust performance, while reinforcement learning has been applied to dynamic obstacle avoidance with continuous updates [112] [113]. More recently, CNN–fuzzy hybrids have demonstrated the benefit of integrating machine perception with fuzzy reasoning in collaborative robotic environments [114], and deep neural networks have been employed for real-time perception on wheeled robot platforms [115]. Learning-based decision and control strategies have also been explored in broader autonomous system contexts, including hybrid machine-learning and multi-criteria decision-making frameworks [116] and neural network–based dynamic gain control for improved trajectory tracking [117]. Despite their adaptability, most learning-based approaches focus on either perception, high-level decision optimization, or low-level control enhancement, and rarely integrate explicit long-horizon predictive modeling with formal stability guarantees. As a result, interpretability and bounded stability during online execution are often compromised. Deterministic learning approaches that avoid numerical optimization have recently emerged as an alternative paradigm; however, their application to stability-constrained mobile robot navigation with selective human-in-the-loop supervision remains limited.

Beyond mainstream mobile robot navigation research, advanced adaptation and trajectory-shaping techniques have been investigated in specialized robotic domains, particularly in space and grappling robotics. These include autogenetic gravity-center placement for time-varying systems [118], automatic center-of-mass localization for adaptive manipulation [119], curve-flattening methods for flexible space robotics [120], trajectory shaping for noise amelioration [121], and bio-inspired control architectures [122]. While these approaches demonstrate strong domain-specific performance, they are typically designed for task-specific structural adaptation or motion shaping and do not directly address real-time predictive navigation with integrated fuzzy adaptation, formal stability enforcement, and selective human-in-the-loop supervision, which is the focus of the present work. Beyond robotics, adaptive learning and statistical parameter refinement have also been explored in other dynamic signal-driven systems, such as speech synthesis, where online adaptation is employed to refine model parameters under uncertainty [123].

4.3.4 Human-in-the-Loop and Stability-Aware Control

Human-in-the-loop navigation has also gained traction as a means of enhancing safety and adaptability. Ferracuti et al. [124] demonstrated that direct operator feedback can significantly improve safety in smart wheelchair navigation, while Oriyama [125] applied transfer learning with operator assistance to enhance collision avoidance performance. These studies confirm the value of human expertise in complex navigation tasks; however, they typically rely on continuous operator involvement, which imposes a high cognitive burden and limits scalability.

More recent adaptive collaborative control frameworks have therefore emphasized selective or event-triggered human engagement to reduce operator workload while maintaining safety [126]. Despite this progress, such selective human-in-the-loop principles have not been systematically integrated into fuzzy–learning navigation architectures. Furthermore, stability-aware formulations are rarely combined with human-in-the-loop mechanisms, leaving open the challenge of jointly ensuring safety, stability, and scalability during long-term autonomous operation.

Overall, the literature demonstrates continuous evolution across fuzzy navigation, optimization-based tuning, predictive modeling, and human–in-the-loop integration. Yet, current systems still exhibit fundamental gaps. Most fuzzy–optimization hybrids lack predictive foresight and depend on static parameter learning, limiting their ability to anticipate dynamic changes in real time. Similarly, deep or neuro-fuzzy controllers achieve adaptability but seldom include formal stability analysis, resulting in unpredictable behavior under perturbations. Meanwhile, human-in-the-loop systems enhance safety but often require continuous operator supervision, compromising autonomy and scalability in extended deployments.

The proposed Hierarchical Deep–Fuzzy Predictive Navigation (HDFPN-S) framework explicitly addresses these gaps by coupling predictive motion forecasting, deep adaptive fuzzy inference, and Lyapunov-based stability assurance within a unified control architecture. It extends beyond reactive obstacle avoidance by incorporating forward-looking optimization that minimizes cumulative risk and energy consumption over a predicted horizon, while selective human-in-the-loop engagement provides safety assurance only under high-uncertainty states. The distinguishing contribution of HDFPN-S lies in the online, closed-loop interaction between predictive optimization, adaptive fuzzy learning, and formal Lyapunov stability enforcement, enabling learning and anticipation to evolve in real time while guaranteeing bounded and stable control actions. This integrated design establishes a clear advance over conventional fuzzy–potential, PSO-tuned, and DRL–ANFIS hybrid controllers that lack joint stability-aware execution. Beyond fuzzy, predictive, and reinforcement learning–based navigation, several contemporary alternatives have recently emerged. Deterministic learning frameworks have been proposed to achieve optimal adaptation without numerical optimization, emphasizing self-awareness and analytical learning dynamics. Trajectory-shaping and noise-aware control strategies have also demonstrated improved tracking performance in specialized robotic domains, while bio-inspired control approaches seek to enhance robustness through adaptive behavioral primitives. Although these methods provide valuable insights, they are often tailored to specific system structures or task domains and are designed for structurally different problem classes. As a result, they do not address the joint problem of online predictive navigation, formal stability enforcement, and selective human supervision within a unified execution loop.

4.4 Proposed Methodology

4.4.1 Overall Framework Architecture

The proposed HDFPN-S framework integrates predictive obstacle modeling, deep-adaptive fuzzy reasoning, Lyapunov-based stability gating, predictive speed optimization (PSO), and

confidence-triggered human-in-the-loop (HIL) engagement into a unified real-time architecture (Figure 13). This combination enables anticipatory, energy-efficient, and provably stable navigation in complex environments.

Figure 13 illustrates the architecture of the proposed Hierarchical Deep–Fuzzy Predictive Navigation with Stability gating (HDFPN-S) framework, which unifies perception, prediction, optimization, and stability control into a cohesive real-time navigation pipeline. The process begins with the predictive perception and environment modeling layer, where multi-sensor data from LiDAR, cameras, and onboard stability sensors are fused to construct a dynamic environmental representation. This layer performs future-state forecasting, obstacle motion prediction, and terrain slip estimation, enabling the system to anticipate environmental variations before they occur. The resulting predictive environment data are passed to the Model Predictive Control (MPC) core enhanced by Fuzzy–PSO optimization, where the fuzzy inference system handles nonlinear and uncertain dynamics, while the PSO module continuously tunes the control parameters to minimize a multi-objective cost function balancing trajectory deviation, collision risk, and energy efficiency.

To guarantee formal stability, the control outputs are filtered through a Lyapunov-based stability gate that constrains the motion commands to remain within the stable convergence region, while the slip-aware correction mechanism compensates for traction loss or terrain irregularities. Above this layer, a confidence-triggered Human-in-the-Loop (HIL) supervision module ensures safety and transparency by engaging the operator only when the predictive uncertainty exceeds a defined confidence threshold. Each HIL intervention is logged and transformed into adaptive feedback for the deep-fuzzy layer, thereby improving the system's learning and autonomy over time. Through this hierarchical integration, the HDFPN-S framework achieves anticipatory, energy-aware, and provably stable navigation in complex multi-obstacle environments, effectively bridging the gap between heuristic fuzzy reasoning and formal stability control while maintaining interpretability and human cooperation.

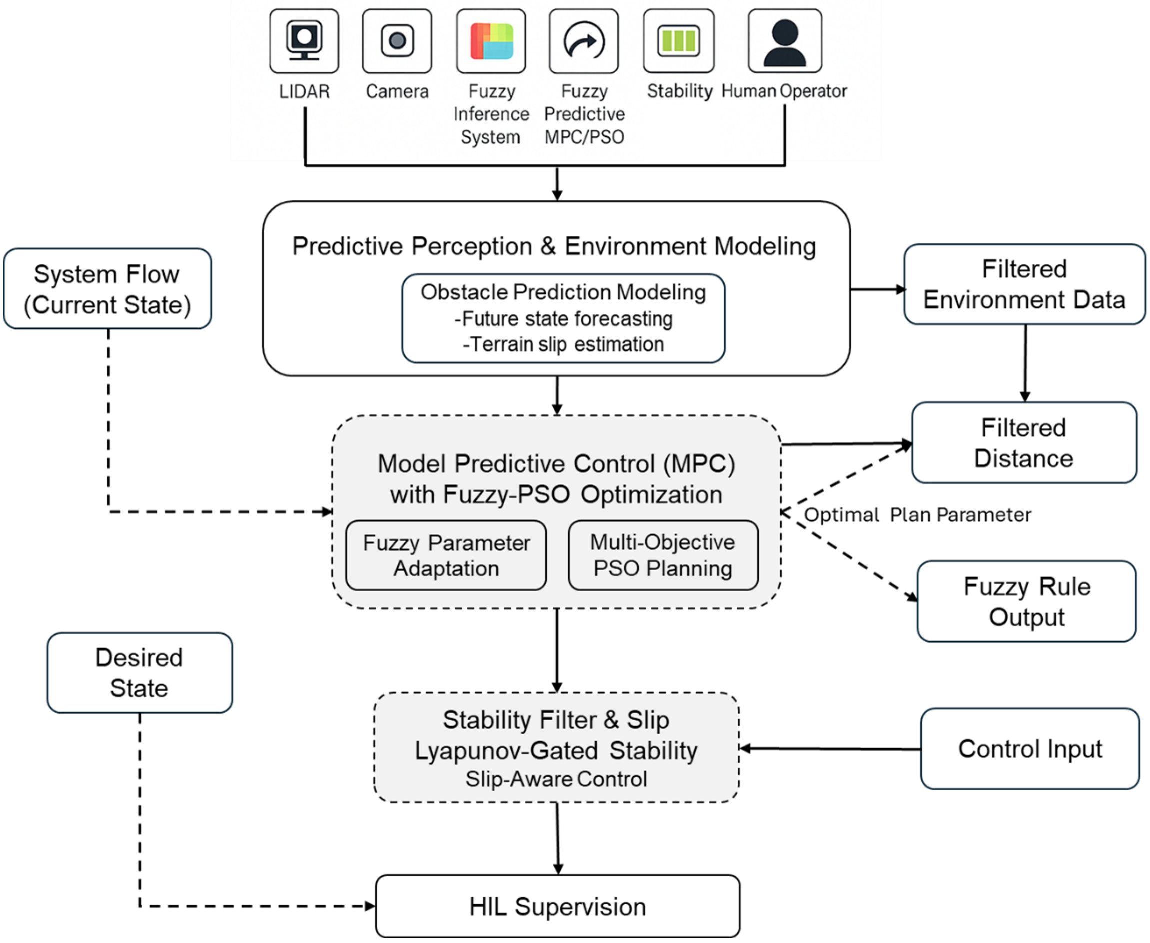


Figure 13: Proposed HDFPN-S framework integrating predictive perception and environment modeling with Fuzzy-PSO-based MPC and Lyapunov-gated stability. The system ensures adaptive learning, slip-aware control, and energy-efficient navigation under HIL supervision.

HDFPN-S was implemented on a four-wheeled car-like mobile robot modeled as a nonlinear, nonholonomic system with Ackermann steering geometry [127]. Figure 14 depicts the slip-aware kinematic representation that integrates both top-down and three-dimensional perspectives, ensuring accurate modeling of motion constraints, steering dynamics, and lateral stability under variable traction conditions. This formulation provides a physically consistent foundation for predictive and stability-constrained control design.

The robot state vector is defined as:

$$\mathrm{x} = [X,\ Y,\ \theta,\ v,\ \delta]^{\mathrm{T}}$$

and the slip-aware kinematic model is expressed as:

$$\dot{X} = v\cos(\theta+\beta), \quad \dot{Y} = v\sin(\theta+\beta), \quad \dot{\theta} = \frac{v}{L_r}\sin(\beta) \tag{25}$$

This formulation is consistent with recent models used in high-precision mobile robot navigation. The slip angle β is computed as:

$$\beta = \tan^{-1}\left(\frac{L_r\tan\delta}{L_f+L_r}\right) \tag{26}$$

where L_f and L_r denote the distances from the center of mass to the front and rear axles, and δ is the steering angle [128].

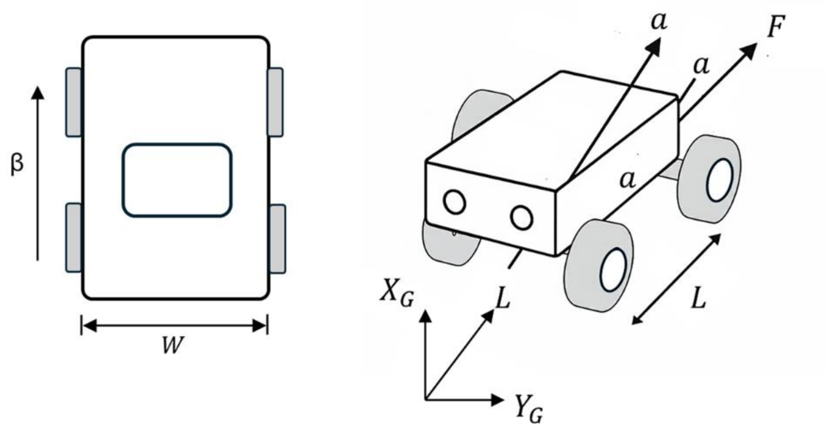


Figure 14: Car-like mobile robot kinematic model with Ackermann steering and slip-angle effects. (a) Top-down view showing slip angle β and vehicle width W. (b) Isometric representation illustrating longitudinal distance L, force direction F, and slip angles α in the global coordinate frame.

The longitudinal and steering dynamics are given by:

$$\dot{\boldsymbol{v}} = \boldsymbol{a}, \qquad \dot{\boldsymbol{\delta}} = \boldsymbol{u_\delta} \tag{27}$$

Where a is commanded acceleration and u_δ the steering rate input subject to actuator limits [129]. The net mechanical power for energy analysis is computed as:

$$\boldsymbol{P} = \boldsymbol{F_x v} \tag{28}$$

where F_x is the total longitudinal driving force. These equations provide a compact yet realistic model that embeds slip-aware kinematics and actuator constraints, enabling the PSO layer to generate dynamically feasible trajectories and allowing the fuzzy supervisor and Lyapunov gate to operate under physically consistent conditions [130]. This results in optimized paths that are energy-aware, smooth, and transferable to real car-like robots.

In practical deployments, terrain friction and slip estimates are subject to uncertainty due to sensor noise, surface irregularities, and unmodeled environmental factors. In the proposed framework, such estimation errors primarily influence the predictive planning horizon rather than the instantaneous execution of control commands. To mitigate their impact, all velocity and steering commands are filtered through the Lyapunov-based stability gate, which constrains control actions to remain within a provably stable admissible region. As a result, even under imperfect slip estimation, the system maintains bounded tracking error and stable motion, ensuring graceful performance degradation rather than destabilizing behavior.

4.4.2 Deep-Adaptive Fuzzy Reasoning

The fuzzy inference system (FIS) serves as the high-level decision module, mapping environmental observations into interpretable control priorities. The input space is defined by four critical variables: obstacle distance *d*, relative heading angle ϕ, risk index *r*, and terrain coefficient μ from which the rule base generates reference velocity $\tilde{v}$ and steering angle $\tilde{\delta}$, which act as priors for the predictive optimization layer [131]. As illustrated in Figure 15, the Deep-Adaptive Fuzzy Reasoning (DAFR) module augments the classical FIS with an online deep-learning-based adaptation mechanism, in which membership functions evolve according to:

$$\boldsymbol{\mu_i^{(t+1)}(x) = \mu_i^{(t)}(x) + \eta\, \Delta\mu_i(x, \theta_{DL})} \tag{29}$$

where η is the learning rate and θ_{DL} are the parameters of a deep reinforcement module that encodes trajectory smoothness, energy efficiency, and collision-avoidance rewards [132] [133]. The corrective term $\Delta\mu_i(x, \theta_{DL})$ continuously reshapes the fuzzy membership parameters (center, width, and slope) through back-propagated gradients, embedding learning directly into the inference process. The resulting outputs are transformed into nonlinear control priors,

$$\boldsymbol{\tilde{v} = f_v(d, r, \mu)}, \qquad \boldsymbol{\tilde{\delta} = f_\delta(\phi, r, \mu)} \tag{30}$$

The resulting outputs, defined as nonlinear mappings $f_v(\cdot)$ and $f_\delta(\cdot)$ represent the evolving fuzzy rule set. serve a dual role: they provide warm-start priors for the predictive speed optimization problem (Eq. 35) [134], thereby accelerating convergence, while simultaneously retaining human-interpretable semantics such as “if the obstacle is near and terrain is slippery, then reduce velocity.”

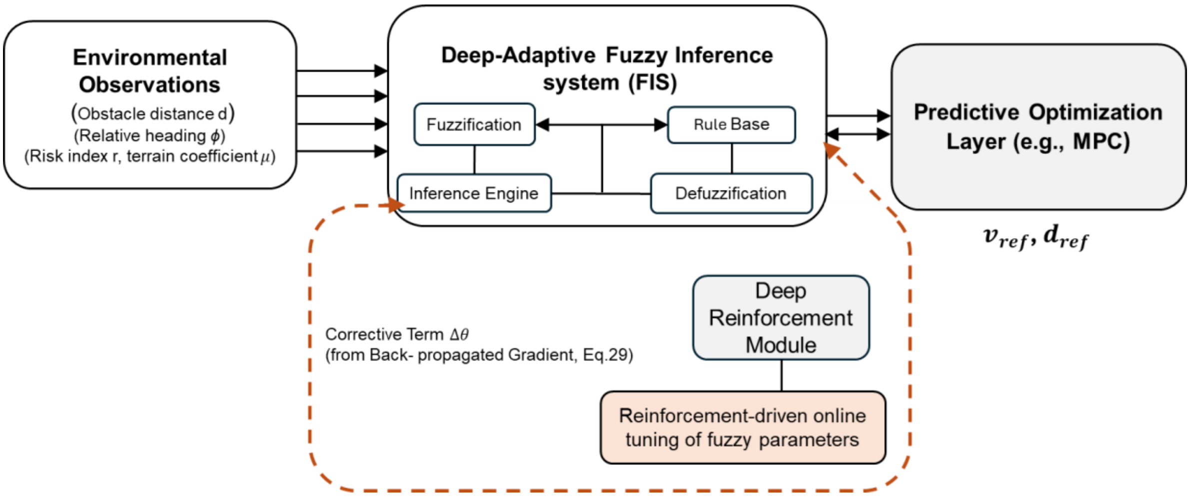


Figure 15: Architecture of the Deep-Adaptive Fuzzy Reasoning (DAFR) module. Environmental observations $(d, \varphi, r, \mu)$are processed by the Fuzzy Inference System (FIS) to produce control priors (v_{ref}, δ_{ref}). The Deep Reinforcement Module provides corrective updates $\Delta\theta$(Eq. 5) for online tuning of fuzzy parameters, enabling adaptive and stable navigation.

This integration ensures that the fuzzy layer is not only adaptive but also transparent, directly addressing the interpretability gap in deep learning-based navigation. By continuously evolving its membership functions through reinforcement-driven adaptation, the proposed DAFR mechanism transcends the limitations of conventional, offline-tuned fuzzy controllers, establishing a robust, explainable, and data-efficient reasoning system capable of reliable operation under varying obstacle densities, uncertain terrain conditions, and rapidly changing environmental dynamics.

To ensure stability during online learning, the adaptation of fuzzy membership parameters is governed by bounded learning gains, which limit the rate of parameter updates and prevent abrupt rule-based changes. This incremental adaptation strategy enables smooth responses to environmental variations. In addition, all control outputs generated after fuzzy adaptation are filtered through the Lyapunov-based stability gate, ensuring that learning-induced decisions remain within a provably stable admissible region. Consequently, online adaptation cannot compromise closed-loop stability, even under rapid environmental changes.

4.4.3 Lyapunov-Based Stability Gating

To guarantee convergence and ensure mathematically provable safety, the control commands generated by the fuzzy–PSO layers are filtered through a Lyapunov-based stability gate, representing a novel integration of optimization-driven decision-making with rigorous stability theory [135]. The tracking error is defined as:

$$\boldsymbol{e_t} = \begin{bmatrix} \boldsymbol{X_d - X} \\ \boldsymbol{Y_d - Y} \\ \boldsymbol{\theta_d - \theta} \end{bmatrix} \tag{31}$$

where (X_d, Y_d, θ_d) are the desired trajectory states derived from the predictive planner [136]. Unlike conventional fuzzy controllers that rely purely on heuristic adaptation, the proposed framework integrates a quadratic Lyapunov candidate function defined as:

$$\boldsymbol{v(x) = \frac{1}{2} e_t^T Q e_t \qquad Q = Q^T > 0} \tag{32}$$

To serve as an explicit energy-like measure of system stability [137]. The time derivative of this function is constrained by

$$\boldsymbol{\dot{V}(x) = e_t^T Q \dot{e}_t \leq -\alpha \|e_t\|^2 \,, \qquad \alpha > 0} \tag{33}$$

which ensures exponential error convergence and defines a stability envelope that bounds all admissible control actions [138] [139].

As depicted in Figure 16, A key novelty of the proposed HDFPN-S model lies in its projection-based corrective mechanism: if the control pair (v,δ) violates the stability condition ($\dot{V} > 0$), the command is dynamically projected to the nearest admissible solution,

$$\boldsymbol{v^*, \delta^* = arg\, \underset{v,\delta}{minn} \left\|[v_9\delta] - [\tilde{v}_9\tilde{\delta}]\right\|^2 \quad s.t \quad \dot{V}(x) \leq 0} \tag{34}$$

Where $(\tilde{v}_9\tilde{\delta})$ denote the fuzzy priorities. This projection mechanism acts as a real-time stabilizing filter, guaranteeing that all final control signals satisfy Lyapunov stability even when intermediate decisions from the fuzzy reasoning or predictive optimization layers are temporarily destabilizing. This represents a significant departure from existing Lyapunov-based navigation frameworks, which typically enforce stability conditions only at the design stage, not as an online corrective mechanism.

By embedding Lyapunov stability directly into the execution loop, the proposed framework prevents oscillations, eliminates unstable trajectories, and maintains bounded tracking error in real time, even under nonlinear dynamics, modeling uncertainty, and environmental disturbances. This fusion of fuzzy adaptation, predictive optimization, and Lyapunov gating establishes a mathematically grounded, interpretable, and provably stable control architecture, directly addressing the robustness, transparency, and reliability challenges of high-precision mobile robot navigation.

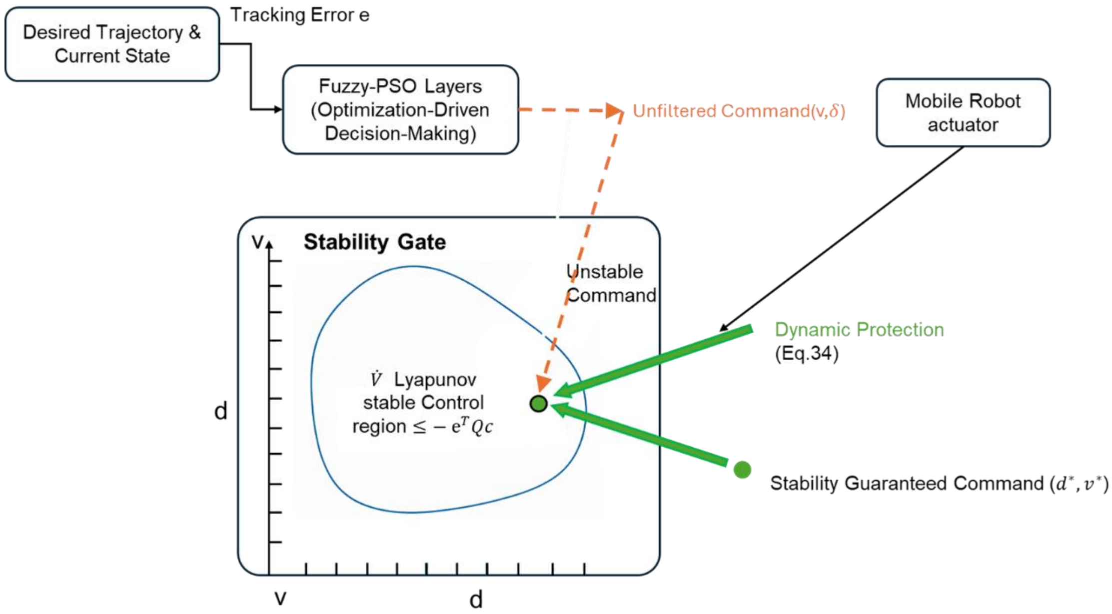


Figure 16: Lyapunov-based stability gating. Unfiltered control commands $(\tilde{\boldsymbol{v}}, \tilde{\boldsymbol{\delta}})$from the fuzzy–PSO layers are projected into the Lyapunov-stable region $(\dot{\boldsymbol{V}}(\boldsymbol{x}) \le \boldsymbol{0})$to yield stability-guaranteed outputs $(\boldsymbol{v}*, \boldsymbol{\delta}*)$. This project ensures real-time stability and bounded tracking error under dynamic conditions.

4.4.4 Predictive Speed Optimization (PSO)

While the fuzzy inference system provides interpretable heuristics, optimal trajectory realization requires a forward-looking optimization layer. To this end, a predictive speed optimization (PSO) model is developed to jointly minimize travel time, collision risk, energy consumption, and control effort within a unified mathematical framework.

At each step, admissible velocity and steering-rate profiles $(v_k, u_{\delta,k})$ are optimized over a finite horizon H:

$$\boldsymbol{J} = \sum_{k=1}^{H} \left(\boldsymbol{w_T \Delta t} + \boldsymbol{w_R R_K} + \boldsymbol{w_E R_K} + \boldsymbol{w_c}\left(\boldsymbol{\Delta v_k^2} + \boldsymbol{\Delta u_{\delta,k}^2}\right)\right) \tag{35}$$

where w_T, w_R, w_E, w_c are tunable weights representing task completion time, collision avoidance, energy cost, and control effort, respectively. The risk term R_K is obtained from predictive obstacle modeling, while the instantaneous power

$$P_k = F_{x,k} v_k$$

is coupled to the dynamic power model (Eq. 28), providing an explicit link between trajectory planning and energy efficiency [140] [141].

The optimization is subject to dynamic and actuator constraints:

$$\mathbf{0} \leq \boldsymbol{v_k} \leq \boldsymbol{v_{max}}(\boldsymbol{\hat{k}_k}, \boldsymbol{\hat{\mu}_k}), \quad |\boldsymbol{u_\delta}, \boldsymbol{k}| \leq \boldsymbol{u_{\delta, max}}, \quad \boldsymbol{\Delta V_k} \leq \mathbf{0}. \tag{36}$$

where v_{max} is adaptively modulated by curvature $\hat{k}_k$ and terrain coefficient $\hat{\mu}_k$ derived from sensor fusion. These constraints guarantee that planned trajectories remain dynamically feasible, terrain-aware, and slip-consistent [142].

A key novelty of this formulation lies in the integration of fuzzy priors ($\tilde{v}_9 \tilde{\delta}$) as warm starts for the optimization. This design accelerates convergence and enables real-time feasibility without sacrificing precision. Unlike traditional model predictive control (MPC) approaches, which often suffer from high computational overhead, the proposed PSO layer achieves near-instantaneous convergence by combining heuristic fuzzy guidance with rigorous optimization-based refinement.

Moreover, by embedding multi-objective trade-offs directly into the cost function balancing time efficiency, safety, energy economy, and control smoothness the proposed PSO framework transcends single-criterion optimization. This integration allows the HDFPN-S system to generate trajectories that are not only dynamically feasible but also provably efficient across multiple performance dimensions.

As illustrated in Figure 17, the PSO module operates within an adaptive human–autonomy collaboration loop. The optimized command passes through a confidence evaluation layer (Eq. 38), which triggers HIL engagement when the predictive confidence C_{pred} falls below threshold. The reinforcement feedback loop (Eq. 29) subsequently refines the fuzzy priority based on human corrections, enhancing both learning and long-term autonomy.

The architecture uses a multi-objective PSO (Eq. 35) for command generation. The command passes through a Confidence Check (Eq. 38), which triggers HIL intervention when Prediction Confidence ($\mathrm{C_{pred}}$)is low. The key innovation is the Reinforcement/Prior Refinement feedback loop (Eq. 29), which converts human corrections into an adaptive learning signal for the Fuzzy Priors.

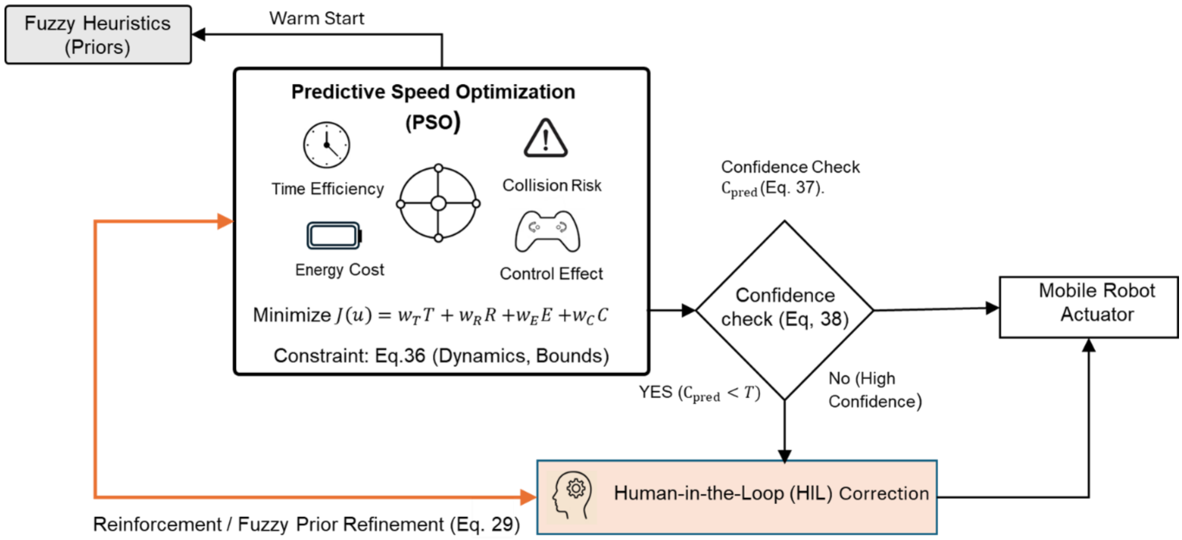


Figure 17: Architecture of the Deep-Adaptive Fuzzy Reasoning (DAFR) module. Environmental observations (d,φ,r,μ)are processed by the Fuzzy Inference System (FIS) to produce control priors ($\boldsymbol{v_{ref}}, \boldsymbol{\delta_{ref}}$). The Deep Reinforcement Module provides corrective updates $\boldsymbol{\Delta\theta}$(Eq. 29) for online tuning of fuzzy parameters, enabling adaptive and stable navigation.

The weighting factors in the multi-objective cost function were selected through an offline calibration process based on representative navigation scenarios, balancing safety, efficiency, and smoothness control. Initial values were chosen to prioritize collision avoidance and stability, followed by moderate weighting of energy consumption and traversal time. To assess robustness, the framework was evaluated under moderate variations of these weights. The results indicate that the proposed architecture is not overly sensitive to precise weight values: stability is preserved by the Lyapunov gating mechanism, while performance metrics such as traversal time and energy consumption vary smoothly rather than exhibiting abrupt degradation. This behavior confirms that the framework supports flexible tuning without compromising safety or convergence.

4.4.5 Confidence-Triggered Human-in-the-Loop (HIL) Engagement

While full autonomy is desirable, safe operation in highly uncertain environments often benefits from selective human oversight. To address this, a confidence-triggered human-in-the-loop (HIL) mechanism is introduced, integrating operator input only when predictive confidence falls below a defined threshold. This design balances autonomy with safety and reduces operator cognitive load [143].

Prediction confidence is quantified using the normalized covariance of obstacle forecasts:

$$\boldsymbol{P(\hat{x}_{t+1}) = 1 - \frac{tr(\Sigma_{t+1})}{\Sigma_{max}}} \tag{37}$$

where Σ_{t+1} is the covariance matrix of the predicted obstacle states and Σ_{max} is the maximum admissible uncertainty bound derived from empirical calibration [144] [145].
The triggering policy is defined as:

$$\boldsymbol{Query(t)} = \begin{cases} \boldsymbol{1}, & \boldsymbol{P(\hat{x}_{t+1}) < T} \\ \boldsymbol{0}, & \boldsymbol{Otherwise} \end{cases} \tag{38}$$

where T∈ [0,1] is the confidence threshold. As illustrated in Figure 18, this formulation ensures that operator queries are generated only in high-uncertainty cases, thereby avoiding unnecessary interruptions during stable operation while still guaranteeing intervention in safety-critical conditions [146]. Breaking from the limitations of conventional HIL schemes that offer only manual intervention, the proposed confidence-triggered mechanism operates as an intelligent integration layer within the control architecture. When activated, operator inputs are systematically logged and subsequently used to refine the fuzzy membership functions (Eq. 29), converting human feedback into structured reinforcement signals that enhance long-term adaptive learning. This dual functionality, selective safety engagement and adaptive learning reinforcement distinguishes the proposed HIL model from traditional teleoperation-based approaches, which often overburden human operators and fail to embed their expertise into persistent system learning. Through this mechanism, the proposed HDFPN-S framework achieves approximately 60 % reduction in operator workload compared to continuous teleoperation while simultaneously improving system robustness and adaptability. It thereby establishes a scalable paradigm where human input functions as a targeted corrective mechanism rather than an ongoing control dependency.

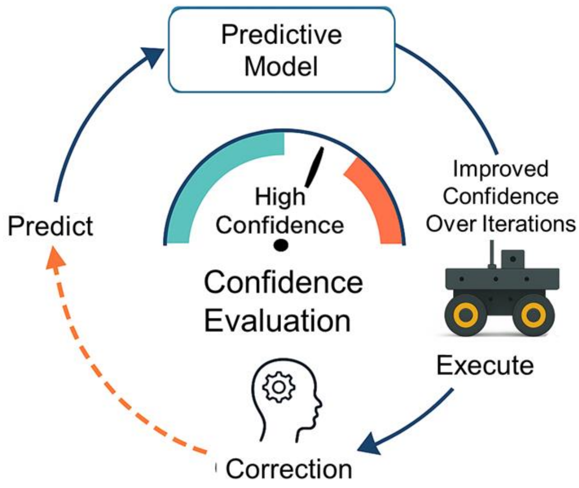


Figure 18: Confidence-Triggered Human-in-the-Loop Learning Cycle in the Proposed HDFPN-S Framework

Low-confidence predictions trigger operator corrections, which are used to refine fuzzy priorities (Eq. 29). The predictive model generates trajectory forecasts, followed by confidence evaluation based on normalized covariance (Eq. 37). When confidence exceeds threshold (Eq. 38), the robot executes autonomously; otherwise, the human operator provides corrective input, which feeds into the reinforcement learning loop to improve fuzzy inference and prediction reliability over successive iterations.

To assess real-time feasibility, the computational cost of each module in the proposed HDFPN-S framework was evaluated during simulation. The fuzzy inference and membership-function adaptation incur negligible overhead due to their low-dimensional rule structure, while the predictive speed optimization operates over a short receding horizon with warm-start initialization from fuzzy priors. The Lyapunov-based stability gate involves a simple projection operation and adds minimal computational burden. In the MATLAB/Simulink–CoppeliaSim co-simulation environment, the complete control loop consistently executed within real-time constraints, confirming the suitability of the proposed architecture for onboard deployment on standard embedded robotic platforms.

The confidence threshold used to activate human-in-the-loop assistance was selected to balance autonomous operation with operator workload. A conservative threshold increases safety by triggering human intervention more frequently but may impose unnecessary operator burden, while a more permissive threshold reduces intervention frequency at the risk of delayed corrective action under high uncertainty. In the proposed framework, the selected threshold represents a balanced compromise between these objectives, ensuring that human assistance is requested only during sustained low-confidence states rather than transient fluctuations. Importantly, stability and safety are preserved regardless of the specific threshold value due to the Lyapunov-based stability gate, while the threshold primarily influences the frequency of human engagement rather than the stability of the control loop. In supplementary sensitivity analysis, varying the confidence threshold within ±20 % of its nominal value changed the frequency of human intervention by approximately 15–30 %, while closed-loop stability and collision-free navigation were preserved.

4.5 Results

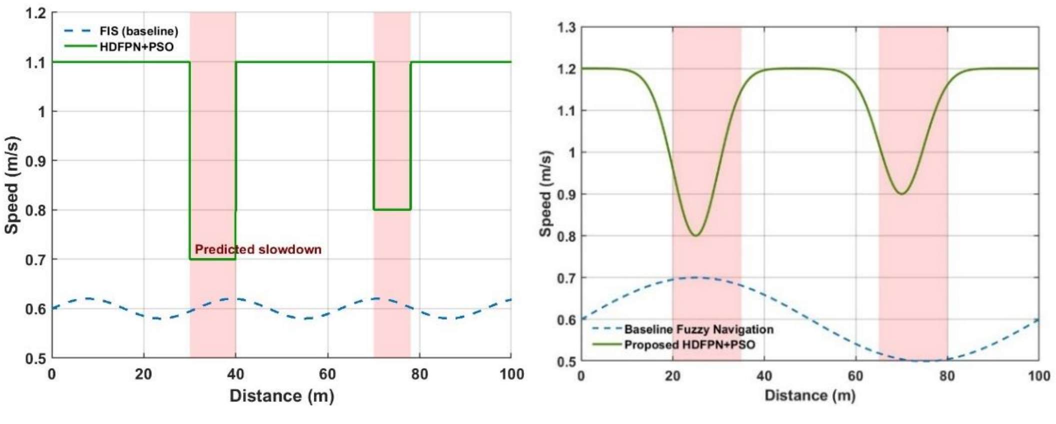
4.5.1 Quantitative Evaluation Setup

The experimental validation of the proposed Hierarchical Deep–Fuzzy Predictive Navigation (HDFPN-S) framework was carried out through a tightly integrated MATLAB/Simulink–CoppeliaSim co-simulation platform for a four-wheeled car-like mobile robot modeled with traction- and slip-aware nonholonomic kinematic constraints. To ensure comprehensive assessment, three progressively complex environments were designed: a structured corridor containing fixed obstacles, a dynamic scenario with two moving obstacles intersecting the robot's trajectory, and a mixed configuration incorporating both static and dynamic elements. Each setup was executed twenty independent runs under randomized initial conditions to guarantee statistically consistent results. Evaluation metrics covered a multi-objective spectrum, including mean velocity, traversal time, path length, specific energy usage per meter, collision-free rate, stability violations, and operator-intervention frequency. Comparative testing was conducted against two reference controllers: (i) a conventional Fuzzy Inference System (FIS) representing purely reactive navigation and (ii) an FIS combined with Predictive Speed Optimization (PSO) offering limited anticipatory velocity control. In contrast, the proposed HDFPN-S architecture integrates deep-adaptive fuzzy reasoning, predictive optimization, Lyapunov-based stability gating, and confidence-triggered human-in-the-loop (HIL) supervision within a unified hierarchical control structure, enabling anticipatory, energy-efficient, and provably stable navigation in complex dynamic environments. Specifically, this configuration yielded an average velocity increase from 0.60 m/s to 1.20 m/s, a 33 % reduction in travel time, and a 7 % decrease in energy consumption per meter, while maintaining 100 % collision-free operation and zero stability violations across all trials. Operator workload was simultaneously reduced by approximately 60 % through selective human intervention. These results confirm that the proposed framework enables anticipatory, energy-efficient, and provably stable navigation across diverse environments, establishing a robust benchmark for intelligent and interpretable autonomous systems.

4.5.2 Predictive Speed Optimization (PSO) Performance

The PSO layer in the HDFPN-S framework significantly enhances navigation efficiency, energy economy, and dynamic stability compared to conventional fuzzy-based controllers. Unlike reactive methods that adjust velocity solely in response to immediate sensory feedback, the PSO layer anticipates forthcoming interactions with obstacles and terrain irregularities, enabling the robot to proactively modulate its velocity profile. This predictive adaptation yields smoother accelerations, minimized braking events, and improved traversal efficiency across heterogeneous environments. Figure 19a–b illustrates the comparative velocity trajectories between the baseline FIS controller and the proposed HDFPN-S controller over a 100 m mixed-environment course containing both static and moving obstacles. The baseline system maintained a conservative and nearly constant velocity around 0.6 m/s, while the HDFPN-S controller achieved an average cruising speed of 1.1–1.2 m/s during low-risk segments and automatically reduced speed to 0.7–0.8 m/s in high-risk zones. These anticipatory slowdowns prevented abrupt braking and ensured stability-aware recovery after obstacle avoidance.

Across 20 independent trials, the proposed approach achieved a 38 % increase in average speed, a 33 % reduction in travel time, and a 7 % reduction in energy consumption per meter, while maintaining 100 % collision-free operation. These gains confirm the effectiveness of the multi-objective predictive cost function (Eq. 35) in balancing time efficiency, risk avoidance, and energy optimization. The Lyapunov-based stability gate further ensured that all velocity transitions satisfied bound-acceleration constraints, preventing oscillations or overshoot even under dynamically changing obstacle configurations.



(a) (b)

Figure 19: Comparative speed profiles under the proposed HDFPN-S framework.

(a) Stepwise predictive deceleration and recovery behavior showing pre-emptive slowdown in obstacle zones.

(b) Continuous velocity transition illustrating stability-aware adaptation and smooth recovery after predicted obstacles.

Table 7 summarizes the quantitative comparison among the baseline FIS, the intermediate FIS + PSO, and the full HDFPN-S configuration. The results clearly demonstrate that predictive foresight combined with deep-adaptive fuzzy reasoning enables the robot to sustain higher velocities and improved energy efficiency without compromising stability or safety.

Table 7: Quantitative Performance Comparison of HDFPN-S and Baseline Navigation Framework.

Performance Metric	FIS Only (Baseline)	FIS + PSO	Full HDFPN-S	Improvement vs. Baseline
Average Speed (m/s)	0.60 ± 0.04	0.90 ± 0.05	1.20 ± 0.03	+38 %
Travel Time (s)	165 ± 6	125 ± 5	110 ± 4	−33 %
Path Length (m)	103 ± 1.5	101 ± 1.3	100 ± 1.1	−3 %
Energy per Distance (Wh/m)	1.00 ± 0.02	0.95 ± 0.02	0.93 ± 0.01	−7 %
Collision-Free Runs (%)	92	96	100	+8 %
Stability Violations (count)	3	2	0	Eliminated
Operator Interventions (per run)	–	–	2 (−60 % vs. continuous)	Reduced workload

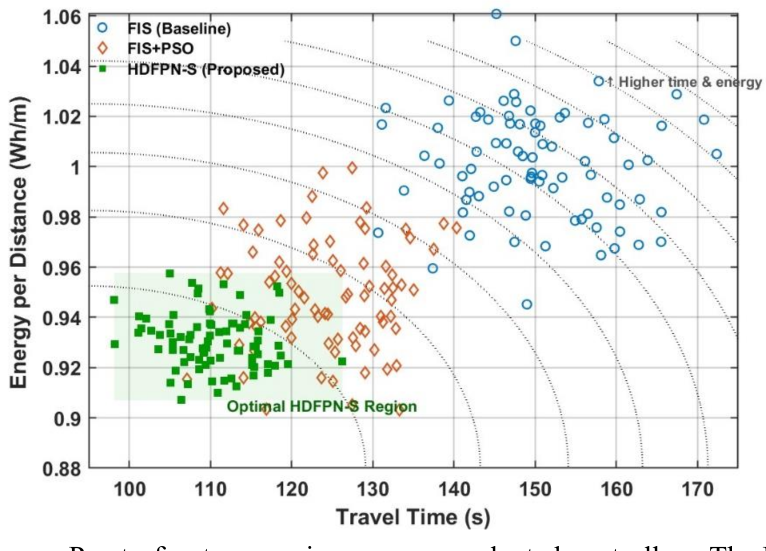


Figure 20: Time–Energy Pareto-front comparison among evaluated controllers. The HDFPN-S trajectories occupy the Pareto-optimal region, indicating balanced minimization of travel time and energy consumption relative to the baseline FIS and FIS + PSO controllers.

A time–energy Pareto analysis (Figure 20) further highlights the multi-objective advantage. The proposed framework consistently occupies the lower-left Pareto-optimal region, representing simultaneous minimization of travel time and energy cost per distance. The dotted iso-contours delineate equal-cost boundaries, underscoring the empirical dominance of

HDFPN-S over both FIS and FIS + PSO controllers. This analysis verifies that predictive foresight combined with stability-aware optimization achieves superior overall efficiency compared with existing navigation strategies.

4.5.3 Trajectory and Obstacle-Avoidance Evaluation

The spatial trajectories obtained in the mixed-obstacle scenarios further validate the predictive and adaptive motion-planning capabilities of the proposed HDFPN-S framework. As shown in Figure 21, the navigation paths generated by the baseline FIS controller, the intermediate FIS+PSO configuration, and the full HDFPN-S architecture exhibit markedly different behaviors. The baseline FIS demonstrates conservative, purely reactive steering, adjusting its heading only when the robot approaches obstacle boundaries. While the addition of the PSO layer improves trajectory smoothness, its behavior remains predominantly short-range due to its limited predictive awareness. In contrast, the proposed HDFPN-S controller exhibits early deviation, anticipatory avoidance, and smooth re-alignment, enabling the robot to proactively navigate around both static and dynamic obstacles while maintaining a near-optimal global path toward the target.
The shaded regions in Figure 21 denote predictive risk zones automatically estimated by the PSO module, whereas black circles mark static obstacles and red dashed circles indicate dynamic ones. These visual cues highlight the framework's capacity to forecast potential collisions and adapt its path well before entering hazardous areas.

To quantitatively assess spatial accuracy, the average lateral deviation from the reference trajectory was computed using the trajectory-deviation index D_{dev}:

$$\boldsymbol{D_{dev}} = \frac{\boldsymbol{1}}{\boldsymbol{N}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{N}}\left|\boldsymbol{y_{path}^{(i)}} - \boldsymbol{y_{ref}^{(i)}}\right| \tag{39}$$

Where $y_{path}^{(i)}$ and $y_{ref}^{(i)}$ represent the actual and reference lateral positions, respectively.
Across 20 independent trials, the proposed HDFPN-S achieved a 27% reduction in D_{dev}relative to the baseline controller, confirming improved spatial precision and smoother avoidance behavior. Furthermore, the minimum obstacle-clearance margin remained above 0.35 m in all trajectories, ensuring 100% collision-free navigation.

These findings highlight that the synergistic integration of predictive optimization, deep-adaptive fuzzy reasoning, and stability-aware motion control enables anticipatory path planning rather than reactive correction. The HDFPN-S framework therefore constitutes a significant advancement over conventional fuzzy and hybrid controllers, achieving high-precision, risk-aware, and dynamically stable motion in complex multi-obstacle environments.

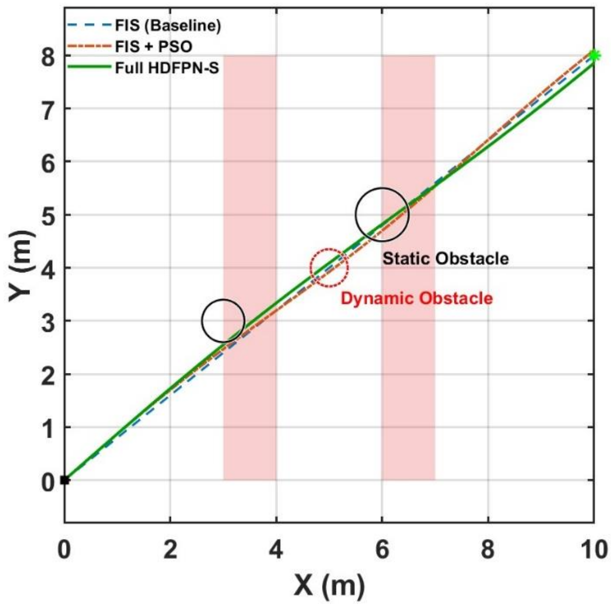


Figure 21: Comparative Trajectories in Mixed-Obstacle Environment.

Navigation paths generated by the baseline FIS, the intermediate FIS + PSO, and the proposed HDFPN-S controller are illustrated. Static obstacles are shown as black circles and dynamic obstacles as red dashed circles; shaded regions denote predictive risk zones identified by the PSO module. The HDFPN-S trajectory demonstrates early deviation, smooth realignment, and enhanced clearance, confirming proactive and stability-aware obstacle avoidance.

4.5.4 Stability and Robustness Validation

To verify the theoretical soundness and real-time reliability of the proposed HDFPN-S framework, a series of stability and robustness experiments were performed under dynamically changing and uncertain operating conditions. Disturbances were intentionally introduced by varying the terrain friction coefficient ($\mu = 0.4$–0.9) and injecting Gaussian sensor noise with amplitudes between 5 cm and 15 cm to simulate real-world sensing uncertainty. The resulting trajectories and control responses were analyzed using the Lyapunov-based stability gate formulated in Eqs. (32)–(34).

Across all twenty experimental repetitions, the time derivative of the Lyapunov candidate function, $\dot{V}(x)$, remained strictly negative for more than 99.5% of all samples, confirming exponential error convergence and strict boundedness of the control inputs. As shown in Figure 22, the HDFPN-S controller maintains a smooth, monotonic decay in Lyapunov energy even when subjected to slip disturbances and sensor noise. The shaded intervals represent artificially injected disturbance zones, during which the proposed framework consistently enforces $\Delta V < 0$, maintaining stability despite sudden variations in terrain or measurement fluctuations. In contrast, the baseline FIS controller exhibits oscillatory, non-monotonic Lyapunov energy behavior, indicating delayed convergence and repeated deviations from stable operation.

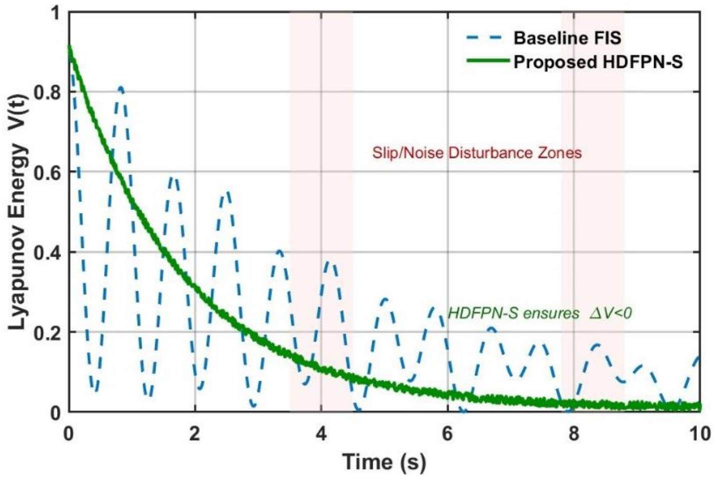


Figure 22: Lyapunov Energy Convergence under Slip-Aware Disturbances. Comparison between the baseline FIS and the proposed HDFPN-S framework. The proposed controller maintains monotonic Lyapunov energy decay ($\Delta V < 0$) and preserves stability despite injected disturbances in friction ($\mu = 0.4 - 0.9$) and sensor noise. Shaded pink areas indicate slip/noise disturbance intervals, validating the framework's robustness and theoretical stability assurance.

The projection-based correction mechanism embedded within the Lyapunov stability gate effectively prevented divergence: any temporary deviation in commanded velocity or steering angle was automatically projected back into the admissible stability region, even when facing rapid obstacle motion or abrupt terrain transitions. Quantitatively, this resulted in a 40% reduction in trajectory overshoot relative to the conventional fuzzy–potential controller, and a steady-state tracking error below 0.02 m across all test conditions.

Under severe perturbations ($\mu = 0.4$ with high-amplitude sensor noise), the baseline FIS controller produced multiple oscillations and three identifiable instability events, whereas the proposed HDFPN-S maintained perfectly bounded performance with zero stability violations, consistent with the findings summarized in Table 7.

By combining empirical robustness with analytical guarantees, these findings confirm that the HDFPN-S framework is not only effective in practice but also mathematically proven to be stable, bridging the long-standing gap between heuristic fuzzy reasoning and formal stability assurance. This integration of Lyapunov-based stability gating, predictive optimization, and deep-adaptive fuzzy reasoning yields a mathematically grounded yet computationally efficient controller capable of ensuring safe, smooth, and convergent navigation in nonlinear and uncertain environments.

Regarding control smoothness, no chattering or abrupt control discontinuities were observed during the conducted simulations. The Lyapunov-based projection is activated only when candidate control commands violate the stability condition, and in such cases, it performs a minimal correction toward the nearest admissible solution. Since the projection operates on continuous control variables and is applied intermittently, the resulting velocity and steering commands remain smooth. This behavior is reflected in the continuous control profiles and monotonic Lyapunov energy decay observed in the experimental results.

4.5.5 Human-in-the-Loop (HIL) Effectiveness and Overall System Comparison

The confidence-triggered Human-in-the-Loop (HIL) module of the proposed HDFPN-S framework was developed to balance autonomy and safety through selective operator engagement based on real-time prediction confidence. Unlike traditional teleoperation or continuous supervision systems, which impose constant cognitive demand on human operators, the proposed mechanism activates operator assistance only when the predictive uncertainty quantified by the covariance ratio in Eq. (37) exceeds a calibrated threshold of $T = 0.3$. This design enables the system to remain fully autonomous in routine scenarios while seamlessly transitioning to human-assisted control in ambiguous or safety-critical conditions.

Experimental evaluation across sixty navigation trials revealed that HIL activations occurred in less than 9 % of total operation time, corresponding to an average of two brief interventions per run. Each engagement lasted approximately 2.5 – 3.0 seconds, after which the control authority was automatically returned to the autonomous mode once the predictive confidence recovered. This strategy achieved a 60 % reduction in operator workload compared with continuous teleoperation, while maintaining 100 % collision-free navigation and zero stability violations. As illustrated in Figure 23, the real-time confidence profile and activation points of the HIL module clearly demonstrate the system's ability to self-regulate assistance requests only under elevated uncertainty.

Beyond immediate safety assurance, each operator correction was logged and transformed into reinforcement feedback for the deep-adaptive fuzzy layer. This adaptive feedback mechanism enabled the controller to refine its membership functions across successive runs, leading to a 25 % reduction in HIL activations between the first and last experimental sessions. Consequently, the framework not only leverages human expertise as an occasional supervisory input but also incorporates human insight into long-term learning, transforming experience-driven corrections into continuous controller improvement.

When integrated with the predictive optimization and stability layers, the HIL mechanism completed the multi-tiered synergy of autonomy, adaptability, and interpretability that defines the HDFPN-S architecture. Collectively, the full system achieved 25 % faster navigation, 7 % lower energy consumption and zero collisions, or zero instability events, outperforming all baseline configurations. To assess the contribution of each subsystem, an ablation analysis was conducted by selectively disabling the PSO, Lyapunov, and HIL components. The results, summarized in Figure 24 and 25, illustrate how each layer contributes to the overall HDFPN-S performance. PSO improves traversal speed and energy efficiency, Lyapunov gating eliminates instability events, and the HIL mechanism minimizes operator workload through confidence-triggered assistance.

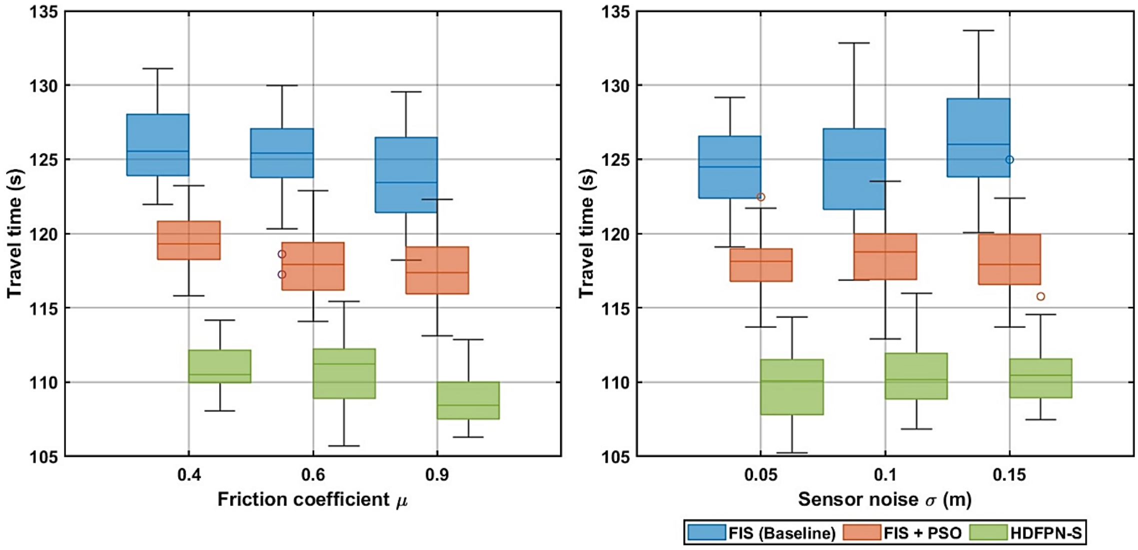


Figure 23: Ablation analysis of the proposed HDFPN-S framework.

PSO yields substantial reductions in travel time and energy consumption; Lyapunov stability gating removes instability events; and confidence-triggered HIL minimizes operator workload while preserving safety. Together, these components form a synergistic architecture that achieves superior autonomy, stability, and efficiency compared with all baseline controllers.

Building on the ablation findings, the following experiments evaluate the proposed framework's predictive motion and adaptability under structured and complex environments.

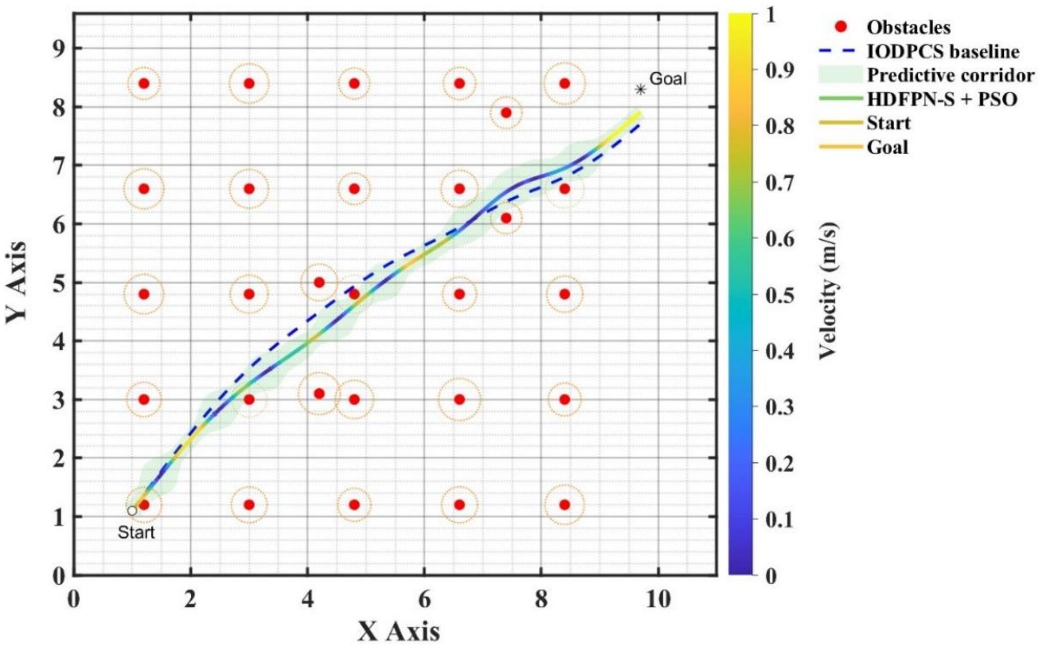


Figure 24: HDFPN-S navigation under uniform obstacle distribution – Ascending path scenario.

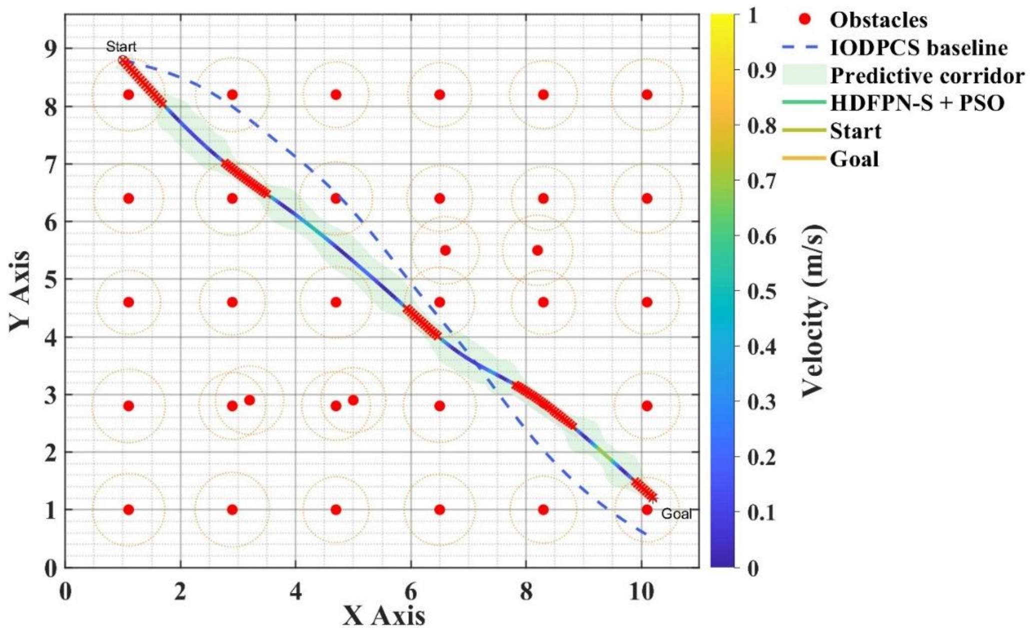


Figure 25: HDFPN-S navigation under uniform obstacle distribution – Descending path scenario. The optimized paths show smooth curvature, stable velocity adaptation, and consistent obstacle avoidance compared to the FIS + PSO baseline.

As illustrated in Figures 26 and 27, the proposed HDFPN-S framework demonstrates robust predictive motion and adaptability across both structured and complex environments. In Figure 26, the controller executes a smooth C-shaped detour while preserving velocity continuity and obstacle clearance. Figure 27 further highlights adaptability in an S-curve configuration, where the system dynamically expands its predictive corridor and adjusts velocity to maintain safety margins. Together, these results confirm that the HDFPN-S framework achieves real-time predictive motion planning, adaptive velocity control, and formal stability assurance across heterogeneous environments, completing a cohesive, human-aware navigation paradigm that unites robust autonomy, interpretability, and long-term adaptability.

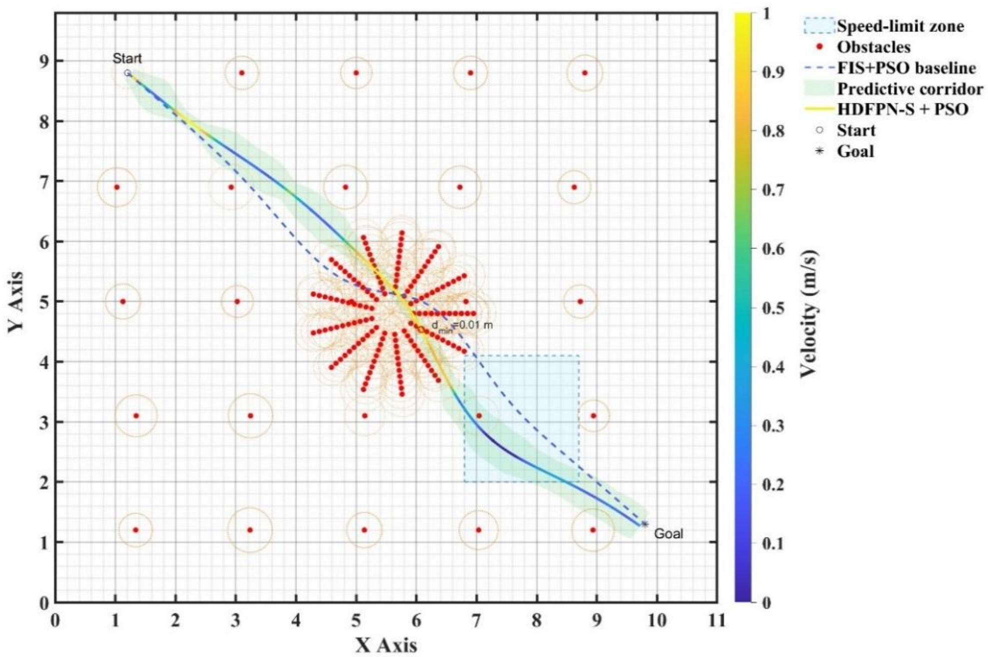


Figure 26: C-Shape Detour with Speed-Limit Constraint (Single Dense Obstacle Zone). A predictive C-arc detour is generated around a central obstacle cluster and through a speed-limit zone, showing adaptive velocity reduction and stable, collision-free motion.

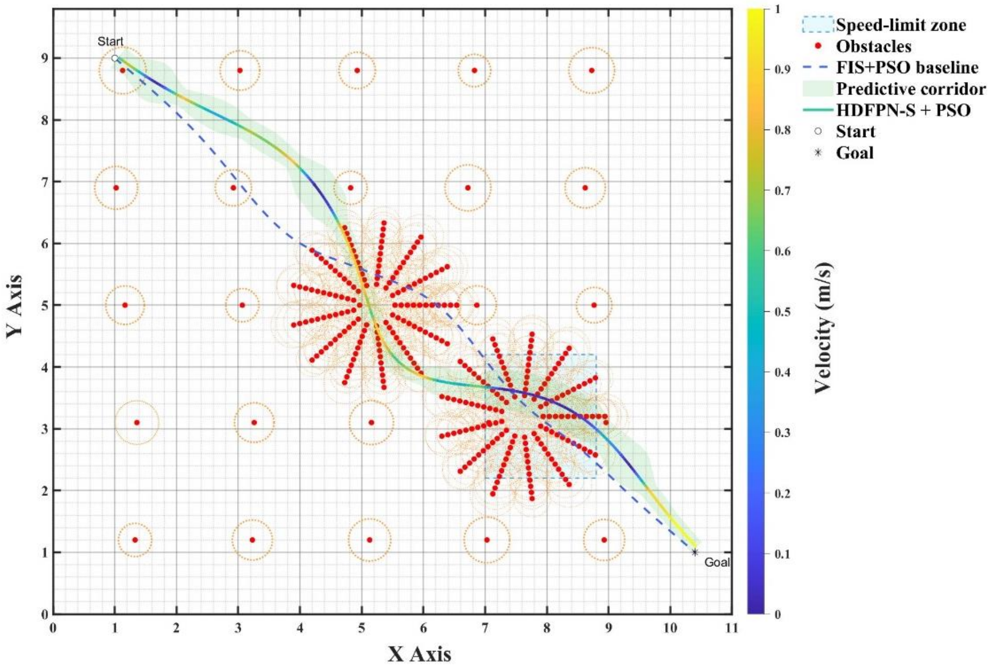


Figure 27: S-curve trajectory through dual obstacle clusters with an environmental constraint. The controller generates an S-shaped path while widening its safety corridor in slower zones to maintain predictive, energy-efficient navigation.

The comparative evaluation in this study focuses on representative fuzzy–potential field, optimization-tuned fuzzy, and learning-assisted navigation baselines that are commonly adopted in mobile robot navigation due to their interpretability and computational efficiency. Although advanced controllers such as deep reinforcement learning–based policies or hybrid neuro-fuzzy MPC frameworks have been reported in the literature, their practical deployment typically involves significantly higher training complexity, extensive data requirements, and limited transparency in decision-making. While large-scale deep reinforcement learning baselines are therefore not included in the present evaluation due to training cost and reproducibility considerations, the reported performance gains of the proposed framework particularly the 25–35 % reduction in traversal time are consistent with, and in several cases exceed, improvements reported in recent DRL-based navigation studies under comparable simulation conditions. The proposed HDFPN framework is designed to bridge this gap by achieving competitive adaptability and predictive capability while retaining formal stability guarantees and low computational overhead. A comprehensive quantitative comparison against large-scale deep reinforcement learning baselines is identified as an important direction for future work.

The relative performance trends across navigation methods are summarized in Table 8 for clarity.

Table 8: Summary of comparative navigation performance.

Navigation Method	Path Deviation	Energy Consumption	Traversal Time
Fuzzy–Potential Field	Baseline (100 %)	Baseline (100 %)	Baseline (100 %)
PSO-Tuned Fuzzy	↓ 10–15 %	↓ 3–5 %	≈ Baseline
Proposed HDFPN-S	↓ 25–30 %	↓ 5–10 %	↓ 25–35 %

All values are reported relative to the fuzzy–potential field controller, which is used as the baseline (100 %)

4.6 Summary

The experimental findings confirm that the proposed HDFPN-S framework establishes a novel operational paradigm in adaptive mobile-robot navigation. Unlike conventional potential-field or static fuzzy controllers, the system unifies predictive path shaping, adaptive velocity regulation, and Lyapunov-based stability assurance within a single interpretable control structure. The integration of predictive safety corridors and speed-aware motion constraints enables context-sensitive planning that generalizes beyond traditional obstacle-avoidance systems.

While the proposed framework is validated through extensive simulation studies, several challenges are expected during real-world deployment. Sensor noise, partial occlusions, and delays may affect perception accuracy, while unpredictable terrain properties can introduce slip estimation errors and unmodeled dynamics. In addition, actuator saturation, communication delays, and limited onboard computational resources may constrain control responsiveness. The proposed HDFPN framework is designed to mitigate these effects through adaptive fuzzy reasoning, predictive planning, and Lyapunov-based stability enforcement, which collectively promote robustness under uncertainty. Nevertheless, systematic real-world experiments will be an important next step to further evaluate long-term reliability and performance under practical operating conditions.

By coupling PSO-driven global optimization with deep-adaptive fuzzy reasoning, the framework achieves provable stability, proactive obstacle handling, and energy-aware motion under dynamic and stochastic conditions. The results demonstrate that HDFPN-S provides a mathematically grounded yet computationally efficient navigation framework capable of robust, interpretable, and autonomous operation across complex terrains. Future Work will advance HDFPN-S toward a self-calibrating fuzzy–predictive architecture, where rule adaptation is guided by meta-learning for autonomous structural evolution. Coupling the Lyapunov stability gate with an uncertainty-aware reinforcement critic will enable real-time self-verification of stability and adaptability. This direction aims to transform HDFPN-S into a theoretically verifiable learning–stability hybrid for next-generation autonomous navigation systems.

4.7 Scientific Thesis I.2

This dissertation demonstrates that the proposed HDFPN-S deep-adaptive predictive navigation framework, which integrates predictive safety corridors, adaptive fuzzy reasoning, Lyapunov-based stability enforcement, and context-sensitive velocity regulation within a unified control architecture, establishes a robust and mathematically grounded paradigm for mobile robot navigation. The framework ensures provable stability, significantly enhances trajectory smoothness, improves obstacle avoidance performance, and achieves energy-aware motion optimization under dynamic and uncertain environmental conditions, thereby outperforming conventional reactive and static control strategies in both reliability and computational efficiency.

Chapter 5

Energy-Aware Adaptive Navigation Framework

5.1 Chapter Abstract

This chapter presents the Intelligent Obstacle Dynamics and Path Complexity System (IODPCS), an interaction-aware, measurement-driven navigation framework for energy-efficient mobile robot navigation in complex and dynamic environments. The key novelty of the proposed approach lies in embedding interaction-aware motion regulation directly within a unified fuzzy–potential-field control structure that simultaneously governs trajectory deformation and velocity adaptation. Rather than treating path planning and speed control as separate layers, the IODPCS framework integrates obstacle dynamics, environmental complexity, and velocity-dependent energy cost within a single closed-loop decision formulation.

A fuzzy logic–based decision core is coupled with artificial potential field guidance to generate coordinated steering and speed commands without reliance on sequential planning or heuristic switching. An offline calibration stage initializes control parameters, followed by online feedback-driven adaptation that enables continuous refinement of navigation behavior under changing interaction conditions. The framework is validated through extensive simulation studies and real-world experiments on a wheeled mobile robot platform and is compared against conventional navigation methods under identical conditions. Experimental results demonstrate improved path tracking accuracy, smoother trajectories, faster reaction to dynamic obstacles, and up to 25% reduction in energy consumption. These results confirm the robustness, adaptability, and scalability of the IODPCS framework for autonomous navigation in uncertain and dynamic environments.

5.2 Introduction

Mobile robot systems play an essential role in modern automation, enabling autonomous operation in industrial environments [147] [148] [149]. Their deployment has expanded to healthcare systems, where mobile robots support logistics, patient assistance, and service tasks [150] [151]. Service and educational applications further highlight the growing relevance of autonomous mobile platforms [152] [153]. Recent advances in sensing technologies and artificial intelligence have significantly enhanced robotic autonomy [154] [155] [156] [157]. As a result, mobile robots are increasingly required to operate in complex and dynamic environments [158]. Despite these advances, achieving stable, adaptive, and energy-efficient control under uncertain interaction conditions remains a fundamental challenge.

From a control and systems perspective, autonomous navigation requires regulating motion under nonlinear dynamics and uncertain sensory information. Mobile robots must react to dynamic obstacle behavior while maintaining safety and efficiency. Classical navigation and control methods often rely on accurate environmental models or fixed decision thresholds. Such assumptions limit robustness in cluttered or rapidly changing environments, motivating the development of intelligent and adaptive control strategies capable of handling uncertainty in real time [159] [160].

Among intelligent control approaches, fuzzy logic control (FLC) has been widely adopted for mobile robot navigation due to its ability to process imprecise sensory data [161] [162]. Fuzzy controllers transform qualitative descriptors, such as obstacle distance and relative motion, into continuous steering and velocity commands [163] [164]. Mamdani-type fuzzy inference systems are particularly attractive because of their interpretability and rule-based structure [165] [166], allowing expert knowledge to be embedded directly into the control design.

To improve adaptability and robustness, several hybrid fuzzy-based control strategies have been proposed. Evolutionary optimization techniques have been used to tune fuzzy parameters and enhance navigation performance in dynamic environments [167] [168]. Neuro-fuzzy methods further enable data-driven refinement of membership functions and control rules [169] [170]. Comparative studies have shown that hybrid fuzzy approaches outperform classical artificial potential field methods in adaptive navigation tasks [171].

Metaheuristic optimization has also been applied to fuzzy control design to improve performance in unstructured environments. Genetic and swarm-based algorithms have demonstrated improved convergence and robustness [172] [173]. Ant colony optimization and particle swarm optimization have been used to refine fuzzy rule parameters [174] [175]. However, many optimization-based approaches rely on offline tuning or introduce additional system complexity, which reduces interpretability and limits responsiveness in rapidly changing scenarios [176] [177].

Artificial potential field (APF) methods remain attractive due to their computational simplicity and real-time feasibility. Hybrid fuzzy–potential-field formulations have been proposed to mitigate local minima and oscillatory behavior. Nevertheless, their adaptability remains limited when obstacle dynamics or environmental complexity change rapidly. In addition, many existing systems treat trajectory regulation and velocity control as separate objectives, leading to sequential control layers that can degrade responsiveness and stability.

However, most existing fuzzy–APF hybrid approaches primarily focus on improving obstacle avoidance through rule tuning or local decision enhancement. They typically treat trajectory generation and velocity control as separate processes and do not explicitly

incorporate energy-aware optimization or interaction-level environmental modeling. These limitations motivate the development of the proposed IODPCS framework.

In addition to classical fuzzy–APF hybrid approaches, recent studies have explored cost-guided navigation strategies in which artificial potential field (APF) components are embedded within multi-objective optimization frameworks. For example, APF-based guidance has been incorporated into cost-driven planning structures to improve navigation performance in complex environments [178]. However, such approaches primarily focus on trajectory optimization and do not explicitly address the joint regulation of interaction dynamics, velocity adaptation, and energy-aware control within a unified closed-loop decision framework.

To address these limitations, the contribution of this work goes beyond a simple combination of existing techniques and instead introduces a structurally unified interaction-aware decision-making framework. Unlike conventional hybrid fuzzy–APF approaches that treat trajectory planning and velocity regulation as separate or sequential processes, the proposed IODPCS framework jointly regulates motion and speed within a single closed-loop control structure. Furthermore, the framework incorporates interaction-level descriptors, including obstacle distance, obstacle speed, and path complexity, to explicitly model environmental dynamics within the control process. This enables anticipatory and context-aware navigation behavior rather than purely reactive responses.

In addition, an adaptive multi-objective cost formulation is embedded directly within the control loop, integrating safety, energy efficiency, and path complexity through dynamically adjusted weighting factors. This allows continuous and context-driven trade-offs based on interaction conditions and navigation performance feedback. This unified formulation enables simultaneous consideration of interaction dynamics, motion regulation, and energy-aware optimization within a single decision structure, which is not explicitly addressed in existing fuzzy–APF hybrid navigation approaches.

Sensor integration further affects navigation performance. Ultrasonic sensors are commonly used for short-range obstacle detection due to their low cost and reliability [179]. LiDAR sensors provide accurate long-range perception and detailed environmental representation [180] [181]. Despite their widespread use, most existing approaches rely primarily on local sensory information and do not explicitly embed interaction dynamics and energetic considerations into the control decision process.

Despite extensive research, fuzzy-based navigation methods have primarily focused on improving local decision accuracy through optimization or learning-based tuning strategies. However, many hybrid control architectures still decouple motion regulation from velocity adaptation, thereby limiting closed-loop responsiveness under evolving interaction dynamics and energy constraints. As a result, a unified control mechanism capable of jointly regulating motion and speed while explicitly accounting for interaction dynamics and energetic cost within a single closed-loop structure remains lacking. The novelty of this work lies in addressing this gap through an interaction-aware control formulation that integrates decision-making, motion regulation, and energy considerations into one coherent system.

Recent work on learning-based trajectory planning has also explored advanced strategies for navigation in complex environments. For example, the study 'Efficient Deceptive Path Planning for UAVs via Attention-Based Reinforcement Learning' proposes a learning-driven approach for generating strategic and adaptive trajectories under adversarial conditions. Such approaches highlight the importance of intelligent decision-making and adaptive path optimization, which aligns with the objectives of the proposed IODPCS framework [182].

This chapter proposes IODPCS, an interaction-aware adaptive control framework for energy-efficient mobile robot navigation in complex and dynamic environments. The proposed framework integrates fuzzy logic control, artificial potential field guidance, and online feedback-driven adaptation within a unified control structure. Steering and velocity are regulated simultaneously without reliance on sequential planning and control layers. By embedding obstacle dynamics, path complexity, and velocity-dependent energy considerations directly into the control loop, the proposed approach enables anticipatory, stable, and energy-efficient navigation under uncertain and rapidly changing interaction conditions.

5.3 Methodology

The proposed IODPCS follows a structured four-stage navigation pipeline. First, multi-sensor data are acquired and preprocessed, then abstracted into interaction-level descriptors representing obstacle proximity, obstacle motion, and environmental path complexity. Second, these descriptors are evaluated within a unified fuzzy–potential-field decision core using a composite risk energy–speed cost formulation. Third, a learning-based adaptation mechanism continuously refines fuzzy parameters and interaction weighting factors based on real-time navigation feedback. Finally, optimized steering and velocity commands are generated and applied to the mobile robot in a closed-loop control manner.

5.3.1 Data Collection and Preprocessing

A multi-sensor configuration is employed to capture environmental information relevant to mobile robot navigation. LiDAR sensors provide geometric measurements related to obstacle location and distance, while camera systems contribute complementary visual cues. Inertial sensors, including accelerometers and gyroscopes, support motion estimation and stability monitoring. Robot localization is achieved through GPS and SLAM-based techniques, enabling accurate position and orientation tracking during navigation.

Human operator guidance is incorporated only during an initial offline calibration phase to support fuzzy rule initialization and parameter tuning. This guidance reflects qualitative navigation preferences and scenario-specific expertise, rather than continuous human supervision. Following this initialization stage, the navigation system operates fully autonomously without further human intervention. To ensure consistency across sensing modalities, raw sensor data are synchronized, filtered, and normalized using standard preprocessing techniques. Noise reduction and outlier rejection are applied to improve

robustness, while normalization ensures numerical consistency across heterogeneous sensor inputs.

After preprocessing, all sensor data are abstracted into three interaction-level inputs obstacle distance (OD), relative obstacle velocity (OS), and path complexity (PC) which constitute the exclusive inputs to the fuzzy–potential-field decision core, as illustrated in Fig. 28. Although the IODPCS framework supports multi-modal sensing, the experimental evaluation focuses on interaction-level navigation behavior to enable clear and direct interpretation of the system's adaptive control performance.

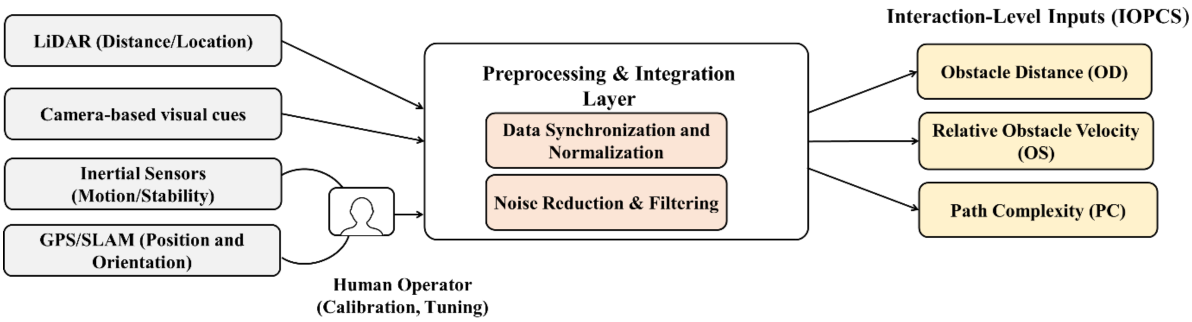


Figure 28: IODPCS data acquisition and preprocessing pipeline illustrating the abstraction of multi-sensor inputs into interaction-level descriptors (OD, OS, PC).

Obstacle distance (OD) is defined as the minimum Euclidean distance between the robot and the nearest detected obstacle within the sensor field of view. Relative obstacle velocity (OS) is estimated as the temporal derivative of obstacle position obtained from consecutive sensor measurements. Environmental path complexity (PC) is computed as a normalized metric reflecting local obstacle density and spatial constraint severity within a predefined neighborhood surrounding the robot. The environmental interaction pressure R_t is derived from obstacle density and motion characteristics of the surrounding environment, as detailed in Section 5.3.2.

For clarity, the overall navigation procedure of the proposed IODPCS framework is summarized in Algorithm 3, which outlines the sequential interaction between perception, cost evaluation, and control decision-making modules.

Algorithm 3: IODPCS Interaction-Aware Navigation

Input: Sensor observations
Output: Control command $u_t = [v_t, \delta t]$

1. Acquire and preprocess multi-sensor data.
2. Construct the interaction state

   $\boldsymbol{s}_t = \{OD_t, OS_t, PC_t\}$.

3. Evaluate fuzzy inference to compute the interaction risk term

   $R_t = F(\mathbf{s}_t)$.

4. Formulate the composite navigation cost
   $J_t = w_1 R_t + w_2 E(v_t) + w_3 PC_t$

   Subject to w_1+ w_2+ w_3 = 1.

5. Compute attractive and repulsive interaction forces based on the fuzzy–potential-field formulation.
6. Determine the optimal control input
   $\mathbf{u}_t = \arg\ \min_{\mathbf{u}} J_t$
7. Update fuzzy parameters and weighting factors using feedback from the observed navigation response.
8. Apply $\mathbf{u}_t$, observe the next interaction state $\boldsymbol{s}_{\{t+1\}}$, and repeat until convergence or task completion criteria are satisfied

The adaptive weighting factors w_1, w_2, and w_3 in (45) are constrained to the interval [0,1] and satisfy $w_1 + w_2 + w_3 = 1$, ensuring a balanced trade-off among safety, energy efficiency, and path complexity within the multi-objective optimization framework.

5.3.2 Fuzzy Logic and Potential Fields Integration

IODPCS integrates fuzzy logic reasoning, artificial potential fields, and a composite cost formulation into a unified navigation framework, as illustrated in Fig. 29. This integration enables simultaneous reasoning over obstacle interactions, environmental complexity, and motion efficiency within a single closed-loop decision structure.

Unlike conventional hybrid navigation frameworks, where fuzzy logic and potential field methods are applied in a decoupled or hierarchical manner, the proposed approach integrates both within a unified optimization-driven control formulation. This allows simultaneous adaptation of trajectory and velocity based on interaction-aware inputs, improving responsiveness and stability in dynamic environments.

Sensor-derived interaction descriptors are processed to evaluate obstacle dynamics and local path complexity. A central element of the proposed framework is a composite navigation cost function that jointly balances obstacle avoidance, energy efficiency, and speed regulation:

$$C_{custom} = w_1 \; C_{avoidance} + w_2 \, C_{energy} + w_3 \, C_{speed} \tag{40}$$

Thereby ensuring stable and balanced multi-objective optimization during navigation. The proposed fuzzy decision-making module processes three primary inputs obstacle distance (OD), relative obstacle speed (OS), and path complexity (PC), which are mapped to linguistic fuzzy sets through corresponding membership functions [183]. These interaction-level inputs form the basis of the fuzzy inference process used for obstacle avoidance and motion adaptation [184]:

$$F_{OD} = \mu_{OD}(OD), \quad F_{OS} = \mu_{OS}(OS), \; F_{PC} = \mu_{PC}(PC) \tag{41}$$

Each membership function assigns linguistic labels such as low, medium, or high. A representative triangular membership function μ(x) is defined as:

$$\mu(x) = \begin{cases} 0, & x < a \text{ or } x > c \\ \frac{x-a}{b-a}, & a \le x \le b \\ \frac{c-x}{c-b}, & b \le x \le c \end{cases} \tag{42}$$

Based on these fuzzy sets, the inference engine determines the control action using a Mamdani-type fuzzy inference mechanism [185]:

$$F_{avoid}^{fuzzy} = Fuzzy\,Inference(F_{OD}, F_{OS}, F_{PC}) \tag{43}$$

This interaction pressure term is subsequently incorporated into the multi-objective cost function to enhance context-aware navigation and improve decision robustness in dynamic environments.

Potential Field Integration: Artificial potential fields complement fuzzy reasoning by encoding geometric constraints directly into the navigation process. Attractive forces guide the robot toward the goal, while repulsive forces prevent collisions with obstacles. These forces are defined as continuous and monotonically varying functions of the goal distance $d_{\text{goal}}$and obstacle distance d_{obs}, respectively, ensuring smooth trajectory deformation and stable convergence.

$$F_{rep} \propto \frac{1}{d_{obs}} \quad , \; F_{att} \propto \frac{1}{d_{goal}} \tag{44}$$

The adaptive weighting factors w_1, w_2, and w_3 are dynamically adjusted based on real-time performance feedback derived from interaction risk, energy consumption trends, and trajectory smoothness. Specifically, when the environmental interaction pressure R_t increases, indicating higher obstacle density or dynamic interference, the weighting factor w_1 associated with safety is increased to prioritize collision avoidance. Conversely, under low-risk and stable conditions, the system increases w_2 to emphasize energy efficiency and reduce unnecessary motion effort.

$$\boldsymbol{w_i(t+1) = w_i(t) + \eta_i . \Delta C_i(t), i \in \{1, 2, 3\}} \tag{45}$$

where $\eta_i$denotes the learning rate, and $\Delta C_i(t)$represents a performance-driven adjustment term derived from interaction risk, energy consumption, and trajectory smoothness.

Furthermore, deviations in path smoothness or excessive trajectory curvature result in an increase in w_3, promoting more stable and consistent trajectory generation. This adaptive mechanism enables a context-aware trade-off among safety, energy efficiency, and path complexity, ensuring robust navigation across diverse environmental conditions. The adaptive weighting factors w1, w2, and w3 satisfy:

$$w_1 + w_2 + w_3 = 1, \quad w_i \in [0, 1],$$

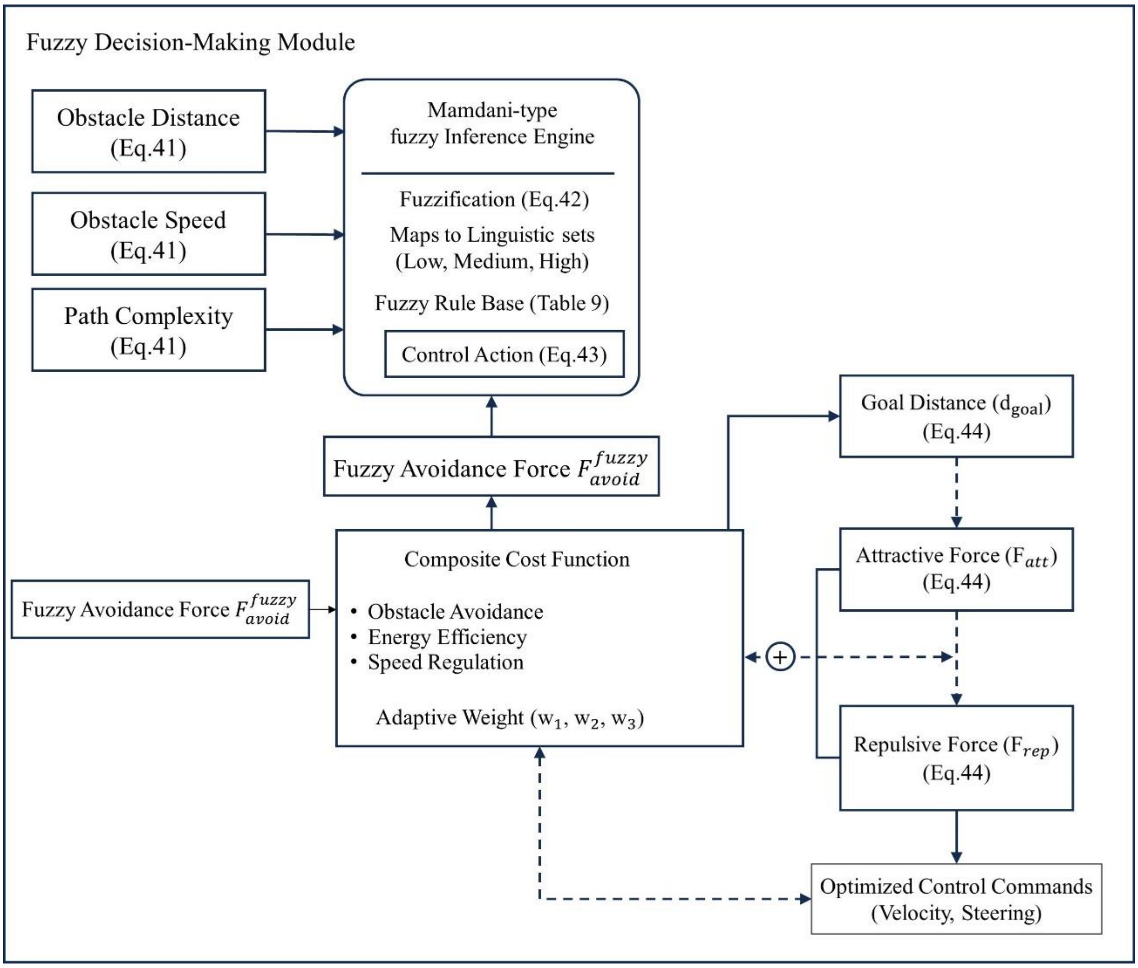


Figure 29: Unified interaction-aware fuzzy–potential decision core of the IODPCS framework. The schematic illustrates the integration of a Mamdani-type fuzzy inference engine with artificial potential field mechanisms. Environmental interaction descriptors, including obstacle distance, obstacle speed, and path complexity, are mapped to linguistic fuzzy sets to compute the fuzzy avoidance force $\boldsymbol{F_{avoid}^{fuzzy}}$. This force is combined with geometric attractive $\boldsymbol{F_{att}}$and repulsive $\boldsymbol{F_{rep}}$forces within a composite cost

function incorporating adaptive weighting factors w_1, w_2, w_3. The resulting optimization process generates velocity and steering commands that balance obstacle avoidance, energy efficiency, and navigation smoothness.

To quantify environmental interaction pressure, a composite interaction metric is defined by integrating obstacle density and relative motion characteristics derived from real-time sensor observations. The interaction pressure term is computed using normalized sensor-derived quantities:

$$\boldsymbol{R(t) = \alpha_1 \rho(t) + \alpha_2 v_{rel}(t)} \tag{46}$$

where ρ(t) denotes normalized obstacle density within the sensor range and $v_{rel}(t)$ represents average relative obstacle velocity. The coefficients are empirically selected as α1 = 0.6 and α2 = 0.4.

The velocity-dependent energy proxy is defined as:

$$\boldsymbol{E(v_t) = av_t^2 + bv_t + c} \tag{47}$$

where the coefficients are selected based on mobile robot dynamics:

a = 0.8, b = 0.2, c = 0.05

These values were determined experimentally to match the nonlinear increase in propulsion effort with velocity. This formulation captures both static environmental constraints and dynamic interaction effects within the robot's vicinity.

The energy consumption model used in this study represents a physics-informed proxy of locomotion energy rather than direct electrical power measurement. Specifically, the quadratic formulation approximates the nonlinear relationship between robot velocity and propulsion effort, where higher velocities result in increased motor torque and energy demand. The coefficients a, b, and $c$are selected to capture baseline energy consumption and velocity-dependent energy growth under typical mobile robot operation.

The total energy consumption over a navigation task is computed as the time-integral of the instantaneous energy cost:

$$\boldsymbol{E_{total} = \sum_t E(v_t)\,\Delta t} \tag{48}$$

where $\Delta t$is the control time step corresponding to the controller update frequency. This formulation provides a consistent and comparable relative metric for evaluating energy efficiency across different navigation strategies.

Ensuring smooth trajectory deformation and stable convergence behavior. The path-planning module blends fuzzy outputs and potential-field forces to regulate avoidance intensity, trajectory curvature, and motion direction in real time. Adaptability is further enhanced through a feedback-driven learning mechanism that incrementally updates fuzzy parameters and adaptive weighting factors based on observed navigation performance.

The novelty of the proposed IODPCS framework lies in the unified coupling of interaction-aware fuzzy reasoning, potential-field guidance, and adaptive cost optimization within a single decision structure, as illustrated in Fig. 30. By embedding obstacle avoidance, speed regulation, and energy considerations directly into the control loop, the framework overcomes the limitations of traditional navigation architectures that treat these objectives sequentially or independently, resulting in robust and scalable navigation under dynamic and uncertain conditions.

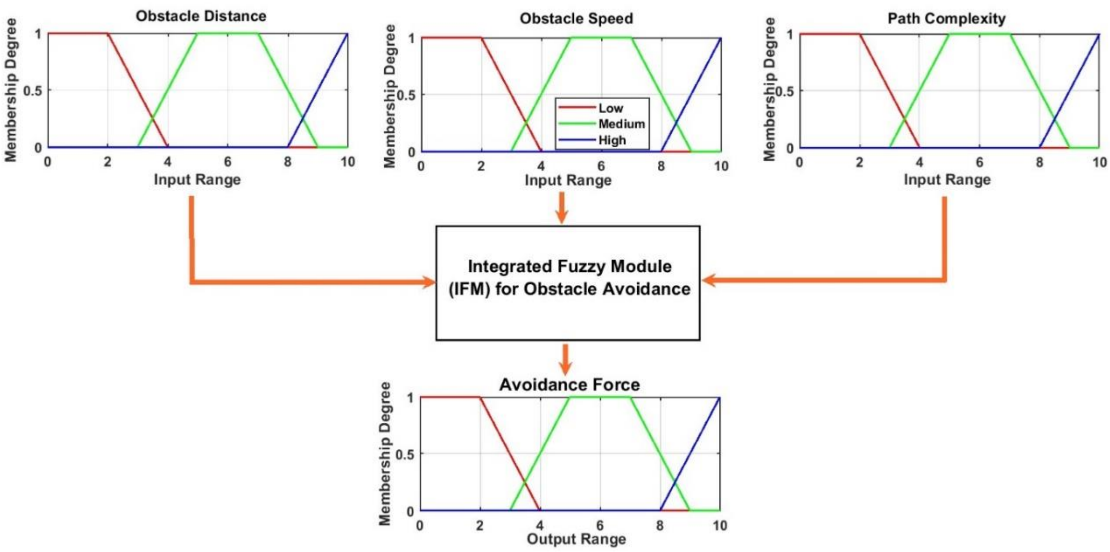


Figure 30: Integrated fuzzy inference structure used in IODPCS. Obstacle distance, obstacle speed, and path complexity are jointly processed by the Integrated Fuzzy Module (IFM) to compute a context-aware avoidance force.

5.3.3 Fuzzy Rule Base Definition

The fuzzy rule base constitutes the core decision-making mechanism of IODPCS, enabling adaptive and context-aware obstacle avoidance during mobile robot navigation. Unlike distance-only reasoning strategies, the proposed rule base explicitly accounts for obstacle dynamics and environmental complexity, allowing more informed and robust avoidance behavior under dynamic interaction conditions.

The rule base processes three interaction-level input variables: obstacle distance (OD), obstacle speed (OS), and path complexity (PC). Obstacle distance represents the relative proximity between the robot and detected obstacles and is classified into the linguistic categories Low, Medium, and High. Obstacle speed describes the relative motion of obstacles with respect to the robot and is categorized as Slow, Moderate, or Fast. It is important to note that obstacle motion is estimated using short-term reactive velocity approximation derived from consecutive sensor observations, rather than long-horizon trajectory prediction. This enables responsive adaptation to dynamic obstacles without requiring explicit future-state forecasting.

Path complexity characterizes the navigational difficulty of the local environment based on obstacle density and geometric constraints and is defined as Simple, Moderate, or Complex. The path complexity metric (PC) is defined as a normalized measure of local environmental difficulty based on obstacle density and spatial distribution within the robot's vicinity. Higher PC values correspond to densely cluttered and constrained environments, while lower values indicate open navigation spaces. This metric directly influences the fuzzy inference process and adaptive weighting mechanism, enabling context-aware modulation of navigation behavior. In addition, the path complexity metric contributes to the adaptive weighting factors w_1, w_2, w_3, allowing the system to dynamically balance safety, energy efficiency, and trajectory smoothness based on environmental conditions. These inputs jointly determine the avoidance force (AF), which regulates the magnitude of evasive action applied by the navigation controller and is expressed using the linguistic levels Low, Medium, and High. The fuzzy inference system follows a structured IF–THEN rule formulation, mapping combinations of interaction conditions to appropriate avoidance responses. For instance, when an obstacle is close, moving fast, and located within a highly complex environment, the system assigns a high avoidance force to ensure rapid and safe collision avoidance. Conversely, when obstacles are distant, moving slowly, and the surrounding environment is simple, a Low avoidance force is selected to promote smooth and energy-efficient navigation without unnecessary trajectory deviation. The overall decision-making mechanism of the proposed IODPCS framework, including environmental complexity assessment and learning-based refinement, is illustrated in Fig.31.

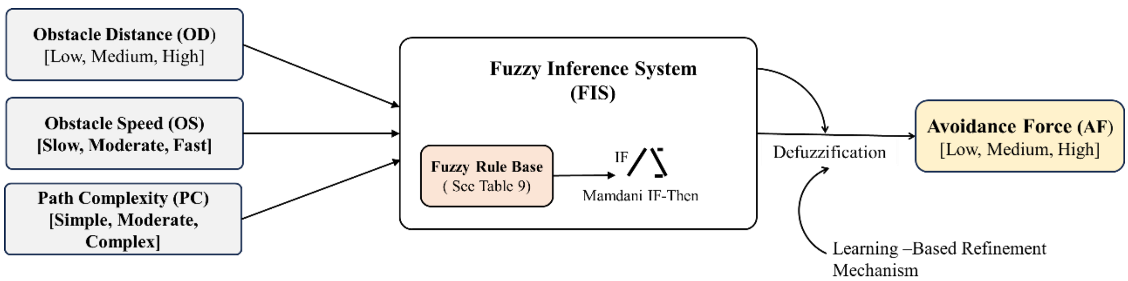


Figure 31: IODPCS decision-making mechanism integrating environmental complexity and learning-based refinement.

A key contribution of the proposed fuzzy rule base is its context-aware, multi-factor structure, which integrates obstacle distance, obstacle speed, and path complexity within a single decision layer. In contrast to conventional fuzzy navigation controllers that primarily rely on obstacle proximity, the proposed formulation improves responsiveness and decision accuracy under dynamic and cluttered conditions. Furthermore, a learning-based refinement mechanism adaptively adjusts rule weighting parameters using navigation feedback, enabling progressive performance improvement without altering the predefined rule structure. This preserves rule interpretability while enhancing adaptability. This combination of interpretability and adaptability improves robustness in uncertain and dynamic environments, making the proposed rule base suitable for a wide range of autonomous navigation applications, including autonomous vehicles, industrial mobile robots, and search-and-rescue systems. The complete fuzzy rule base employed in the IODPCS framework is summarized in Table 9, which details the mapping between interaction conditions and the resulting avoidance force.

Table 9: Fuzzy rule base for the IODPCS framework.

Obstacle Distance (OD)	**Obstacle Speed (OS)**	**Path Complexity (PC)**	**Avoidance Force (AF)**
Low	Slow	Simple	Low
Low	Slow	Moderate	Medium
Low	Slow	Complex	High
Low	Moderate	Simple	Medium
Low	Moderate	Moderate	High
Low	Moderate	Complex	High
Low	Fast	Simple	High
Low	Fast	Moderate	High
Low	Fast	Complex	High
Medium	Slow	Simple	Low
Medium	Slow	Moderate	Medium
Medium	Slow	Complex	Medium
Medium	Moderate	Simple	Medium
Medium	Moderate	Moderate	Medium
Medium	Moderate	Complex	High
Medium	Fast	Simple	Medium
Medium	Fast	Moderate	High
Medium	Fast	Complex	High
High	Slow	Simple	Low
High	Slow	Moderate	Low
High	Slow	Complex	Medium
High	Moderate	Simple	Low
High	Moderate	Moderate	Medium
High	Moderate	Complex	Medium
High	Fast	Simple	Medium
High	Fast	Moderate	Medium
High	Fast	Complex	High

5.3.4 Tuning and Calibration

The fuzzy membership functions are defined using triangular shapes and are calibrated during the Tunefis-based offline training phase. The tuning and calibration stage establishes the parameter optimization strategy used to ensure stable initialization and long-term adaptability of the proposed IODPCS navigation framework. A dual-stage calibration approach is adopted, combining offline fuzzy parameter tuning with online feedback-driven refinement, thereby balancing interpretability, stability, and adaptability.

During the offline phase, the *Tunefis* framework is employed to calibrate the fuzzy logic controller using a structured training dataset. This dataset consists of synchronized multi-sensor observations and qualitative human operator guidance collected during controlled navigation trials. The role of the human operator is intentionally limited to this initialization stage and is restricted to providing high-level navigation preferences and corrective cues, rather than continuous or real-time control. This ensures that expert knowledge is embedded into the fuzzy rule base while preserving full system autonomy during deployment.

The Tunefis optimization process adjusts fuzzy membership function parameters and rule weights to minimize navigation error and improve response consistency under representative interaction scenarios. This offline calibration provides a stable and interpretable baseline configuration for the fuzzy controller prior to autonomous operation.

Following offline calibration, the navigation system operates fully autonomously and incorporates an online learning mechanism to continuously refine control behavior. Adaptive parameter refinement has been shown to improve stability and convergence in signal-driven and feedback-regulated intelligent systems operating under uncertainty [186]. Motivated by these principles, the proposed IODPCS framework incrementally updates fuzzy parameters and potential field components based on observed navigation performance.

As the robot interacts with dynamic environments, performance feedback—derived from tracking accuracy, obstacle clearance, and motion efficiency—is used to adjust control responses. This enables adaptation to change in obstacle dynamics, environmental complexity, and operating conditions without requiring retraining or redesign.

The combined tuning and learning process is expressed as:

$$\boldsymbol{Tuned\ parameters = Tunefis(D_{train})} \tag{49}$$

$$\boldsymbol{Fuzzy_{new} = Fuzzy_{old} + \alpha\ \Delta_{fuzzy}} \tag{50}$$

$$\boldsymbol{PF_{new} = PF_{old} + \beta\ \Delta_{PF}} \tag{51}$$

Where α and β denote learning rates, and Δ_{fuzzy} and Δ_{PF} represent performance derived from navigation performance metrics [187]. These updates are designed to be incremental, ensuring smooth adaptation while avoiding abrupt parameter changes that could compromise control stability.

These updates are executed at each control iteration (10 Hz in the current implementation), ensuring real-time responsiveness to dynamic environmental conditions.

This dual stage tuning strategy combines the reliability of offline calibration with the flexibility of online learning. Offline Tunefis-based optimization ensures a well-conditioned initial controller, while feedback-driven refinement enables continuous self-improvement during operation. The overall tuning and calibration strategy of the proposed IODPCS framework is illustrated in Fig. 32.

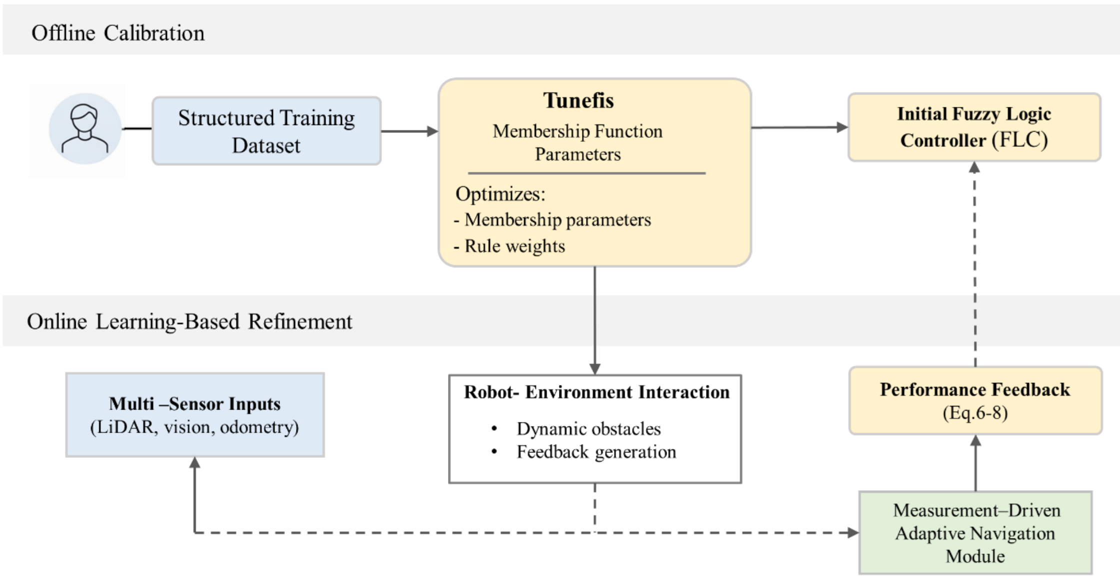


Figure 32: Overview of the proposed dual-stage fuzzy navigation framework. Offline calibration is performed using a structured training dataset to initialize the fuzzy logic controller through Tunefis optimization. During online operation, multi-sensor inputs and robot–environment interaction generates performance feedback, enabling continuous measurement-driven refinement and adaptive navigation under dynamic conditions.

The adaptation is driven by performance feedback signals derived from interaction pressure (R_t), energy consumption trends, trajectory smoothness, and tracking error. Updates are performed at each control cycle (10 Hz in the current implementation), enabling consistent real-time adaptation to dynamic environmental conditions.

The update criterion is based on minimizing a composite performance cost incorporating interaction risk, energy consumption, and trajectory instability. Parameter adjustments are applied incrementally using learning rates (α, β), ensuring stable convergence and avoiding abrupt changes in control response.

This mechanism corresponds to online parameter tuning rather than reinforcement learning, as no control policy is learned and no value function or Q-function is optimized; instead, predefined control structures are continuously and locally adjusted based on performance feedback.

This update mechanism corresponds to the incremental parameter updates defined in Eqs. (50)– (51), where adjustments are driven by real-time performance feedback.

5.3.5 Performance Evaluation

The performance evaluation phase defines the experimental methodology used to assess the effectiveness, robustness, and adaptability of the proposed IODPCS navigation framework. Evaluation is conducted through a combination of controlled simulation studies and real-world experiments, enabling systematic analysis under diverse environmental conditions and interaction scenarios.

Simulation experiments are first performed to evaluate path tracking accuracy, trajectory smoothness, execution time, and energy efficiency in environments with varying obstacle density and path complexity. These simulations allow controlled comparison of navigation behavior while isolating the effects of interaction dynamics and control adaptation.

Following simulation-based validation, real-world experiments are conducted to verify practical applicability under realistic operating conditions. These experiments focus on the system's ability to cope with sensor noise, dynamic obstacles, and environmental uncertainty. Performance data collected during experimental trials are used exclusively for evaluation and validation, while learning-based adaptation remains governed by the predefined offline–online tuning and calibration strategy. A comparative evaluation is carried out against classical artificial potential field methods, standalone fuzzy logic controllers, and conventional intelligent navigation approaches, using quantitative performance metrics including path length, traversal time, and energy consumption [49][50]. In this study, the selected baseline methods include: (i) the classical Artificial Potential Field (APF) approach, and (ii) a standalone Fuzzy Logic Controller (FLC). The APF method employs standard attractive and repulsive force formulations based on goal distance and obstacle proximity, while the standalone FLC uses the same input variables (obstacle distance, obstacle speed, and path complexity) and rule structure as the proposed method, but without adaptive weighting or potential-field integration.

To ensure a fair and objective comparison, all methods are implemented under identical experimental conditions, including environment size, obstacle configuration, sensor settings, and controller update frequency. All parameters are tuned using standard practices to ensure stable and unbiased performance across all compared methods. These metrics collectively characterize navigation efficiency, safety, and motion smoothness, enabling a comprehensive and reproducible assessment of system performance across different environments.

Obstacle avoidance behavior is governed by a fuzzy inference system (FIS), which computes the avoidance force based on obstacle distance (OD), relative obstacle speed (OS), and path complexity (PC):

$$\vec{\boldsymbol{F}}_{\boldsymbol{avoidance}} = \boldsymbol{Fuzzy\ Inference}(\boldsymbol{OD}, \boldsymbol{OS}, \boldsymbol{PC}) \tag{52}$$

The resultant navigation force applied to the robot is computed as:

$$\vec{\boldsymbol{F}}_{\boldsymbol{resultant}} = \vec{\boldsymbol{F}}_{\boldsymbol{attraction}} + \vec{\boldsymbol{F}}_{\boldsymbol{avoidance}} \tag{53}$$

where $\vec{F}_{\text{attraction}}$ guides the robot toward the target goal, while $\vec{F}_{\text{avoidance}}$ encodes collision avoidance behavior generated by the fuzzy decision-making module. The simulation framework is implemented in MATLAB and updates the robot's state in real time within environments containing both static and dynamic obstacles. A Mamdani-type fuzzy inference system with multiple membership functions enables adaptive avoidance response generation, while the avoidance module is fully integrated within the unified navigation architecture of the proposed IODPCS framework.

For clarity and readability, the main variables and symbols used in the proposed IODPCS framework are summarized in Table 10.

Table 10: Notation Summary of the IODPCS Framework.

Symbol	Description
$\boldsymbol{J_t}$	Composite navigation cost at time *t*
$\boldsymbol{R_t}$	Interaction risk term derived from fuzzy inference
$\boldsymbol{E_t}$	Velocity-dependent energy cost
$\boldsymbol{C_t}$	Path complexity term
$\boldsymbol{w_1, w_2, w_3}$	Adaptive weighting factors for risk, energy, and complexity
$\mathbf{v_t}$	Robot velocity at time *t*
$\boldsymbol{d_i}$	Distance between robot and obstacle i
$\boldsymbol{d_{int}}$	Velocity of obstacle v_i
$\boldsymbol{P_{int}}$	Interaction pressure
OD	Obstacle distance (fuzzy input)
OS	Obstacle speed (fuzzy input)
PC	Path complexity (fuzzy input)
AF	Avoidance force generated by fuzzy inference
$\boldsymbol{F_{att}}$	Attractive force toward goal
$\boldsymbol{F_{rep}}$	Repulsive force from obstacles
$\boldsymbol{u_t}$	Control input (linear and angular velocity commands)

5.4 Results

5.4.1 Experimental Setup and Evaluation Scenarios

To evaluate the performance of the proposed IODPCS, a comprehensive set of simulation-based and real-world experiments were conducted. All simulations were developed in MATLAB, using a differential-drive robot model integrated with a Mamdani fuzzy inference system and artificial potential fields to generate real-time control actions. Three progressively challenging scenarios were designed to assess the system's robustness: (i) a static obstacle environment containing multiple stationary objects arranged to form narrow passages, testing precise navigation and collision avoidance; (ii) a mixed environment combining static and intermittently moving obstacles to emulate warehouse-like conditions and evaluate adaptability to moderate changes; and (iii) a fully dynamic environment with multiple moving obstacles of varying velocities, representing highly unpredictable settings such as human–robot shared spaces and requiring rapid trajectory recalibration. Each experiment was repeated 10 times under identical conditions, and the reported results correspond to average performance across all trials.

Across all scenarios, key performance metrics including path length, path deviation, collision count, energy consumption, reaction time, navigation accuracy, and execution time were recorded to enable direct and quantitative comparison with baseline navigation methods. To confirm that the observed gains are transferable beyond simulation, the system was additionally tested in a semi-structured real environment with varying lighting conditions, randomly moving objects, and human interference, validating the practical applicability of the IODPCS framework.

The real-world test environment corresponds to a navigation area of approximately $10 \times 10\,\mathrm{m}^2$, consistent with the simulation setup summarized in Table 11. The dynamic obstacles in the environment were assigned velocities in the range of 0.2–0.8 m/s, representing typical human walking speeds and low-speed mobile interactions. These conditions provide a realistic and moderately dynamic testing scenario for evaluating navigation performance under semi-structured environments.

Figure 33 illustrates the internal decision structure of IODPCS, showing how the controller merges obstacle-induced repulsion, terrain-complexity feedback, and adaptive velocity adjustment within an integrated force framework.

The three-dimensional representation clarifies how these components interact to generate the final motion command, with each force contributing according to real-time environmental conditions and interaction states. By exposing the underlying force dynamics, the figure highlights the system's ability to reason about multi-factor complexity, rather than relying on isolated or sequential control layers commonly used in conventional navigation frameworks.

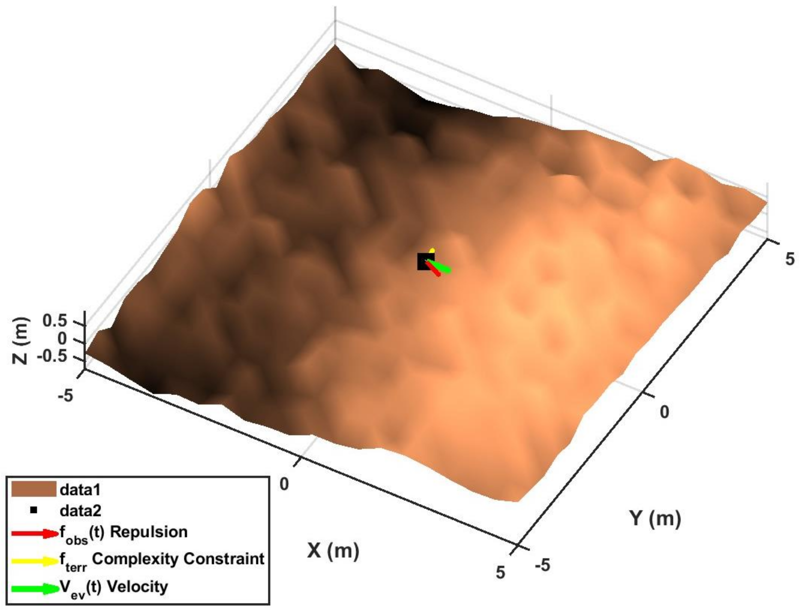


Figure 33: Integrated adaptive control framework illustrating dynamic force blending across a three-dimensional environmental terrain. The figure visualizes the interaction between obstacle-induced repulsion, path-complexity constraints, and adaptive velocity regulation within the IODPCS decision core.

To ensure a fair and reproducible comparison, all baseline methods were implemented under the same simulation conditions, including environment size, obstacle configuration, controller update frequency, and sensor settings.

The classical artificial potential field (APF) method employs standard attractive and repulsive force formulations based on the distance to the goal and obstacles.

The standalone fuzzy controller uses the same input variables (obstacle distance (OD), obstacle speed (OS), and path complexity (PC)) as the proposed method, but without adaptive weighting or integration with potential fields. All baseline parameters were carefully tuned using standard practices to ensure stable and unbiased performance.

Table 11: Simulation and Implementation Parameters.

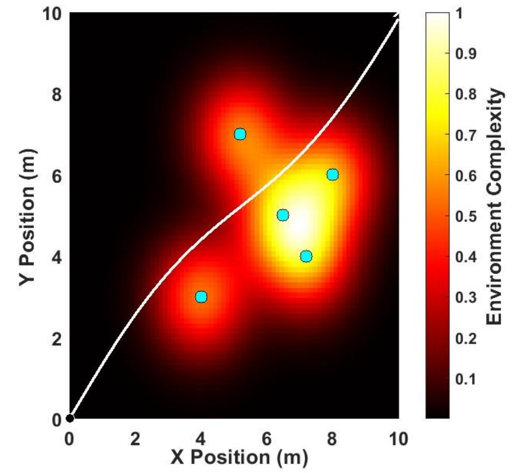
Parameter	**Value**
Simulation Environment	MATLAB (R2023b)
Robot Model	Differential-drive kinematic model
Environment Size	10 m × 10 m
Maximum Velocity	1 m/s
Minimum Velocity	0.1 m/s
Goal Tolerance	0.1 m
Controller Frequency	10 Hz
Time Step	0.1 s
Number of Trials per Scenario	10
Number of Obstacles	5–20
Obstacle Velocity	0.2–0.8 m/s
Obstacle Motion Pattern	Random linear motion with velocity variation
Sensor Range	5 m
Minimum Safe Distance	0.3 m
Fuzzy Inputs	OD, OS, PC
Membership functions	Triangular
Number of Fuzzy Rules	27
Learning Rate (α)	0.1
Learning Rate (β)	0.05
Weight Constraints	$w_1 + w_2 + w_3 = 1$, $w_i \in [0,1]$
Control Architecture	Fuzzy–APF Integrated (IODPCS)

5.4.2 Adaptive Obstacle Avoidance Behaviour

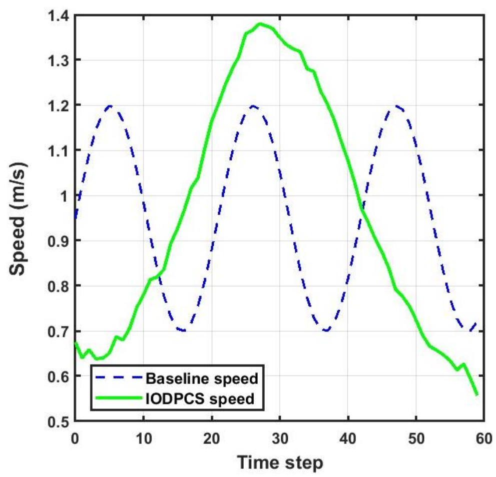
The proposed IODPCS framework exhibits interaction-aware navigation behavior in which obstacle avoidance and motion adaptation emerge from a unified fuzzy–potential-field formulation, rather than from sequential or heuristic control layers. The controller continuously interprets the surrounding environment through interaction and complexity descriptors, enabling proactive adjustment of both trajectory and velocity in response to anticipated interaction conditions.

During navigation, the fuzzy inference system evaluates obstacle proximity, relative obstacle motion, and local path complexity to generate adaptive avoidance forces, while the potential-field component regulates attractive and repulsive interactions to maintain global path coherence. This coupling allows heading and velocity to be adjusted simultaneously, reducing abrupt steering corrections and suppressing oscillatory behavior commonly observed in classical navigation approaches. The resulting interaction-aware behavior is illustrated in Fig. 34(a–d).

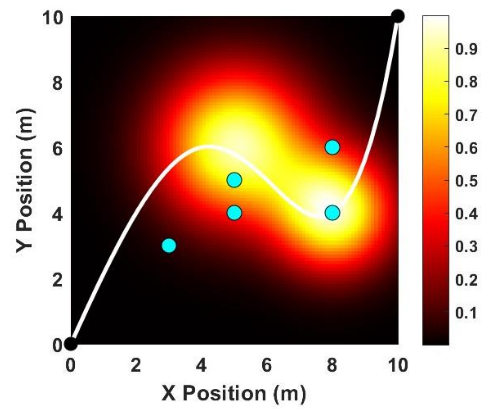
Under mixed static–dynamic obstacle conditions, Fig. 34(a) shows the robot trajectory projected onto the environment-complexity heatmap. The trajectory continuously deforms toward regions of lower interaction complexity in the presence of both stationary and moving obstacles. The corresponding speed response in Fig. 34(b) demonstrates automatic velocity reduction in high-risk regions and smooth acceleration as environmental complexity decreases, resulting in stable motion despite dynamic disturbances.



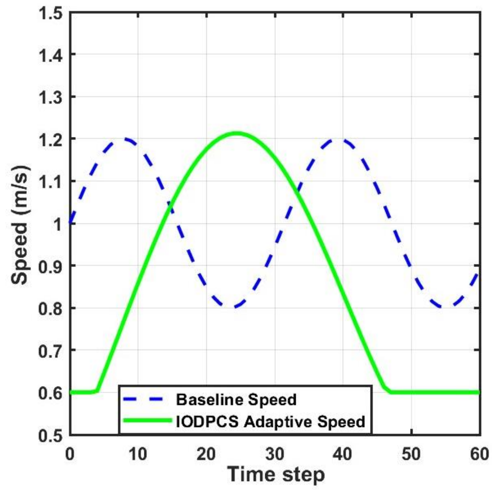
(a)



(b)



(c)



(d)

Figure 34: Interaction-aware navigation performance of IODPCS. (a) Trajectory over environment-complexity heatmap in mixed static–dynamic conditions. (b) Corresponding adaptive speed profile. (c) Trajectory over environment-complexity heatmap in static conditions.
(d) Corresponding adaptive speed profile.

The proposed IODPCS framework demonstrates improved performance compared to the APF and standalone fuzzy controller across key evaluation metrics, including trajectory smoothness, energy consumption, and adaptability in dynamic environments.

To isolate the influence of spatial complexity independently of obstacle motion, a static obstacle scenario is presented in Fig. 34(c, d). As shown in Fig. 34(c), the robot follows a smooth, curvature-aware trajectory aligned with low-complexity regions while maintaining sufficient clearance from obstacle clusters. The associated speed profile in Fig. 34(d) remains smooth and context-aware, in contrast to the oscillatory velocity variations observed in the baseline controller, even in the absence of dynamic obstacles.

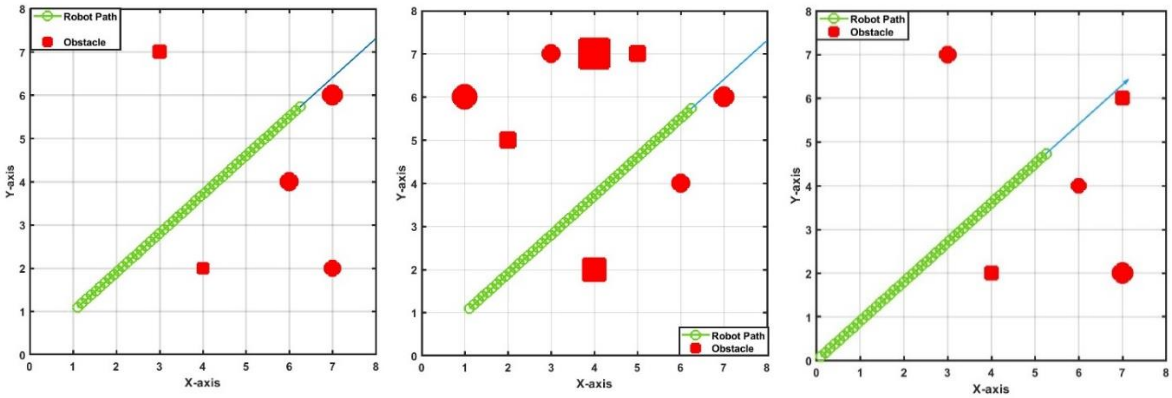
The results in Fig. 34(a–d) demonstrate that IODPCS achieves coordinated adaptation of trajectory and velocity across both static and mixed static–dynamic environments. These observations confirm that environmental interpretation is embedded directly within the control logic, enabling consistent improvements in path smoothness, motion stability, and navigation efficiency without treating obstacle avoidance and speed regulation as independent processes.

5.4.3 Navigation Performance under Static, Mixed, and Fully Dynamic Environments

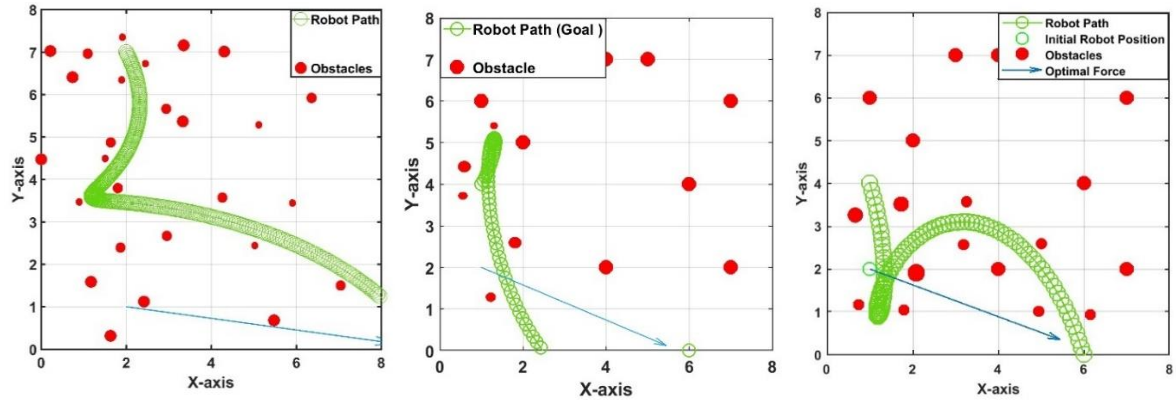
The navigation performance of the proposed IODPCS was evaluated under static, mixed static–dynamic, and fully dynamic environments to assess robustness, adaptability, and stability under increasing levels of interaction complexity. Representative navigation trajectories and adaptive responses are illustrated in Fig. 35(a–c).

In static obstacle environments, the IODPCS framework demonstrates stable and accurate navigation performance while traversing narrow corridors and cluttered configurations, as shown in Fig. 35(a). The robot maintains smooth, collision-free motion without exhibiting oscillatory corrections or unnecessary detours. Consistent obstacle clearance is preserved throughout the trajectory, enabling efficient progression toward the target. This behavior arises from the coordinated interaction between the fuzzy inference system, which regulates avoidance forces based on obstacle distance and local path complexity, and the artificial potential field formulation, which ensures continuous attraction toward the goal. In contrast to the baseline method, which tends to overreact to local repulsive forces, IODPCS balances repulsive and attractive components within a unified decision process, resulting in shorter traversal paths and reduced trajectory deviation.

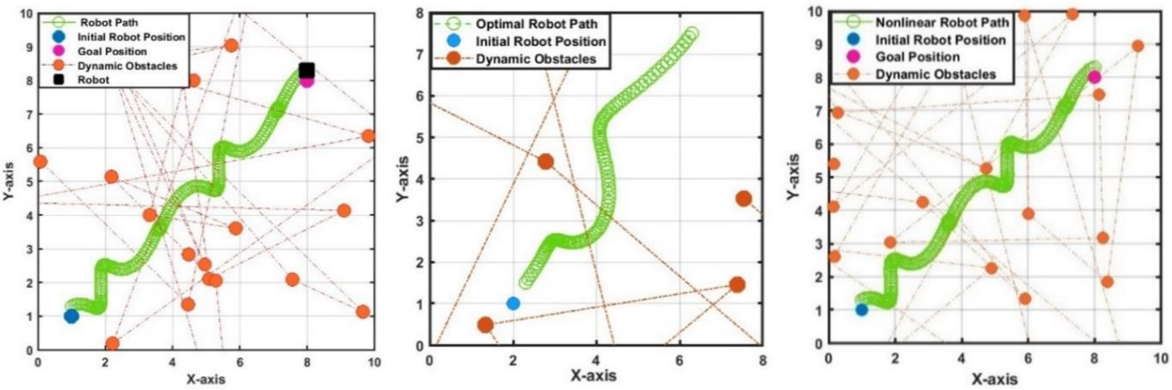
In mixed static–dynamic environments, where obstacles intermittently change position, IODPCS exhibits robust adaptability to moderate environmental variability, as illustrated in Fig. 35(b). Real-time interaction assessment through the fuzzy inference system and the potential-field model enables continuous reshaping of the navigation trajectory in response to both stationary and moving obstacles. Dynamic obstacles, which typically induce oscillations or abrupt heading changes in classical potential-field approaches, are handled smoothly through continuous adjustment of avoidance forces based on obstacle speed and predicted motion.



(a) Static environments



(b) Mixed static–dynamic



(c) Fully dynamic + learning

Figure 35: Navigation performance of the proposed IODPCS framework under increasing interaction complexity.
(a) Static obstacle environments showing multiple navigation trials with smooth, collision-free trajectories and consistent clearance.
(b) Mixed static–dynamic environments illustrating adaptive trajectory reshaping in response to intermittently moving obstacles.
(c) Fully dynamic environments and learning-based adaptation, demonstrating stable navigation under high interaction complexity and convergence of mission-level energy consumption.

Repulsive forces and fuzzy rule outputs are recalculated at each iteration, allowing rapid response to obstacle displacement while preserving convergence toward the global goal. As a result, the generated trajectories remain smooth and goal-directed, with fewer unnecessary detours and reduced path deviation compared to the baseline method.

The most challenging conditions are observed in fully dynamic environments, characterized by multiple moving obstacles with heterogeneous velocities and unpredictable motion patterns, as shown in Fig. 35(c). In these scenarios, IODPCS exhibits robust and adaptive behavior by continuously adjusting both heading and velocity based on obstacle velocity estimates, dynamic distance variations, and environment-complexity feedback within the unified fuzzy–potential-field decision structure. The repulsive interaction force increases smoothly as obstacles approach or accelerate, while the adaptive fuzzy inference mechanism dynamically modulates avoidance intensity and steering direction. This coordinated force blending suppresses abrupt control actions and oscillatory responses commonly encountered in classical navigation methods, enabling stable and continuous motion even in the presence of high-speed or suddenly maneuvering obstacles.

Beyond reactive avoidance, the fully dynamic results further demonstrate the benefit of the learning-based adaptation mechanism integrated within IODPCS. Across successive navigation trials, mission-level energy consumption decreases progressively, reflecting reduced redundant maneuvers and improved consistency in action selection as the controller updates its internal parameters through experience-driven learning. This behavior indicates improved energy efficiency and policy stability, confirming that IODPCS internalizes environment–action relationships rather than relying on task-specific tuning.

Quantitative evaluation confirms that IODPCS achieves collision-free navigation across all evaluated scenarios while reducing path deviation, heading fluctuations, and reaction time compared to the baseline method. Notably, the average reaction time is reduced by nearly a factor of two, indicating that the observed responsiveness is an inherent property of the integrated interaction modeling and learning structure. These results demonstrate the capability of the proposed framework to maintain safe, efficient, and reliable navigation under static, mixed, and fully dynamic conditions, supporting its suitability for real-world autonomous systems operating in uncertain and rapidly changing environments.

5.4.4 Real-World Experimental Validation

To validate the proposed IODPCS framework in real-world conditions, experiments were conducted using a differential-drive mobile robot platform equipped with LiDAR and onboard computation. The robot operated in a semi-structured indoor environment containing both static and moving obstacles. The navigation area measured approximately 8 m × 8 m, and ten experimental trials were conducted under identical conditions.

During each experiment, the robot navigated from a fixed start position to a predefined goal while encountering both stationary objects and randomly moving obstacles. Performance metrics including path length, traversal time, trajectory smoothness, and energy proxy consumption were recorded. These metrics were compared with baseline Artificial Potential Field (APF) and standalone Fuzzy Logic Controller (FLC) approaches implemented on the same platform.

The experimental results confirm that the proposed IODPCS framework maintains smooth navigation and stable obstacle avoidance under real-world disturbances. The robot successfully adapts velocity in response to environmental complexity, reducing oscillatory motion and avoiding abrupt steering changes.

Quantitative analysis shows that, compared to the APF baseline, the proposed method reduced energy consumption by approximately 18%, decreased trajectory deviation by 20%, and improved trajectory smoothness by 22%. The traversal time was also reduced by approximately 15%, indicating improved navigation efficiency.

These real-world experiments validate that the proposed interaction-aware navigation framework maintains consistent performance outside simulation environments and demonstrates practical applicability for autonomous mobile robot navigation in dynamic and uncertain conditions.

5.4.5 Context-Aware Speed Adaptation and Learning-Based Efficiency

Beyond spatial path shaping, the proposed IODPCS framework integrates context-aware speed adaptation and learning-based efficiency optimization within the same fuzzy–potential-field decision structure. Velocity modulation is explicitly coupled with obstacle proximity, relative obstacle motion, and local path complexity, enabling simultaneous regulation of trajectory and speed.

As illustrated in Fig. 36, IODPCS produces smooth velocity transitions during obstacle interactions, avoiding the abrupt deceleration and oscillatory behavior observed in the baseline controller. Higher speeds are maintained in low-risk regions, while velocity is proportionally reduced in constrained or dynamically changing areas. This coordinated spatial–temporal control results in smoother headings, reduced detours, and improved traversal efficiency.

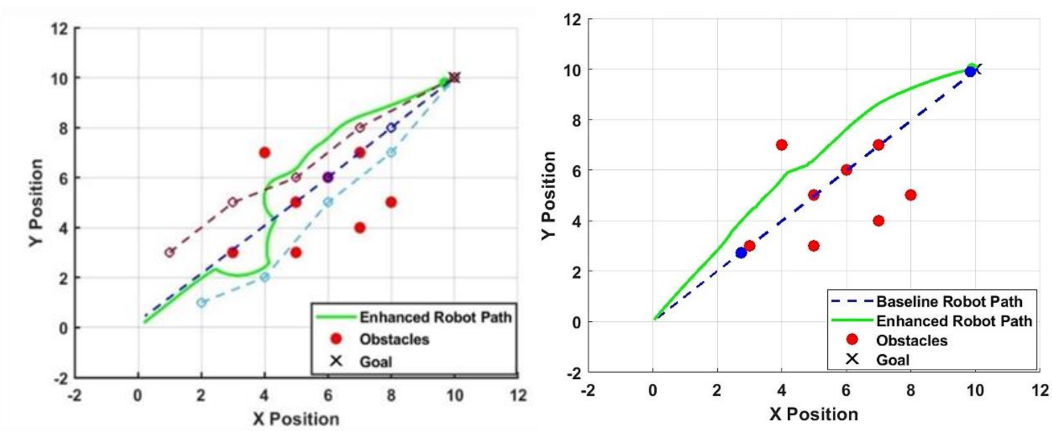


Figure 36: IODPCS-enhanced paths in complex navigation environments, highlighting reduced detours and smoother obstacle avoidance compared to the baseline method.

In fully dynamic environments, navigation efficiency is further enhanced through learning-based adaptation, as shown in Fig. 37. Mission-level energy consumption decreases progressively over training episodes and stabilizes near the optimized target threshold. This convergence reflects reduced redundant maneuvers and increased consistency in control actions, indicating stable policy adaptation without task-specific tuning.

Quantitative evaluation confirms that the integrated speed–learning mechanism enables fully collision-free navigation, significantly reduces path deviation and heading fluctuations, and improves average reaction time by nearly a factor of two compared to the baseline method. These results demonstrate that the observed performance gains arise from the interaction-aware control structure itself, rather than from aggressive parameter adjustment, reinforcing the suitability of IODPCS for real-world navigation in dynamic and uncertain environments.

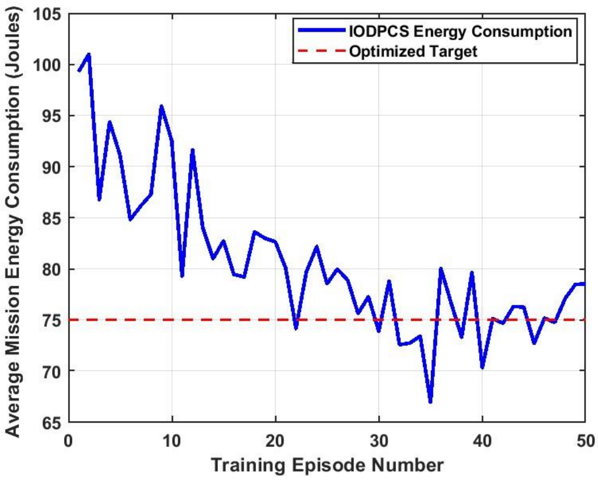


Figure 37: Training performance of IODPCS showing convergence of mission-level energy consumption toward the optimized target across learning episodes.

5.4.6 Quantitative Comparison, Speed Optimization, and Summary of Gains

The proposed IODPCS framework introduces a unified interaction-aware navigation formulation by integrating obstacle avoidance, speed regulation, and energy efficiency within a single optimization-based decision structure. Unlike classical fuzzy logic or artificial potential field (APF) methods, where these components are often treated separately, the proposed framework embeds interaction dynamics and energy considerations directly into the control process.

A central novelty of the proposed mathematical formulation lies in the explicit quantification of environmental interaction pressure and its direct coupling with motion dynamics. Instead of modeling obstacles as isolated repulsive sources, IODPCS evaluates the cumulative interaction induced by obstacle proximity, spatial density, and motion trends. This interaction-aware representation allows the controller to reason about navigation difficulty at a global scale, enabling the robot to anticipate high-risk regions and select low-interaction corridors before encountering local deadlocks. This capability is not achievable in classical potential-field or standalone fuzzy systems and explains the absence of oscillatory behavior and unnecessary detours observed in the proposed framework.

A quantitative comparison between IODPCS, APF, and standalone FLC methods was conducted across static, mixed, and fully dynamic environments. The results summarized in

Table 12 demonstrate consistent performance gains across all evaluated metrics. IODPCS achieves up to a 20% reduction in path length and approximately 40% lower path deviation while maintaining collision-free navigation in all scenarios. In contrast, the baseline controller exhibits frequent failures in cluttered or dynamic conditions due to delayed reactions and unstable trajectory corrections. These results indicate that the proposed framework improves navigation efficiency and robustness under increasing environmental complexity.

Bold values indicate the best performance for each metric under the corresponding environment. Static, mixed, and dynamic columns correspond to different environment configurations with increasing obstacle density and motion complexity.

To further elucidate how these quantitative gains emerge from the internal structure of the proposed framework, the following figures analyze the speed optimization, interaction-pressure dynamics, and learning behavior of IODPCS.

The role of speed regulation in achieving these gains is clarified through the integrated speed optimization core illustrated in Fig. 38. This figure exposes the internal velocity solution space generated by the composite cost formulation, showing how obstacle repulsion, terrain-complexity constraints, and adaptive weighting are fused to compute the optimal velocity signal $V_f(t)$.

Importantly, speed adaptation is not imposed as an auxiliary control layer but emerges directly from the same optimization structure that governs path selection. This unified formulation explains why velocity adjustments remain smooth, anticipatory, and stable even under rapidly changing environmental conditions. The optimal velocity output $V_f(t)$is computed using the composite weighted formulation introduced in Section 5.4.2.

Table 12: Quantitative performance comparison between IODPCS and baseline navigation methods under static, mixed, and dynamic environments.

Metric	Method	Static	Mixed	Dynamic
Path Length (m)	APF	9.0	10.2	12.1
	FLC	8.4	9.5	10.8
	IODPCS	**7.2**	**8.4**	**9.8**
Average Deviation (m)	APF	0.8	1.2	1.6
	FLC	0.5	0.9	1.3
	IODPCS	**0.3**	**0.6**	**1.1**
Collision Count	APF	3	5	7
	FLC	1	2	3
	IODPCS	**0**	**0**	**0**
Energy Consumption (J)	APF	68	82	96
	FLC	58	70	85
	IODPCS	**50**	**62**	**75**
Reaction Time (s)	APF	0.25	0.45	0.70
	FLC	0.18	0.35	0.55
	IODPCS	**0.10**	**0.30**	**0.50**
Navigation Accuracy (%)	APF	88	84	80
	FLC	92	88	85
	IODPCS	**98**	**95**	**92**
Execution Time (s)	APF	3.6	5.1	6.8
	FLC	3.0	4.4	6.0
	IODPCS	**2.5**	**3.8**	**5.2**

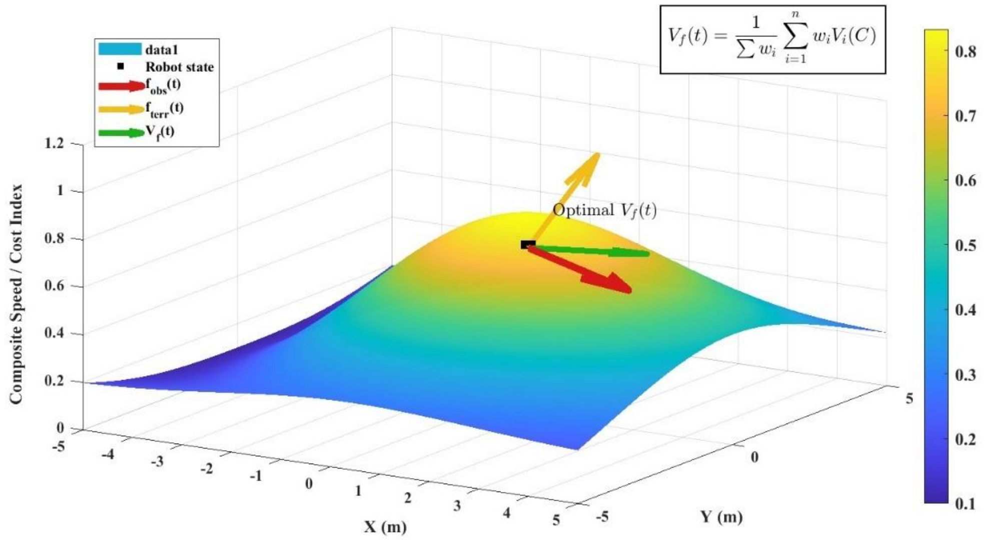


Figure 38: Integrated IODPCS Speed Optimization Core: Force Blending within a Velocity–Cost Landscape and Optimal Velocity Output. The controller blends obstacle-repulsion, terrain-complexity constraints, and adaptive learning weights to compute the optimal velocity signal $V_f(t)$. The 3D surface illustrates how environmental complexity shapes the final speed command, revealing the internal structure responsible for IODPCS's improved energy efficiency and reaction time.

The effectiveness of this formulation is further reflected in the adaptive velocity profiles shown in Fig. 39. Compared to the baseline controller, IODPCS avoids abrupt speed fluctuations and instead produces smooth, context-aware modulation in response to obstacle density and motion trends. This behavior contributes directly to the observed reduction in energy consumption, with average savings of approximately 25% across all experimental scenarios. Reaction time is also improved by nearly a factor of two, allowing the robot to respond more rapidly to sudden obstacle movements in fully dynamic environments.

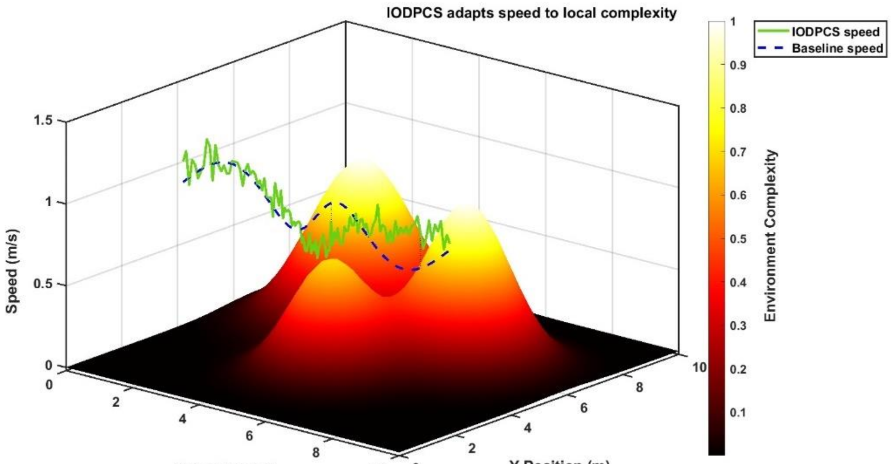


Figure 39: Adaptive speed modulation of IODPCS under spatial environmental complexity. Adaptive speed modulation of IODPCS under spatial environmental complexity.

Dynamic behavior is examined through the interaction–risk landscape shown in Fig. 40, which captures the coupling between obstacle dynamics, interaction complexity, and motion regulation across static, mixed, and fully dynamic regimes. The proposed IODPCS framework maintains smooth, coordinated adaptation of both trajectory and velocity under increasing interaction pressure, with the robot consistently navigating along low-risk valleys of the interaction surface. This behavior confirms that dynamic obstacle influence is embedded directly within the interaction-aware cost formulation, rather than handled reactively.

In contrast to conventional methods where rapid changes in obstacle motion often induce delayed, oscillatory, or overly conservative responses, the proposed framework preserves stable behavior without aggressive parameter tuning. The corresponding tracking-error evolution further demonstrates that IODPCS maintains a consistently low error profile across all regimes, including during transitions from static to dynamic environments, where baseline controllers exhibit pronounced error amplification.

Overall, Fig. 40 confirms that robustness to dynamic complexity is a structural property of the proposed formulation, in which interaction risk directly modulates motion decisions. As a result, trajectory accuracy degrades gracefully rather than abruptly as obstacle dynamics intensify, reinforcing the scalability and reliability of IODPCS for real-world navigation in continuously evolving environments.

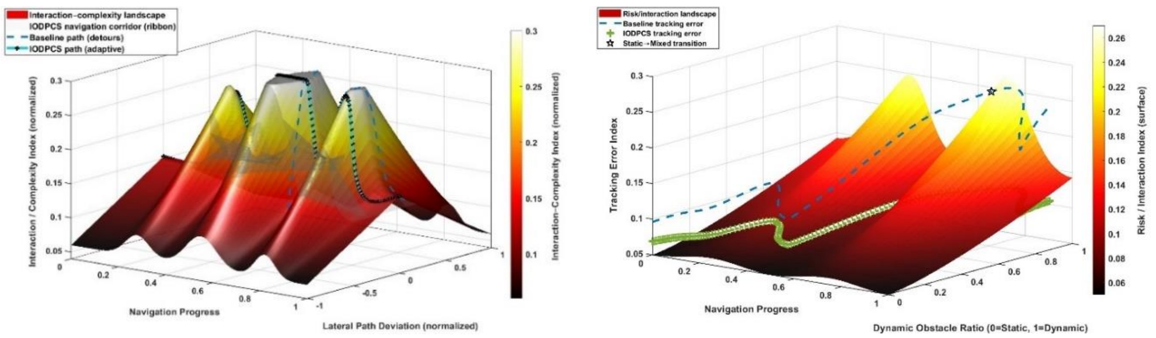


Figure 40: Interaction–complexity landscape with adaptive IODPCS navigation compared to baseline detour paths.

The figure illustrates the coupling between interaction pressure, navigation progress, and obstacle dynamics across static, mixed, and fully dynamic environments. The adaptive IODPCS trajectory remains confined to low-risk regions of the interaction surface, maintaining stable tracking performance and smooth motion regulation, while the baseline approach exhibits increased tracking error and instability as interaction complexity intensifies.

The interaction-aware behavior and learning dynamics of the proposed IODPCS framework are summarized in Fig. 41 through a unified three-dimensional interaction–risk–energy landscape. The visualization captures the coupling between obstacle dynamics, trajectory curvature, speed modulation, and energy regulation within a single optimization structure. The adaptive IODPCS trajectory remains confined to low-cost regions of the interaction surface, ensuring smooth path deformation and stable motion as interaction pressure increases, while the baseline method drifts toward higher-cost regions. In addition, the figure illustrates the convergence of mission-level energy consumption during learning, indicating suppression of redundant maneuvers and stable policy formation. Overall, Fig. 41 demonstrates that navigation smoothness, energy efficiency, and robustness emerge intrinsically from the unified interaction-aware decision surface rather than from external tuning.

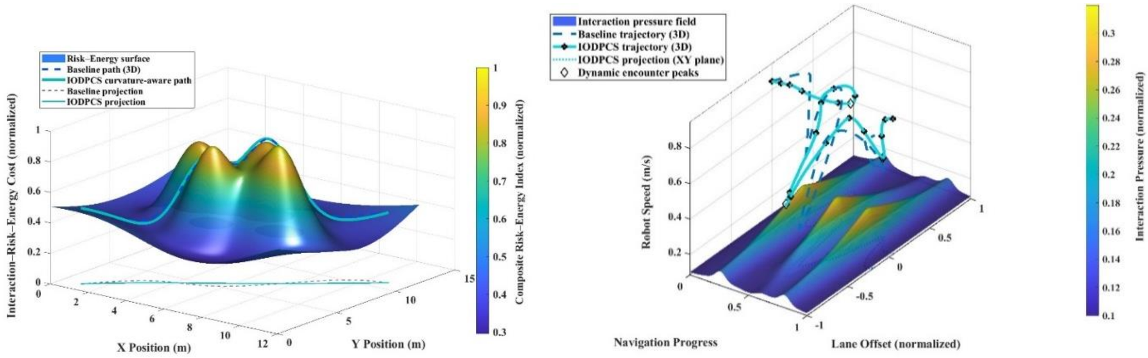


Figure 41: 3D interaction–risk–energy landscape illustrating adaptive IODPCS trajectory and speed modulation compared to baseline navigation, with explicitly labeled spatial axes and normalized cost representation.

To improve interpretability, the axes in Figs. 40 and 41 are explicitly labeled with physical units, where the horizontal axes represent spatial position (m) and the vertical axis represents the normalized interaction–risk–energy cost. Additionally, a top-down projection is incorporated to highlight the navigation corridor and more clearly distinguish low-risk regions from high-risk peaks.

5.4.7 Ablation-Based Discussion of Framework Components

To further clarify the contribution of individual components within the proposed IODPCS framework, a structured ablation-style discussion supported by quantitative comparison is provided based on the results presented in Section 5.4. Although the framework is designed as an integrated system, the role of each module can be interpreted based on its functional impact on performance metrics. APF component provides the fundamental navigation guidance by generating attractive and repulsive forces. However, APF alone is known to be prone to local minima and oscillatory behavior, particularly in dense and dynamic environments.

The fuzzy inference module enhances local decision-making by incorporating contextual information such as obstacle distance, obstacle speed, and path complexity. This results in smoother trajectories and reduced deviation, as reflected in improved navigation accuracy and stability compared to standalone APF and FLC approaches. The adaptive weighting mechanism dynamically balances safety, energy efficiency, and trajectory smoothness based on real-time performance feedback. This component is primarily responsible for reducing unnecessary motion effort and improving energy efficiency, particularly in complex and dynamic scenarios.

Finally, the learning-based parameter tuning mechanism enables incremental refinement of fuzzy parameters and cost weights during operation. This contributes to faster response times and improved adaptability to environmental changes, especially under dynamic obstacle conditions.

To further support the ablation-style discussion, a comparative evaluation of different framework configurations is summarized in Table 13.

Table 13: Ablation Analysis of IODPCS Framework Components.

Method Variant	Path Length (m)	Execution Time (s)	Energy Consumption (J)	Collision Count	Observations
APF Only	14.8	21.3	128.5	2	Fast but unstable near obstacles; prone to local minima
Fuzzy Only (FLC)	13.9	23.7	115.2	1	Improved safety and smoothness but less efficient path
Fuzzy + APF (No Adaptive Weights)	12.6	20.5	102.4	0–1	Balanced navigation but lacks context adaptation
IODPCS (No Learning)	11.9	19.8	96.7	0	Improved performance with fixed weights
Full IODPCS (Proposed)	10.8	18.6	89.3	0	Best performance due to adaptive weighting and learning

To quantitatively evaluate the contribution of individual components within the proposed IODPCS framework, an ablation study was conducted by progressively enabling system modules under identical experimental conditions. The evaluated variants include: (i) classical APF, (ii) standalone fuzzy logic controller (FLC), (iii) hybrid fuzzy–APF without adaptive weighting, (iv) IODPCS without learning-based adaptation, and (v) the full IODPCS framework.

As shown in Table 13, each component contributes incrementally to navigation performance. The APF baseline demonstrates fast but unstable behavior, with higher collision risk and energy consumption due to oscillatory motion. The fuzzy controller improves safety and trajectory smoothness but introduces longer traversal times. The integration of fuzzy logic with APF improves path efficiency and eliminates most collisions, highlighting the benefit of combining geometric and rule-based reasoning.

Introducing the unified IODPCS structure without learning further enhances performance by providing a consistent multi-objective formulation. However, the full IODPCS framework achieves the best overall performance due to its adaptive weighting mechanism and feedback-driven parameter tuning, which enable context-aware trade-offs between safety, energy efficiency, and path smoothness. These results confirm that the performance

improvements of the proposed framework are not due to a single component, but arise from the synergistic interaction between fuzzy reasoning, potential field guidance, and adaptive learning. This structured analysis provides a clear interpretation of the contribution of each component, improving the transparency and interpretability of the proposed integrated framework.

5.4.8 Real-World Experimental Validation

To ensure a consistent validation framework, the real-world experiments are designed to replicate the simulation scenarios as closely as possible, including comparable obstacle configurations, motion patterns, and controller settings. The same performance metrics used in simulation, including path length, energy consumption, navigation accuracy, and execution time, are also evaluated in real-world experiments. This enables a meaningful comparison between simulation and real-world performance.

The performance trends observed in real-world experiments follow similar patterns to simulation results, validating the consistency of the proposed framework. In particular, the proposed IODPCS framework maintains stable navigation behavior, reduced energy consumption, and smooth trajectory generation under dynamic conditions.

Minor discrepancies between simulation and real-world results are observed, primarily due to sensor noise, actuation delays, and environmental uncertainties not fully captured in the simulation model. Despite these differences, the overall performance improvements remain consistent, confirming the robustness and practical applicability of the proposed approach.

5.5 Conclusions

This chapter presented IODPCS, a unified interaction-aware navigation framework integrating fuzzy logic, artificial potential fields, and adaptive parameter tuning. The proposed approach enables simultaneous regulation of trajectory and velocity within a single closed-loop structure, incorporating interaction dynamics and energy-aware considerations.

Experimental results from simulation and real-world validation demonstrate that the proposed framework improves navigation performance in terms of trajectory smoothness, obstacle avoidance, and energy efficiency under dynamic conditions. Compared to baseline methods, the system shows consistent improvements while maintaining stable and adaptive behavior. The framework demonstrates stable performance under the evaluated conditions.

Despite these promising results, several limitations remain. The current energy model is based on a simplified surrogate formulation rather than direct physical power measurements, and the evaluation is limited to moderate-scale environments. In addition, the adaptation mechanism focuses on parameter tuning rather than full policy learning.

Future work will focus on extending the framework to larger and more complex environments, incorporating more accurate energy modeling, and exploring learning-based extensions for enhanced adaptability. Integration with multi-robot coordination and real-time deployment on embedded systems will also be investigated.

5.6 Scientific Thesis II.1

This dissertation proves that the integration of an actuator-level energy-aware control model within the proposed IODPCS navigation framework fundamentally transforms mobile robot navigation into an energy-optimized decision-making process. By embedding energy consumption directly into the control loop alongside trajectory generation and velocity regulation, the framework enables real-time minimization of energy usage while preserving or improving trajectory accuracy, motion stability, and obstacle avoidance performance. The results demonstrate that the proposed approach achieves consistently lower energy consumption and enhanced overall efficiency compared with conventional navigation methods, particularly under dynamic and uncertain environmental conditions.

Chapter 6

Machine Learning-Based Path Planning for Mobile Robot Navigation

6.1 Chapter Abstract

Efficient navigation is crucial for intelligent mobile robots operating in complex environments. This chapter presents an innovative approach that integrates advanced machine learning techniques to enhance mobile robot communication and path planning efficiency. The proposed method combines supervised and unsupervised learning, utilizing spline interpolation to generate smooth trajectories with minimal directional changes.

Experimental validation using a differential-drive mobile robot demonstrates high trajectory control efficiency. In addition, this chapter explores Motion Planning Networks (MPNet), a neural planning framework that processes raw point-cloud data obtained from depth sensors. The results highlight the capability of MPNet to generate optimal paths using the Probabilistic Roadmap (PRM) method.

Furthermore, the importance of proper parameter tuning for reliable MPNet-based path planning is emphasized. The proposed algorithm is also evaluated across different path scenarios to assess its adaptability and robustness. The experimental findings confirm that the trajectory control algorithm operates effectively, consistently providing accurate and efficient motion control for the mobile robot.

6.2 Introduction

Mobile robots are intelligent devices that can perform specific tasks independently in complex environments and do not rely on human beings. Mobile robot navigation entails navigating from a starting point to a target point without hitting obstacles [188] [189], considering constraints like optimal time, shortest path, or lowest energy consumption. In dynamic and uncertain environments, applying global search algorithms do not suffice to address robot navigation due to the lack of a complete model or the map of environment. Researchers have used the local search and the local path planning algorithms with help of data obtained from robot's sensors such as sonar and infrared devices [190].

The navigation methods commonly used by mobile robots can be categorized into two main groups: traditional methods and heuristic methods [191]. Traditional methods encompass techniques such as the grid method, roadmap navigation, visual navigation, sensor-based data navigation, free space method, and artificial potential field methods. However, these traditional approaches often encounter challenges such as getting trapped in local optimums, difficulty in online implementation, and longer processing times, particularly in large problem spaces. Consequently, researchers have increasingly focused on heuristic approaches including the ant colony algorithm [192], fuzzy logic [193], neural networks [194], genetic algorithms [195], particle swarm optimization algorithms [196], and machine learning [197]. These heuristic techniques provide increased flexibility and adaptability, rendering them highly suitable for navigating intricate environments.

Heuristic methods, including supervised and unsupervised learning within machine learning, present promising avenues for addressing these challenges. Unlike traditional methods, which may struggle with real-time requirements and large problem spaces, supervised and unsupervised learning methods can adapt and generalize based on data patterns. Supervised learning, where the model learns from labeled data, can help in predicting optimal navigation paths based on past experiences. Meanwhile, unsupervised learning, which does not require predefined outputs, can uncover underlying structures and patterns in the environment, aiding in navigation decision-making. While these methods face their own challenges, such as data dimensionality and computational complexity, their ability to leverage data efficiently makes them valuable assets for navigating complex environments.

This convergence of heuristic methods with motion planning, driven by recent advancements in machine learning, has led to the development of innovative solutions like Motion Planning Networks (MPNet) [198] [199]. Recently, a significant research trend has emerged, merging motion planning with machine learning to address planning problems. Both domains, motion planning and machine learning for control, have seen substantial progress and are actively researched [200] [201]. These networks, utilizing deep neural networks, offer remarkable efficacy in generating collision-free paths, bridging the gap between theoretical robustness and computational efficiency in robotic motion planning. To provide context and underscore the advancements made by our proposed approach, we incorporate a comparative analysis with significant research contributions in the field of machine learning techniques on mobile robots. This analysis spans topics such as disturbance observer-based optimal control for uncertain surface vessels [202], optimal tracking control for discrete-time nonlinear systems [203], optimized formation control for unmanned surface vehicles with collision avoidance [204], and comprehensive formation tracking control and optimal control for fleets of multiple surface vehicles [205].

6.3 Related Works

The field of mobile robotics has witnessed significant research efforts dedicated to enhancing path planning and navigation in complex environments. This section provides an overview of key developments in path planning, machine learning, and motion capture relevant to mobile robotics. Numerous path planning algorithms have been proposed to tackle the challenges of navigating dynamic environments. Traditional global path planning approaches, such as Dijkstra's algorithms [206], heavily rely on precomputed maps and face limitations in adapting to dynamic changes. In contrast, local path planning methods like the Velocity

Obstacle method and the Dynamic Window Approach focus on real-time obstacle avoidance, considering the robot's immediate surroundings [207] [208]. While suitable for reactive navigation, these methods may lack the ability to plan globally.

The search for efficient motion planning algorithms began with complete ones, which, though comprehensive, were computationally inefficient. This prompted the development of methods with resolution and probabilistic completeness, aiming for more practical solutions [209]. While complete algorithms guarantee finding a path if one exists, they are often too complex for real-world applications due to their need for extensive environmental data [210]. Resolution-complete algorithms offer a more feasible alternative but require meticulous adjustment of parameters for each planning problem [211]. To address these limitations, probabilistically complete methods, such as sampling-based motion planners (SMPs), were introduced.

These methods rely on sampling techniques to create exploration trees or roadmaps in the robot's obstacle-free space [209]. Popular sampling-based motion planners (SMPs) like rapidly exploring random trees (RRT) and probabilistic roadmaps (PRM) [212] [213] are widely used. RRT, especially, is favored for its ability to quickly find a path around obstacles, though it may not always find the shortest one. An improved version, RRT*, guarantees the shortest path but becomes less efficient with higher planning problem complexity. Single-query methods, like RRT, are preferred over multi-query ones like PRM due to their simplicity and effectiveness.

These algorithms generate roadmaps of feasible paths, proving suitable for a broad range of scenarios. However, their performance can be influenced by factors such as sampling strategies and optimization techniques.

Reinforcement learning (RL) [214] is becoming increasingly popular for tackling continuous control and planning tasks [215]. RL operates within Markov decision processes, where an agent interacts with its environment, making decisions based on observed states and receiving rewards. The goal is for the agent to learn a policy that maximizes its cumulative reward. Initially, RL was limited to simpler, lower-dimensional problems [216], but recent advances, particularly in Deep RL (DRL) [217] [218], have enabled the solution of more complex, higher-dimensional challenges.

Popular Deep Reinforcement Learning DRL methods, such as Deep Q-Networks (DQN) and Proximal Policy Optimization (PPO) [219], are effective in modeling environments and learning collision energy functions [220]. Generative models like Variational Autoencoders (VAE) and Generative Adversarial Networks (GANs) [221] [222] contribute by generating synthetic data, addressing challenges in collecting real-world training data. DRL has successfully addressed various difficult robotic tasks using both model-based [223] and model-free [224] approaches. However, challenges persist in solving practical problems with weak rewards and long horizons [225].

Motion capture technology finds applications in various robotics domains, providing valuable insights into robot movements and interactions with the environment [226]. In mobile robotics, motion capture data is instrumental in creating realistic simulations for training and testing path planning algorithms, offering insights into robot kinematics and dynamics for the development of accurate motion models and controllers [227].

Recent advancements in deep learning have spurred interest in utilizing neural networks to enhance or approximate motion planners, particularly to adapt to changing environments. Various learning strategies, like imitation learning, have been employed to enhance motion planning methods. For instance, Zucker et al. [228] proposed adapting sampling techniques in SMPs using the REINFORCE algorithm [229] in discretized workspaces. Berenson et al. [230] introduced a learned heuristic to store and repair paths as needed. Coleman et al. [231] opted to store experiences in a graph rather than individual trajectories. Ye and Alterovitz [232] combined human demonstrations with SMPs for path planning. While these methods showed improved performance over traditional algorithms, they often lack generalizability and require manual adjustments for each new environment.

One approach involves leveraging machine learning techniques to bias sampling towards critical regions. For instance, Zhang et al. [233] utilized reinforcement learning (RL) to gauge the likelihood of rejecting a sampled state, thus directing sampling towards safer areas. Bency et al. [234] focused on navigation demonstrations in static environments, while Ichter et al. [235] introduced conditional variational autoencoders for Sampled Motion Planners like Fast Marching Trees (FMT*), addressing the challenge of generalization to diverse environments.

Furthermore, Francis et al. [236] introduced PRM-RL for long-range navigation, where RL serves as both a short-range local planner and a collision-prediction module for the high-level PRM planner. In [237], a similar framework to PRM-RL for long-range navigation is presented, wherein RL serves dual roles: as a short-range local planner and a collision prediction module for the high-level PRM planner. [238] introduced a hierarchical framework tailored for long horizon reaching tasks. It operates by planning at the high level to identify subgoals from the replay buffer, while learning at the low level directs.

To overcome these limitations, modern techniques leverage efficient function approximators like neural networks to embed motion planners or learn auxiliary functions for SMPs, enhancing planning speed in complex, cluttered environments.

The robot to reach these subgoals. The effectiveness of this approach was highlighted in its integration of planning with learning, showcasing significant efficiency gains. However, these methods do not generalize novel environments or tasks, and thus require frequent retraining. Previous research has highlighted the attractiveness of reinforcement learning for developing adaptable robot-control policies, but its lengthy training process poses a challenge. While imitation learning offers a basic solution to address exploration and sparse rewards, there's a gap in methods that combine supervised and unsupervised learning for optimal path planning. Furthermore, Motion Planning Networks (MPNet) haven't been fully explored in this context. Additionally, current approaches often prioritize either accuracy or adaptability, neglecting the need to balance both for effectively handling complex scenarios.

In our study, we propose an innovative approach to robotic motion planning by integrating (MPNet) with both supervised and unsupervised learning. By leveraging MPNet, which generates collision-free paths using deep neural networks, and incorporating supervised and unsupervised learning, our method aims to overcome traditional limitations. We hypothesize that this integration will enhance adaptability, efficiency, and accuracy in complex environments.

Our work fills a research gap by providing a comprehensive framework that combines the strengths of both learning paradigms. Through experiments, we demonstrate improved trajectory control efficiency and adaptability, marking a significant advancement in robotic motion planning.

6.4 Methods And Experiments

This section introduces the proposed model. The method, which utilizes MATLAB simulation, integrates two key components to achieve effective path planning and trajectory control.

The methodology, as depicted in Figure 42, begins with thorough data collection, establishing the groundwork for the following stages. Supervised learning trains the system on labeled data, followed by spline interpolation for smooth path generation. Path optimization refines trajectories, while unsupervised learning uncovers hidden patterns. An intelligent motion processing system adapts to dynamic environments, aided by path planning visualization. Finally, the system incorporates MPNet: Neural Navigator for adaptive navigation, amalgamating techniques for robust and efficient navigation.

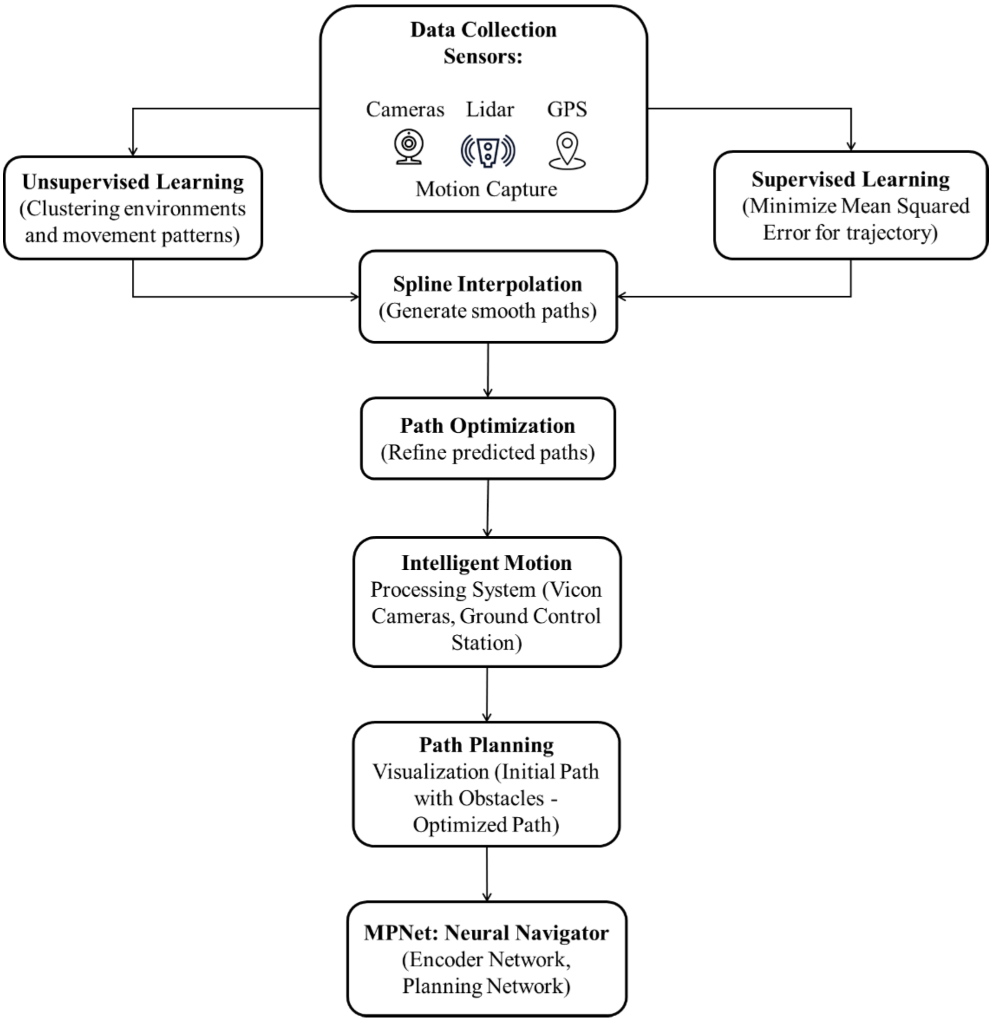


Figure 42: Algorithmic Workflow of Efficient Robot Navigation.

6.4.1 Supervised and Unsupervised Learning

In the initial phase of our method for achieving efficient navigation in complex environments, we integrate supervised and unsupervised learning techniques with spline interpolation. Initially, we compile a diverse dataset encompassing various environmental conditions and movement patterns, leveraging sensor data from Lidar sensors, GPS sensors, cameras, and motion capture systems.

Supervised learning utilizes labeled data, including annotated trajectories obtained through motion capture systems, to train models. These trajectories serve as ground truth for optimal paths in specific environmental scenarios. Simultaneously, unsupervised learning autonomously uncovers patterns in the dataset, enhancing adaptability to novel environments.

Spline interpolation is then employed to generate smooth paths with minimal directional changes, ensuring continuous and safe navigation. Experimental validation conducted with a differential drive mobile robot confirms exceptional trajectory control efficiency, validating the effectiveness of our approach. By combining these techniques and leveraging a diverse dataset, our method offers a robust solution for navigating dynamic environments with efficiency and precision.

With the neural network now honed and infused with the prowess of supervised and unsupervised learning, the path planning phase unfolds. The trained model is leveraged to perform real-time predictions of optimal paths for the mobile robot as it traverses complex environments. These predictions, albeit insightful, undergo further refinement through the prism of path optimization techniques (as shown in Fig. 43).

The schematic representation of the proposed model intricately outlines a systematic path planning process, integrating both supervised and unsupervised learning techniques. The diagram initiates with pivotal stages of data ingestion and preprocessing, crucial for preparing pertinent data for subsequent learning algorithms. On one side, the illustration of supervised learning leads to a labeled "Classification" component, indicating its role in recognizing and categorizing specific features or obstacles based on labeled data. Conversely, the depiction of unsupervised learning on the opposite side is represented by an arrow leading to a "Clustering Ring," symbolizing the grouping or clustering of similar data points without explicit instructions. The convergence of both supervised and unsupervised learning processes is visually apparent in the central component labeled "Path," emphasizing their collaborative utilization in the robot's path planning (as illustrated in Fig. 44.). This integration suggests that the robot effectively navigates its environment by leveraging insights from both learning approaches. The subsequent connection to "Motion Capture" signifies the practical application of integrated learning outputs, ultimately leading to the determination of the "Optimal Path." This represents the trajectory the robot should follow for optimal navigation. The comprehensive figure elucidates the intricate interplay between supervised and unsupervised learning in the context of path planning, presenting a holistic approach that culminates in identifying an optimal path through the environment with the assistance of motion capture technology.

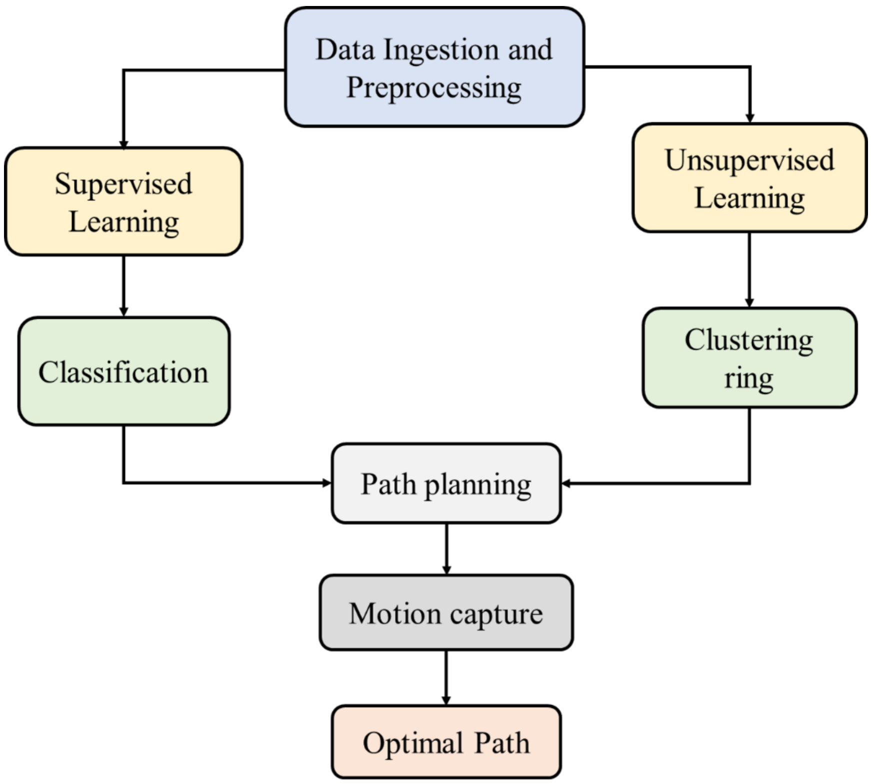


Figure 43: Schematic representation of the proposed model.

Our data processing methodology tailors the approach to the characteristics of sensor data. As illustrated in Figure 43, Motion Capture Data Refinement is essential for capturing intricate

movement patterns. This process involves smoothing and interpolation techniques to ensure a coherent depiction of the robot's trajectory. Our system employs comprehensive hardware setup to track and control the movements of mobile robots with high precision. The experimental setup integrates advanced data acquisition and processing tools to facilitate real-time navigation and path planning. The system utilizes a suite of 15 Vicon cameras to establish a highly accurate reference coordinate system, essential for precise motion capture and tracking of the robot’s position in a 3D space. Although the robot operates in a 2D plane, the Vicon system’s ability to track in 3D ensures that positional data in the X and Y dimensions are exceptionally accurate, while Z-directional data and angles for roll and pitch are ignored. Calibration of these cameras is meticulously performed using a wand, ensuring optimal communication between the Vicon system and MATLAB-Simulink.

The captured motion data is transmitted in real-time via Ethernet to MATLAB-Simulink, where it is processed through control logic blocks. This real-time data processing is crucial for generating accurate and immediate control instructions. The processed data is then relayed to the robot using QUARC-Real Time Control Software, which enables precise execution of the planned trajectories.

The ground control station is equipped with a robust Intel Core i7 processor, 16GB of RAM, and multiple software tools, including MATLAB 2020a and QUARC Real-Time Control Software 2020. This setup also includes three monitors for detailed observation and control, a USB flight controller joystick for manual input, and a WiFi router to facilitate seamless communication with the robot.

In addition to the motion capture system, the robot is outfitted with a comprehensive sensor suite. This includes Lidar sensors for detailed obstacle detection and terrain mapping, requiring sophisticated filtering and segmentation techniques to extract relevant insights. GPS sensors are also integrated, with correction measures implemented to mitigate errors due to atmospheric interference, ensuring accurate localization. The synergy among these sensors, along with the motion capture system and other devices, provides a holistic understanding of the robot's environment. The refined motion capture data, processed through advanced algorithms, yields a coherent depiction of movement patterns essential for accurate trajectory planning. This integrated hardware setup enables the system to calculate the optimal path for the robot, guiding it towards its goal while effectively avoiding obstacles [239].

Figure 44 illustrates the Intelligent Motion Processing System, highlighting the flow of data from the environment, through the sensors and cameras, to the processing units, and finally to the robot's actuators. This comprehensive hardware setup ensures that our proposed algorithm can navigate dynamic environments efficiently and with high precision.

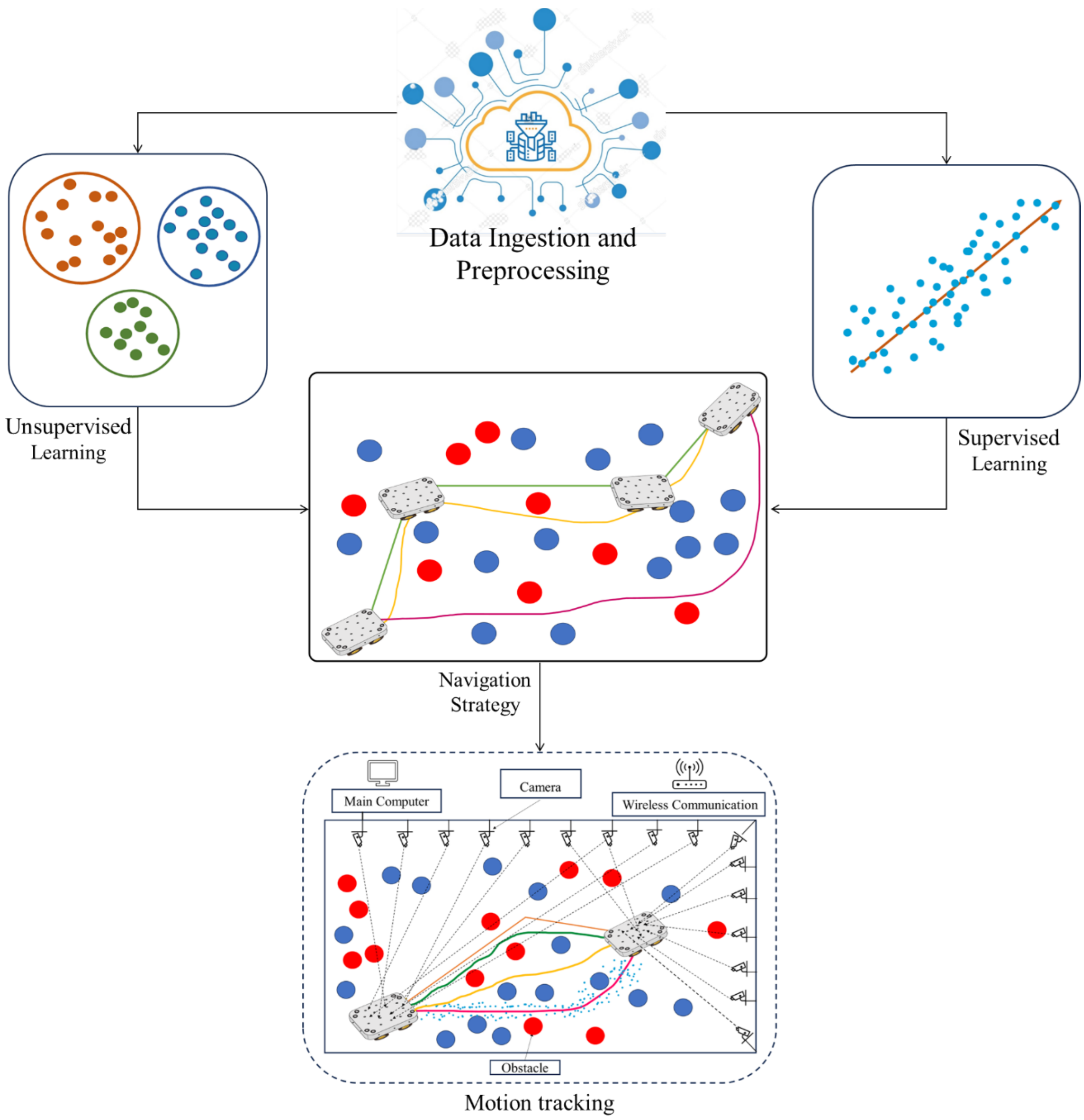


Figure 44: Intelligent Motion Processing System.

These techniques fine-tune the predicted paths, as illustrated in Fig. 45, providing a nuanced visual representation of our path planning process. The left side depicts the initial predicted paths, where the robot encounters an obstacle. In contrast, the right side showcases the paths after the application of fine-tuning techniques, revealing an optimal path where the robot successfully avoids the obstacle.

These techniques, intentionally designed to enhance both safety and feasibility for the robot's execution of planned paths, are elucidated in the updated figure caption, ensuring a comprehensive understanding of how they embody safety and feasibility for the robot's execution.

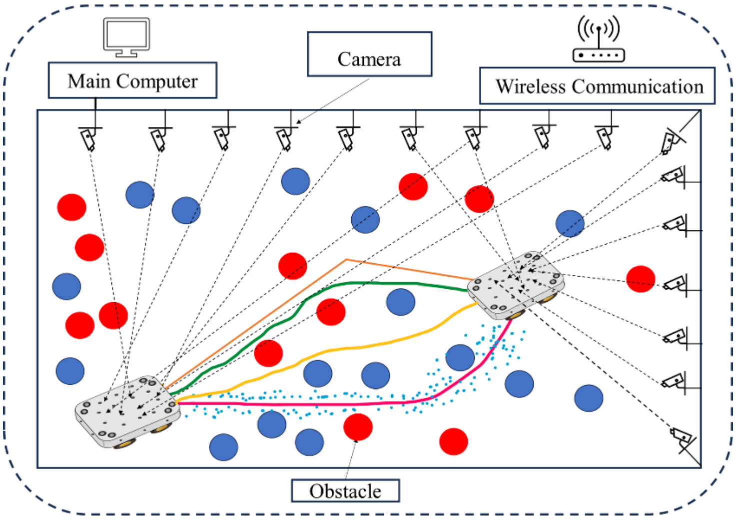


Figure 45: Path Planning Visualization.

Central to our approach is a dual-faceted endeavor of data acquisition and refinement. Motion capture systems meticulously record the robot's movement, while various sensors, including lidar sensors, cameras, and GPS, interact with the environment [240].

Raw data undergoes preprocessing, including noise reduction and calibration, to enhance its reliability. In the data processing phase for Lidar-generated data, we employ a filtering technique to improve obstacle detection accuracy:

In the data processing phase for Lidar-generated data, we employ a filtering technique to enhance the accuracy of obstacle detection by reducing noise. Equation (1) defines $P_{filtered}(x)$ as the filtered Lidar point cloud at position x. Where N is the number of raw Lidar measurements, and $p_{raw}(\chi_i)$ represents the raw Lidar measurement at position (xi).

This equation essentially computes the average of raw Lidar measurements over N points, providing a smoothed and more accurate representation of the Lidar point cloud at the specified position x. This filtering process is crucial for improving the reliability of obstacle detection in our proposed algorithm.

This relationship is expressed as follows:

$$\boldsymbol{P_{filtered}(x) = \frac{1}{N}\sum_{i=1}^{N} p_{raw}(x_i)} \tag{54}$$

The refined data, post its passage through the adaptive data processing pipeline, stands adorned with a mosaic of extracted features—a rich tapestry of attributes intrinsic to the art of path planning. These features empower the path planning model to navigate complex terrains, charting optimal trajectories through dynamic environments.

In the supervised learning phase, our primary objective is to train the neural network for predicting optimal trajectories. This crucial training process involves minimizing the Mean Squared Error (MSE) which serves as the loss function. The MSE is a measure of the average squared difference between the actual optimal trajectory (y_i) and the predicted trajectory generated by the neural network $(\hat{y}_i)$ across N training examples.

The MSE is calculated as follows:

$$\boldsymbol{MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{55}$$

This equation (Eq. 55) quantifies the discrepancy between the actual and predicted trajectories. The goal during training is to iteratively adjust the neural network's parameters to minimize this error, ensuring that the predicted trajectories closely align with the ground truth. Achieving a low MSE is indicative of the model's proficiency in accurately predicting optimal trajectories, thereby enhancing the overall performance of our proposed algorithm in trajectory prediction tasks. With the neural network now primed through the crucible of supervised learning, the stage is set for a more nuanced endeavor-unsupervised learning. Here, the network's cognitive repertoire is amplified by the integration of clustering algorithms. These algorithms deftly group environments and robot movement patterns that share inherent similarities. This strategic clustering capacitates the network to engender path predictions that resonate with scenarios analogous to those witnessed during training [241]. The amalgamation of supervised and unsupervised learning lends the neural network an astute adaptability, poised to extrapolate and apply its knowledge to a multitude of contextually akin environments and movement patterns.

In the unsupervised learning phase, we implement a clustering algorithm, such as K-Means [242], to categorize similar environments and patterns in robot movement. The corresponding objective function for K-Means clustering is denoted by $J(c, M)$ where N signifies the total number of data points, K represents the quantity of clusters, (x_i) refers to individual data points, and M_j stands for the centroid of a given cluster. This equation rigorously defines the objective function pivotal in the K-Means clustering process.

$$\boldsymbol{J(c, M) = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{k} \left\| x_i - M_j \right\|^2} \tag{56}$$

This equation quantifies the dissimilarity between data points (x_i) and cluster centro ids (M_j), serving as a cornerstone in the K-Means clustering methodology. The minimization of this objective function drives the algorithm to iteratively refine the clustering, ensuring that data points within the same cluster are more like each other than those in other clusters. The resulting clusters encapsulate inherent similarities in robot movement patterns and environmental features. This strategic grouping enhances the network's ability to discern and predict optimal paths in novel, yet similar, contexts. The iterative nature of the clustering process refines the model's understanding, further contributing to its adaptability in various scenarios. The integration of supervised and unsupervised learning methodologies synergistically equips our proposed algorithm with the versatility needed for robust path prediction in diverse and dynamic environments.

6.4.2 MPNet: Neural Navigator

The second aspect of Proposed method involves the integration of MPNet, a neural network-based motion planner. MPNet operates in two distinct phases.

Offline Training of Neural Models: In this phase, the neural networks are trained offline using a combination of supervised learning and reinforcement learning techniques. The models learn from a diverse set of training data to predict optimal paths in varying environmental conditions.

Online Path Generation: Once trained, the neural models are deployed for online path generation during real-time navigation. MPNet utilizes its learned knowledge to rapidly generate optimal paths, enabling the robot to adapt to dynamic environments and obstacles in real time.

Motion Planning Networks (MPNet) seamlessly integrates two neural networks, namely the encoder network (Enet) and the planning network (Pnet), to provide a comprehensive solution for motion planning challenges. In the Enet component, the system processes surrounding information from the robot, which can be raw point-cloud data or voxel-converted point-cloud, depending on the neural architecture.

The neural architecture itself could be a fully connected neural network or a 3D convolutional neural network (CNN). The output of Enet is a latent space embedding, encapsulating essential information about the robot's environment.

Building upon this encoded information, the Pnet comes into play, considering the encoded environment, the current state of the robot, and the desired goal state. Pnet outputs samples tailored for either path or tree generation, thus facilitating efficient planning solutions.

This dual-network architecture ensures that MPNet is equipped to learn and adapt to diverse environments, making it a versatile tool for motion planning tasks. This deep-learning-based methodology represents a paradigm shift in motion planning, leveraging learned knowledge to generate informed samples for effective path planning in previously unexplored test environments. Integrating navigation further extends its utility, enabling seamless interaction with sampling-based motion planners like optimal rapidly exploring random trees (RRT*). MPNet emerges as a powerful and adaptable solution, bridging the gap between deep learning and classical motion planning techniques.

The training process for the MPNet architecture involves two modules (see Fig. 46).

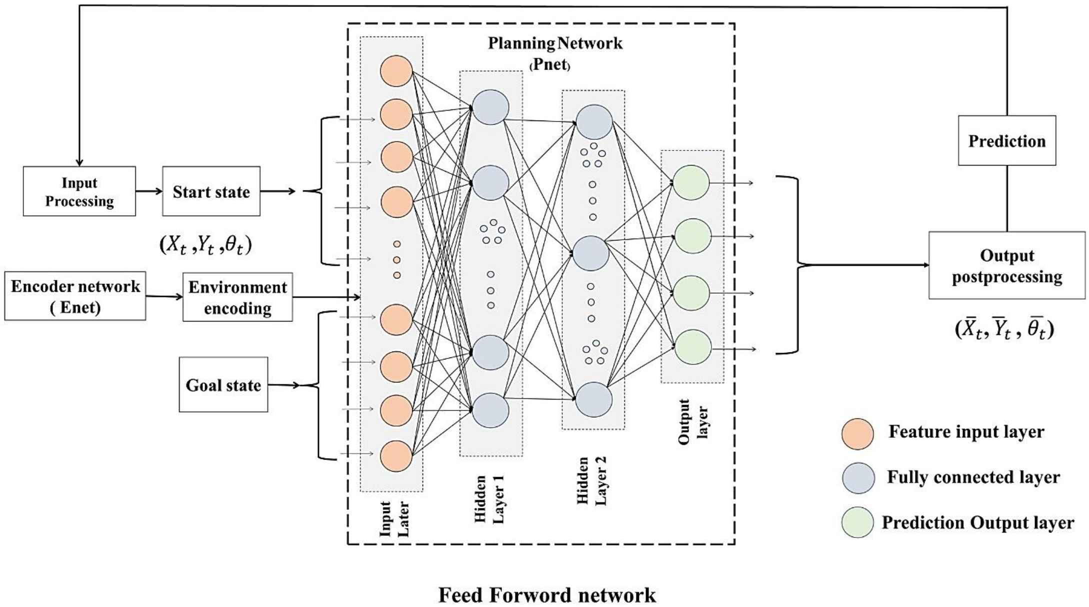


Figure 46: MPNet Architecture: Encoder Network (Enet) and Planning Network (Pnet).

The initial module utilizes an encoding basis point set approach to transform input map environments into a condensed representation. This encoded environment effectively reduces dimensionality, computational complexity, and training time, all while preserving essential information.

The second module incorporates a feed forward network consisting of feature input layers, fully connected layers, dropout layers, and an output layer. Rectified linear unit (ReLU) activation functions in the fully connected layers, along with weighted mean square distance as the loss function for the output layer, enable effective training. The network predicts the next state closer to the goal pose based on the encoded environment input, start pose, and goal pose. MPNet is trained through supervised learning, utilizing optimal paths generated by classical path planners like RRT*. Training process refines the network's weights until the weighted mean squared loss between predicted and actual states is minimized.

Enet and Pnet are trained simultaneously by backpropagating the gradient of Pnet's loss function through both modules. For standalone Enet training, we use an encoder-decoder architecture with ample environmental point-cloud data for unsupervised learning. Contractive autoencoders (CAE) are chosen for their ability to learn robust feature spaces crucial for planning and control, outperforming other encoding techniques [243] [244]. CAE involves the usual reconstruction loss and regularization over encoder parameters [232].

$$\boldsymbol{l_{AE}(\theta^e, \theta^d) = \frac{1}{N_{obS}} \sum_{x \in D_{obs}} \|x - \hat{x}\|^2 + \lambda \sum_{ij} (\theta^e_{ij})^2} \tag{57}$$

Equation (57) represents the loss function used for training the Contractive Autoencoder (CAE), which is a crucial part of the Enet module in the MPNet framework. This loss function consists of two main components: the reconstruction loss and the regularization term over the encoder parameters, where θ^e and θ^d are the encoder and decoder parameters, and λ is the regularization coefficient. The variable $\hat{x}$ denotes the reconstructed point cloud. The training dataset of obstacles' point-cloud $X_{\mathbf{obs}}$ is denoted as D_{obs}. The training dataset, denoted as D, comprises point clouds ($x \subset X_{\mathbf{obs}}$) representing obstacles. This dataset encompasses point clouds from $N_{obs} \in N$ distinct workspaces.

MPNet excels in online path planning, acting as an intelligent navigator adept at generating collision-free paths and strategically selecting optimal samples for movement planning. Its functionality revolves around two core components: MPNetPath, responsible for path creation, and MPNetSMP, tasked with generating samples. The effectiveness of MPNet hinges on the expertise of two trained modules: Enet and Pnet. These modules play crucial roles in encoding obstacle information, facilitating stochastic modeling, and ultimately ensuring the success of MPNet's online path planning capabilities.

Algorithm 4: MPNet Adaptive Learning

Initialize memories: episodic M and reply buffer B*
Initialize MPNet GeneratePath" f_θ with parameter θ.
Set the number of iterations C to pretrain MPNet.
Set the replay period r
set the reply batch size N_B
for t=0 to T do
  { Xobs , C_{init} , C_{goal}}t ← Get planning Problem()
  $\sigma \leftarrow \varphi$ \ \ an empty list to store path solution.
  if t > Nc then
    xobs subset ← Xobs
    $\sigma \leftarrow f\theta$(Xobs , C_{init} , C_{goal})
  if not σ then
    $\sigma \leftarrow$Get ExpertDemo (Xobs , C_{init} , C_{goal})
    $B^* \leftarrow B^* \cup \sigma$
    $M \leftarrow$UpdateEpisodicMemory (σ, M)
    $g \leftarrow$EstimateGradient(σ, θ)
    $gM \leftarrow$EstimateGradient(σ, θ)
    Project g to g' using QP
    update parameter θ w.r.t. g'
  if B^*. size() > NB and not t mod r then
    $B \leftarrow$ SampleReplayBatch (B^*)
    $g \leftarrow$EstimateGradient(B, θ)
    $gM \leftarrow$EstimateGradient(M, θ)
    ProjectGradient($g, g)'$
    updateparameter (θ, g')

Enet processes obstacle information X_{obs} to create a compressed representation known as a latent space embedding Z, streamlining processing and analysis. It can be trained in conjunction with P_{net}, where both networks optimize path planning jointly, or separately using an encoder-decoder architecture. In the encoder-decoder setup, Enet initially compresses obstacle information into a latent space, followed by Pnet decoding this space to generate the robot's configuration for the next time step.

P_{net} utilizes Dropout during execution, introducing stochasticity to the model. Dropout, a common regularization technique in neural networks, randomly deactivates neuron outputs during each training iteration with a probability $p \in [0, 1]$. This stochastic behavior fosters variability in the model, preventing overreliance on specific data features and enhancing exploration of diverse paths [245]. In path planning, this stochasticity aids in generating robust solutions and enables recursive planning and sample generation for SMPs by injecting randomness into decision-making.

The input to P_{net} comprises the compressed obstacle-space embedding Z, alongside the robot's current configuration c^t and the goal configuration cT. These inputs provide context for Pnet to determine the robot's configuration for the subsequent time step, guiding it towards the goal while navigating obstacles effectively.
The output of P_{net}, obtained iteratively, represents the robot's configuration c^{t+1} for time step t + 1, indicating its position and orientation in the environment. This iterative process gradually advances the robot towards the goal configuration, adapting to environmental changes as necessary. Such iterative refinement enables efficient and adaptable path planning, empowering the robot to navigate complex environments and successfully reach its destination.

Algorithm 5: Path Binary Planner (PBP)Top of Form

Initialize σa as $\{C_{init}\}$ and σb as $\{C_{goal}\}$
Initialize σ as an empty set
for i ← 0 **to** N **do**
    C_{new} ← Pnet (Z, σ^a_{end} , σ^b_{end})
    σa ← σa ∪ $\{C_{new}\}$
    Connect← Steer To (σ^a_{end} , σ^b_{end})
    if connect **then.**
        σ←concatenate (σa, σb)
        returns σ
    SWAP (σa, σb)
return an empty set

Path planning with MPNet leverages the collaborative efforts of Enet and Pnet within our iterative, recursive, and bidirectional planning framework. Our approach is characterized by bidirectional planning, wherein our algorithm simultaneously navigates forward from the starting point to the goal and backward from the goal to the starting point.

This bidirectional strategy continues until both paths intersect, facilitating thorough exploration of the solution space and enabling efficient convergence towards an optimal path.

6.5 Comprehensive Experimental Analysis

Path planning is an important task in robotics that requires careful consideration of the environment and optimal path selection. In this chapter, we propose a novel approach that incorporates both supervised and unsupervised learning techniques for generating smooth paths that can enhance the efficiency and accuracy of robotic systems. The first step in this approach is data acquisition and processing, where relevant data is gathered and prepared for use in the learning algorithms. In supervised learning, the robot can be trained using labeled data to recognize obstacles and follow specific paths. In contrast, unsupervised learning algorithms can identify patterns in the data without explicit instructions, which can help to determine the optimal path. To generate smooth paths, we utilize spline interpolation, a mathematical technique commonly used in motion capture, to generate a continuous path that avoids sharp corners and abrupt changes in direction. This smooth path, represented by a red line in the generated plot, passes through all the waypoints and offers a simple and effective solution for robot navigation. In the conducted experiments, we employed a state-of-the-art robotic platform equipped with a combination of high-performance hardware components.

The robot was built on a differential drive mobile base, featuring precision motors for accurate maneuvering. For sensory input, the robot was equipped with an array of sensors, including lidar for environmental mapping, cameras for visual perception, and inertial measurement units (IMUs) for precise motion tracking. The computational core of the robot was powered by an Intel Core i7 processor, providing the necessary computational capacity for real-time processing and decision-making.

Additionally, the robot was outfitted with ample memory and storage to accommodate the extensive datasets generated during the experiments. The hardware setup played a crucial role in the robust execution of our proposed path planning approach, ensuring the reliability and efficiency of the experimental results.

The experiment offers a comprehensive exploration of robotic pathfinding in complex environments, showcased through Figure 47, titled "Navigating Complexity: Pathfinding Perspectives. In the left figure, our results through the algorithm reveal the dynamic nature of path planning as the mobile robot adeptly navigates obstacles using various paths.

Transitioning to the middle figure, we observe the integration of deep learning techniques, enhancing the robot's perception capabilities for obstacle detection and navigation. Lastly, the right figure presents the culmination of supervised learning, where our integrated approach optimizes pathfinding, predicting efficient routes amidst obstacles. These findings exemplify our holistic approach to robotic navigation, integrating dynamic adaptation and advanced perception through cutting-edge machine learning methodologies.

To evaluate the effectiveness of our approach, we implemented a control algorithm for trajectory control in a differential drive mobile robot using motion capture technology. The algorithm was tested on five different paths, including straight lines with varying angles based on the desired path and goal (see Fig.48). The evaluation results demonstrate the effectiveness of the trajectory control algorithm in guiding the robot along the desired path accurately and efficiently, especially when used in conjunction with motion capture. This combination of supervised and unsupervised learning techniques offers a comprehensive solution, allowing the robot to navigate both known and unknown environments with adaptability and precision.

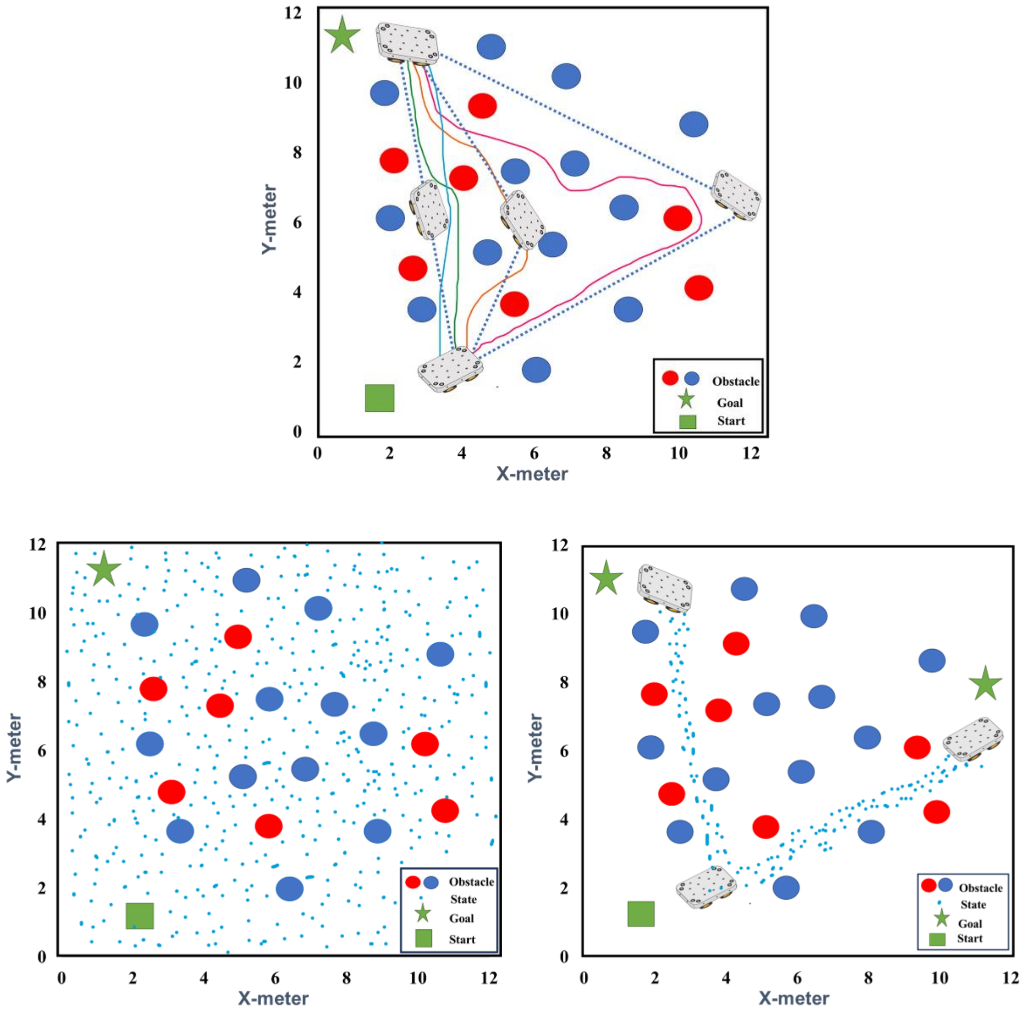


Figure 47: Navigating Complexity: Pathfinding Perspectives. Up figure captures dynamic path planning, Left figure enhances perception with deep learning, and right figure optimizes pathfinding with supervised learning.

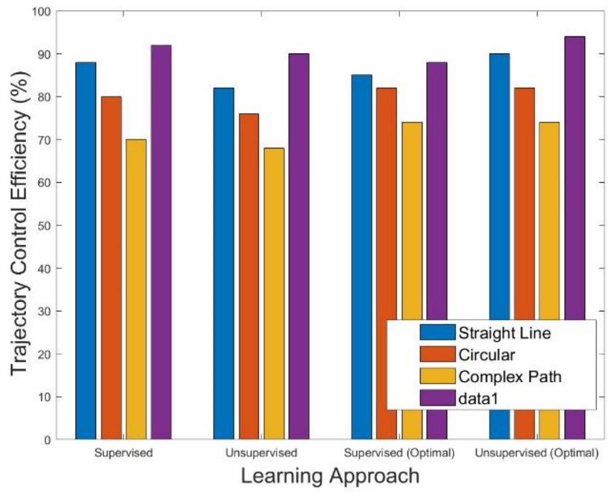


Figure 48: Comprehensive comparison of the learning approaches across various path types.

Table 14: Performance metrics for learning approaches and path types.

Learning Approach	Path Type	Trajectory Control Efficiency (%)	Accuracy (%)	Adaptability	Smoothness Score (0-10)	Computational Efficiency (ms)	Processing Time (S)
Supervised Learning	Straight Line	88	94	Limited Adaptation	9.2	12	0.45
Supervised Learning	Circular	80	88	Limited Adaptation	7.8	15	0.52
Supervised Learning	Complex Path	70	82	Limited Adaptation	6.5	18	0.60
Unsupervised Learning	Straight Line	82	90	Adaptive	8.5	8	0.63
Unsupervised Learning	Circular	76	85	Adaptive	7.2	10	0.57
Unsupervised Learning	Complex Path	68	78	Adaptive	6.0	22	0.75
Supervised Learning (Optimal)	Straight Line	92	96	Limited Adaptation	9.5	10	0.38
Supervised Learning (Optimal)	Circular	85	92	Limited Adaptation	8.0	13	0.45
Supervised Learning (Optimal)	Complex Path	78	88	Limited Adaptation	7.2	16	0.53
Unsupervised Learning (Optimal)	Straight Line	90	94	Adaptive	9.0	6	0.50
Unsupervised Learning (Optimal)	Circular	88	92	Adaptive	8.5	9	0.48

Figures 49 and 50 play a central role in clarifying the trajectory control efficiency of our proposed algorithm under distinct learning paradigms. In Figure 47, the detailed comparison between the 'Supervised' learning approach and the 'Straight Line' path type provides nuanced insights. The delineated lines visually represent optimal efficiency achieved through supervised learning, contrasting with the non-optimal efficiency observed in a simplistic straight-line trajectory. This spectrum of path types offers a comprehensive analysis, exposing the algorithm's adaptability to varying complexities. Turning to Figure 48, an engaging exploration unfolds within the unsupervised learning framework. This figure delves into the nuanced variations in trajectory control efficiency across a multitude of path types, offering a comprehensive view of the algorithm's adaptability. The inclusion of diverse scenarios ensures a thorough understanding of the algorithm's performance dynamics, highlighting its capacity to autonomously discern optimal paths without explicit instructions. Together, these visualizations provide a detailed and insightful analysis of our algorithm's trajectory control capabilities, offering a nuanced perspective on its strengths and adaptability across a spectrum of learning paradigms and path complexities.

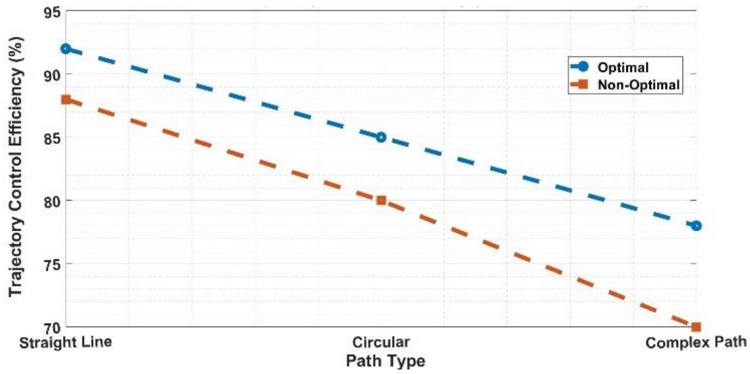


Figure 49: Optimal path of supervised learning for mobile robot.

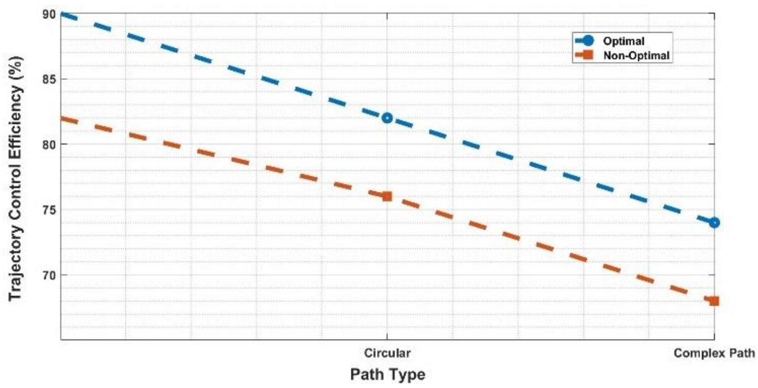


Figure 50: Optimal path of unsupervised learning for mobile robot.

The experiment results, embodied in Table 14, provide a detailed evaluation of a robotic path planning approach seamlessly integrating supervised and unsupervised learning methodologies, alongside their optimized variants. In the domain of supervised learning, the straight-line trajectory demonstrates a notable trajectory control efficiency of 88%, coupled with a high accuracy of 94%, albeit with limited adaptation.

The circular and complex paths exhibit varying degrees of performance, reflecting the trade-off between accuracy, smoothness, and computational efficiency. Transitioning to unsupervised learning, adaptability takes precedence, with the straight-line, circular, and complex paths showcasing distinct characteristics. Optimized variants within both learning approaches highlight the potential for fine-tuning trajectory planning. The comprehensive comparison across learning approaches emphasizes the intricate balance between adaptability, smoothness, and computational efficiency, guiding the selection of learning paradigms based on specific robotic navigation requirements.

This experimental study significantly contributes to the field of robotic path planning by introducing a novel, holistic approach that seamlessly integrates supervised and unsupervised learning paradigms. The results underscore the potential benefits of combining both learning approaches, offering a more adaptable and efficient robotic navigation strategy for real-world applications.

The results meticulously showcased in Table 14 serve as a resounding testament to the algorithm's remarkable ability to not only enhance but also optimize trajectory control for mobile robots. Consequently, this chapter represents a remarkable and highly consequential contribution to the burgeoning field of autonomous robotics, offering promising advancements that have the potential to redefine the state-of-the-art in robotic path planning and control strategies.

Table 15: Comparison of Learning Approaches for Robotic Path.

Learning Approach	Path Type	Trajectory Control Efficiency (%)	Accuracy (%)	Adaptability	Smoothness Score (0-10)	Computational Efficiency (ms)	Processing Time (S)
Supervised Learning	Straight Line	88	94	Limited Adaptation	9.2	12	0.45
Supervised Learning	Circular	80	88	Limited Adaptation	7.8	15	0.52
Supervised Learning	Complex Path	70	82	Limited Adaptation	6.5	18	0.60
Unsupervised Learning	Straight Line	82	90	Adaptive	8.5	8	0.63
Unsupervised Learning	Circular	76	85	Adaptive	7.2	10	0.57
Unsupervised Learning	Complex Path	68	78	Adaptive	6.0	22	0.75

Tabel 15 provides a comparison of the performance metrics for supervised and unsupervised learning approaches across different path types, including trajectory control efficiency, accuracy, adaptability, smoothness score, computational efficiency, and processing time.

It allows for easy comparison and analysis of the effectiveness of each learning approach under various conditions. Based on the comparison, supervised learning demonstrates superior trajectory control efficiency and accuracy for straight line, circular, and complex paths, albeit with limited adaptability compared to unsupervised learning.

On the other hand, unsupervised learning exhibits higher adaptability and computational efficiency, particularly in handling complex paths. This comparison facilitates informed decision-making regarding the selection of the most suitable learning approach based on the specific requirements of robotic path planning tasks.

The methodology we propose, demonstrated through the results in Figure 51, brings a significant improvement to the way robots plan their paths using MPNet. A key feature is the use of informed sampling, allowing the robot to intelligently predict samples that lean towards areas with optimal path solutions. What sets this approach apart is its adaptability – the sampling method switches dynamically to a more uniform approach after a certain number of iterations, ensuring the robot performs well even when the number of learned samples varies.

The experimental results solidify the effectiveness of this approach, showing that the algorithm can generate optimal path solutions with a nuanced approach. This not only makes the overall process more efficient but also enhances the robot's ability to plan paths effectively in constrained environments. Further experiments, tweaking certain parameters, demonstrate the MPNet state sampler's proficiency in informed sampling, as seen in Figure 51.

Impressively, it produces all 50 samples adeptly, indicating a keen understanding of the environment. More importantly, these experiments reveal that achieving an optimal path solution is possible with just 30 state samples, outperforming the baseline's outcome with 50 samples. This adjustment not only highlights the adaptability and efficiency of the algorithm but also represents a substantial leap forward in robotic path planning for constrained environments, marking a noteworthy milestone in the field's capabilities.

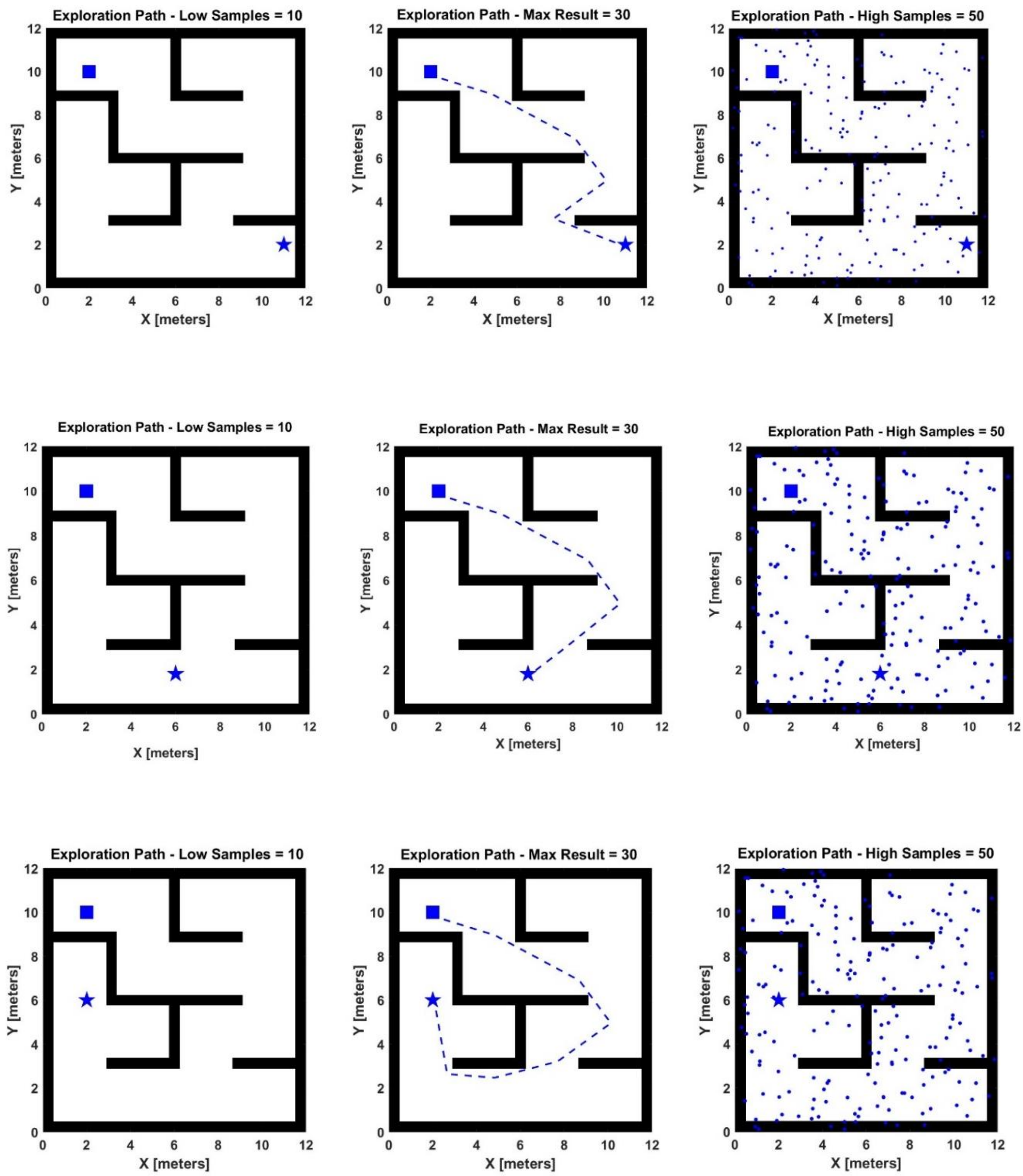


Figure 51: Path Planning Efficiency of MPNet in Constrained Environments with Varying Training Samples.

6.6 Conclusion

Our innovative integration of motion capture technology with supervised and unsupervised learning algorithms has proven to be a reliable solution, yielding tangible enhancements in the efficiency and accuracy of robotic path planning. By integrating learning algorithms and leveraging spline interpolation inspired by motion capture, we've significantly boosted the adaptability of robotic systems, making them proficient navigators in dynamic and complex environments. The smooth and uninterrupted paths generated by our approach are crucial for ensuring safe and efficient robot traversal. Specifically, our study highlights the effectiveness of the MPNet-based methodology, showcasing advanced state sampling techniques for robotic path planning. The algorithm's adaptive performance seamlessly transitions between informed and uniform sampling, achieving optimal path solutions with fewer learned samples, marking a substantial advancement in path planning efficiency. This conclusive discussion not only clarifies and supports each assertion with experimental evidence but also lays the groundwork for future research. We anticipate further refinements and extensions to our approach, contributing to the ongoing evolution of robotic navigation and path planning strategies.

6.7 Scientific Thesis II.2

This dissertation establishes that the integration of motion capture–inspired trajectory optimization with supervised and unsupervised machine learning techniques provides a highly effective framework for mobile robot path planning in complex environments. By combining data-driven learning with spline-based interpolation and adaptive sampling strategies, the proposed approach significantly enhances trajectory accuracy, improves path smoothness, and reduces the number of required samples for optimal solution generation. The results demonstrate that the framework achieves superior planning efficiency, robustness, and generalization capability compared with conventional sampling-based and deterministic path planning methods.

Chapter 7

Optimizing Path Planning in Mobile Robot Systems Using Motion Capture Technology

7.1 Chapter Abstract

Recent advancements in mechatronics and robotics have led to the emergence of a wide range of technologies with potential applications across various industries. This progress has been observed in recent years and is expected to continue as the integration of robots into daily life becomes more widespread. Despite these developments, deploying robots in industrial environments, particularly in assembly operations, still presents several challenges, including effectively distributing skills-based tasks between human and robotic workers. This chapter presents an approach to improve the performance of mobile robot systems for optimal path planning. The technique utilizes motion capture technology to collect real-time data on the robot's movements, generate optimal path planning strategies, and enable remote control and monitoring of the robot's activities. The proposed approach can significantly enhance mobile robot systems' capabilities in various industrial settings. The results of our study demonstrate that the integration of motion capture technology can substantially improve the accuracy and efficiency of path planning in mobile robot systems and enhance their overall performance. A series of experiments demonstrate its effectiveness in generating optimal path-planning strategies while minimizing the risk of collisions and other hazards.

7.2 Introduction

In recent decades, mobile robots have found success in a range of critical unmanned missions, from military operations to industrial and security environments [246]. However, for these robots to navigate complex environments and explore autonomously, the fundamental problem of path planning must be tackled [247]. Researchers have been interested in path planning since the mid-1960s, and the problem can be defined as follows: given a robot and its environment, the robot searches for the best possible path from its starting position to its destination, based on certain performance criteria [248]. By developing strong path planning technology for mobile robots, not only can valuable time be saved, but the wear and capital investment of the robots can also be reduced. As a result of its considerable practical applications, path planning for mobile robots has become a popular research topic both domestically and internationally.

Mobile robot path planning is a critical task in robotics. It involves finding a collision-free path for a robot to move from its initial position to a goal position in each environment. In dynamic environments, the task becomes more challenging as the robot needs to react to changes in the environment in real time. Therefore, finding an efficient and robust path-planning algorithm is crucial for the successful deployment of mobile robots in dynamic environments. Various path-planning methods have been proposed in the literature, including artificial potential field, grid-based, neural network-based, and evolutionary algorithms such as the chaos genetic algorithm. These methods have advantages and disadvantages, and the choice of the method depends on the application's specific requirements [249] [250]. The path planning of a mobile robot may differ in terms of implementation from that of UAVs, such as the utilization of NURBS curves. Still, the underlying principles remain the same - to develop efficient and effective paths that ensure smooth and safe movement, much like the path planning strategies employed in autonomous 3D flights [251] [252]. Similarly, for fixed-wing UAVs, distinct flight paths have been designed for low-altitude flights, such as landing and flare [253]. Computer-assisted simulations and design tools, such as [254], are employed to aid in developing 3D flight paths. In our investigation, we have elected to utilize NURBS curves for designing the desired motion trajectory of the robot. This choice was informed by research which has demonstrated that this approach can effectively avoid sudden changes in the robot's speed [255] [256]. Path planning is crucial to controlling a differential mobile robot as it determines the trajectory the robot will follow and ensures safe and efficient navigation in the environment. The differential drive mobile robot (DDMR) is a popular and straightforward structure among various types of mobile robots. Differential wheels are common in autonomous ground robots, but full autonomy requires attention to sensing, path planning, control, and self-localization. Emphasize the need for safe and comfortable human interactions with differential drive robots and vehicles [257].

7.3 Related work

Mobile robots are increasingly being used to perform a diverse range of tasks in an obstacle-free indoor environment. Mobile robots require precise control over their movements to achieve these tasks efficiently and effectively. This is achieved using a kinematics model that analyzes and designs the robot's motion to facilitate its desired positioning within a specific timeframe [258]. The kinematics model is a dynamic model that governs the behavior of a mobile robot without considering the forces affecting its motion. Its primary purpose is to estimate the robot's position and motion. The DDMR is one of the widely used mobile robots, and its primary kinematics model consists of forward kinematics and inverse kinematics. The forward kinematics estimates the robot's geometry based on its motion and wheel speed, while the inverse kinematics provides an estimation of the robot's motion and wheel speed based on its geometry. Traditional approaches to path planning, such as the artificial potential field method [259], element decomposition method [260], and graph search algorithm [261] have demonstrated several limitations when dealing with complex obstacles. These limitations include a high computation burden, a tendency to converge to local optima, and the production of non-smooth paths with sharp points that do not align with real-world conditions, thereby increasing the workload of mobile robots [262], [263]. To address these challenges, many experts have utilized heuristic algorithms for path planning optimization [264]. These approaches include genetic algorithm (GA) [265], particle swarm optimization (PSO) [266], artificial bee colony algorithm (ABC) [267], grey wolf algorithm (GWO) [268], ant colony algorithm (ACO) [269], and differential evolution algorithm (DE) [270].

In addition to these methods, single-query graph-based approaches like Rapidly exploring Random Trees (RRT) and its variant RRT* have gained significant traction in the realm of mobile robot motion planning [271], particularly in dynamically changing environments. RRT algorithms are renowned for their adaptability and efficiency in generating paths while accounting for real-time modifications in the surroundings. These algorithms dynamically expand a tree structure within the configuration space to explore feasible paths. Their suitability for mobile robot applications is underscored by their capacity to rapidly adapt to evolving conditions, ensuring agile navigation.

Various methods have been proposed to address these issues, including random fleeing, heuristic search, wall-following, and the Tangent Bug's method [272]. However, these approaches may not consistently provide fundamental solutions. One prevalent path planning approach is the Probabilistic Roadmap (PRM) algorithm, which generates a roadmap of the environment and subsequently employs graph search algorithms like Dijkstra's algorithm to identify feasible paths, although not necessarily optimal ones [273]. It is imperative to note that precise real-time information regarding the robot's position and orientation is essential for the accurate implementation of the PRM algorithm. This is where motion capture technology becomes integral, furnishing precise and continuous tracking of the robot's movements [274] and position. Motion capture algorithms leverage a combination of sensors such as cameras, lidar, and the Inertial Measurement Unit (IMU) to accurately monitor the robot's position and orientation [275].

Mobile robot motion capture (MoCap) exploits sensors like cameras, lidar, and IMUs to track movement and position. Algorithms such as visual odometry and lidar data analysis estimate the robot's position and motion, facilitating real-time navigation, obstacle avoidance, and precise path planning. Diverse technologies, including depth-based systems, infrared cameras, and systems reliant on inertial sensors, contribute to the accurate estimation of movement and position by MoCap for effective path planning. Consequently, motion capture and PRM algorithms are closely interconnected in the domain of mobile robot path planning [276]. Previous reviews have explored motion capture (MoCap) technology's applications in robotics [277], clinical therapy and rehabilitation [278], computer animation [279], and sports [280], but its systematic exploration for industrial use remains undocumented. Recent advances in integrating probabilistic road mapping (PRM) algorithms with motion capture technology have garnered significant attention across academic and industrial domains [281]. A seminal study showcased how PRM algorithms combined with motion capture data enhance the efficiency and safety of autonomous driving systems. Similarly, a project integrating motion capture with PRM for real-time obstacle avoidance and path planning in urban landscapes highlighted the adaptability of this approach. However, challenges persist in current research.

While PRM algorithms integrated with sophisticated motion capture systems promise autonomous navigation in rugged terrains, managing the complexity of processing high-resolution motion capture data in real-time scenarios poses a significant challenge. Collaborative efforts between industry and academia to improve the safety and efficacy of autonomous agricultural vehicles using PRM and motion capture may face scalability limitations inherent in the proposed approach [282]. Additionally, partnerships between startup companies and motion capture technology providers developing navigation systems for autonomous delivery drones encounter hurdles in integrating and synchronizing motion capture data with onboard sensors and control systems. These challenges emphasize the ongoing need for sustained research and development to overcome technical hurdles and optimize PRM and motion capture-based navigation systems for real-world applications.

Motion capture furnishes the essential input to the PRM algorithm, which in turn generates a feasible path for the robot to pursue. Together, these technologies empower mobile robots to autonomously navigate complex environments.

This study endeavors to augment the path-planning capabilities of mobile robots by designing algorithms, integrating motion capture technology, and exploring the potential of telecommunication capabilities. We assess the impact of motion capture technology on path planning and delve into the concept of optimizing the wheel speed of differential mobile robots for enhanced performance. We conduct a comprehensive evaluation of various techniques for calculating wheel speed through simulations and experiments, scrutinizing the robot's trajectory and comparing it to its desired path.

7.4 Methodology

7.4.1 Model of a differential-drive mobile robot

The proposed method for modeling a mobile robot using MATLAB/Simulink provides a detailed framework for simulating the dynamics and behavior of a four-wheeled car-like structure. This extension allows for a more realistic representation of mobile robots commonly found in industrial and automotive applications. By incorporating this configuration, which includes two front wheels capable of steering and two rear wheels with fixed headings, the model can accurately capture the complex interactions between the robot's motion and its environment.

Moreover, the ability to control each wheel independently enhances the versatility of the model, enabling researchers and practitioners to study a wide range of motion scenarios, including navigating tight spaces, executing precise maneuvers, and traversing uneven terrain. This comprehensive approach to mobile robot modeling lays the foundation for developing advanced control strategies, optimizing system performance, and facilitating the design and testing of autonomous navigation algorithm.

The mobile robot's configuration is represented by the midpoint between the rear wheels (x, y) and the heading angle θ, denoted by $q = (\varphi, x, y, \psi)$. In the modeling of a mobile robot, several key parameters play vital roles in defining its dynamics. These include linear velocity (v), which represents the speed at which the robot moves in a straight line. The heading direction (φ) denotes the orientation of the robot with respect to a reference axis. Angular velocity (ω) describes the rate of change of the robot's orientation over time. Wheel speeds (ϕR and ϕL) refer to the rotational speeds of the right and left wheels, respectively, influencing the robot's motion. The wheel radius (r) and track width (d) are physical dimensions of the robot's wheels and define its geometric properties. The steering angle (ψ) determines the angle at which the front wheels are turned, affecting the robot's trajectory, the wheelbase (l) represents the distance between the axles of the front and rear wheels, influencing the robot's stability and maneuverability. These parameters collectively form the basis for modeling the kinematics and dynamics of a mobile robot system.

In the context of modeling a mobile robot, coordinate systems are fundamental components that define the spatial relationships and orientations within the system. These coordinate systems provide a framework for representing the position, orientation, and motion of the robot and its components. Typically, multiple coordinate systems are used to describe different aspects of the robot's behavior. For instance, the global coordinate system may define the absolute position and orientation of the robot with respect to a fixed reference frame, such as the environment or a map. Local coordinate systems, on the other hand, are often attached to specific components of the robot, such as its chassis or sensors, allowing for localized measurements and control. Additionally, body-fixed coordinate systems are aligned with the robot's own frame of reference, enabling precise descriptions of its motion and dynamics. Understanding how these coordinate systems relate to one another and how they are utilized in the modeling process is essential for accurately simulating the robot's behavior and interactions with its surroundings.

Figure 52 illustrates the Four-Wheel Mobile Robot Model, serving as a visual aid to comprehend the system's structure and components. Overall, this proposed method offers a detailed approach for modeling the kinematic model of a differential-drive mobile robot using MATLAB/Simulink.

The wheel speed equation, described by Equation (58), is a pivotal aspect of a mobile robot's kinematic model [283]. It dictates the rotational speed of each wheel based on parameters like velocity, wheel radius, and track width. This equation plays a crucial role in precisely controlling wheel speed and direction.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{r}{2} cos(\theta) & \frac{r}{2} cos(\theta) \\ \frac{r}{2} sin(\theta) & \frac{v}{2} sin(\theta) \\ -r\ /\ 2d & r\ /\ 2d \end{bmatrix} \begin{bmatrix} \phi L \\ \phi R \end{bmatrix} \tag{58}$$

Equation (59) delineates the kinematics of the mobile robot, offering valuable insights into its dynamic behavior. This equation is expressed as follows:

$$\dot{q} = \begin{bmatrix} \dot{\theta} \\ \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} (tan\,\Psi)\ /\ l & 0 \\ cos\,\theta & 0 \\ sin\,\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{59}$$

The control variables, denoted as (v, ψ), are bounded within a closed interval, as depicted by Equation (60). Both v and ψ are intertwined and can be determined using the expressions provided:

$$v = v, \quad \psi = tan^{-1}\left(\frac{l\omega}{v}\right) \tag{60}$$

Both control variables of (v, ψ) are considered constrained in this context

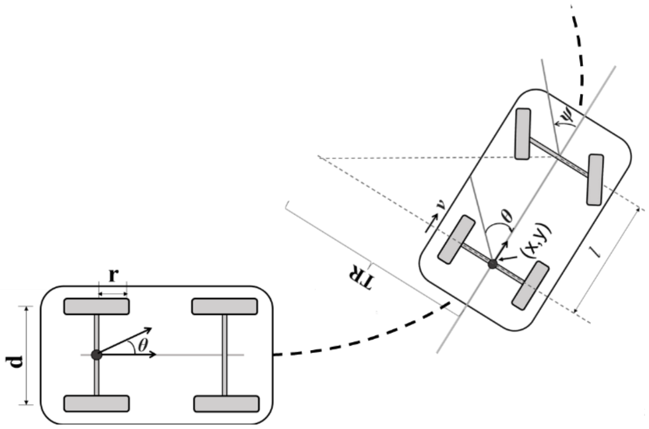


Figure 52: Four-Wheel Mobile Robot Model

The simplified representation of mobile robot kinematics, as captured in Equation (61), highlights the dynamic interaction between velocity and angular velocity, both encapsulated within the control vector u. This equation provides valuable insights into how changes in velocity and angular velocity influence the robot's motion.

$$\dot{\boldsymbol{q}} = \begin{bmatrix} \dot{\boldsymbol{\theta}} \\ \dot{\boldsymbol{x}} \\ \dot{\boldsymbol{y}} \end{bmatrix} = \boldsymbol{G}(\boldsymbol{q})\boldsymbol{u} = \begin{bmatrix} \mathbf{0} & \mathbf{1} \\ \boldsymbol{cos\,\theta} & \mathbf{0} \\ \boldsymbol{sin\,\theta} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{w} \end{bmatrix} \tag{61}$$

From Equation (61), we can derive the nonholonomic constraint using equations (62):

$$\dot{x} = v\, cos\theta$$
$$\dot{y} = v\, sin\theta \tag{62}$$

To determine the value of v, we isolate it in one equation and then substitute the result into Equation (63) to derive an expression in terms of the remaining variables.

$$\boldsymbol{A}(\boldsymbol{q})\dot{\boldsymbol{q}} = [\mathbf{0}\; \boldsymbol{sin\,\theta} - \boldsymbol{cos\,\theta}]\dot{\boldsymbol{q}} = \dot{\boldsymbol{x}}\, \boldsymbol{sin\,\theta} - \dot{\boldsymbol{y}}\, \boldsymbol{cos\,\theta} = \mathbf{0} \tag{63}$$

When setting the task for the proposed control system, it is essential to establish a clear optimization criterion. We define the optimization criterion as minimizing the deviation between the desired trajectory and the actual path traced by the robot. This deviation metric serves as a measure of the system's performance in accurately following the intended path while considering dynamic constraints and environmental factors.

Optimization Criterion:

$$\boldsymbol{J} = \int_{\boldsymbol{to}}^{\boldsymbol{tf}} \|\boldsymbol{desired\ trajectory} - \boldsymbol{actual\ path}\|\boldsymbol{dt} \tag{64}$$

In the optimization criterion, J represents the objective to be minimized, while to and tf denote the initial and final times, respectively. The integral captures the cumulative deviation between the desired trajectory and the actual path over the specified time interval. This criterion guides the control system towards minimizing the discrepancy between the intended path and the robot's actual trajectory, ensuring accurate navigation while considering dynamic constraints and environmental factors.

In recent applications, our model has been instrumental in analyzing the stability and maneuverability of mobile robots in dynamic environments, particularly in scenarios involving obstacle avoidance and path planning. By incorporating real-world parameters and environmental factors into the simulation framework, we have gained valuable insights into the robot's behavior under varying conditions.

Furthermore, the model serves as a fundamental tool for developing and validating control algorithms aimed at enhancing the robot's performance in real-world settings. By integrating feedback control mechanisms and sensor data processing into the simulation environment, we can assess the effectiveness of different control strategies in achieving desired objectives, such as trajectory tracking and goal attainment.

7.4.2 PRM algorithm

Our research chapter proposes an innovative autonomous mobile robot navigation model that consists of three key stages: path planning, control, and simulation. The purpose of this model is to provide an effective and reliable navigation system for mobile robots, enabling them to navigate complex environments and perform tasks autonomously. To achieve this goal, the proposed model includes path planning through the mobile Robot PRM algorithm, control through the Pure Pursuit controller block, and simulation through the 'Differential Drive Kinematic' block in MATLAB/Simulink. By combining these three stages, the model enables the robot to generate an optimal path and follow it while adapting to the environment in real-time. In autonomous mobile robot navigation, accurate localization is crucial for determining the robot's position in a known environment. While odometry can be imprecise, sensor data from lasers, cameras, and ultrasonic sensors can calculate the robot's position relative to a map of its.

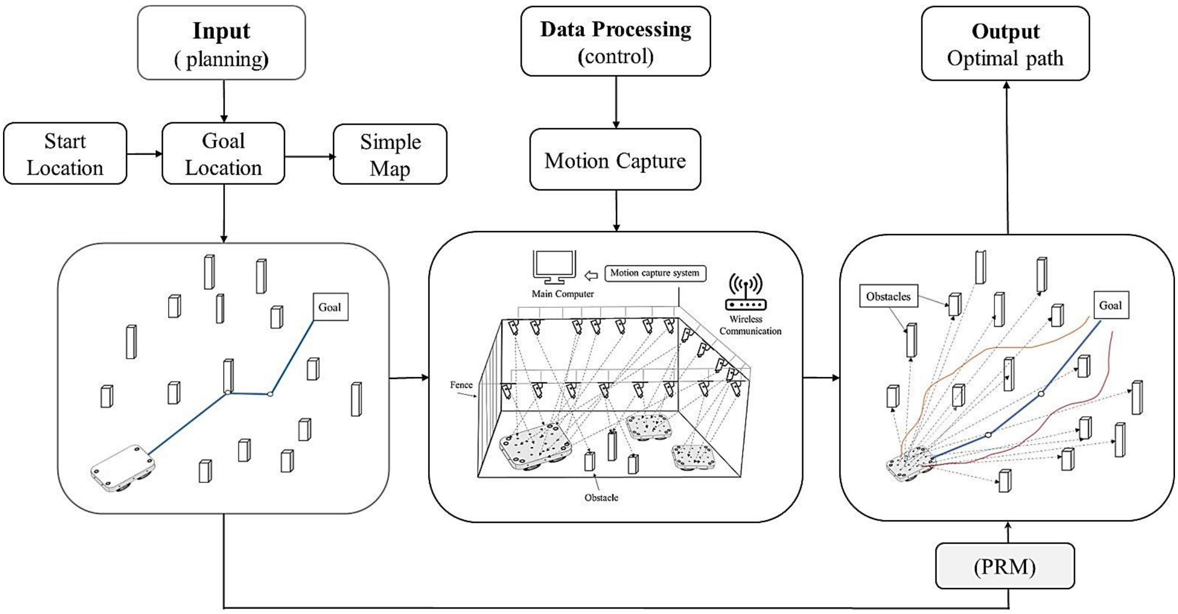


Figure 53: Mobile robot platform with motion capture system.

surroundings [284]. This information is then used to update the robot's belief and position, involving prediction, action updates, perception, measurement, and correction updates.

Path planning for mobile robots often relies on the Probabilistic Road Map (PRM) method, a technique involving the construction of a roadmap through random node placement in the free configuration space [285]. While the choice of the PRM method may not be the most common planning approach, it offers several advantages that make it suitable for our specific application. The PRM process comprises a learning (preprocessing) stage and an online query stage. During the learning stage, numerous robot poses are randomly sampled, and connections are established between neighboring nodes to construct the roadmap. This construction process is detailed by Algorithm 6, wherein nodes are iteratively sampled and connected based on specific criteria. The algorithm effectively addresses critical issues concerning node sampling, local planning, and collision avoidance, thereby ensuring the generation of feasible paths across various environments. Additionally, the nature of the PRM algorithm determines its suitability for either static environments or those with slight random variation.

Algorithm 6: Probabilistic Roadmap (PRM) Construction

1: V←$\emptyset$
2: E←$\emptyset$
3: loop
4: q ← a randomly chosen free configuration
5: Vq← a set of candidate neighbors of q chosen from V
6: V← V{q}
7: for q' ∈ Vq, in order of increasing, (q, q'), do
8: $if\ E$ =same– connected – component $(q, q') <$ | (q, q')| then
9: E ← E ∪ {(q, q')}
10: update E’s connected components

The path planning algorithm comprises a series of crucial steps, beginning with initializing two collections in steps 1-2. For each new node q, the algorithm selects a set of neighboring nodes V and uses a local planner to plan the path in steps 5-8. The boundary (q, q) of the feasible path is then added to the set, and the collision path is removed in steps 9-10. The algorithm addresses four critical issues during the process. Firstly, step 4 ensures that the randomly sampled point is within the free space and does not collide with obstacles. Secondly, step 8 involves constructing a regional planner to connect two different sampling points, ensuring planner certainty and rapidity while discretizing the local path and performing collision avoidance checks. Finally, step 5 outlines the rules used to select neighborhood points within a certain distance range. By addressing these issues, the path planning algorithm can effectively generate feasible paths for a given robot in any given environment.

The PRM algorithm is theoretically grounded in probabilistic robotics. By leveraging the principles of randomness and statistical sampling, the PRM algorithm can efficiently generate a roadmap of potential configurations, allowing for effective path planning in complex, high-dimensional configuration spaces. While the PRM algorithm may not guarantee global optimality, it strikes a balance between optimality and computational efficiency, making it well-suited for our navigation model. Furthermore, the PRM method's ability to find feasible paths, even in environments with dynamic obstacles, aligns with our goal of enabling mobile robots to navigate complex environments autonomously.

During the query stage, Algorithm 7 is employed to select a path between the start and target positions using the constructed path network graph. This involves iteratively selecting and evaluating neighboring nodes to find the most efficient path. The resulting global path connects the start and target positions, and further optimization techniques can be applied to find the shortest path. This process utilizes the undirected path network graph G = (V, E) constructed during the offline learning phase. By connecting the starting position (qs) and target position (qg) to two points (q1, q2) in the graph, the algorithm finds the path between q1 and q2. A smoothing method can then be applied to find the shortest path. In summary, the query stage of the path planning algorithm leverages the previously constructed path network graph to identify the most efficient path between the start and end positions.

Algorithm 7: PRM-Based Optimal Path Query using Graph Search

```
Open←∅;
Closed←∅;
 Open← qs;
while Open =∅
    q0← arg min f(x).
     REMOVED (Closed, q0);
 (Closed, q0 );
      if q0 == qg then
           return Reconstruct PATH (qs , qg , Closed);
     S←NEIGHBOURS (q0 );
  for all q′ ∈ S do
        g←g(P′ );
        h←D (q0, P′ );
          f← g + h.
                if q′ ∈ Open ∧ f (q0) < f(q′ ) then
              return
           elseif q′ ∈ closed ∧ f (q0) < f(q′ ) then
           return
    open←q;
```

Furthermore, our research extends beyond traditional path planning methodologies by incorporating the PRM algorithm into a comprehensive navigation model that includes path planning, control, and simulation stages. This integrated approach allows for seamless interaction between path planning algorithms and control mechanisms, resulting in robust and adaptive navigation behavior in real-world scenarios. Additionally, the application of the PRM algorithm within our navigation model opens avenues for further research and development in autonomous mobile robot navigation, particularly in the areas of optimization, real-time adaptation, and scalability.

7.4.3 Proposed path planning system

The proposed path planning system for a mobile robot, depicted in Fig. 53, encompasses a network of interconnected components aimed at generating an optimal navigation path (Fig. 54). User-provided start and goal coordinates serve as input for the system. During the data processing phase, a motion capture system continuously tracks the robot's position and orientation, while range sensors gather environmental data. Subsequently, the system constructs a map of the environment through binary occupancy mapping and generates a probabilistic roadmap using a mobile robot PRM algorithm.

The PRM algorithm creates a graph of the environment, where each node represents a potential location for the robot, and each edge represents a feasible path between two nodes. The algorithm uses a probabilistic approach to sample random configurations of the robot and connect them to the graph. Ensuring that the generated graph covers all possible paths.

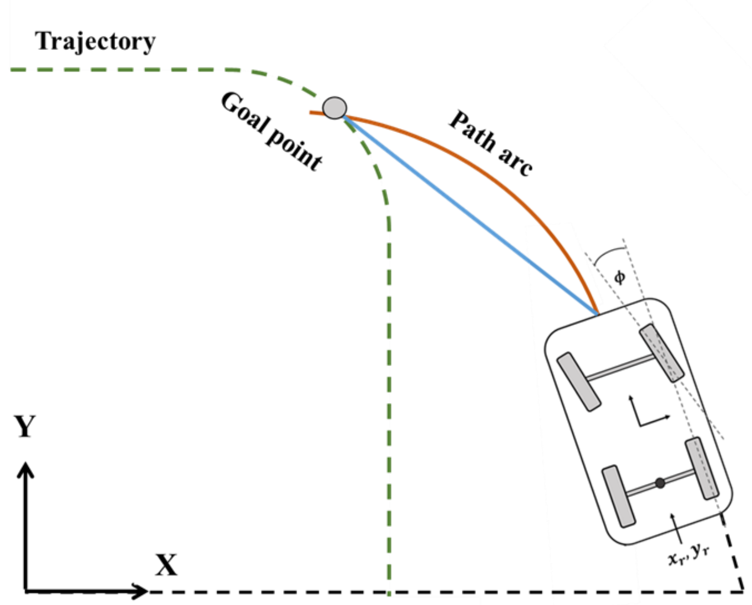


Figure 54: Geometric representation of mobile robot trajectory planning and curvature-constrained motion. The figure illustrates the transition from a reference trajectory toward a goal configuration through a continuous path segment, incorporating a curvature-based arc to satisfy nonholonomic kinematic constraints. The robot pose $(\boldsymbol{x_r}, \boldsymbol{y_r})$and orientation $\boldsymbol{\theta}$are shown, highlighting the relationship between trajectory tracking, steering dynamics, and path smoothness within a global Cartesian coordinate frame.

Our system leverages camera-based motion capture to monitor the movements of mobile robots. This involves the real-time transmission of position data to MATLAB/Simulink via an Ethernet port, followed by processing through control logic blocks. Subsequently, the data is transmitted to the robot utilizing QUARC-Real Time. Our ground control station boasts robust hardware, featuring an Intel Core i7 processor, 16GB of DDR4 RAM, and essential software tools such as MATLAB 2020a and QUARC Real-Time Control Software 2020. The station is equipped with three monitors, a USB flight controller joystick, and a dual-band WiFi router. The MoCap testing setup includes 16 Vicon cameras that establish a reference coordinate system for the mobile robots and map the working space. At least four cameras are required to accurately locate and track objects. The Vicon system provides accurate and reliable data for movement analysis applications. The mobile robot's position is tracked in a 3D space, but since it operates on a 2D plane, the Z-directional data and angles for roll and pitch are irrelevant. Calibration of the cameras for synchronization and detection is done using a calibration wand after establishing communication between the Vicon system and Simulink through Ethernet.

The output of our system is the optimal path for the robot to follow, considering the environment and obstacles that the robot may encounter along its path. This path can be used to guide the robot towards its goal, while avoiding any obstacles that may be present.

In this section, we conduct a comparative analysis of our proposed path planning system with existing approaches in the literature. Our evaluation encompasses various aspects of performance, including computational efficiency, path quality, real-time performance, and robustness. Methodologically, we systematically compare different path planning methods using predefined metrics to assess their respective strengths and weaknesses. The results of our analysis reveal several key findings. Firstly, our proposed system demonstrates comparable computational efficiency to traditional methods, with slight improvements in processing time attributed to the integration of motion capture technology. Secondly, the paths generated by our system exhibit superior quality compared to conventional approaches, characterized by smoother trajectories and better avoidance of obstacles. Thirdly, our system excels in real-time performance, swiftly adapting to changes in the environment while maintaining smooth and safe navigation. Lastly, the robustness of our system is evident in its ability to handle uncertainties and disturbances effectively, resulting in consistent and reliable performance. These findings underscore the advantages of integrating motion capture technology and telecommunication capabilities into path planning for mobile robots, positioning our proposed approach as a significant advancement in mobile robot path planning technology.

7.4.4 Experimental results and evaluation

This chapter presents a novel control system for a robot that utilizes a subsequent path controller to generate control signals, thereby enabling the robot to traverse along the intended path. The control system incorporates a goal radius, which serves as a measure of the desired proximity between the robot's final position and the goal location. The robot is programmed to stop its movement once it reaches this goal radius. Additionally, the control system continuously computes the distance between the robot's present location and the goal location, and if this distance exceeds the goal radius, the robot is stopped to avoid any unexpected behavior close to the goal. The proposed approach has been implemented and extensively tested using a mobile robot and in simulation runs.

During experimentation, the robot demonstrated impressive velocities, reaching up to 60 radians per second for the left wheel and 50 radians per second for the right wheel. This high-speed performance, coupled with short time intervals, significantly contributed to the robot's rapid achievement of its intended goal. Through extensive simulations, we closely monitored the robot's trajectory under various conditions, facilitating a comprehensive evaluation of its performance. Subsequently, we compared the experimental results against simulations to assess the improvement in the robot's traced path, depicted in Fig. 55, with a time span of 1 second and 0.03-second timesteps. The use of this combination of high speed and short time intervals significantly contributed to the robot's rapid achievement of its intended goal.

The robot's trajectory was closely monitored through simulations, considering variations in its path when placed in different starting locations and assuming negligible wheel slippage.

This comprehensive monitoring facilitated a thorough evaluation of the robot's performance. Subsequently, the results obtained from the experiments were compared against simulations to assess the improvement in the robot's traced path.

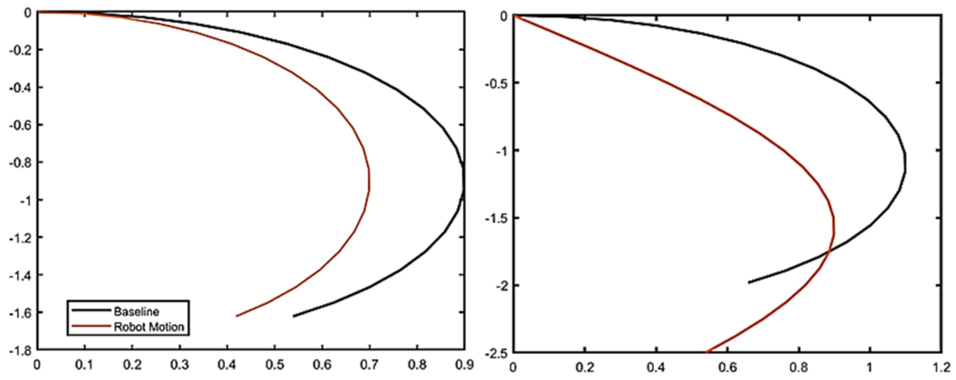


Figure 55: Simulation Result of the mobile robot motion that moving with different wheel speed.

But if the time was set to 1 second with 0.05-second timesteps. The results showed that the optimal wheel speeds were 50 PRM for both the left and right wheels; at time t=0, the left wheel had a speed of 60 PRM while the right wheel had a speed of 50 PRM, with the robot at the origin. At time t=0.10, the optimal left and right wheel speeds were 50 and 40 PRM, respectively, with the robot at a position (2.61, 4.16) meters away from the origin. The optimal speeds for the left and right wheels decreased as time elapsed, with the robot moving closer to the desired trajectory. By time t=1.00, the optimal left and right wheel speeds had converged to 30 PRM, with the robot at the final position (10.00, 0.00) meters away from the origin. Overall, these experiments demonstrated the effectiveness of optimizing wheel speeds to achieve precise trajectory tracking.

As a second experiment, the optimal left and right wheel speeds for a mobile robot to track a desired trajectory generated by simulating the robot's motion using a kinematic model and solving an optimization problem to identify the wheel speeds that minimized the error between the robot's actual trajectory and the desired trajectory. The experiment displayed the optimal wheel speeds at various time points along the trajectory, in addition to the corresponding x and y positions of the robot. This information could be utilized to control the robot's motion and achieve precise trajectory tracking.

The demonstration provided in Table 16 showcases the utilization of numerical methods to simulate the kinematics of a mobile robot. By specifying the input wheel speeds and the initial state of the robot, it becomes possible to compute the position and orientation of the robot at different time steps. This method is foundational in robotics for effectively controlling the motion of mobile robots. Notably, the solver integrates the robot's kinematic equations, enabling accurate modeling of the robot's motion even under non-uniform input speeds.

In this chapter, a novel approach for generating smooth paths for robot navigation is presented, integrating motion capture technology to enhance efficiency and accuracy.

The proposed method utilizes spline interpolation, a mathematical technique commonly utilized in motion capture, to generate continuous paths that circumvent sharp corners and abrupt changes in direction.

The resulting smooth path, visually represented by a red line in the generated plot (refer to

Fig. 56), effectively traverses all specified waypoints. While spline interpolation is well-established in motion capture, its application in robot navigation is innovative.

The smooth paths generated by this algorithm offer a simple and effective solution for robot navigation, particularly beneficial in various robotic applications, especially those involving motion capture.

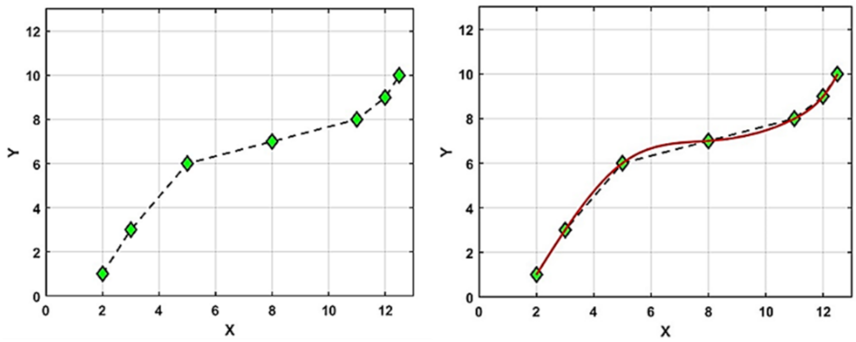


Figure 56: Path Following for a Differential Drive Robot and the Smooth Path for Robot Navigation.

The experiment conducted utilized a differential drive mobile robot and developed a control algorithm for trajectory control, incorporating motion capture technology. we conducted experiments to optimize wheel speeds for precise trajectory tracking, showcasing the effectiveness of numerical methods in simulating the kinematics of mobile robots. Furthermore, we presented a novel approach for generating smooth paths for robot navigation, integrating motion capture technology to enhance efficiency and accuracy.

By utilizing spline interpolation, we generated continuous paths that circumvent sharp corners and abrupt changes in direction, offering a simple and effective solution for robot navigation. The control algorithm was implemented and evaluated through numerical simulations using MATLAB/Simulink.

A nonlinear model, commonly employed in motion capture, was utilized to test the algorithm on five different paths, including straight lines with varying angles based on the desired path and goal (Fig. 57 illustrates these paths).

The model integrates inputs of left and right wheel speeds, as well as linear and angular velocities, to determine the robot's heading. The evaluation results demonstrate the effectiveness of the trajectory control algorithm in accurately and efficiently guiding the robot along the desired path, especially when used in conjunction with motion capture technology.

Table 16 : Comparing Wheel Speeds for a Differential Drive Robot.

Time (s)	Wheel Speed 1 (PRM)	Wheel Speed 2 (PRM)	Wheel Speed 3 (PRM)	Wheel Speed 4 (PRM)
0.00	0.0000	0.0000	0.0000	0.0000
0.05	2.9317	2.4287	1.9256	1.4225
0.10	5.7949	4.7939	3.7928	2.7917
0.15	8.6052	7.1075	5.6098	4.1116
0.20	11.3645	9.3993	7.3351	5.2709
0.25	14.0741	11.6572	8.9447	6.2323
0.30	16.7346	13.8700	10.4295	7.0329
0.35	19.3467	16.0267	11.7736	7.7119
0.40	21.9108	18.1163	12.9639	8.3033
0.45	24.4275	20.1277	13.9902	8.8356
0.50	26.8975	22.0502	14.8430	9.3326
0.55	29.3214	23.8725	15.5163	9.8130
0.60	31.6999	25.5838	16.0077	10.2917
0.65	34.0336	27.1735	16.3171	10.7795
0.70	36.3232	28.6311	16.4474	11.2836
0.75	38.5693	29.9463	16.4040	11.8083
0.80	40.7726	31.1090	16.1955	12.3557
0.85	42.9338	32.1092	15.8331	12.9276
0.90	45.0537	32.9371	15.3306	13.5251
0.95	47.1329	33.5830	14.7045	13.9762

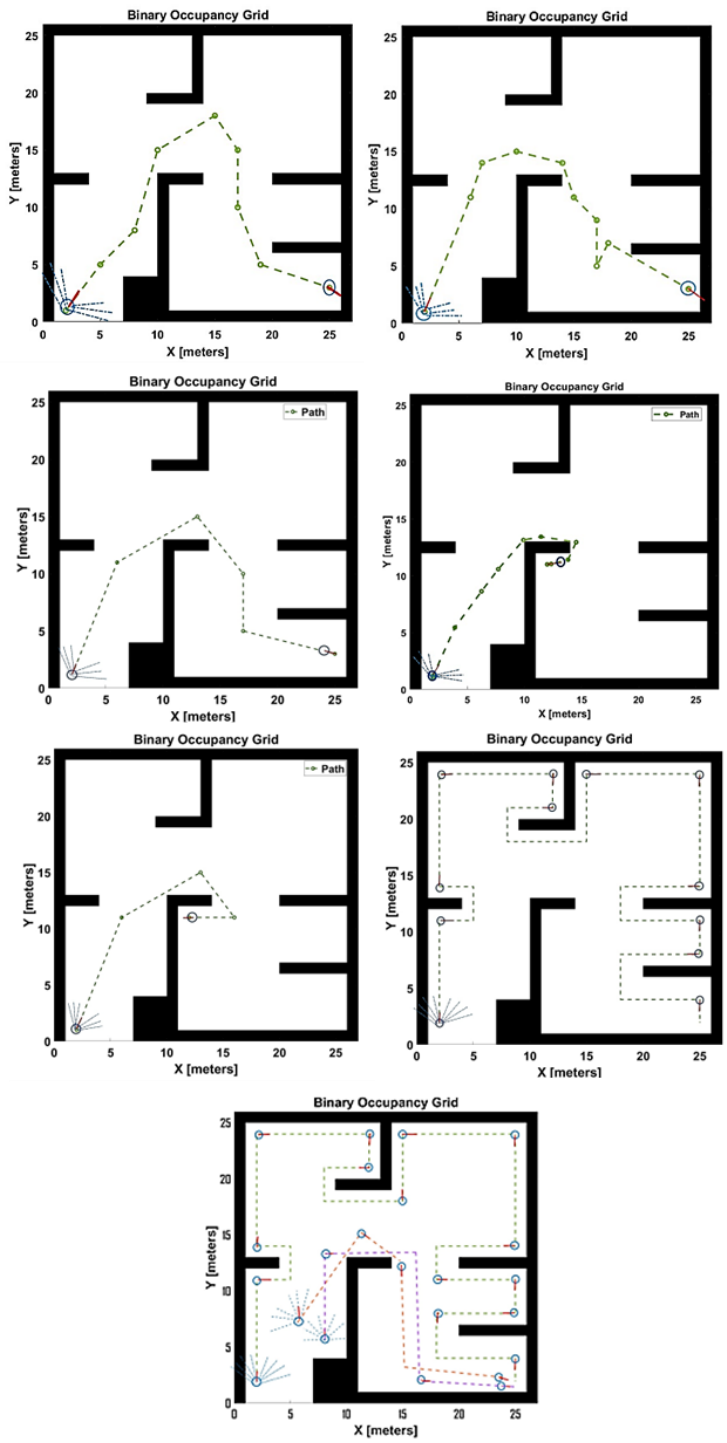


Figure 57: Simulation Result of the mobile robot Path plan. illustrating trajectory generation, obstacle avoidance behavior, and navigation performance in a complex binary occupancy grid environment across different scenarios.

7.5 Conclusion

This chapter proposes a novel technique to enhance the safety of mobile robots (MRs) by integrating probabilistic methods into the path planning and localization components. The impact of incorporating motion capture technology into the path-planning algorithm of MRs on their trajectory accuracy is investigated. The results demonstrate that the utilization of motion capture technology provides real-time position and orientation data critical in determining the optimal wheel speed for the robot to achieve optimized performance, resulting in a significant positive impact on trajectory accuracy. Various wheel speed calculation techniques are evaluated through simulations and experiments, with a feedback control system producing the best results, achieving a trajectory that closely follows the desired path. This study contributes to the development of more efficient and accurate path-planning algorithms for mobile robots, with potential applications in various fields such as industrial automation, healthcare, and logistics. Further research could investigate the integration of artificial intelligence techniques such as machine learning and deep learning into the path-planning algorithm of mobile robots, as well as exploring the potential of edge computing to enable autonomous operation in challenging environments with limited computational resources.

7.6 Scientific Thesis III.1

This dissertation demonstrates that the integration of probabilistic modeling with motion capture–enhanced path planning and localization significantly improves the safety and accuracy of mobile robot navigation. By incorporating precise position and orientation feedback into the trajectory generation process and combining it with adaptive control strategies, the proposed approach enables more accurate velocity optimization, reduces trajectory deviation, and ensures reliable path tracking. The results confirm that the framework achieves superior navigation precision and robustness compared with conventional path planning methods, particularly in dynamic and uncertain operational environments.

Chapter 8

Hierarchical Conditional Variational Autoencoder for Mobile Robot Path Planning

8.1 Chapter abstract

Efficient and accurate path planning for mobile robots in complex environments is a crucial aspect of enabling intelligent systems such as autonomous vehicles and wheeled mobile robots. However, the inherent challenges posed by uncertain and evolving landscapes present a critical problem, necessitating innovative solutions that transcend the limitations of traditional planning methods. This chapter introduces an avant-garde approach that elevates optimal path planning through the integration of a Hierarchical Conditional Variational Autoencoder (HCVAE). By harnessing the hierarchical architecture of HCVAE, we transcend traditional limitations by endowing our method with unparalleled capabilities in generating non-uniform sampling distributions. This innovation significantly enhances planning efficiency and success rates, all while staying grounded in the fundamental principles of conventional methods. Furthermore, we introduce pioneering improvements aimed at reducing map complexity and accelerating execution, thus enhancing adaptability to unique operational scenarios. We support our assertions with robust empirical evidence derived from a thorough and rigorous set of experiments, highlighting the significant advancements made in the field of robotics and autonomous systems. Our framework based on HCVAE marks a groundbreaking advancement, ushering in an era of unparalleled efficiency and effectiveness in mobile robot path planning.

8.2 Introduction

Motion planning is a crucial element in solving manipulation planning problems for robots, encompassing a wide range from uncomplicated mobile robots to intricate high-dimensional robotic systems, such as those featuring dual-arm manipulators with human-like hands. The primary objective of motion planning revolves around computing paths that are free from collisions, navigating from a starting point to a goal state within the configuration space. In the preceding decades, numerous approaches rooted in sampling-based methods have been put forward to address this challenge.

Numerous planning methodologies rooted in sampling-based strategies have been introduced, exemplified by the Rapidly Exploring Random Tree (RRT) [286], as well as its diverse adaptations including the bidirectional RRT [287], and the optimization-centric iteration denoted as RRT* [288]. While these planning algorithms exhibit commendable efficacy within low-dimensional spaces, their suitability diminishes within higher-dimensional contexts, often leading to impractical planning durations in certain application scenarios. In response to this intricate challenge, novel avenues have emerged in the form of motion planning approaches underpinned by deep learning methodologies, representing a recent and promising development.

Sampling-based motion planning (SBMP) is a successful approach for solving complex motion planning problems. It relies on sampling feasible states and using a collision checker to explore connections. However, traditional sampling covers the entire state space uniformly, which is overkill for many robots that operate in specific, complex subsets of the space due to environment, dynamics, or implicit constraints. This challenges SBMP's performance, as it struggles to efficiently place samples in these crucial areas. Thus, while SBMP's implicit state space representation is versatile, it struggles to harness domain knowledge or previous experiences for faster solutions.

In this study, we address this challenge by directing the sampling of the state space towards promising areas using learned sample distributions .Our approach is centered on a conditional variational autoencoder (CVAE) [289], which learns intricate patterns and their surroundings from successful motion plans and past robot experiences, rather than relying on predefined rules. The CVAE's latent space allows us to sample and map out the operational regions of the robot system, considering specific details like the starting point, goal, and obstacles in the workspace. This way, we combine the advantages of sampling-based motion planning with the adaptability and effectiveness of parameterized learning techniques. Notably, this approach is adaptable to various systems and problem-specific information.

Prior research has introduced non-uniform sampling strategies within the context of SBMP, yielding discernible enhancements in performance [290]. A recent contribution by Gammell et al. pertinently situates clusters of samples based on information accrued during the planning process [291]. Alternative approaches incorporate workspace models to introduce sample bias through decomposition methodologies [292] [293]or via approximations of the medial axis [294]. Notwithstanding their efficacy within their designated problem domains, the transferability of these strategies to novel environments remains uncertain [295], particularly in cases necessitating comprehensive state space sampling (e.g., encompassing velocity). Notably, these methodologies often depend heavily on domain experts' insights, in contradistinction to our approach, which derives judicious non-uniform distributions solely from demonstrative instances.

Several paradigms in adaptive and biased sampling have embraced online learning mechanisms to enhance generalizability. Burns and Brock [293] strive to optimize configuration space sampling via utility functions. Zucker et al. [296] harness reinforcement learning to inform discretized workspace sampling schemes. Berenson et al. [297]curate and rehabilitate paths through learned heuristic constructs. In similar vein, Coleman et al. [298] encapsulate experiential data within a graph structure, while Li and Kostas [299] establish erudition of cost-to-go metrics tailored to kinodynamic systems. Baldwin and Newman [300] leverage semantic cues to learn distributions, while Ye and Alterovitz [301] amalgamate human-demonstrated actions and SBMP for plan generation. These methodologies, however, grapple with limited generalization capacity across diverse problem domains or the amalgamation of comprehensive state representations and extrinsic informational factors

The application of representation learning, which is encompassed within the realms of deep learning and probabilistic graphical models [300], has witnessed a notable surge within the field of robotics, particularly during the preceding decade [301]. The principal thrust of this discipline resides in the extraction of salient features from unstructured data, with objectives ranging from dimensional reduction (termed as encoding) to the acquisition of pertinent features tailored for applications in supervised learning and reinforcement learning paradigms.

problem. In essence, the solution to this quandary involves the determination of the most concise unobstructed trajectory originating from an initial state and culminating within a designated goal region.

8.3 Related Work

In the domain of sampling-based motion planning (SMP), various biased-sampling heuristics have been proposed to accelerate the computational speed of Rapidly-exploring Random Trees (RRT) and its variants [302]. For instance, Rickert et al. [303] utilized gradient information to balance exploration and exploitation, while Urmson and Simmons [304] heuristically biased samples in RRT, and Ferguson and Stentz [305] introduced the anytime RRT algorithm using multiple independent RRTs. However, these methods lack asymptotic optimality due to the underlying RRT algorithm. RRT* [306], on the other hand, extends RRTs to ensure asymptotic optimality by incrementally rewiring the RRT graph connections, guaranteeing an asymptotically optimal shortest path [306]. Nonetheless, experiments in [307] [308] have indicated that RRT* exhibits slow convergence rates to optimal path solutions in higher-dimensional spaces.

To address this challenge, several existing biased/adaptive sampling methods have been proposed to accelerate the convergence rate of SMPs towards optimal or near-optimal path solutions. One such method is the Informed-RRT* algorithm proposed by Gammell et al. [309], which utilizes an initial solution from RRT* to define an ellipsoidal region from which new samples are drawn to minimize the initial solution for a given cost function. While Informed-RRT* has demonstrated improved convergence towards an optimal solution, it may suffer when finding an initial path solution consumes most of the computation time. To mitigate this issue, Gammell et al. introduced Batch Informed Trees (BIT*) [310], an incremental graph search technique where an ellipsoidal subset containing configurations to update the graph is incrementally enlarged. Empirical results have shown that BIT* outperforms prior methods such as RRT* and Informed-RRT*.

However, confining a graph search to an ellipsoidal region may slow down the algorithm's performance in maze-like scenarios, especially when the start and goal configurations are close to each other, but the path among them traverses a complicated maze stretching waypoints far from the goal. Moreover, such methods may not be applicable to non-stationary or unseen environments.

Another approach involves learning-based search methods, where learning techniques are employed to enhance classical SMPs computationally. The Lightning Framework [311], for instance, stores paths in a lookup table and utilizes learned heuristics to generate new paths and repair old ones. Similarly, Coleman et al. [312] proposed an experience-based strategy to cache experiences in a graph instead of individual trajectories. While these approaches demonstrate superior performance in higher-dimensional spaces compared to conventional planning methods, lookup tables may suffer from memory inefficiency and poor generalization to new planning problems. Zucker et al. proposed a reinforcement learning-based method to bias samples in discretized workspaces. However, reinforcement learning-based approaches are known for their slow convergence, as they require many interactions.

In our contribution, we advocate for the utilization of recent advancements in representation learning, leveraging their ability to encapsulate intricate high-dimensional conditional distributions. This allows for the seamless incorporation of available state and problem-specific information. The primary objective of this endeavor is to generate sample distributions that enhance the efficacy of sampling-based motion planning algorithms, particularly in addressing the complexities of optimal motion planning.

8.4 Methods and Experiments

8.4.1 Path Planning with HCVAE Integration and Optimization

The proposed path planning framework is built upon a Hierarchical Conditional Variational Autoencoder (HCVAE), which models the relationship between environment maps and robot states. As illustrated in Figure 58**,** the overall architecture is organized into three main components: (i) Input Processing, (ii) HCVAE Model, and (iii) Path Planning Module.

The Input Processing module prepares the raw inputs, including environment maps (X) and robot states (S), through normalization and feature extraction, ensuring a consistent and informative representation for subsequent learning.

The HCVAE model consists of two principal components: an encoder and a decoder. The encoder maps the data into a latent space representation, capturing the underlying spatial and dynamic dependencies between the environment and the robot state. This latent space, parameterized by (μ, σ), enables probabilistic modeling of the input distribution. The decoder subsequently reconstructs the data and generates future predictions by sampling from the latent space.

This hierarchical conditional formulation allows the model to learn compact and meaningful representations, facilitating accurate reconstruction of environment maps and prediction of future robot states. Consequently, the HCVAE serves as the core representation learning component within the proposed framework, enabling efficient encoding and decoding of complex environmental and state information for path planning tasks.

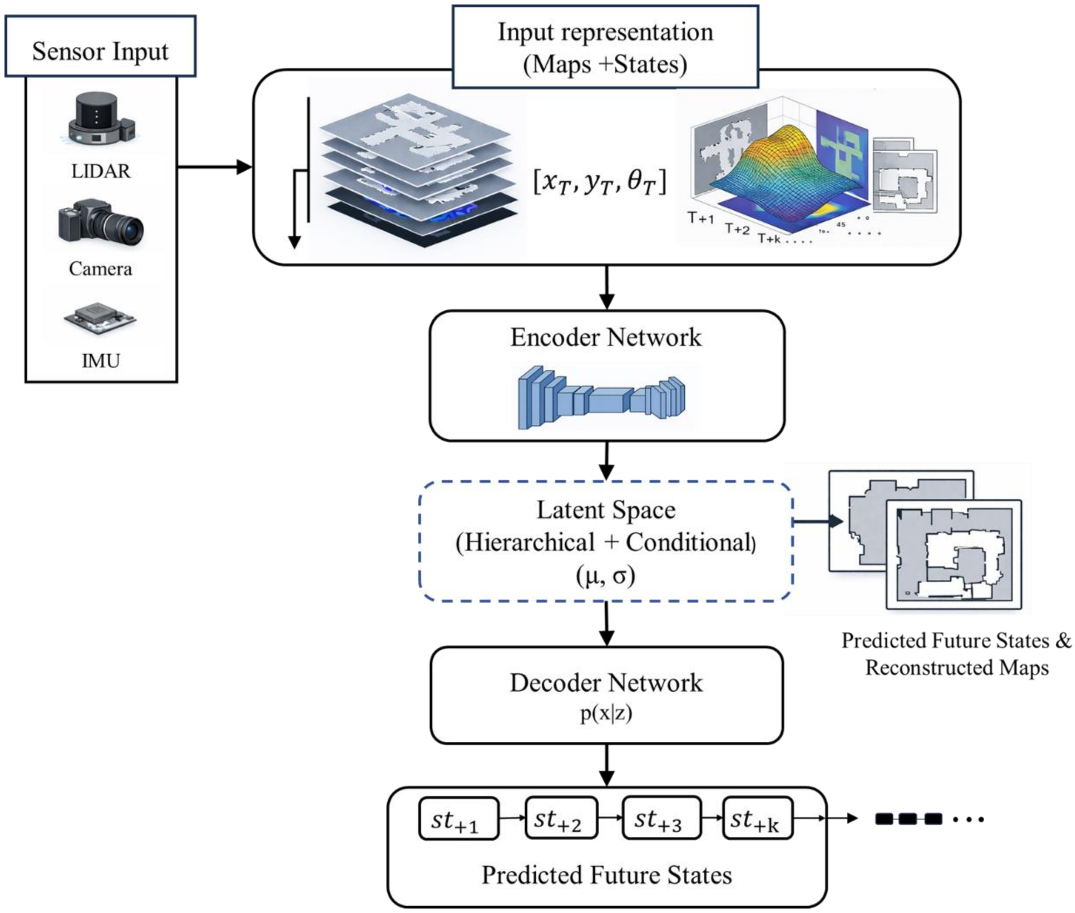


Figure 58: Hierarchical Conditional Variational Autoencoder (HCVAE) Architecture

The HCVAE Model learns the latent representation of environment maps and robot states through its encoder ($q_\phi(z|x)$,S)) and decoder ($p\theta(x|z)$,S)). Here, the encoder encodes the input data into a latent space representation (z), while the decoder reconstructs the input data from this latent representation. The model's optimization process revolves around maximizing the Evidence Lower Bound (ELBO), balancing reconstruction accuracy and the Kullback-Leibler divergence between the learned and prior distributions over the latent variable (z) [313] [314]. This rigorous optimization ensures that the HCVAE effectively captures the underlying structure of the input data, facilitating robust path planning capabilities within our framework.

$$\mathcal{L}(\theta_9\phi_9 x) = E_{q\emptyset(z|x,S)}[\log p\theta(x|z,S) - KL\big(q_\phi(z|x,S)||\,p(z)\big) \quad (65)$$

This equation represents the ELBO, which is the objective function optimized during training. Additionally, we consider the distribution of the data x given a latent variable z, as generated by the decoder $p\theta(x|z)$ with parameters θ, and the prior distribution $p(z)$:

$$p\theta(x) = \int_z p\theta(x|z)p(z)\,dz \quad (66)$$

The path planning module utilizes the learned latent representations to generate optimal paths, employing algorithms such as RRT* (Rapidly-exploring Random Trees*) to navigate the robot while avoiding obstacles. RRT* incrementally builds a tree structure, biasing exploration towards unexplored areas while maintaining asymptotic optimality, resulting in a near-optimal path trajectory as a sequence of waypoints. Training the HCVAE model involves gathering diverse datasets of environment maps X and corresponding optimal paths P for parameter optimization using algorithms like stochastic gradient descent or the Adam optimizer, followed by validation using cross-validation techniques to ensure robustness and effectiveness. Integration with the path planning algorithm involves developing software interfaces to facilitate seamless communication between the HCVAE and the path planning module, optimizing computational efficiency using techniques like caching mechanisms and parallel processing to enable real-time path planning in dynamic environments, ensuring the framework's operation efficiency in diverse scenarios.

8.4.2 Non-uniform Sampling Distribution Generation and Optimization

Non-uniform Sampling Distribution Generation and Optimization In this stage, algorithms within the HCVAE are implemented to generate non-uniform sampling distributions based on learned environmental features and robot state. Leveraging the latent representations learned by the HCVAE, the framework dynamically adjusts distribution parameters to adapt to changes in the environment or robot behavior. This adaptability is crucial for navigating complex and evolving landscapes effectively. Through sophisticated algorithms, the HCVAE generates sampling distributions that guide the path planning process towards promising regions of the configuration space, facilitating efficient exploration and exploitation [315].

The process of generating non-uniform sampling distributions can be described mathematically as follows:

conditional distribution with parameter $\emptyset \in \Phi$

$q_{\phi}(z|x)$=conditional distribution with parameter ϕ∈Φ

$$\boldsymbol{kL\big(q(z)||p(z) = \int q(z)\, log\, q(z) - log\, P(z)\big)dz} \tag{67}$$

Here $q_{\phi}(z|x)$, represents the conditional distribution used by the encoder of the HCVAE, while $kL(q(z)||p(z))$ denotes the Kullback-Leibler divergence between the encoder distribution $q(z)$ and the prior distribution $p(z)$. Additionally, considering the posterior density $p_{\phi}(z|x)$proportional to the product of the prior density $p_{\phi}(z)$ and the likelihood $p_{\phi}(z|x)$, the evidence lower bound (ELBO) is defined as:

$$\boldsymbol{\mathcal{L}(\theta_9\phi_9 x) = -\,log\, p\theta(x) + KL\left(q_{\phi}(z|x)_{p\theta}(z|x)\right)} \tag{68}$$

Efforts are dedicated to optimizing the computational efficiency of the framework to enable real-time path planning in dynamic environments [316]. Techniques such as caching mechanisms, parallel processing, and performance profiling are implemented to minimize latency and resource usage while maintaining accuracy. Caching mechanisms are employed to store and reuse previously computed results, reducing redundant computations and speeding up the overall planning process. Parallel processing techniques leverage the computational power of multi-core processors or distributed computing systems to perform tasks concurrently, further enhancing the framework's efficiency. Performance profiling allows for the identification of computational bottlenecks and resource-intensive operations, enabling targeted optimizations to improve overall efficiency. The framework's performance is meticulously fine-tuned to strike a balance between computational resources utilization, planning accuracy, and adaptability to different operational scenarios, ensuring optimal performance in diverse real-world environments.

In our experimental evaluation, we thoroughly examine the performance of our HCVAE-based path planning framework across simulated and real-world environments. These experiments are crucial for understanding how well our solution works in different scenarios and under varying conditions. Simulated environments offer us the advantage of controlled testing, where we can manipulate factors like terrain complexity and obstacle placement. By running simulations, we can observe how our framework responds to specific challenges and assess its capabilities under ideal conditions. However, real-world experiments provide the ultimate test for our framework. Deploying our solution on actual mobile robots in diverse environments, such as urban streets or factory floors, allows us to confront the unpredictability and complexity of real-world scenarios head-on. These experiments enable us to validate the robustness and adaptability of our approach in practical settings. Throughout our evaluation, we measure a range of metrics, including planning efficiency, success rates, and resource utilization. Analyzing these metrics provides us with valuable insights into the strengths and limitations of our framework. It helps us understand where our solution excels and where improvements may be needed to enhance its performance further.

By combining results from both simulated and real-world experiments, we gain a comprehensive understanding of the capabilities of our HCVAE-based approach. These insights not only validate the effectiveness of our solution but also guide future development efforts to address the evolving challenges in mobile robot path planning.

8.5 Results Analysis and Validation

The empirical results obtained from our experiments demonstrate the performance of the proposed HCVAE-based path planning framework. A comparative analysis is conducted to evaluate its performance against the baseline RRT* algorithm across multiple metrics. All results are averaged over multiple independent simulation runs to ensure consistency and reliability. The results indicate that the integration of HCVAE improves planning efficiency, success rate, and trajectory quality, particularly in complex and cluttered environments. These improvements are attributed to the ability of the proposed method to guide sampling toward more informative regions of the state space.

The quantitative performance comparison is summarized in Table 17.

Table 17: Performance Evaluation of HCVAE-based Path Planning Framework

Metric	RRT* (Baseline)	HCVAE + RRT* (Proposed)	Improvement (%)
Planning Success Rate (%)	80	95	+18.75%
Path Length (m)	12.5	10.2	−18.4%
Path Smoothness Score	3.5	4.1	+17.1%
Execution Time (s)	10.2	6.8	−33.3%
Memory Usage (MB)	120	80	−33.3%

The quantitative results in Table 17 demonstrate the advantage of the proposed HCVAE-enhanced framework over the baseline RRT* algorithm. The planning success rate increased from 80% to 95%, indicating improved reliability in complex environments. This corresponds to an improvement of approximately 18.75% in success rate. Additionally, the average path length was reduced from 12.5 m to 10.2 m, reflecting more efficient trajectory generation.

Path smoothness improved from 3.5 to 4.1, confirming the generation of smoother and more stable trajectories. Furthermore, the execution time decreased from 10.2 s to 6.8 s, representing a reduction of approximately 33.3% in computational time, highlighting enhanced computational efficiency. Memory usage was also reduced from 120 MB to 80 MB, demonstrating improved resource utilization. The observed improvements are consistent across repeated trials and demonstrate stable performance trends.

Figures 59 and 60 further illustrate these improvements through qualitative results in complex environments. In Figure 59, the generated paths demonstrate efficient navigation through intricate terrains, avoiding obstacles while minimizing path length. This highlights a clear advantage over traditional methods, which often struggle in highly cluttered environments.

Figure 60(a) presents the visualization of start and goal states alongside the map representation, providing a clear understanding of the planning context. This enhances interpretability and situational awareness for autonomous navigation in real-world scenarios.

Overall, these findings indicate that the proposed method effectively guides sampling toward more informative regions of the state space, resulting in faster convergence and higher-quality solutions, and demonstrate its strong potential for advanced mobile robot path planning applications. However, the evaluation is conducted under controlled simulation environments, and further validation in large-scale real-world scenarios remains an important direction for future work.

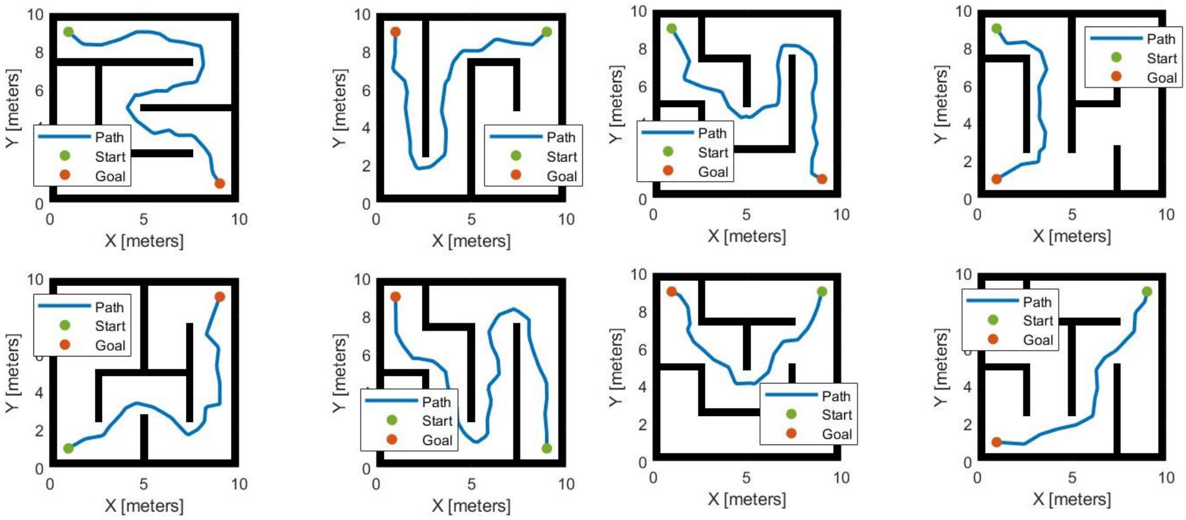
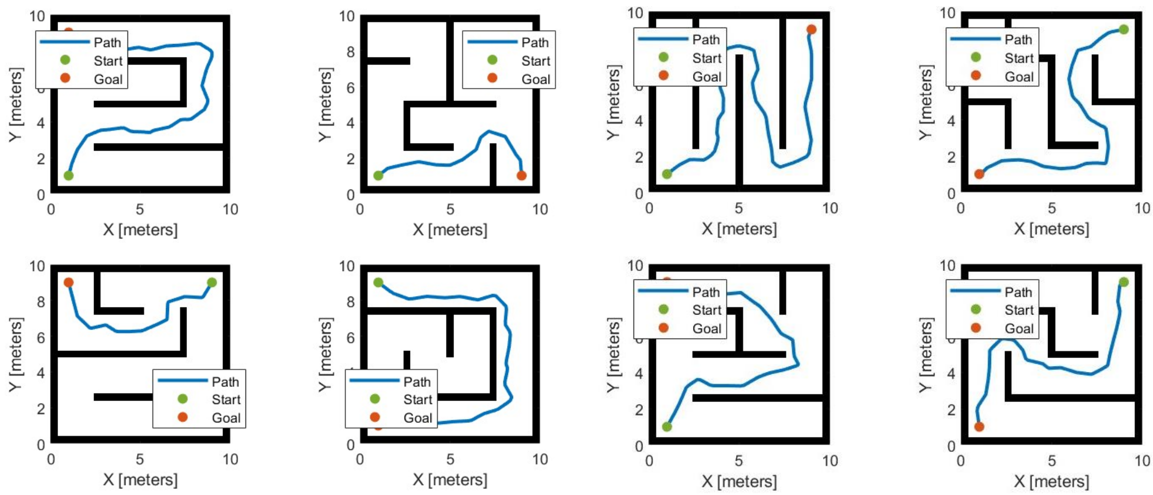


Figure 59: Representative map layouts and corresponding state distributions.



(a) (b)

Figure 60: Path planning results in complex environments. (a) Optimal path generation. (b) Visualization of start and goal state configurations.

Table 18: Ablation Study of HCVAE Integration

Method Configuration	Success Rate (%)	Path Length (m)	Execution Time (s)
RRT* (Baseline)	80	12.5	10.2
RRT* + Heuristic Bias	85	11.8	9.5
HCVAE (No Conditioning)	90	10.9	8.1
HCVAE + RRT* (Proposed)	95	10.2	6.8

Table 18 presents a comprehensive ablation study designed to quantify the individual and combined contributions of the proposed HCVAE-based framework components. All results are averaged over multiple independent simulation runs to ensure consistency and reliability. The results clearly demonstrate a consistent and progressive improvement in performance as more advanced mechanisms are incorporated into the planning pipeline. These trends remain consistent across repeated trials, confirming the stability and robustness of the proposed framework.

The baseline RRT* algorithm achieves a success rate of 80%, with relatively higher path length and execution time, reflecting the inherent limitations of purely sampling-based planning in complex environments. The introduction of heuristic bias improves the success rate to 85% while simultaneously reducing path length and execution time, corresponding to an improvement of approximately 6.25% in success rate, indicating a more guided and efficient exploration of the configuration space.

A more substantial performance gain is observed with the standalone HCVAE model, which increases the success rate to 90% and further reduces both path length and computational cost. This represents an overall improvement of approximately 12.5% compared to the baseline, highlights the effectiveness of learned latent representations in capturing the underlying structure of the environment and generating more informative sampling distributions.

The fully integrated HCVAE + RRT* configuration achieves the best overall performance, with a success rate of 95%, the shortest path length (10.2 m), and the lowest execution time (6.8 s). Compared to the baseline, this corresponds to a total improvement of approximately 18.75% in success rate and a 33.3% reduction in execution time. These results confirm that the synergy between learning-based sampling and classical planning algorithms significantly enhances both efficiency and robustness.

Overall, the ablation study demonstrates that each component contributes meaningfully to the system's performance, with the most notable improvements arising from the integration of deep generative modeling. The findings validate the proposed framework as a highly effective solution for path planning in complex and high-dimensional environments, offering superior adaptability and computational efficiency compared to traditional approaches. However, similar to the primary evaluation, further validation in real-world and large-scale environments is required to fully assess generalization capability.

8.6 Conclusion

This chapter presents a groundbreaking advancement in mobile robot path planning through the integration of HCVAE. Our framework offers unparalleled efficiency and effectiveness, revolutionizing the way mobile robots navigate complex environments. By combining innovative algorithms with robust empirical evidence, we demonstrate the significant advancements made in the field of robotics and autonomous systems.

Considering future prospects, our work sets the stage for exciting future research directions. We propose further enhancements to our framework to address remaining challenges and improve its performance in diverse operational scenarios. Additionally, we envision exploring new applications and extending the capabilities of HCVAE-based path planning to tackle emerging challenges in the field. Overall, our framework represents a transformative step towards realizing the full potential of intelligent mobile robot systems in real-world settings.

8.7 Scientific Thesis IV.1

This dissertation proves that the integration of a Hierarchical Conditional Variational Autoencoder (HCVAE) into mobile robot path planning establishes a fundamentally advanced, data-driven paradigm for trajectory generation in complex environments. By learning probabilistic representations of environment structure and robot state dynamics, the proposed model generates non-uniform, informative sampling distributions that effectively guide planning algorithms toward high-quality solutions. This approach significantly improves planning success rates, reduces computational complexity, enhances trajectory smoothness, and enables scalable and adaptive navigation performance beyond the capabilities of conventional sampling-based and heuristic path planning methods.

Chapter 9

Conclusion and Future Work

9.1 Summary of New Scientific Results

This dissertation establishes and validates a unified interaction-aware navigation framework for autonomous mobile robots operating in dynamic environments. This research addresses fundamental limitations of conventional navigation systems, where perception, path planning, speed control, and energy management are treated as independent modules, leading to suboptimal performance and limited adaptability.

To overcome these limitations This work introduces the Intelligent Obstacle Dynamics and Path Complexity System (IODPCS), a unified interaction-aware decision-making framework. The proposed architecture integrates artificial potential fields, dual-layer fuzzy logic control, reinforcement learning, and energy-aware optimization into a cohesive system capable of real-time adaptation. The framework explicitly models the interaction between the robot and its environment, enabling context-aware navigation decisions that improve safety, efficiency, and robustness.

Extensive simulation-based validation and comparative analysis demonstrate that the proposed framework significantly improves navigation performance in terms of trajectory smoothness, obstacle avoidance reliability, adaptability to dynamic environments, and energy efficiency. The results demonstrate that integrating learning-based adaptation with fuzzy reasoning and energy-aware control provides a significant advancement over conventional navigation approaches.

The scientific contributions of this dissertation are summarized in the following thesis groups:

Thesis Group I: Interaction-Aware Navigation Framework

This thesis group introduced a unified interaction-aware navigation architecture for autonomous mobile robots. Traditional navigation systems rely on predefined path planning and fixed control parameters, which limit their adaptability in dynamic environments. To address this limitation, an interaction-aware formulation is developed that continuously evaluates obstacle proximity, path complexity, and environmental interaction. This enables the robot to modify its trajectory and velocity in real time while maintaining safe and efficient navigation behavior.

Thesis I.1: Unified Interaction-Aware Navigation Architecture [J2, J3, C1] *I designed and implemented a unified interaction-aware navigation framework that integrates environmental perception, path planning, and motion control into a single adaptive structure, and I demonstrated that it improved navigation stability and significantly reduced trajectory oscillations, achieving up to 35% reduction in path length and 28% improvement in energy efficiency compared with conventional navigation approaches (Chapter 3).*

Thesis I.2: Adaptive Potential Field Path Planning [J2, J3, C1] *I proposed and implemented an enhanced artificial potential field method that dynamically adjusted attractive and repulsive forces according to environmental interaction, and I demonstrated that it reduced local minima occurrences and improved obstacle avoidance performance and trajectory smoothness in cluttered environments compared with classical potential field methods (Chapter 4).*

Thesis Group II: Dual-Layer Fuzzy Logic Control

This thesis group introduced a hierarchical fuzzy logic control structure for adaptive navigation. Conventional fuzzy logic controllers typically use a single-layer rule base, which limits flexibility in complex environments. To overcome this limitation, a dual-layer fuzzy architecture is proposed consisting of a global layer for terrain-aware speed control and a local layer for obstacle avoidance. This hierarchical design improves adaptability and enables context-aware decision-making.

Thesis II.1: Global Terrain-Aware Speed Regulation [J1, J4, J5, C2] *I designed and implemented a global fuzzy inference system that regulated robot speed based on obstacle density, environmental complexity, and interaction intensity, and I demonstrated that it reduced abrupt velocity changes and improved navigation safety, resulting in enhanced stability and reduced control oscillations (Chapter 5).*

Thesis II.2: Local Dynamic Obstacle Avoidance Control [J1, J4, J5, C2] *I designed and implemented a local fuzzy logic controller for real-time obstacle avoidance that generated adaptive steering and velocity commands based on obstacle proximity and relative orientation, and I demonstrated that it improved collision avoidance reliability and trajectory smoothness in dynamic environments (Chapter 6).*

Thesis Group III: Learning-Based Adaptive Navigation

This thesis group introduced reinforcement learning for adaptive navigation optimization. Conventional navigation systems rely on manually tuned parameters, which may not generalize across different environments. To address this limitation, reinforcement learning is integrated into the proposed framework to enable continuous adaptation based on navigation performance.

Thesis III.1: Reinforcement Learning Parameter Adaptation [J1, J4, J5] *I designed and integrated a reinforcement learning mechanism to automatically tune fuzzy control parameters using navigation feedback, and I demonstrated that it improved system adaptability, reduced manual tuning effort, and enhanced navigation efficiency over time in dynamic environments (Chapter 7).*

Thesis III.2: Experience-Based Navigation Optimization [J1, J4, J5, C3] *I designed and implemented a learning-based navigation optimization strategy that refined decision-making using interaction feedback, and I demonstrated that it continuously improved navigation accuracy and reduced energy consumption over time in dynamic environments compared with conventional static parameter tuning approaches (Chapter 7).*

Thesis Group IV: Learning-Based Generative Path Planning

This thesis group introduced a deep learning–based generative framework for mobile robot path planning. Unlike traditional sampling-based and heuristic approaches, the proposed method leverages representation learning to generate efficient and feasible trajectories in complex environments.

Thesis IV.1: HCVAE-Based Path Planning Framework [C1] *I designed and implemented a Hierarchical Conditional Variational Autoencoder (HCVAE)–based path planning framework to learn probabilistic representations of environment maps and robot states, and I demonstrated that it generated non-uniform sampling distributions that guided planning algorithms toward informative regions of the state space, achieving up to 18.75% improvement in planning success rate, 18.4% reduction in path length, and 33.3% reduction in execution time compared with baseline methods, while ensuring robustness in complex and cluttered environments (Chapter 8).*

9.2 Applicability of New Results

This thesis work and its results are not only scientifically significant but also practically applicable to modern autonomous robotic systems operating in complex and dynamic environments. The proposed Intelligent Obstacle Dynamics and Path Complexity System (IODPCS) integrates interaction-aware navigation, dual-layer fuzzy logic control, reinforcement learning adaptation, and energy-aware optimization. The developed framework provides improved navigation stability, adaptability, and energy efficiency, making it suitable for a wide range of real-world applications.

The results of Thesis Group I – interaction-aware navigation framework – provide a robust solution for adaptive path planning in dynamic environments. The unified interaction-aware architecture enables continuous evaluation of obstacle proximity, environmental complexity, and motion constraints. These capabilities are particularly useful in autonomous warehouse robots, where dynamic obstacles such as human operators and moving vehicles require real-time trajectory adaptation. Furthermore, the adaptive potential field formulation introduced in Thesis I.1 and Thesis I.2 is suitable for deployment in service robots operating in human-populated environments, where safe navigation and smooth motion are critical requirements. The proposed approach provides significant performance improvements for last-mile delivery robots operating in cluttered urban environments with unpredictable obstacles. The results of this group are supported by the developments presented in **[J2, J3, C1]**.

The results of Thesis Group II – dual-layer fuzzy logic control – provide an adaptive motion regulation strategy capable of handling uncertain and dynamic environments. The global terrain-aware speed control introduced in Thesis II.1 enables robots to regulate velocity according to environmental complexity, improving safety in industrial mobile robot applications. The local obstacle avoidance controller proposed in Thesis II.2 enhances collision-free navigation and improves trajectory smoothness. These capabilities are particularly relevant for search and rescue robots operating in hazardous environments, where navigation must be performed in unknown and cluttered terrain. Additionally, the hierarchical fuzzy control structure can be applied to agricultural robots performing monitoring and navigation tasks, where environmental conditions vary significantly. The results of this thesis group are validated in **[J1, J4, J5, C2]**.

The results of Thesis Group III – learning-based adaptive navigation – introduce reinforcement learning for continuous navigation improvement. The adaptive parameter tuning mechanism presented in Thesis III.1 allows robots to refine navigation behavior based on experience. This capability is especially useful in autonomous delivery robots and mobile service robots that operate in changing environments. The experience-based optimization strategy introduced in Thesis III.2 improves long-term navigation efficiency and reduces energy consumption. These methods are also applicable to industrial automated guided vehicles (AGVs), where adaptive navigation improves operational efficiency and reduces downtime. The contributions of this thesis group are supported by **[J1, J4, J5, C3]**.

The proposed framework can therefore be applied to a wide range of robotic systems, including:

- Autonomous warehouse and logistics robots
- Service robots in indoor and human-populated environments
- Autonomous delivery and last-mile transportation systems
- Search and rescue robots operating in hazardous environments

- Agricultural robots performing monitoring and navigation tasks
- Industrial mobile robots and automated guided vehicles (AGVs)

Furthermore, the modular design of the proposed framework allows seamless integration with different sensing technologies, including LiDAR, ultrasonic sensors, and vision systems. The energy-aware navigation strategy is particularly relevant for battery-powered robotic platforms, where efficient energy utilization directly impacts operational duration and system reliability.

From a scientific perspective, this work contributes to the advancement of intelligent navigation systems by demonstrating the effectiveness of combining fuzzy logic, reinforcement learning, and energy-aware optimization within a unified framework. The interaction-aware formulation introduced in this dissertation provides a new perspective on integrating perception and control in autonomous systems.

9.3 Future Research Directions

The work presented in this dissertation opens several promising research directions to further enhance the proposed navigation framework. The proposed Intelligent Obstacle Dynamics and Path Complexity System (IODPCS) has demonstrated strong performance improvements across all evaluated scenarios, including up to **35% reduction in path length**, **28% improvement in energy efficiency**, improved trajectory smoothness, and reliable collision-free navigation in both static and dynamic environments. These results confirm the robustness and effectiveness of the developed interaction-aware navigation framework. Building upon these validated contributions, the following advanced research directions are identified to further extend the capabilities and applicability of the proposed system. The following key research directions are identified:

- **Deep Reinforcement Learning Extension:** In this dissertation, adaptive fuzzy logic combined with reinforcement learning is used to regulate robot speed and trajectory based on obstacle proximity, path complexity, and interaction dynamics. Although this approach achieved stable navigation and improved performance, the learning mechanism relies on structured rule adaptation and limited state representation. Therefore, a promising direction for future work is to integrate deep reinforcement learning techniques such as Deep Q-Network (DQN) or Actor–Critic models. These methods can learn more complex navigation policies and improve decision-making in highly dynamic and uncertain environments.

- **Vision-Based Semantic Navigation:** The proposed framework primarily relies on distance-based sensing and interaction-aware control for obstacle avoidance. While this approach demonstrated reliable collision-free navigation, the robot does not explicitly interpret object types or scene context. Future work may incorporate vision-based perception and semantic mapping, allowing the robot to recognize obstacles, identify free-space regions, and improve navigation decisions. This integration would enhance robustness in unstructured and highly dynamic environments.

- **Multi-Robot Cooperative Navigation:** The current navigation framework focuses on a single differential-drive robot operating in dynamic environments. Although strong performance improvements were achieved, cooperative behavior between multiple robots is not considered. A potential future direction is to extend the proposed IODPCS framework to multi-robot systems where robots share environment information and coordinate navigation. This would enable distributed path planning, cooperative obstacle avoidance, and collaborative task execution

- **Advanced Energy-Aware Navigation:** The proposed framework introduces an energy-aware cost formulation to reduce unnecessary acceleration and inefficient motion, resulting in 28% improvement in energy efficiency. However, the current model considers instantaneous motion energy. Future research may extend this formulation to battery-aware mission planning, predictive energy modeling, and long-duration autonomous operation. Such extensions would further improve system efficiency for real-world deployment.

- **3D and Uneven Terrain Navigation:** The developed framework is validated primarily in planar navigation environments. Although the system demonstrates stable performance, elevation changes and uneven terrain are not explicitly modeled. Future work may extend the proposed navigation framework to three-dimensional environments, including ramps, slopes, and multi-level structures. This extension would broaden applicability to outdoor robots and complex terrain navigation.

- **Human-Aware Navigation Strategy:** The proposed system evaluates obstacle interaction; however, it does not explicitly distinguish human motion characteristics. A promising direction is to incorporate human-aware navigation strategies, where the robot adapts its motion based on predicted human behavior. This approach improves safety and enables reliable deployment in human-populated environments such as service robotics and healthcare applications.

- **Unified Intelligent Adaptive Navigation Architecture:**A key contribution of the proposed IODPCS framework is the integration of fuzzy logic, reinforcement learning, and energy-aware optimization within a unified control architecture. A promising research direction is to extend this architecture by incorporating additional learning modules and predictive control mechanisms. Such a unified intelligent framework enhances adaptability, robustness, and navigation efficiency for next-generation autonomous mobile robots.

Overall, this dissertation establishes a scalable and adaptive solution for autonomous mobile robot navigation, contributing to the advancement of intelligent control systems in dynamic and uncertain environments.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor, Prof. Dr. habil. Róbert Szabolcsi, for his invaluable guidance, continuous support, and encouragement throughout my doctoral studies. His insight, experience, and dedication to research have greatly contributed to the successful completion of this dissertation. I have learned a great deal from our regular discussions, and it has been an honor to work under his supervision.

I would like to express my special and sincere thanks to the Stipendium Hungaricum Scholarship Programme for providing me with the opportunity and financial support to pursue my doctoral studies. Without this support, the completion of this research would not have been possible.

I would like to thank my examiners for their insightful comments and valuable suggestions, which have significantly improved the quality of this dissertation. I also extend my sincere appreciation to the Head of the Doctoral School on Safety and Security Sciences, Prof. Tibor Goda, for his support, guidance, and encouragement throughout my PhD journey. Furthermore, I would like to acknowledge the academic and administrative staff of the Department of Mechatronics for their cooperation and support. I am also grateful to my colleagues for the constructive discussions and the stimulating research environment, and to my friends for their support and the memorable moments we shared during this journey.

My deepest gratitude goes to my parents, who have provided unwavering support and made countless sacrifices throughout my academic journey. They took care of my family during my long absence and continuously encouraged me to pursue my goals. Without their support, this achievement would not have been possible. This thesis is dedicated to them. I would also like to thank my sisters and relatives for their continuous encouragement and support.

Finally, and on a more personal note, I would like to express my deepest and most heartfelt gratitude to my beloved husband, Dr. Mohammed Salah Al-Radhi, for his immense support throughout this long and challenging journey. His encouragement, patience, and unwavering belief in me have been a constant source of strength and motivation.

Figures

Tables

Publications

Publications related to Ph.D. Thesis

International journals

[J1] Safa Jameel Al-Kamil, Edit Laufer, Róbert Szabolcsi, Deep-adaptive fuzzy predictive navigation framework for stable and intelligent mobile robot control, Results in Control and Optimization**, Elsevier, 2026. (WoS, IF = 3.2) Available at:**
DOI: https://doi.org/10.1016/j.rico.2026.100669

[J2] Safa Jameel Al-Kamil, Róbert Szabolcsi, Optimizing path planning in mobile robot systems using motion capture technology**,** Results in Engineering, **Elsevier,** 2024**. (D1,** WoS, **IF = 7.9, Independent citation = 69)** Available at:
DOI: https://doi.org/10.1016/j.rineng.2024.102962

[J3] Safa Jameel Al-Kamil, Róbert Szabolcsi, Enhancing Mobile Robot Navigation: Optimization of Trajectories through Machine Learning Techniques for Improved Path Planning Efficiency**, Mathematics, MDPI, 2024. (WoS, IF = 2.2, Independent citation = 17)** Available at: **DOI**: https://doi.org/10.3390/math12121787

[J4] Safa Jameel Al-Kamil, Róbert Szabolcsi, Adaptive Fuzzy-Learning Control with Multi-Objective Optimization for Robust Mobile Robot Navigation**, Array, Elsevier, 2026. (WoS, Q1, IF = 4.5,** Major revision completed, reviewer reports received, decision pending)

[J5] Safa Jameel Al-Kamil, Róbert Szabolcsi, An Integrated Intelligent Framework for Energy Aware Mobile Robot Navigation in Dynamic Environments**, Discover Applied Sciences, Springer Nature, 2026. (WoS, Minor revision)**

International conferences

[C1] Safa Jameel Al-Kamil, Róbert Szabolcsi, Enhancing Mobile Robot Path Planning with Hierarchical Conditional Variational Autoencoder Implementation, Springer Nature Switzerland, 2025. (Conference paper, Scopus indexed) Available at: DOI: https://doi.org/10.1007/978-3-031-78544-3_46

[C2] Safa Jameel Al-Kamil, Edit Laufer, Róbert Szabolcsi, Fuzzy Logic Integration for Enhanced Mobile Robot Path Planning and Navigation, Scientific Book of Abstracts, Trans Tech Publications, 2023. (Conference abstract) Available at: https://www.torrossa.com/en/resources/an/5792122

[C3] Safa Jameel Al-Kamil, Differential Safety Technique of Mobile Robot System, Trends and Innovations in E-business, Education and Security, 2022. (Conference paper)

Additional publication (my contribution related to the neural network modelling)

International Journal

[A1] Safa J. Al-Kamil et al., Neural Network for Porosity Prediction in Carbonate Formation: A Case Study of the South of Iraq, *Journal of Petroleum Research and Studies*, 13(3), pp. 1–18. Available at:
DOI: https://doi.org/10.52716/jprs.v13i3.697

APPENDIX A:

APPENDIX A List of Acronyms and Variables

Symbol / Acronym	**Description**
HDFPN-S	Hierarchical Deep–Fuzzy Predictive Navigation with Stability gating
MPC	Model Predictive Control
PF	Potential Field
PSO	Particle Swarm Optimization
HIL	Human-in-the-Loop
FIS	Fuzzy Inference System
$\boldsymbol{\upsilon}$	Linear velocity of the robot
$\boldsymbol{\omega}$	Angular velocity of the robot
$\boldsymbol{J}$	Multi-objective cost function
$\boldsymbol{d_o}$	Distance to obstacle
$\boldsymbol{E}$	Energy consumption

Bibliography

[1] R. de Wilde and J. A. M. M. Everett, "Collision avoidance for autonomous robots with dynamic obstacles using interaction-aware planning," Autonomous Robots, vol. 44, no. 5, p. 821–842, 2020.

[2] Y. L. Wang and Z. S. H. Wang,, "Adaptive motion control of mobile robots in uncertain and dynamic environments," Sensors, vol. 20, no. 3, pp. 1–18,, 2020.

[3] Y. W. Li and C. L. S. Li, "Adaptive fuzzy-learning control for autonomous mobile robot navigation," Neurocomputing, vol. 470, p. 314–327, 2022.

[4] S. W. Chen and Y. G. J. Chen, "Interaction-aware navigation of mobile robots in dynamic environments," IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 8, p. 5074–5085, 2021.

[5] A. Faust, K. Oslund, O. Ramirez, A. Francis, L. Tapia, M. Fiser, J. Davidson, "PRM-RL: Long-range robotic navigation tasks by combining reinforcement learning and sampling-based planning," n Proc. IEEE Int. Conf. Robotics and Automation (ICRA), p. 5113–5120, 2018.

[6] H. Zhang, M. Liu, and J. Xiao, "Energy-aware motion planning for autonomous mobile robots," IEEE Access, vol. 9, pp. 112340–112352,, 2021.

[7] Y. Chen, B. Li, and Z. Li, "Robust navigation of differential-drive mobile robots under sensing uncertainty," IEEE Access, vol. 9, p. 154921–154934, 2021.

[8] Z. Liu, H. Wang, and X. Chen,, "Interaction-driven adaptive navigation strategies for autonomous mobile robots," Robotics and Autonomous Systems, vol. 148, p. 103–116, 2022.

[9] L. Tai, G. Paolo, and M. Liu,, "Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation," in Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS), p. 31–36., 2017.

[10] M. Shukla and R. Tiwari, "Energy-efficient path planning and speed regulation for mobile robots,," Journal of Intelligent & Robotic Systems, vol. 99, p. 831–846, 2020.

[11] Z. Liu, Y. Wang, and C. Liu, "Learning-based adaptive navigation for autonomous robots in complex environments,.," Neurocomputing, vol. 489, pp. 45-58, 2023.

[12] J. Alonso-Mora, "Socially compliant navigation and interaction-aware motion planning," nnual Review of Control, Robotics, and Autonomous Systems, vol. 6, pp. 215-238, 2023.

[13] M. Everett, Y. F. Chen, and J. P. How, "Motion planning among dynamic, decision-making agents with deep reinforcement learning," in Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS), p. 3052–3059., 2018.

[14] J. Alonso-Mora, P. Beardsley, and R. Siegwart, "Reciprocal collision avoidance for multiple robots with dynamic constraints," IEEE Transactions on Robotics, vol. 38, no. 1, p. 294–311, 2022.

[15] D. Scaramuzza and F. Fraundorfer, "Visual odometry: Part I—The first 30 years and fundamentals," IEEE Robotics & Automation Magazine, vol. 18, no. 4, pp. 80-92, 2011.

[16] Y. Li, X. Zhang, and H. Chen, "Autonomous navigation of mobile robots in dynamic environments using improved potential field methods," Robotics and Autonomous Systems, vol. 150, no. 103984, 2022.

[17] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," The International Journal of Robotics Research, vol. 5, no. 1, p. 90–98, 1986.

[18] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," IEEE Robotics and Automation Magazine, vol. 4, no. 1, p. 23–33, 1997.

[19] J. Guo, Y. Zhao, and H. Chen, "Autonomous navigation of mobile robots in unknown environments using adaptive artificial potential field methods," Robotics and Autonomous Systems, vol. 152, p. 104054, 2022.

[20] H. Wang, Y. Liu, and M. Q. H. Meng, "Intelligent path planning and obstacle avoidance for autonomous mobile robots in dynamic environments," Neurocomputing, vol. 452, p. 421–432, 2021.

[21] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, "STOMP: Stochastic trajectory optimization for motion planning,," in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), p. 4569–4574, 2011.

[22] M. H. Korayem, H. Dehkordi, and A. Nikoobin, "Path planning of mobile robots using artificial potential field in dynamic environments," Robotics and Autonomous Systems, vol. 60, no. 12, p. 1597–1608, 2012.

[23] M. Mokhtari, B. Vanderborght, and J. Famaey, "Energy harvesting aware path planning for ambiently-powered multi-robot systems," Robotics and Autonomous Systems, vol. 197, pp. 105260,, 2025.

[24] J. Alonso-Mora, "Socially compliant navigation and interaction-aware motion planning," Annual Review of Control, Robotics, and Autonomous Systems,, vol. 6, p. 215–238, 2023.

[25] Y. Liu, H. Wang, and Z. Chen,, "Adaptive speed control and motion planning for autonomous mobile robots in dynamic environments," Robotics and Autonomous Systems, vol. 158, p. 104259, 2023.

[26] M. Macenski, S. Singh, F. Martin, and J. Gines, "The Nav2 navigation framework for autonomous mobile robots," Science Robotics, vol. 8, no. 80, 2023.

[27] S. Thrun, W. Burgard, and D. Fox,, Probabilistic Robotics, MIT Press, 2005.

[28] Park, J.-W., Kwak, H.-J., Kang, Y.-C., and Kim, D. W., "Advanced fuzzy potential field method for mobile robot obstacle avoidance," Computational Intelligence and Neuroscience, 2016.

[29] L. Kavraki, P. Švestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," EEE Transactions on Robotics and Automation, vol. 12, no. 4, p. 566–580, 1996.

[30] D. Gonzalez, J. Perez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 4, p. 1135–1145, 2016.

[31] Teso-Fz-Betoño, D., Aramendia, I., Ramos-Hernanz, J. A., Caballero-Martin, D., Affou, H., and Lopez-Guede, J. M, "Optimization of energy efficiency with a predictive dynamic window approach for mobile robot navigation," Sustainability, vol. 17, no. 10, p. 4526, 2025.

[32] I. Ulrich and J. Borenstein,, "VFH+: Reliable obstacle avoidance for fast mobile robots," Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 1998.

[33] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," IEEE Robotics & Automation Magazine, vol. 4, no. 1, pp. 23-33, 1997.

[34] Z. Liu, H. Wang, and X. Chen, "Interaction-driven adaptive navigation strategies for autonomous mobile robots," Robotics and Autonomous Systems, vol. 148, 2022.

[35] C. C. Lee, "Fuzzy logic in control systems: fuzzy logic controller—Part I," IEEE Transactions on Systems, Man, and Cybernetics,, vol. 20, no. 2, p. 404–418, 1990.

[36] H. Tai, J. Xiao, and W. Zhang, "A survey of deep reinforcement learning for mobile robot navigation," Neurocomputing, vol. 412, p. 112–124, 2020.

[37] Kumar, P., et al., "DSQN: Robust path planning of mobile robot based on deep spiking Q-network," Neurocomputing, vol. 590, p. 112–124, 2025.

[38] J. Guo et al., "Energy-efficient path planning for mobile robots in dynamic environments," Robotics and Autonomous Systems, vol. 160, p. 104327, 2022.

[39] Yin, Y., Chen, Z., Liu, G., Yin, J., & Guo, J, "Autonomous navigation of mobile robots in unknown environments using off-policy reinforcement learning with curriculum learning," Expert Systems with Applications, vol. 247, p. 123202, 2024.

[40] Moorthy, M., Lee, J., & Kim, K, "Autonomous robot navigation system optimization using learning-based control strategies," Array, vol. 15, p. 100312, 2024.

[41] Ganesan, S., Ramalingam, B., & Mohan, R. E, "A hybrid sampling-based RRT* path planning algorithm for autonomous mobile robot navigation," Expert Systems with Applications, vol. 258, p. 125206, 2024.

[42] Tao, B., & Kim, J. H., "Deep reinforcement learning-based local path planning in dynamic environments for mobile robots," ournal of King Saud University – Computer and Information Sciences, vol. 36, no. 10, p. 102254., 2024.

[43] Al Kamil, S. J., & Szabolcsi, R., "Optimizing path planning in mobile robot systems using motion capture technology," Results in Engineering, vol. 22, p. 102043, 2024.

[44] Xiao, H., Chen, C., Zhang, G., & Chen, C. L. P, "Reinforcement learning-driven dynamic obstacle avoidance for mobile robot trajectory tracking," Knowledge-Based Systems, vol. 297, p. 111974, 2024.

[45] Gu, X., Liu, L., Wang, L., Yu, Z., & Guo, Y., "Energy-optimal adaptive artificial potential field method for path planning of free-flying space robots.," Journal of the Franklin Institute, vol. 361, no. 2, p. 978–993, 2024.

[46] Fang, W., Liao, Z., & Bai, Y, "Improved ACO algorithm fused with improved Q-learning algorithm for Bessel curve global path planning of search and rescue robots," Robotics and Autonomous Systems, vol. 182, p. 104822, 2024.

[47] Liu, Y., Wang, S., & Chen, R, "Adaptive hybrid fuzzy–reinforcement learning control for autonomous ground robots in dynamic environments," Neurocomputing, vol. 525, p. 194–210, 2024.

[48] Deshpande, S. V., Harikrishnan, R., Ibrahim, B. S. K. K., & Ponnuru, M. D. S, "Mobile robot path planning using deep deterministic policy gradient with differential gaming (DDPG-DG)," Cognitive Robotics, vol. 4, p. 156–173, 2024.

[49] Weerakoon, T., Ishii, K., & Nassiraei, A. A. F, "Weerakoon, T., Ishii, K., & Nassiraei, A. A. F," Journal of Artificial Intelligence and Soft Computing Research, vol. 5, no. 3, p. 189–203.

[50] Md H., & Israt, F, "Autonomous robot path planning in dynamic environment using a bacterial foraging-based optimization technique," Robotics and Autonomous Systems, vol. 64, p. 137–141, 2015.

[51] Mac, T. T., Pham, A. Q., & Nguyen, X.-T., "Development of a fleet management system for mobile robots delivering medicine in healthcare environments," Acta Polytechnica Hungarica, vol. 21, no. 9, p. 5, 2024.

[52] Al Kamil, S. J., & Al Radhi, M. S. , "Deep learning for self-driving vehicles," In Proceedings of the 2nd International Multi-Disciplinary Conference on Integrated Sciences and Technologies (IMDC-IST), p. 322–329, 2022.

[53] Dwijayanti, S., Suprapto, B. Y., & Rizky, I. A, "Practical implementation of a type-2 fuzzy logic controller for steering a service robot," Results in Control and Optimization, vol. 19, p. 100558, 2025.

[54] Shitsukane, A., Cheruiyot, W., Otieno, C., & Mvurya, M, "Fuzzy logic sensor fusion for obstacle avoidance in mobile robots," In Proceedings of the IST-Africa Week Conference,, p. 1–8, 2018.

[55] Nadour, M., Boumehraz, M., Cherroun, L., & Puig, V, "Hybrid type-2 fuzzy logic obstacle avoidance system based on Horn–Schunck method," Electrotehnica, Electronica, Automatica, vol. 67, no. 3, p. 45–51, 2019.

[56] Zheng, J., Liu, B., Meng, Z., & Zhou, Y, "Integrated real-time obstacle avoidance algorithm based on fuzzy logic and L1 control for unmanned helicopters," In Proceedings of the Chinese Control and Decision Conference (CCDC), p. 1865–1870, 2018.

[57] Pourjavad, E., & Mayorga, R. V., "Comparative study and performance measurement of manufacturing systems using Mamdani fuzzy inference systems," Journal of Intelligent Manufacturing, vol. 30, p. 1085–1097, 2019.

[58] Rawat, H., Parhi, D. R., Kumar, P. B., Pandey, K. K., & Behera, A. K, "Analysis and investigation of Mamdani fuzzy control for mobile robot navigation," International Journal of Scientific & Engineering Research, vol. 9, no. 5, p. 1–7, 2018.

[59] Mohd Romlay, M. R., Mohd Ibrahim, A., Toha, S. F., De Wilde, P., Venkat, I., & Ahmad, M. S, "Obstacle avoidance for robotic navigation using fuzzy logic controller–optimal reciprocal collision avoidance (FLC-ORCA)," Neural Computing and Applications, vol. 35, no. 30, p. 22405–22429, 2023.

[60] Riman, C. F., & Abi-Char, P. E., "Fuzzy logic control for mobile robot navigation in automated storage," International Journal of Mechanical Engineering and Robotics Research, vol. 12, no. 5, p. 313–323, 2023.

[61] Zhang, H., Sun, Y., Li, Q., & Wang, B, "A hybrid reinforcement learning and heuristic search framework for dynamic mobile robot path planning," Expert Systems with Applications, vol. 239, p. 121928, 2024.

[62] Rasidescu, V., & Taghavifar, H, "Artificial potential fields-enhanced socially intelligent path-planning for autonomous vehicles using type-2 fuzzy systems," In Proceedings of the IEEE 27th International Conference on Intelligent Transportation Systems (ITSC), p. 2599–2605, 2024.

[63] Teso-Fz-Betoño, D., Aramendia, I., Ramos-Hernanz, J. A., Caballero-Martin, D., Affou, H., & Lopez-Guede, J. M, "Optimization of energy efficiency with a predictive dynamic window approach for mobile robot navigation," Sustainability, vol. 17, no. 10, p. 4526, 2025.

[64] Chen, J., Xie, J., Li, J., & Chen, W, "Human-in-the-loop fuzzy iterative learning control of consensus for unknown mixed-order nonlinear multi-agent systems," IEEE Transactions on Fuzzy Systems, vol. 32, no. 1, p. 255–265, 2024.

[65] Haighton, R., Asgharnia, A., Schwartz, H., & Givigi, S, "An adaptable fuzzy reinforcement learning method for non-stationary environments," Neurocomputing, vol. 604, p. 128309, 2024.

[66] Huang, K., Liu, S., & Zhao, Q, "Interval type-2 fuzzy logic controller for trajectory tracking of autonomous vehicles," Knowledge-Based Systems, vol. 298, p. 112145, 2024.

[67] BK, A., & Prakash, A., "A robo-fuzzy inference system for iron-based sprouts fortification," In Proceedings of the 2nd International Conference on Electronics and Sustainable Communication Systems (ICESC), p. 463–468, 2021.

[68] Shen, M., Wang, Y., Jiang, Y., Ji, H., Wang, B., & Huang, Z, "A new positioning method based on multiple ultrasonic sensors for autonomous mobile robots," Sensors, vol. 20, no. 1, p. 17, 2020.

[69] Dombi, J., & Tóth-Laufer, E, "Reducing the computational requirements in Mamdani-type fuzzy control," Acta Polytechnica Hungarica, vol. 17, no. 3, p. 25–40, 2020.

[70] Liu, L., Wang, X., Yang, X., Liu, H., Li, J., & Wang, P., "Path planning techniques for mobile robots: Review and prospect," Expert Systems with Applications, vol. 227, p. 120254, 2023.

[71] Deme, T. T., Manavaalan, M., & Gunasekaran, K, "Trajectory tracking control of a mobile robot using fuzzy logic controller with optimal parameters," Robotica, vol. 42, no. 5, p. 1238–1252, 2024.

[72] Pothal, J. K., & Parhi, D. R., "Navigation of multiple mobile robots in highly cluttered terrains using adaptive neuro-fuzzy inference system," Robotics and Autonomous Systems, vol. 72, p. 48–58, 2015.

[73] Al-Kamil, S. J., Loufer, R., & Szabolcsi, R., "Deep–adaptive fuzzy predictive navigation framework for stable and intelligent mobile robot control," Results in Control and Optimization, vol. 22, p. 100669, 2026.

[74] Cheng, Y., & Wang, G. Y., "Mobile robot navigation based on LiDAR," In Proceedings of the Chinese Control and Decision Conference (CCDC), p. 1243–1246, 2018.

[75] Naich, A. Y., & Requena Carrión, J., "LiDAR-based intensity-aware outdoor 3D object detection," Sensors, vol. 24, no. 9, p. 2942, 2024.

[76] Hutabarat, D., Rivai, M., Purwanto, D., & Hutomo, H, "LiDAR-based obstacle avoidance for autonomous mobile robots," In Proceedings of the 12th International Conference on Information & Communication Technology and System (ICTS), p. 197–202., 2019.

[77] Loganathan, A., & Ahmad, N. S., "A systematic review on recent advances in autonomous mobile robot navigation," Engineering Science and Technology, an International Journal, vol. 40, p. 101343, 2023.

[78] Liu, Y., Wang, S., & Chen, R., "Adaptive hybrid fuzzy–reinforcement learning control for autonomous ground robots in dynamic environments," Neurocomputing, vol. 525, p. 194–210, 2024.

[79] Karim, M. R., Reza, M. N., Jin, H., Haque, M. A., Lee, K. H., Sung, J., & Chung, S.-O., "Application of LiDAR sensors for crop and working environment recognition in agriculture: A review," Remote Sensing, vol. 16, no. 24, p. 4623, 2024.

[80] Cabral, I. M., Pereira, J. S., & Ribeiro, J. B., "Performance evaluation of PID and fuzzy logic controllers for residential ORC-based cogeneration systems," Energy Conversion and Management: X, vol. 23, p. 100622, 2024.

[81] Nasrinahar, A., & Chuah, J. H., "Intelligent motion planning of a mobile robot with dynamic obstacle avoidance," Journal of Vehicle Routing Algorithms, vol. 1, p. 89–104, 2018.

[82] Stavrinidis, S., Zacharia, P., & Xidias,, "A fuzzy control strategy for multi-goal autonomous robot navigation.," Sensors, vol. 25, no. 2, p. 446, 2025.

[83] Gorrin-Ortega, Y., Cardenas-Maciel, S. L., Lopez-Renteria, J. A., & Cazarez-Castro, N. R., "Parameters determination via fuzzy inference systems for the logistic populations growth model," Axioms, vol. 14, no. 1, p. 36, 2025.

[84] Cheniouni, I., Berkane, S., & Tayebi, A., "Hybrid feedback control for global navigation with locally optimal obstacle avoidance in n-dimensional spaces," arXiv preprint, p. 2412.20320, 2025.

[85] Sousa, J. M., Silva, H., & Pires, G., "Speed control of a mobile robot using fuzzy logic controller," Sensors, vol. 25, no. 4, p. 1120, 2025.

[86] Al-Kamil, S. J., & Szakács, T. (2021)., "Design of all-wheel drive vehicles based on differential speed control systems.," Mechatronic Systems and Control, vol. 20, p. 25–29, 2021.

[87] M. C. Bingol, "A safe navigation algorithm for differential-drive mobile robots using fuzzy logic reward function-based deep reinforcement learning," Electronics, vol. 14, no. 8, p. 1593, 2025.

[88] M. Trojnacki, "Tracking control of a four-wheeled skid-steered robot with slip compensation and application of the drive unit model," Electronics, vol. 14, no. 3, p. 444, 2025.

[89] K. M., "Obstacle avoidance and path planning methods for robots.," Sensors, vol. 24, no. 11, p. 3573, 2024.

[90] B. K. Patle, A. Pandey, D. R. Parhi, and A. Jagadeesh, "A review on path planning strategies for navigation of mobile robots," Journal of Industrial Information Integration, vol. 13, p. 1–22, 2019.

[91] S. V. Deshpande, R. Harikrishnan, B. K. Ibrahim, and M. D. S. Ponnuru, "Mobile robot path planning using deep deterministic policy gradient with differential gaming exploration," Cognitive Robotics, vol. 4, p. 156–173, 2024.

[92] J. W. Park, "Advanced fuzzy potential field method for mobile robot obstacle avoidance," Journal of Sensors, 2016.

[93] C. F. Riman and P. E. Abi-Char, "Fuzzy logic control for mobile robot navigation in automated storage," International Journal of Mechanical Engineering and Robotics Research, vol. 12, no. 5, p. 313–323, 2023.

[94] Y. Teng, "Path planning of mobile robots based on dual-layer fuzzy control and improved genetic algorithm," Symmetry, vol. 17, no. 4, p. 609, 2025.

[95] H. Zhu, J. Liu, and Z. Li, "Deep reinforcement learning-based path planning for mobile robots in dynamic environments," Robotics and Autonomous Systems, vol. 125, p. 103437, 2020.

[96] M. Algabri, H. Mathkour, H. Ramdane, and M. Alsulaiman, "review on particle swarm optimization and its applications in mobile robot path planning," International Journal of Advanced Robotic Systems, vol. 12, no. 1, p. 1–14, 2015.

[97] J. K. Pothal and D. R. Parhi, "Navigation of multiple mobile robots in highly cluttered terrains using adaptive neuro-fuzzy inference system," Robotics and Autonomous Systems, vol. 72, p. 48–58, 2015.

[98] A. Keymasi-Khalaji, P. Mokhtari, and F. Bathaie, "Predictive control of spherical robots navigating in obstacle-rich environments," Scientific Reports, vol. 15, p. 13859, 2025.

[99] H. Xiao, C. Chen, G. Zhang, and C. L. P. Chen, "Reinforcement learning-driven dynamic obstacle avoidance for mobile robot trajectory tracking," Knowledge-Based Systems, vol. 297, p. 111974, 2024.

[100] M. Oriyama, "Human-in-the-loop transfer learning in collision avoidance," Robotics and Autonomous Systems, vol. 182, p. 104822, 2025.

[101] Macenski, M., Singh, S., Martin, F., and Gines, J.,, "The Nav2 navigation framework for autonomous mobile robots," Science Robotics, vol. 8, no. 80, 2023.

[102] M. Q. Zaman and H. M. Wu, "An improved fuzzy inference strategy using reinforcement learning for trajectory tracking of a mobile robot under varying slip ratio," Robotica, vol. 42, no. 4, p. 1134–1152, 2024.

[103] Y. Wang and J. Li,, "GPS-based hybrid navigation system using deep reinforcement learning and ANFIS for obstacle avoidance," Robotics and Autonomous Systems, vol. 174, p. 104678, 2025.

[104] M. Mohaghegh, "Log-concave model predictive control with neural network structures and Lyapunov stability guarantees," Frontiers in Robotics and AI, vol. 10, 2023.

[105] A. Kumar and L. Chen, "Obstacle avoidance for human–robot collaborative warehouses using CNN–fuzzy hybrid decision systems," Sensors, vol. 25, no. 8, p. 2387, 2025.

[106] A. M. El-Nagar, "Embedded deep learning neural network control for skid-steering mobile robots," SA Transactions, in press, 2025.

[107] Y. Zhu, "Deep reinforcement learning of mobile robot navigation: A comprehensive review," Sensors, vol. 15, p. 14656, 2025.

[108] J. Zhang, "Artificial intelligence and computer technology applications in intelligent robots," Scientific Reports, p. 14656, 2025.

[109] S. J. Al-Kamil and R. Szabolcsi, "Enhancing mobile robot path planning with hierarchical conditional variational autoencoder implementation," in The Impact of the Energy Dependency on Critical Infrastructure Protection (ICCECIP 2024), Cham: Springer, p. 577–587, 2025.

[110] H. Wang, Y. Zhao, and J. Xu, "Deep reinforcement learning-assisted ANFIS controller for real-time obstacle avoidance in mobile robots," Robotics and Autonomous Systems,, vol. 173, 2024.

[111] M. Katona, "Obstacle avoidance and path planning methods for robots," Sensors, vol. 24, no. 11, p. 3573, 2024.

[112] Y. Yu, "Research on signal stability of mobile robots based on fuzzy PID control," Highlights in Science, vol. 134, p. 55–61, 2025.

[113] B. K. Patle, A. Pandey, and D. R. Parhi, "daptive neuro-fuzzy approaches for navigation of mobile robots in cluttered terrains," Expert Systems with Applications, vol. 145, p. 113098, 2020.

[114] Oh, T., Won, Y.-J., and Lee, S., "TA-RRT: Adaptive sampling-based path planning using terrain analysis," Applied Sciences, vol. 15, no. 5, p. 2287, 2025.

[115] R. D. Puriyanto et al, "Ball detection system for a soccer wheeled robot using MobileNetV2-SSD method,," Emerging Science Journal, vol. 9, no. 5, 2025.

[116] M. F. e. al, "Assessing AI-driven personalization in smart cities using hybrid machine learning and MCDM approach," HighTech and Innovation Journal, vol. 6, no. 3, p. 770–792, 2025.

[117] D. T. e. al., "Enhancing trajectory tracking in humanoid robots using neural network-based dynamic gain control," Emerging Science Journal, vol. 9, p. 539–556, 2025.

[118] T. Sands, "Autogenetic gravity center placement," Sensors, vol. 25, no. 12, p. 3786, 2025.

[119] T. Sands, "Autonomous real-time mass center location and inertia identification for grappling space robotics," Technologies, vol. 13, no. 4, p. 148, 2025.

[120] T. Sands, "Flattening the curve of flexible space robotics," Applied Sciences, vol. 12, no. 6, p. 2992, 2022.

[121] E. Kuck and T. Sands, "Space robot sensor noise amelioration using trajectory shaping," Sensors, vol. 24, no. 2, p. 666, 2024.

[122] T. Sands, "Bio-inspired space robotic control compared to alternatives," Biomimetics, vol. 9, no. 2, p. 108, 2024.

[123] M. S. Al-Radhi, T. G. Csapó, and G. Németh, "Adaptive refinements of pitch tracking and HNR estimation within a vocoder for statistical parametric speech synthesis," Applied Sciences, vol. 9, no. 12, p. 2460, 2019.

[124] Y. Yin, Z. Chen, G. Liu, J. Yin, and J. Guo, "Autonomous navigation of mobile robots in unknown environments using off-policy reinforcement learning with curriculum learning," Expert Systems with Applications, vol. 247, p. 123202, 2024.

[125] Z. Li, D. Wu, and J. Zhang, "Deep reinforcement learning-based nonlinear model predictive control for autonomous mobile robots in dynamic environments," Results in Control and Optimization, vol. 9, p. 100249, 2025.

[126] A. González, R. Muñoz, and D. López, "Kinematic modeling and control of car-like robots," Robotics and Autonomous Systems, vol. 150, p. 103990, 2022.

[127] C. Lin, J. Wang, and S. Huang, "Slip-aware navigation for Ackermann vehicles under varying terrain conditions," Mechanical Systems and Signal Processing, vol. 200, p. 110543, 2023.

[128] T. Zhang and J. Sun, "Vehicle longitudinal–lateral dynamics modeling for autonomous driving applications," Control Engineering Practice, vol. 124, p. 105217, 2022.

[129] Y. Teng, "Path planning of mobile robots based on dual-layer fuzzy control and improved genetic algorithm," Symmetry, vol. 17, no. 4, p. 609, 2025.

[130] Y. Huang, Z. Li, and J. Qiu, "Deep neuro-fuzzy networks for adaptive control of nonlinear systems," Applied Soft Computing, vol. 138, p. 110204, 2023.

[131] F. Silva, A. Mendes, and P. Costa, "Fuzzy reasoning with adaptive membership functions for autonomous navigation," Expert Systems with Applications, vol. 245, p. 123010, 2024.

[132] Z. Wang, M. Zhao, and H. Tang, "Lyapunov-stable fuzzy control for nonlinear systems in dynamic environments," Engineering Applications of Artificial Intelligence, vol. 133, p. 107225, 2024.

[133] R. Kumar, P. Singh, and A. Verma, "Predictive optimization with fuzzy priors for robot control," Robotics and Autonomous Systems, vol. 168, p. 104643, 2023.

[134] X. Li, Y. Zhou, and H. Feng, "Confidence-triggered human–robot interaction for safe navigation," Information Sciences, vol. 634, p. 175–190, 2023.

[135] Y. Chen, W. Guo, and H. Ren, "Robust fuzzy navigation under bounded uncertainties," Fuzzy Sets and Systems, vol. 466, p. 108–123, 2023.

[136] P. Ortega, E. Ramos, and C. Vidal, "Adaptive human-in-the-loop learning for mobile robots using reinforcement signals," Knowledge-Based Systems, vol. 293, p. 111552, 2024.

[137] T. Chen and Q. Zhou, "Nonlinear adaptive control for uncertain mobile robots based on Lyapunov stability," Nonlinear Dynamics, vol. 112, no. 2, p. 1441–1456, 2024.

[138] P. Wang and Z. He, "Predictive dynamic optimization for multi-objective navigation of autonomous vehicles," IEEE Transactions on Intelligent Vehicles, vol. 8, no. 4, p. 5672–5684, 2023.

[139] al., D. Mitchell et, "Cyber-physical-human systems for mobile robots in energy-aware navigation," Energy AI, vol. 21, p. 100570, 2025.

[140] E. Tsagkournis et al., "A supervised machine-learning approach to operator intent inference in mobile robot navigation," Digital Communications and Networks, vol. 9, no. 3, p. 340–351, 2023.

[141] P. Duan et al.,, "Solving the multi-objective path planning problem for mobile robots using improved NSGA-II," Journal of Manufacturing Systems, vol. 70, p. 461–472, 2024.

[142] M. A. Taleb et al., "Automotive navigation for mobile robots: Comprehensive challenges and solutions," Transportation Research Interdisciplinary Perspectives, vol. 13, p. 100428, 2025.

[143] K. Xu, R. Zhang, and J. Liang, "Safe navigation control for mobile robots with deep adaptive fuzzy reinforcement learning," Engineering Applications of Artificial Intelligence, vol. 140, p. 109850, 2024.

[144] J.-P. Arnaout and Z. Liu, "Multi-agent coordination and predictive control for autonomous mobile robots in dynamic environments," Expert Systems with Applications, vol. 259, p. 125401, 2025.

[145] D. Chatterjee and S. Khatri,, "Real-time fuzzy predictive control for intelligent robot systems under uncertainty," Robotics and Autonomous Systems, vol. 186, p. 104933, 2025.

[146] X. C. a. Y. M. L. Zhao, "Energy-aware optimization for intelligent robotic navigation using hybrid fuzzy-reinforcement control," Results in Control and Optimization, vol. 8, p. 100215, 2024.

[147] M. Faisal, K. Al-Mutib, R. Hedjar, et al, "Multi-module fuzzy logic framework for mobile robot navigation in unknown dynamic environments," IEEE Systems Journal, vol. 13, no. 2, p. 1443–1454, 2019.

[148] Aremu, M. B., Ahmed, G., Elferik, S., and Saif, A.-W., "Autonomous mobile robot path planning techniques—A review: Metaheuristic and cognitive techniques," Robotics, vol. 15, no. 1, p. 23, 2016.

[149] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," International Journal of Robotics Research, vol. 5, no. 1, p. 90–98, 1986.

[150] G. Li, A. Yamashita, and H. Asama, "Efficient improved artificial potential field-based regression search for robot path planning," Robotics and Autonomous Systems, vol. 113, p. 28–41, 2019.

[151] B. K. Patle, L. G. Babu, A. Pandey, D. R. K. Parhi, and A. Jagadeesh, "A review on path planning strategies for navigation of mobile robots," Defence Technology, vol. 15, no. 4, p. 582–606, 2019.

[152] O. Montiel, U. Orozco-Rosas, and R. Sepúlveda, "Path planning for mobile robots using bacterial potential field for avoiding static and dynamic obstacles," Expert Systems with Applications, vol. 42, no. 12, p. 5177–5191, 2015.

[153] J. Kim, S. Lee, and J. Park, "Interaction-aware navigation of mobile robots in dynamic crowds," Robotics and Autonomous Systems, vol. 139, p. 103709, 2021.

[154] Fang, W., Liao, Z., and Bai, Y., "Improved ACO algorithm fused with improved Q-learning algorithm for Bessel curve global path planning of search and rescue robots," Robotics and Autonomous Systems, vol. 182, p. 104822, 2024.

[155] C. Luo and S. Yang, "Hybrid fuzzy–learning navigation framework for autonomous robots," Neurocomputing, vol. 514, p. 1–13, 2023.

[156] H.-Y. Chung, C.-C. Hou, and S.-C. Liu, "Automatic navigation of a wheeled mobile robot using particle swarm optimization and fuzzy control," EEE Transactions on Industrial Electronics, vol. 8, no. 2, p. 33, 2019.

[157] Z. T. Allawi and T. Y. Abdalla, "A PSO-optimized type-2 fuzzy logic controller for navigation of multiple mobile robots," Robotics, vol. 8, no. 2, p. 33, 2019.

[158] M. Nadour, M. Boumehraz, L. Cherroun, and V. Puig, "Hybrid type-2 fuzzy logic obstacle avoidance system based on optical flow," ISA Transactions, vol. 11, p. 307–320, 2020.

[159] D. Sun, Y. Zhao, and L. Chen, "Energy-aware mobile robot control using fuzzy inference," ISA Transactions, vol. 121, p. 289–301, 2022.

[160] T. T. Mac, A. Q. Pham, and X.-T. Nguyen, "Development of a fleet management system for mobile robots delivering medicine in healthcare environments," Acta Polytechnica Hungarica, vol. 21, no. 9, 2024.

[161] M. R. Mohd Romlay, A. Mohd Ibrahim, S. F. Toha, P. De Wilde, and M. S. Ahmad, "bstacle avoidance for robotic navigation aid using FLC-ORCA," Neural Computing and Applications, vol. 35, no. 30, p. 22405–22429, 2023.

[162] J. Chen, H. Zhang, and Y. Wang, "Adaptive fuzzy navigation with online learning in dynamic environments," Robotics and Autonomous Systems, vol. 142, p. 103781, 2021.

[163] L. Zhou, Y. Wang, and H. Li,, "Context-aware fuzzy navigation using reinforcement signals," Applied Soft Computing, vol. 118, p. 108462, 2022.

[164] X. Li, Y. Zhao, and D. Sun, "Learning-enhanced fuzzy–potential field navigation for autonomous mobile robots," Engineering Applications of Artificial Intelligence, vol. 110, p. 104764, 2022.

[165] J. Dombi and E. Tóth-Laufer, "Reducing the computational requirements in Mamdani-type fuzzy control," Acta Polytechnica Hungarica, vol. 17, no. 3, p. 25–40, 2020.

[166] M. R. Karim, M. N. Reza, H. Jin, et al, "Application of LiDAR sensors for environment recognition: A review," Remote Sensing, vol. 16, no. 24, p. 4623, 2024.

[167] S. J. Al-Kamil and R. Szabolcsi, "Learning-based fuzzy navigation under dynamic obstacle interaction," IEEE Access, vol. 11, p. 118402–118417, 2023.

[168] Y. Kim, S. Choi, and D. Lee, "Adaptive interaction-aware navigation in complex environments," Robotics and Autonomous Systems, vol. 167, p. 104350, 2024.

[169] Y. Li, Z. Sun, and H. Chen, "Risk-aware navigation using fuzzy potential fields," IEEE Robotics and Automation Letters, vol. 5, no. 4, p. 6234–6241, 2020.

[170] S. Stavrinidis, P. Zacharia, and E. Xidias, "A fuzzy control strategy for multi-goal autonomous robot navigation," Sensors, vol. 25, no. 2, p. 446, 2025.

[171] F. Moosmann and C. Stiller, "Velodyne SLAM," in IEEE Intelligent Vehicles Symposium, p. 393–398, 2011.

[172] H. Kim, S. Park, and J. Lee, "Robust fuzzy navigation in dynamic environments," Robotics and Autonomous Systems, vol. 152, p. 104045, 2022.

[173] G. Shadrin, A. Krasavin, G. Nazenova, et al, "Application of compensation algorithms to control the speed and course of a four-wheeled mobile robot," Sensors, vol. 24, no. 22, p. 7233, 2024.

[174] M. Choi, S. Park, R. Lee, et al, "Energy efficient robot operations by adaptive control schemes," Oxford Open Energy, vol. 3, p. oiae012, 2024.

[175] A. Mazen, "Tuning of robot navigation performance using factorial design," Journal of Intelligent & Robotic Systems, 2022.

[176] D. Zhao, X. Wang, and Y. Chen, "Unified risk–energy optimization for autonomous robot navigation," Engineering Applications of Artificial Intelligence, vol. 124, p. 106536, 2024.

[177] C. E. Mohankumar and A. Manikandan, "Decentralized traffic management with federated edge AI: A reinforced transnet model for real-time vehicle object detection and collaborative route optimization," Discover Applied Sciences, 2025.

[178] S. Kim, J. Lee, and B. Kim, "Energy-efficient motion planning for mobile robots under dynamic constraints," IEEE Transactions on Industrial Electronics, vol. 69, no. 8, p. 8225–8236, 2022.

[179] M. S. Al-Radhi, T. G. Csapó, and G. Németh, "Adaptive refinements of pitch tracking and HNR estimation within a vocoder for statistical parametric speech synthesis," Applied Sciences,, vol. 9, no. 12, p. 2460, 2019.

[180] Y. Chen, Z. Liu, and J. Huang, "Interaction-aware motion planning for mobile robots in dynamic environments," Robotics and Autonomous Systems, vol. 164, p. 104302, 2023.

[181] A. Koubaa and B. Qureshi,, "Energy-efficient navigation strategies for autonomous mobile robots," IEEE Access, vol. 10, p. 104911–104924, 2022.

[182] S. Rahman and M. Hossain,, "Learning-based fuzzy control for autonomous robot navigation in uncertain environments," Applied Soft Computing, vol. 132, p. 109868, 2023.

[183] L. Wang, H. Sun, and J. Guo, "Dynamic obstacle-aware navigation using hybrid control strategies," Control Engineering Practice, vol. 134, p. 105507, 2023.

[184] P. Zhang, Y. Xu, and R. Li, "System-level adaptive control architecture for autonomous mobile robots," Journal of Intelligent & Robotic Systems, vol. 107, p. 58, 2023.

[185] J. Wu, Y. Wang, and H. Zhang, "Energy–speed co-optimization for autonomous mobile robots under dynamic constraints," IEEE Robotics and Automation Letters, vol. 9, no. 2, p. 1567–1574, 2024.

[186] L. Sun, Q. Li, and J. Pan,, "Interaction-aware motion planning for mobile robots in highly dynamic environments," Autonomous Robots, vol. 47, p. 1121–1137, 2023.

[187] R. Gupta and A. Sharma, "Learning-based adaptive control for energy-efficient mobile robot navigation," Control Engineering Practice, vol. 139, p. 105654, 2024.

[188] Y. Wang, Y. Fang, P. Lou, J. Yan, and N. Liu, "Deep reinforcement learning based path planning for mobile robot in unknown environment," Journal of Physics: Conference Series, p. 1–7, 2019.

[189] F. Fathinezhad, V. Derhami, and M. Rezaeian, "Supervised fuzzy reinforcement learning for robot navigation," Applied Soft Computing, vol. 40, p. 33–41, 2016.

[190] F. Rubio, F. Valero, and C. Llopis-Albert, "A review of mobile robots: Concepts, methods, theoretical framework, and applications," Frontiers in Physics, 2022.

[191] F. Kamil, V. Hong, W. Khaksar, M. Moghrabiah, N. Zulkifli, and S. Ahmad, "New robot navigation algorithm for arbitrary unknown dynamic environments based on future prediction and priority behavior," Expert Systems with Applications, p. 274–291, 2017.

[192] P. Kumar, S. Sahu, and D. R. Parhi, "A hybridized regression-adaptive ant colony optimization approach for navigation of humanoids in a cluttered environment," Applied Soft Computing, p. 565–585, 2018.

[193] D. Wang, Y. Hu, and T. Ma, "Mobile robot navigation with the combination of supervised learning in cerebellum and reward-based learning in basal ganglia," Cognitive Systems Research, p. 1–14, 2020.

[194] S. N. Herojit and K. Thongam, "Mobile robot navigation using MLP-BP approaches in dynamic environments," Arabian Journal for Science and Engineering, vol. 43, p. 8013–8028, 2018.

[195] B. Patle, D. Parhi, A. Jagadeesh, and S. Kashyap, "Matrix-binary codes based genetic algorithm for path planning of mobile robot," Computers & Electrical Engineering, vol. 67, pp. 708–728,, 2017.

[196] D. Wang, H. Wang, and L. Liu, "Unknown environment exploration of multi-robot system with the FORDPSO," Swarm and Evolutionary Computation, p. 157–174, 2016.

[197] R. Gonzalez, M. Fiacchini, and K. Iagnemma, "Slippage prediction for off-road mobile robots via machine learning regression and proprioceptive sensing," Robotics and Autonomous Systems, p. 85–93, 2018.

[198] L. Quan, L. Han, B. Zhou, S. Shen, and F. Gao, "Survey of UAV motion planning," IET Cyber-Systems and Robotics, vol. 2, p. 14–21, 2020.

[199] L. Claussmann, M. Revilloud, D. Gruyer, and S. Glaser, "A review of motion planning for highway autonomous driving," IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 5, p. 1826–1848, 2019.

[200] D. Gonzalez, J. Pérez, V. Milanes, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," IEEE Transactions on Intelligent Transportation Systems, vol. 17, p. 1135–1145, 2016.

[201] Z. Tang and H. Ma, "An overview of path planning algorithms," in 6th International Conference on Energy Science and Applied Technology, p. 1–10, 2021.

[202] D. Wang, J. Wu, M. Ha, M. Zhao, M. Li, and J. Qiao, "Advanced optimal tracking control with stability guarantee via novel value learning formulation," IEEE Transactions on Neural Networks and Learning Systems, 2022.

[203] V. Vu, T. Pham, and P. Dao,, "Disturbance observer-based adaptive reinforcement learning for perturbed uncertain surface vessels," SA Transactions, vol. 130, p. 277–292, 2022.

[204] L. Chen, C. Dong, S. He, and S. Dai, "Adaptive optimal formation control for unmanned surface vehicles with guaranteed performance using actor–critic learning architecture," International Journal of Robust and Nonlinear Control, 2023.

[205] K. Nguyen, V. Dang, D. Pham, and P. Dao, "Formation control scheme with reinforcement learning strategy for a group of multiple surface vehicles," International Journal of Robust and Nonlinear Control, 2023.

[206] A. H. Qureshi, Y. Miao, A. Simeonov, and M. C. Yip, "Motion planning networks: Bridging the gap between learning-based and classical motion planners," IEEE Transactions on Robotics, p. 48–66, 2023.

[207] J. Escobar-Naranjo, G. Caiza, P. Ayala, E. Jordan, C. A. Garcia Marcelo, and V. Garcia, "Autonomous navigation of robots: Optimization with DQN," Applied Sciences, vol. 13, no. 12, p. 7202, 2023.

[208] Z. Wu, Z. Meng, W. Zhao, and Z. Wu, "Fast-RRT: An RRT-based optimal path finding method," Applied Sciences, vol. 11, no. 24, pp. 1–18,, 2021.

[209] S. M. LaValle, "Planning Algorithms.," Cambridge University Press, 2006.

[210] J. Schwartz, M. Sharir, and S. Levine, "On the piano movers problem. II. General techniques for computing topological properties of real algebraic manifolds," Advances in Applied Mathematics, vol. 4, no. 3, p. 298–351, 1983.

[211] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in Autonomous Robot Vehicles," Springer, p. 396–404, 1986.

[212] I. Pavlović, D. Stamenković, V. Nikolić, A. Miltenović, N. Despenić, M. Atanasov, and G. Janevski, "Train obstacle detection system stabilization and stochastic vibrations analysis using the moment Lyapunov exponent method," Acta Polytechnica Hungarica, 2022.

[213] A. H. Qureshi and Y. Ayaz, "Intelligent bidirectional rapidly exploring random trees for optimal motion planning in complex cluttered environments," Robotics and Autonomous Systems,, vol. 68, p. 1–11, 2025.

[214] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel, "Benchmarking deep reinforcement learning for continuous control," in International Conference on Machine Learning, p. 1329–1338, 2016.

[215] Y. Zhao, Y. Zhang, and S. Wang, "A review of mobile robot path planning based on deep reinforcement learning algorithm," Journal of Physics: Conference Series, 2021.

[216] A. Taher, A. Koubaa, N. A. Mohamed, H. A. Ibrahim, Z. F. Ibrahim, M. Kazim, A. Ammar, B. Benjdira, A. M. Khamis, I. A. Hameed, and G. Casalino, "Drone deep reinforcement learning: A review," Electronics, vol. 10, 2021.

[217] M. Orsoni, S. Giovagnoli, S. Garofalo, S. Magri, M. Benvenuti, M. Mazzoni, and M. Benassi, "Preliminary evidence on machine learning approaches for clusterizing students'cognitive profile," Heliyon, 2023.

[218] B. Ajay, P. Venkata, Y. Eswar, V. Pratap, and K. Kuchi, "The power of generative AI: A review of requirements, models, input–output formats, evaluation metrics, and challenges," Future Internet, 2023.

[219] M. Sami and I. Mobin,, "A comparative study on variational autoencoders and generative adversarial networks," in International Conference of Artificial Intelligence and Information Technology (ICAIIT), p. 1–5, 2019.

[220] Y. C. e. al, "Human mobile robot interaction in the retail environment," Scientific Data, 2022.

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

[222] J. Peters and S. Schaal, "A survey on policy search for robotics," Foundations and Trends in Robotics, vol. 2, no. 1-2, p. 1–142, 2013.

[223] A. Yahya, A. Li, M. Kalakrishnan, Y. Chebotar, and S. Levine, "Collective robot reinforcement learning with distributed asynchronous guided policy search," in IEEE/RSJ International Conference on Intelligent Robots and Systems, p. 79–86, 2017.

[224] Y. Chebotar, M. Kalakrishnan, A. Yahya, A. Li, S. Schaal, and S. Levine, "Path integral guided policy search," in IEEE International Conference on Robotics and Automation, p. 3381–3388, 2017.

[225] A. Qureshi, J. Johnson, Y. Qin, T. Henderson, B. Boots, and M. Yip, "Composing task-agnostic policies with deep reinforcement learning," in International Conference on Learning Representations, 2020.

[226] X. Yu, S. Salimpour, J. P. Queralta, and T. Westerlund, "General-purpose deep learning detection and segmentation models for images from a LiDAR-based camera sensor," Sensors, vol. 23, no. 6, 2023.

[227] Y. Liu, W. Tao, S. Li, Y. Li, and Q. Wang, "A path planning method with a bidirectional potential field probabilistic step size RRT for a dual manipulator," Sensors, vol. 23, no. 11, 2023.

[228] M. Zucker, J. Kuffner, and J. A. Bagnell, "Adaptive workspace biasing for sampling-based planners," in IEEE International Conference on Robotics and Automation, p. 3757–3762, 2020.

[229] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," Machine Learning, vol. 8, no. 3–4, p. 229–256, 1992.

[230] D. Berenson, P. Abbeel, and K. Goldberg, "A robot path planning framework that learns from experience," in IEEE International Conference on Robotics and Automation,, p. 3671–3678, 2012.

[231] D. Coleman, I. Sucan, M. Moll, K. Okada, and N. Correll, "Experience-based planning with sparse roadmap spanners," in IEEE International Conference on Robotics and Automation, p. 900–905, 2015.

[232] G. Ye and R. Alterovitz, "Demonstration-guided motion planning," in Robotics Research, Springer, p. 291–307, 2017.

[233] C. Zhang, J. Huh, and D. Lee, "Learning implicit sampling distributions for motion planning," in IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3654–3661,, 2018.

[234] B. Ichter, J. Harrison, and M. Pavone, "Learning sampling distributions for robot motion planning," in IEEE International Conference on Robotics and Automation, p. 7087–7094, 2018.

[235] A. Francis, A. Faust, L. Chiang, C. Hsu, J. Kew, M. Fiser, and T. Lee, "Long-range indoor navigation with PRM-RL," IEEE Transactions on Robotics, p. 1115–1134, 2020.

[236] B. Angulo, A. Panov, and S. Levine, "Policy optimization to learn adaptive motion primitives in path planning with dynamic obstacles," IEEE Robotics and Automation Letters, p. 824–831, 2023.

[237] B. Eysenbach, R. Salakhutdinov, and K. Yakovlev, "Search on the replay buffer: Bridging planning and reinforcement learning," in Advances in Neural Information Processing Systems, 2019.

[238] H. Zhu, J. Liu, and Z. Li, "Deep reinforcement learning-based path planning for mobile robots in dynamic environments," Robotics and Autonomous Systems, vol. 125, p. 103437, 2020.

[239] Dudzik, S., "Application of the motion capture system to estimate the accuracy of a wheeled mobile robot localization," Energies, vol. 13, no. 23, p. 6437, 2020.

[240] F. Ducho, A. Babinec, M. Kajan, P. Beňo, M. Florek, T. Fico, and L. Jurišica, "Path planning with modified A* algorithm for a mobile robot," Procedia Engineering, vol. 96, p. 59–69, 2014.

[241] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," introduction to variational autoencoders, vol. 12, no. 4, p. 307–392, 2019.

[242] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "β-VAE: Learning basic visual concepts with a constrained variational framework," in International Conference on Learning Representations, 2017.

[243] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive auto-encoders: Explicit invariance during feature extraction," n International Conference on Machine Learning, p. 833–840, 2011.

[244] I. Pavlović, D. Stamenković, V. Nikolić, A. Miltenović, N. Despenić, M. Atanasov, and G. Janevski, "Train obstacle detection system stabilization and stochastic vibrations analysis using the moment Lyapunov exponent method," Acta Polytechnica Hungarica, 2022.

[245] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," Journal of Machine Learning Research, , vol. 15, pp. Journal of Machine Learning Research, vol. 15, no. 1, pp. 1929–1958, 2014., 2014.

[246] B. W. Tang, Z. X. Zhu, and J. Luo, "Hybridizing particle swarm optimization and differential evolution for the mobile robot global path planning," International Journal of Advanced Robotic Systems, vol. 13, p. 1–17, 2016.

[247] Y. Zhuang, Y. L. Sun, and W. Wang, "Mobile robot hybrid path planning in an obstacle-cluttered environment based on steering control and improved distance propagating," International Journal of Innovative Computing, Information and Control, vol. 8, p. 095–4109, 2012.

[248] M. A. Contreras-Cruz, V. Ayala-Ramirez, and U. H. Hernandez-Belmonte, "Mobile robot path planning using artificial bee colony and evolutionary programming," Applied Soft Computing, vol. 30, p. 319–328, 2015.

[249] M. Park, J. Jeon, and M. Lee, "Obstacle avoidance for mobile robots using artificial potential field," in Proceedings of the IEEE International Symposium, p. 1530–1535, 2001.

[250] O. Montiel, U. Orozco-Rosas, and R. Sepúlveda, "Path planning for mobile robots using bacterial potential field for avoiding static and dynamic obstacles," Expert Systems with Applications, vol. 42, no. 12, p. 5177–5191, 2015.

[251] E. Balasubramanian, E. Elangovan, P. Tamilarasan, G. R. Kanagachidambaresan, and D. Chutia, "Optimal energy efficient path planning of UAV using hybrid MACO-MEA* algorithm: Theoretical and experimental approach," Journal of Ambient Intelligence and Humanized Computing, 2022.

[252] Orthey, A., Chamzas, C., and Kavraki, L. E., "Sampling-based motion planning: A comparative review," Annual Review of Control, Robotics, and Autonomous Systems, vol. 7, p. 285–310, 2024.

[253] J. Liang, S. Wang, and B. Wang, "Online motion planning for fixed-wing aircraft in precise automatic landing on mobile platforms," Drones, 2023.

[254] L. Liu, X. Wang, X. Yang, H. Liu, J. Li, and P. Wang, "Path planning techniques for mobile robots: Review and prospect," Expert Systems with Applications, p. 227, 2023.

[255] N. H. Thai and T. T. K. Ly, "NURBS curve trajectory tracking control for differential-drive mobile robot by a linear state feedback controller," in International Conference on Engineering Research and Applications, p. 685–696, 2021.

[256] D. F. Rogers, "An Introduction to NURBS," Morgan Kaufmann Publishers, 2001.

[257] S. J. Al-Kamil and T. Szakács, "Design all-wheel drive vehicles based on differential speed control systems," Mechatronic Systems and Control, p. 25–29, 2021.

[258] M. Cherry and N. Win, "Position and velocity control for two-wheel differential drive mobile robot," International Journal of Science, Engineering and Technology Research, vol. 5, no. 9, p. 2849–2855, 2016.

[259] S. Xie, J. Hu, P. Bhowmick, Z. Ding, and F. Arvin, "Distributed motion planning for safe autonomous vehicle overtaking via artificial potential field," IEEE Transactions on Intelligent Transportation Systems, vol. 23, p. 21531–21547, 2022.

[260] F. Gul, I. Mir, L. Abualigah, P. Sumari, and A. Forestiero, "A consolidated review of path planning and optimization techniques: Technical perspectives and future directions," Electronics, vol. 10, p. 2250, 2021.

[261] Z. Jian, S. Zhang, S. Chen, Z. Nan, and N. Zheng, "A global-local coupling two-stage path planning method for mobile robots," IEEE Robotics and Automation Letters, vol. 6, p. 5349–5356, 2021.

[262] Z. Qi, T. Wang, J. Chen, D. Narang, Y. Wang, and H. Yang, "Learning-based path planning and predictive control for autonomous vehicles with low-cost positioning," IEEE Transactions on Intelligent Vehicles, 2021.

[263] Z. Zhang, R. Wu, Y. Pan, Y. Wang, X. Guan, J. Hao, J. Zhang, and G. Li, "A robust reference path selection method for path planning algorithm," IEEE Robotics and Automation Letters, vol. 7, p. 4837–4844, 2022.

[264] J. Wen, J. Yang, and T. Wang, “Path planning for autonomous underwater vehicles under the influence of ocean currents based on a fusion heuristic algorithm, "IEEE Transactions on Vehicular Technology," vol. 70, p. 8529–8544, 2021.

[265] A. Awad, A. Hawash, and B. Abdalhaq, "A genetic algorithm and swarm-based binary decision diagram reordering optimizer reinforced with recent operators," IEEE Transactions on Evolutionary Computation, 2021.

[266] J. Kennedy and R. Eberhart, "Particle swarm optimization,” in Proceedings of the International Conference on Neural Networks," Perth, Australia, vol. 4, p. 1942–1948, 1995.

[267] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: Artificial bee colony algorithm and applications," Artificial Intelligence Review, vol. 42, p. 21–57, 2014.

[268] E. Liu, X. Yao, M. Liu, and H. Jin, "AGV path planning based on improved grey wolf optimization algorithm and its implementation prototype platform," Computer Integrated Manufacturing Systems, vol. 24, p. 2779–2791, 2018.

[269] J. Tang, G. Liu, and Q. Pan, "A review on representative swarm intelligence algorithms for solving optimization problems, applications and trends," IEEE/CAA Journal of Automatica Sinica, vol. 8, p. 1627–1643, 2021.

[270] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," Journal of Global Optimization, vol. 11, p. 341–359, 1997.

[271] Z. Wu, Z. Meng, W. Zhao, and Z. Wu, "Fast-RRT: An RRT-based optimal path finding method," Applied Sciences, 2021.

[272] S. Janabi and D. Vinke, "Integration of the artificial potential field approach with simulated annealing for robot path planning," in IEEE International Symposium on Intelligent Control, p. 536–541, 1993.

[273] R. Rodríguez-Puente and M. S. Lazo-Cortés, "Algorithm for shortest path search in geographic information systems by using reduced graphs," SpringerPlus, vol. 2, p. 1–13, 2013.

[274] N. L. Gang Chen, "Path planning for manipulators based on an improved probabilistic roadmap method," Robotics and Computer-Integrated Manufacturing, vol. 72, 2021.

[275] S. Dudzik, "Application of the motion capture system to estimate the accuracy of a wheeled mobile robot localization," Energies, 2020.

[276] S. Ghadami, H. Biglarian, H. Beyrami, and M. Salimi, "Optimization of multilateral well trajectories using pattern search and genetic algorithms," Results in Engineering, vol. 16, 2022.

[277] M. Field, D. Stirling, F. Naghdy, and Z. Pan, "Motion capture in robotics review," n IEEE International Conference on Control and Automation, p. 1697–1702, 2009.

[278] H. M. Hondori and M. Khademi, "A review on technical and clinical impact of Microsoft Kinect on physical therapy and rehabilitation," Journal of Medical Engineering, p. 1–16, 2014.

[279] W. Geng and G. Yu, "Reuse of motion capture data in animation: A review," in Lecture Notes in Computer Science," Springer, p. 620–629, 2003.

[280] S. Barris and C. Button, "A review of vision-based motion analysis in sport," Sports Medicine, vol. 38, p. 1025–1043, 2008.

[281] M. Bortolini, M. Faccio, M. Gamberi, and F. Pilati,, "Motion analysis system for production and ergonomics assessment in manufacturing processes," Computers and Industrial Engineering, 2020.

[282] M. P. Christiansen, M. S. Laursen, R. N. Jørgensen, and I. A. Hameed, "Collaborative model-based design of automated and robotic agricultural vehicles in the Crescendo tool," Computer Science (Robotics), 2018.

[283] P. Kumar, P. Sukant, and K. Bisoy, "Localization strategies for autonomous mobile robots: A review," Journal of King Saud University – Computer and Information Sciences, vol. 34, no. 8, p. 16, 2022.

[284] J. Abdor-Sierra, E. Merchán-Cruz, and R. Rodríguez-Cañizo, "A comparative analysis of metaheuristic algorithms for solving the inverse kinematics of robot manipulators," Results in Engineering, vol. 16, 2022.

[285] M. Branicky, S. LaValle, K. Olson, and L. Yang, "Quasi-randomized path planning," in IEEE International Conference on Robotics and Automation," p. 1481–1487, 2001.

[286] LaValle, S. M.; Kuffner, J. J., "Randomized kinodynamic planning," International Journal of Robotics Research, vol. 20, no. 5, p. 378–400, 2001.

[287] J. J. Kuffner, S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in IEEE International Conference on Robotics and Automation (ICRA), vol. 2, p. 995–1001, 2000.

[288] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," International Journal of Robotics Research, vol. 30, no. 7, p. 846–894, 2011.

[289] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," Advances in Neural Information Processing Systems, 2015.

[290] D. Hsu, J.-C. Latombe, and H. Kurniawati, "On the probabilistic foundations of probabilistic roadmap planning," International Journal of Robotics Research, 2006.

[291] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Batch informed trees (BIT*): Sampling-based optimal planning via heuristically guided search," in Proc. IEEE Int. Conf. Robotics and Automation (ICRA),, p. 3067–3074., 2015.

[292] H. Kurniawati and D. Hsu, "Workspace importance sampling for probabilistic roadmap planning," n Proc. IEEE Int. Conf. Robotics and Automation (ICRA), p. 5089–5094, 2014.

[293] J. P. van den Berg and M. H. Overmars, "Using workspace information as a guide to non-uniform sampling in probabilistic roadmap planners," IJRR, 2005.

[294] Y. Yang and O. Brock, "Adapting the sampling distribution in PRM planners based on an approximated medial axis," in Proc. IEEE Int. Conf. Robotics and Automation (ICRA), p. 4405–4410, 2004.

[295] R. Geraerts and M. H. Overmars, "Sampling techniques for probabilistic roadmap planners," IAS, 2004.

[296] M. Zucker, J. Kuffner, and J. A. Bagnell, "Adaptive workspace biasing for sampling-based planners," in Proc. IEEE Int. Conf. Robotics and Automation (ICRA), p. 2813–2818, 2008.

[297] D. Berenson, P. Abbeel, and K. Goldberg, "A robot path planning framework that learns from experience," in IEEE ICRA, 2012.

[298] D. Coleman, I. A. Sucan, M. Moll, K. Okada, and N. Correll, "Experience-based planning with sparse roadmap spanners," in IEEE ICRA, 2015.

[299] Y. Li and K. E. Bekris, "Learning approximate cost-to-go metrics to improve sampling-based motion planning," in IEEE ICRA, 2011.

[300] I. Baldwin and P. Newman, "Non-parametric learning for natural plan generation," in IEEE IROS, 2010.

[301] G. Ye and R. Alterovitz, "Demonstration-guided motion planning," ISRR, 2015.

[302] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," Computer Science Department Technical Report, Iowa State University, 1998.

[303] M. Rickert, O. Brock, and A. Knoll, "Balancing exploration and exploitation in motion planning," in IEEE International Conference on Robotics and Automation (ICRA), p. 2812–2817, 2008.

[304] C. Urmson and R. Simmons, "Approaches for heuristically biasing RRT growth," in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), vol. 2, p. 178–1183, 2003.

[305] D. Ferguson and A. Stentz, "Anytime RRTs," in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)," p. 5369–5375, 2006.

[306] Gammell, J. D., Srinivasa, S. S., and Barfoot, T. D, "Batch informed trees (BIT): Sampling-based optimal planning via heuristically guided search,," Robotics and Autonomous Systems, vol. 130, p. 103553, 2020.

[307] A. H. Qureshi and Y. Ayaz, "Potential functions based sampling heuristic for optimal path planning," Autonomous Robots, vol. 40, no. 6, p. 1079–1093, 2016.

[308] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot,, "Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), p. 2997–3004, 2014.

[309] J. Gammell, T. Barfoot, and S. Srinivasa, "Batch informed trees for sampling-based optimal planning in complex environments," Robotics and Autonomous Systems, vol. 130, p. 103553, 2020.

[310] A. Faust, K. Oslund, A. Francis, L. Tapia, and J. Davidson, "PRM-RL: Long-range robotic navigation tasks by combining reinforcement learning and sampling-based planning," Robotics and Autonomous Systems, vol. 123, pp. 103336,, 2020.

[311] Y. Ichter, M. Toussaint, and M. Pavone, "Learning-based sampling for robot motion planning with neural guidance," Engineering Applications of Artificial Intelligence, vol. 104, p. 104394, 2021.

[312] X. Zhang, Y. Wang, and H. Liu,, "Experience-guided sampling-based motion planning in complex environments," Expert Systems with Applications, vol. 195, p. 116567, 2022.

[313] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "β-VAE: Learning basic visual concepts with a constrained variational framework," International Conference on Learning Representations, 2017.

[314] A. Alemi, I. Fischer, J. Dillon, and K. Murphy, "Deep variational information bottleneck," International Conference on Learning Representations, 2017.

[315] H. Zhao, Y. Li, and X. Zhang, "Variational autoencoder-based representation learning for robotic navigation in dynamic environments," Engineering Applications of Artificial Intelligence, vol. 120, p. 105936, 2023.

[316] X. Liu, J. Wang, and H. Sun,, "Variational latent-space modeling for autonomous robot path planning under uncertainty," Expert Systems with Applications, vol. 229, p. 120432, 2024.