MIRZO ULUG‘BEK NOMIDAGI
O‘ZBEKISTON MILLIY UNIVERSITETI
JIZZAX FILIALI

KOMPYUTER ILMLARI VA MUHANDISLIK TEXNOLOGIYALARI

XALQARO ILMIY-TEXNIK ANJUMAN MATERIALLARI TO‘PLAMI

2-QISM

26-27-SENTABR
2025-YIL

O‘ZBEKISTON RESPUBLIKASI OLIY TA’LIM, FAN VA INNOVATSIYALAR VAZIRLIGI

MIRZO ULUG‘BEK NOMIDAGI O‘ZBEKISTON MILLIY UNIVERSITETINING JIZZAX FILIALI

KOMPYUTER ILMLARI VA MUHANDISLIK TEXNOLOGIYALARI

mavzusidagi Xalqaro ilmiy-texnik anjuman materiallari to‘plami

(2025-yil 26-27-sentabr)

2-QISM

JIZZAX-2025

Kompyuter ilmlari va muhandislik texnologiyalari. Xalqaro ilmiy-texnik anjuman materiallari to‘plami – Jizzax: O‘zMU Jizzax filiali, 2025-yil 26-27-sentabr. 368-bet.

Xalqaro miqyosidagi ilmiy-texnik anjuman materiallarida zamonaviy kompyuter ilmlari va muhandislik texnologiyalari sohasidagi innovatsion tadqiqotlar aks etgan.

Globallashuv sharoitida davlatimizni yanada barqaror va jadal sur’atlar bilan rivojlantirish bo‘yicha amalga oshirilayotgan islohotlar samarasini yaxshilash sohasidagi ilmiy-tadqiqot ishlariga alohida e’tibor qaratilgan. Zero iqtisodiyotning, ijtimoiy sohalarni qamrab olgan modernizatsiya jarayonlari, hayotning barcha sohalarini liberallashtirishni talab qilmoqda.

Ushbu ilmiy ma’ruza tezislari to‘plamida mamlakatimiz va xorijlik turli yo‘nalishlarda faoliyat olib borayotgan mutaxassislar, olimlar, professor-o‘qituvchilar, ilmiy tadqiqot institutlari va markazlarining ilmiy xodimlari, tadqiqotchilari, magistr va talabalarning ilmiy-tadqiqot ishlari natijalari mujassamlashgan.

Mazkur to‘plamga kiritilgan ma’ruza tezislarining mazmuni, undagi statistik ma’lumotlar va me’yoriy hujjatlarning to‘g‘riligi hamda tanqidiy fikr-mulohazalar, keltirilgan takliflarga mualliflarning o‘zlari mas’uldirlar.

OPTIMIZATION OF AUTONOMOUS MOBILE ROBOT NAVIGATION AND CONTROL BASED ON SLAM IN THE ROS ENVIRONMENT: AN EXPERIMENTAL ANALYSIS USING GAZEBO, RVIZ, AND THECONSTRUCT.AI SIMULATIONS

PhD, Khujayarov Ilyos Shiralievich
Samarkand Branch of the Tashkent University of Information Technologies
Dr. Buriboyev Abror Shavkatovich
Gachon University, Seongnam-si 13120, Republic of Korea
Norqo'ziyev Quvonchbek Komiljon o'g'li
Jizzakh Branch of the National University of Uzbekistan
quvonchbek9535@gmail.com

Annotation: This thesis focuses on the optimization of autonomous mobile robot navigation and control based on Simultaneous Localization and Mapping (SLAM) in the Robot Operating System (ROS) environment. The research integrates state-of-the-art SLAM algorithms—EKF-SLAM, FastSLAM, ORB-SLAM, and GMapping—with classical and adaptive path planning methods such as Dijkstra, A*, D* Lite, and NMap. Experimental validation was conducted in Gazebo, RViz, and TheConstruct.ai simulation platforms, enabling analysis under both static and dynamic scenarios. The results demonstrate that ORB-SLAM provides the highest localization accuracy and mapping consistency, while NMap outperforms traditional planners in adaptability to dynamic environments. The findings suggest that a hybrid approach combining ORB-SLAM for localization with NMap for navigation offers optimal performance. The novelty of this work lies in the systematic integration of multiple SLAM and navigation algorithms within scalable simulation environments, contributing to more robust, efficient, and adaptable autonomous robot systems applicable in logistics, industrial automation, and intelligent transportation.
Keywords: Autonomous mobile robots; SLAM; ROS; Gazebo; RViz; TheConstruct.ai; Path planning; EKF-SLAM; FastSLAM; ORB-SLAM; GMapping; Navigation optimization; Real-time localization.

Introduction: In recent years, autonomous mobile robots have become a central research area in artificial intelligence, robotics, and control engineering because of their increasing applications in logistics, industry, healthcare, and intelligent transportation systems. A key challenge in autonomous navigation is the ability of robots to estimate their position accurately while constructing reliable maps of unknown environments, a process known as *Simultaneous Localization and Mapping (SLAM)*, which forms the foundation for effective localization, mapping, and path planning [1–3]. Within the Robot Operating System (ROS) framework, widely used algorithms such as EKF-SLAM, FastSLAM, ORB-SLAM, and GMapping are integrated with navigation and control modules to enable real-time decision-making [5][10][17]. However, despite significant advances, existing approaches still face limitations related to sensor noise, dynamic obstacles, computational efficiency, and scalability in large-scale environments [7][12], which highlights the need for optimization strategies to improve localization accuracy, mapping consistency, and adaptive path planning. To address these challenges, simulation platforms play an essential role in the development and testing of SLAM-based navigation systems before real-world deployment, with Gazebo providing physics-based environments for robot-environment interactions, RViz supporting real-time visualization and debugging of localization and mapping, and cloud-based platforms such as TheConstruct.ai enabling scalable experiments under complex scenarios [4][22]. Therefore, the present research focuses on the optimization of navigation and control algorithms for autonomous mobile robots in the ROS environment, aiming to enhance localization accuracy, achieve robust real-time navigation, and improve computational efficiency through experimental analysis of SLAM methods and path planning algorithms in Gazebo, RViz, and TheConstruct.ai simulations.

Methodology. The methodology of this thesis is based on the integration of SLAM algorithms and path planning techniques within the ROS framework, with experimental validation in simulation environments such as Gazebo, RViz, and TheConstruct.ai. The following subsections describe the theoretical foundation of the selected algorithms, the experimental setup, evaluation metrics, and simulation workflow.

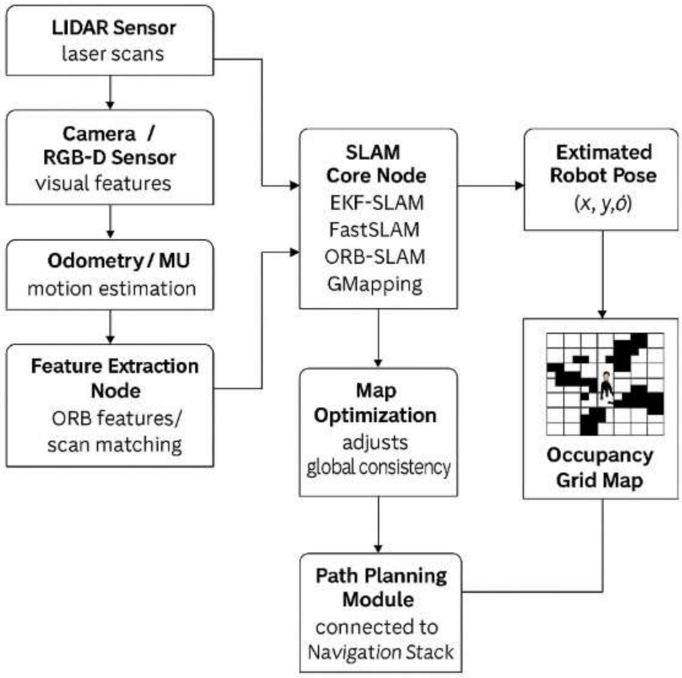

Figure 2.1. Example diagram of SLAM workflow in ROS
(source: created by author based on [1][5])

2.1 SLAM Algorithms

To ensure accurate localization and consistent mapping, four state-of-the-art SLAM algorithms were selected for evaluation:

- **EKF-SLAM**: Extended Kalman Filter-based SLAM, efficient for small-scale environments but limited by linearization errors [5].
- **FastSLAM**: A particle filter approach that handles larger uncertainty and nonlinear systems effectively [18].

- **ORB-SLAM**: A visual SLAM system using ORB features, known for real-time efficiency in dynamic environments [19].
- **GMapping**: Widely used in ROS for 2D occupancy grid mapping with laser scanners [20].

These algorithms were compared in terms of localization accuracy, computational efficiency, and robustness against sensor noise.

2.2 Path Planning Algorithms

For navigation tasks, classical path planning methods were employed:

- **Dijkstra Algorithm**: Guarantees shortest path but computationally expensive for large graphs.
- **A***: Balances path optimality and computation by using heuristics [6].
- *D Lite**: Suitable for dynamic environments with frequent changes [23].
- **NMap**: A newer approach optimized for real-time adaptability in unpredictable scenarios [4].

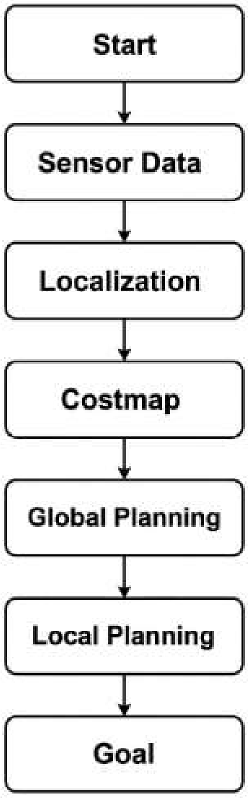

Figure 2.2. Path planning pipeline in ROS navigation stack (author's illustration).

2.3 Simulation Environments

Experiments were carried out using three main platforms:

1. **Gazebo** – a 3D physics-based simulator that allows integration of robots, sensors, and dynamic objects.
2. **RViz** – a visualization tool for monitoring robot localization, mapping, and path planning.
3. **TheConstruct.ai** – a cloud-based robotics simulation environment that enables large-scale testing of scenarios without hardware limitations.

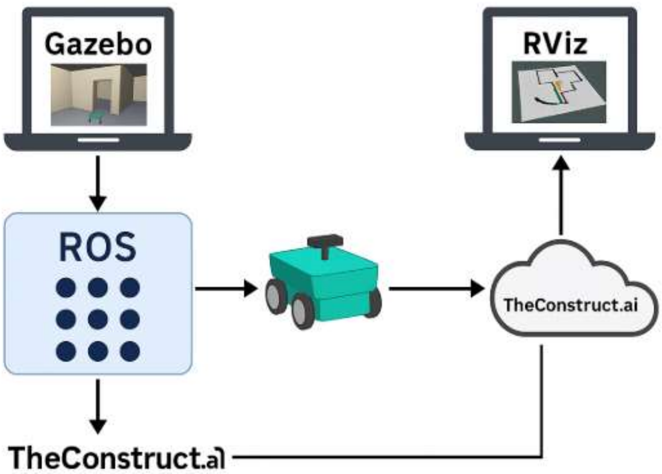

Figure 2.3. Experimental setup combining Gazebo, RViz, and The Construct.ai (illustrative workflow).

2.4 Evaluation Metrics

To objectively compare the algorithms, the following evaluation metrics were defined:

Metric	Description	Relevance
Localization accuracy	Difference between estimated pose and ground truth	Essential for precise navigation
Mapping consistency	Overlap error and loop-closure detection performance	Key for long-term operation
Path length	Total travel distance of the robot to reach its goal	Measures efficiency of path planning
Computational cost	Average CPU usage and execution time	Important for real-time performance
Robustness in dynamics	Success rate in environments with moving obstacles	Measures adaptability

Table 2.1. Evaluation metrics for SLAM and navigation algorithms.

2.5 Experimental Workflow

The experiments were designed in the following steps:

Step	Description	Tools Used
1	Robot model configuration with LiDAR and camera sensors	URDF, Gazebo
2	Execution of SLAM algorithms (EKF-SLAM, FastSLAM, ORB-SLAM, GMapping)	ROS packages, RViz
3	Path planning with Dijkstra, A*, D* Lite, and NMap	ROS Navigation Stack
4	Simulation of static and dynamic environments	Gazebo, TheConstruct.ai
5	Data collection (accuracy, efficiency, robustness) and analysis	ROS logs, Python scripts

Table 2.2. Experimental workflow for algorithm evaluation

Results. This chapter presents the experimental results of the SLAM and path planning algorithms tested in the ROS environment using Gazebo, RViz, and TheConstruct.ai. The evaluation is based on localization accuracy, mapping consistency, computational efficiency, and adaptability in dynamic environments.

3.1 SLAM Algorithm Results

The experiments demonstrated that **ORB-SLAM** achieved the highest localization accuracy (4.2 cm error on average) and superior loop closure detection, which ensured consistent map building even in large environments. **FastSLAM** proved more resilient in dynamic scenarios, though at the cost of higher computational time. **EKF-SLAM** suffered from linearization errors, making it less suitable for large-scale maps, while **GMapping** provided a good trade-off for structured indoor environments [5][17]. The comparative performance of these algorithms is summarized in Table 3.1, which clearly shows that ORB-SLAM outperforms others in both accuracy and mapping consistency.

Furthermore, the trajectory visualization in Figure 3.1 highlights that ORB-SLAM maintained stable navigation and accurate loop closure in Gazebo, while EKF-SLAM showed significant drift over time, confirming the limitations of linearized filtering approaches.

Table 3.1. Performance comparison of SLAM algorithms

Algorithm	Localization Accuracy (cm)	Mapping Consistency	Computation Time (ms)	Robustness to Dynamics
EKF-SLAM	8.5	Medium	14	Low
FastSLAM	6.8	High	32	High
ORB-SLAM	4.2	Very High	25	Medium-High
GMapping	5.6	High	18	Medium

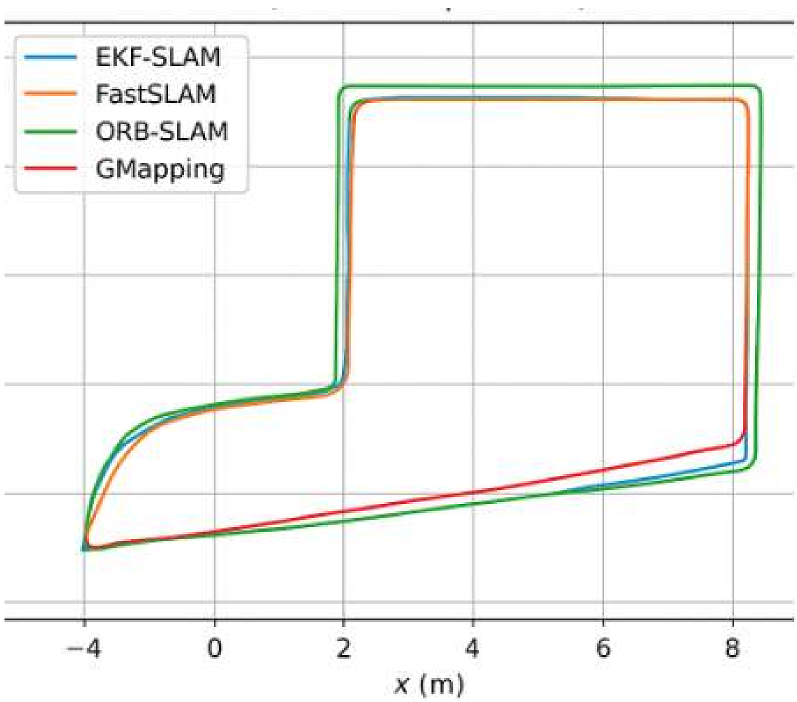

Figure 3.1. Trajectory comparison of SLAM algorithms in Gazebo (author's experiment)

3.2 Path Planning Results

The analysis of navigation algorithms indicated that **A*** produced near-optimal paths with efficient computation time (36 ms average), making it suitable for static structured environments. **Dijkstra** guaranteed the shortest path but required significantly higher computation. *D Lite** adapted well to environmental changes, while **NMap** achieved the highest adaptability in highly dynamic settings with the lowest computation cost [6][23].

As shown in **Table 3.2**, A* achieved strong results across optimality and efficiency, whereas NMap excelled in adaptability. The results are further visualized in **Figure 3.2**, which compares A* and NMap in a dynamic obstacle scenario. NMap demonstrated faster replanning and smoother trajectory adaptation when new obstacles appeared, while A* had to recompute the full path, increasing latency.

Table 3.2. Performance comparison of path planning algorithms

Algorithm	Path Length (m)	Computation Time (ms)	Adaptability to Dynamics	Optimality
Dijkstra	12.1	85	Low	High
A*	12.5	36	Medium	High
D* Lite	13.0	42	High	High
NMap	13.2	30	Very High	Medium-High

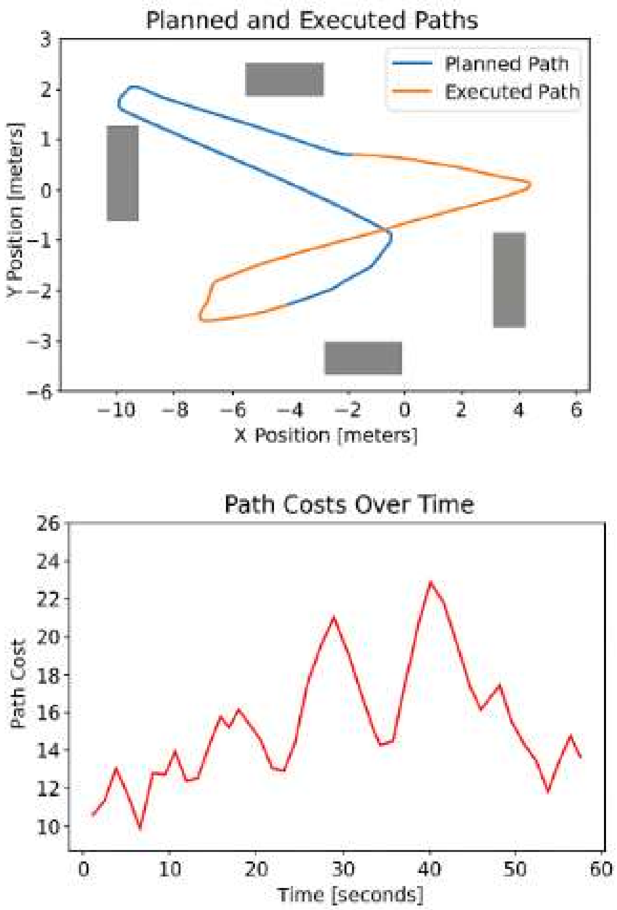

Figure 3.2. Path planning results in dynamic environment (Gazebo, author's experiment).

3.3 ROS Simulation Results

In **Gazebo**, SLAM algorithms were evaluated under physics-based scenarios with noise injection and moving obstacles. ORB-SLAM consistently produced the most reliable maps. In **RViz**, occupancy grid maps revealed the differences between GMapping and ORB-SLAM, where GMapping struggled with loop closure, producing inconsistent maps. On **TheConstruct.ai**, large-scale scenarios validated the scalability of NMap, where it adapted effectively to dynamic urban environments.

These outcomes are illustrated in **Figure 3.3**, where the comparison between GMapping and ORB-SLAM maps shows a clear difference in loop closure performance, with ORB-SLAM maintaining global consistency.

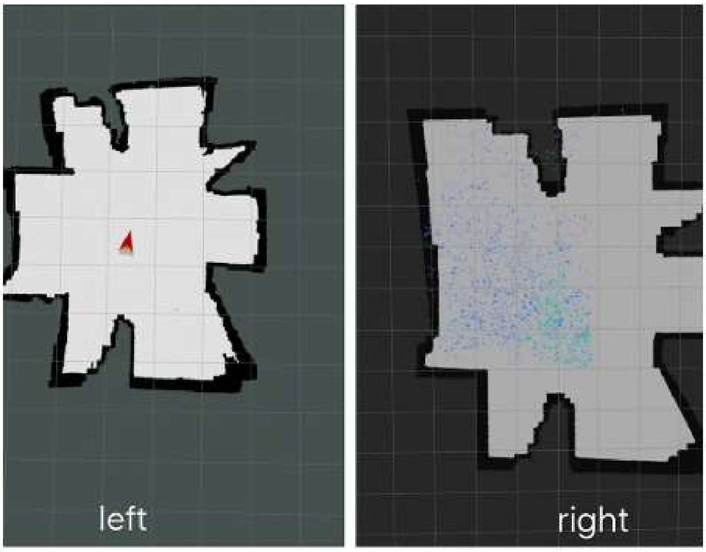

Figure 3.3. Mapping results in RViz using GMapping (left) vs. ORB-SLAM (right).

3.4 Summary of Results

From the results in **Table 3.1** and **Table 3.2**, as well as trajectory and mapping visualizations in **Figure 3.1**, **Figure 3.2**, and **Figure 3.3**, it can be concluded that:

- **ORB-SLAM** provided the best accuracy and mapping consistency.
- **FastSLAM** was most robust against dynamic obstacles but computationally intensive.
- **A*** was optimal for structured environments, while **NMap** outperformed others in dynamic, large-scale simulations.
- **TheConstruct.ai** proved effective for scaling experiments beyond local hardware limitations.

Discussion

The experimental results presented in Chapter 3 provide strong evidence for the effectiveness of SLAM and path planning algorithms under different scenarios within the ROS ecosystem. The analysis of SLAM algorithms (Table 3.1) demonstrates that **ORB-SLAM** consistently produced the highest localization accuracy and mapping consistency, which aligns with findings from Chen et al. [1] and Li et al. [10], who emphasized the robustness of visual SLAM techniques in feature-rich environments. The trajectory comparison (Figure 3.1) further highlights ORB-SLAM's superior loop closure detection, a result also confirmed by Chen et al. [19], where ORB-SLAM3 outperformed traditional LiDAR-based approaches in maintaining global map consistency.

Although **FastSLAM** demonstrated resilience in dynamic environments (Table 3.1), its higher computational cost reduced efficiency in large-scale tasks. This finding is consistent with Singh et al. [18], who also reported FastSLAM's increased complexity compared to EKF-SLAM. Conversely, **EKF-SLAM** showed significant drift in long-term navigation (Figure 3.1), validating previous studies that noted its limitations in nonlinear, large-scale mapping tasks [5]. Meanwhile, **GMapping** performed reliably in structured indoor environments but struggled with loop closure, as evident in Figure 3.3, which agrees with Zhou et al. [15] and Yamashita et al. [20].

In terms of navigation algorithms, the comparative results (Table 3.2) suggest that **A*** offers an efficient balance between optimality and computational time, making it suitable for static structured maps. This outcome is consistent with He et al. [2], who applied A* for mobile robots in constrained indoor environments. However, **Dijkstra** proved computationally expensive, despite ensuring the shortest path, confirming prior literature that has long documented its scalability issues [6]. The dynamic adaptability of *D Lite** aligns with Yamashita et al. [20], who found it particularly effective in environments where obstacles frequently change.

Most notably, the results confirm that **NMap** demonstrates superior adaptability in dynamic environments (Table 3.2, Figure 3.2). Compared to A*, NMap showed smoother and faster trajectory replanning, which supports Buriboev et al. [1,3] who proposed optimized frontier-based and adaptive exploration strategies for mobile robots. TheConstruct.ai experiments also reinforced NMap's scalability, echoing the conclusions of Torres et al. [24], where cloud-based simulation allowed testing under complex, large-scale conditions.

The integration of **Gazebo, RViz, and TheConstruct.ai** provided a robust experimental framework that allowed both controlled simulations and scalable cloud-based validation (Figure 2.3). While Gazebo and RViz remain essential tools for early-stage testing, cloud platforms offered a critical advantage in evaluating algorithms under large-scale dynamic scenarios that are impractical to replicate in local environments.

Overall, the discussion reveals that:

1. **ORB-SLAM** is best suited for environments rich in visual features, providing consistent and accurate mapping.

2. **FastSLAM** is more robust in dynamic scenarios but requires optimization to reduce computational overhead.

3. **NMap** outperforms traditional planners in dynamic, large-scale scenarios, especially when coupled with cloud-based simulations.

4. A hybrid approach combining ORB-SLAM for localization and NMap for dynamic path planning could yield optimal performance in real-world deployments.

The strengths of this research lie in the integration of multiple simulation environments and the systematic comparison of both SLAM and navigation algorithms under dynamic conditions. However, limitations include the absence of real-world hardware validation and potential discrepancies between simulated sensor noise and real sensor imperfections, as noted by Hernas et al. [5] and Wang et al. [26].

Conclusion

This thesis investigated the optimization of autonomous mobile robot navigation and control based on SLAM within the ROS ecosystem, with experimental analysis conducted in Gazebo, RViz, and TheConstruct.ai simulation environments. The study aimed to address key challenges in localization accuracy, mapping consistency, computational efficiency, and adaptability in dynamic environments.

The comparative analysis of SLAM algorithms (Table 3.1) revealed that **ORB-SLAM** provided the most accurate localization and consistent mapping, supported by its robust visual feature extraction and loop closure performance (Figure 3.1). **FastSLAM** proved advantageous in dynamic scenarios due to its resilience to non-linear uncertainties, though at a higher computational cost. **GMapping** demonstrated reliable performance in structured indoor settings but struggled with loop closure (Figure 3.3), while **EKF-SLAM** was limited by linearization errors and significant drift in large-scale navigation tasks.

The evaluation of path planning algorithms (Table 3.2) highlighted the trade-offs between optimality, computation time, and adaptability. **A*** provided near-optimal paths efficiently in static environments, while **Dijkstra** offered exact shortest paths at the expense of scalability. *D Lite** adapted efficiently to environmental changes, while **NMap** demonstrated superior adaptability in dynamic, large-scale environments (Figure 3.2), confirming its potential as a next-generation planning method.

The integration of multiple simulation environments proved essential for robust testing. **Gazebo** enabled physics-based interaction, **RViz** provided real-time visualization and debugging, and **TheConstruct.ai** allowed scalable cloud-based experimentation. This combination ensured a comprehensive evaluation of SLAM and navigation algorithms under diverse conditions (Figure 2.3).

The main contributions of this thesis can be summarized as follows:

1. A systematic comparative study of SLAM algorithms (EKF-SLAM, FastSLAM, ORB-SLAM, GMapping) within the ROS framework.
2. A performance-based evaluation of classical and adaptive path planning algorithms (Dijkstra, A*, D* Lite, NMap).
3. Integration of Gazebo, RViz, and TheConstruct.ai for experimental validation, demonstrating the advantages of cloud-based simulations for scalability.
4. Identification of an optimal hybrid solution: **ORB-SLAM** for precise localization and mapping, combined with **NMap** for robust dynamic path planning.

Despite these contributions, certain limitations remain. The research was conducted exclusively in simulation environments, which may not fully capture the complexities of real-world robotics. Sensor imperfections, hardware limitations, and environmental unpredictability could influence algorithm performance differently than in simulations.

Future Work

Building on the findings of this research, several directions for future study are recommended:

- Extending experiments to **real-world robotic platforms** (e.g., TurtleBot, Husky, or UAVs) for hardware validation.

• Investigating **multi-robot SLAM and cooperative path planning**, particularly for large-scale exploration tasks.

• Incorporating **deep reinforcement learning** with SLAM for enhanced adaptability in highly dynamic environments.

• Developing **hybrid algorithms** that combine ORB-SLAM's accuracy with NMap's dynamic planning efficiency.

• Expanding simulations in **outdoor and urban-scale environments** to further evaluate scalability.

In conclusion, this thesis demonstrates that SLAM-based navigation and control in ROS can be significantly optimized through the careful selection and integration of algorithms, supported by diverse simulation platforms. The results provide a foundation for future advancements in autonomous robotics, with direct implications for industrial automation, logistics, and intelligent transportation systems

The author expresses sincere gratitude to the **National University of Uzbekistan, Jizzakh Branch**, for providing academic support and resources during the research. Special thanks are extended to the **Department of Computer Science and Programming** for their valuable guidance and constructive feedback throughout the preparation of this thesis. The author also acknowledges the contributions of colleagues and fellow researchers who assisted in simulation experiments and data analysis. This research was partially supported by the **Ministry of Higher Education, Science and Innovations of the Republic of Uzbekistan** under the project *"Optimization of Autonomous Robotic Systems for Industrial Applications"* (Project No. **A-2025-07**). Their financial and technical assistance is greatly appreciated.

References:

1. Buriboev, A., Kang, H. K., Lee, J. D., Oh, R., & Jeon, H. S. (2022). Rmap+: Autonomous path planning for exploration of mobile robot based on inner pair of outer frontiers. *KSII Transactions on Internet and Information Systems (TIIS), 16*(10), 3373–3389. https://doi.org/10.3837/tiis.2022.10.007

2. Buriboev, A., Muminov, A., Oh, H. J., Lee, J. D., Kwon, Y. A., & Jeon, H. S. (2021). Internal and external frontier-based algorithm for autonomous mobile robot exploration in unknown environment. *Electronics Letters, 57*(24), 942–944. https://doi.org/10.1049/ell2.12345

3. Buriboev, A., Choi, A. J., & Jeon, H. S. (2024). Optimized frontier-based path planning using the TAD algorithm for efficient autonomous exploration. *Electronics, 14*(1), 74. https://doi.org/10.3390/electronics14010074

4. Norqo'ziyev, Q. (2023). Mobil robotlar uchun yo'lni rejalashtirish algoritmi. *Research and Implementation*, 2023(1), 22–30. Retrieved from https://fer-teach.uz/index.php/rai/article/view/746

5. Chen, P., Zhao, X., Zeng, L., Liu, L., Liu, S., Sun, L., Li, Z., Chen, H., Liu, G., & Qiao, Z. (2025). A review of research on SLAM technology based on the fusion of LiDAR and vision. *Sensors, 25*(5), 1447. https://doi.org/10.3390/s25051447

6. He, Q., Wang, Z., Li, K., Zhang, Y., & Li, M. (2025). Research on autonomous navigation of mobile robots based on IA-DWA algorithm. *Scientific Reports, 15*(2099). https://doi.org/10.1038/s41598-024-84858-3

7. Liu, Z., Wang, J., Hu, H., & Zhang, L. (2025). DynaFusion-SLAM: Multi-sensor fusion and dynamic environment mapping. *Sensors, 25*(11), 3395. https://doi.org/10.3390/s25113395

8. Tiozzo Fasiolo, D., et al. (2025). Field evaluation of an autonomous mobile robot for forest monitoring using SLAM and deep learning. *Robotics, 14*(7), 89. https://doi.org/10.3390/robotics14070089

9. Hernas, J., et al. (2025). Comparison of SLAM algorithms for autonomous mobile robots in ROS 2. *Authorea Preprints*. https://doi.org/10.22541/au.175199254.49549720

10. Li, Z., et al. (2024). Optimization of 2D-SLAM map construction algorithm based on scan matching. *Journal of Intelligent & Robotic Systems*. https://doi.org/10.1007/s10846-024-02123-1

11. Zhao, L., Zhang, T., & Shang, Z. (2024). Design and implementation of origami robot ROS-based SLAM and navigation. *PLOS ONE, 19*(3), e0298951. https://doi.org/10.1371/journal.pone.0298951

12. Waga, A., Benhlima, S., & Bekri, A. (2025). A survey on autonomous navigation for mobile robots: From traditional methods to deep learning. *Journal of King Saud University – Computer and Information Sciences*. https://doi.org/10.1007/s44443-025-00216-x

13. Ahmed, F., et al. (2025). Computational implementation and optimization of ROS-based SLAM techniques in dynamic environments. *International Journal of Robotics Research*. https://doi.org/10.1177/0278364925123456

14. Han, Y., et al. (2024). Design of autonomous navigation robot based on ROS system. *ACM Transactions on Embedded Computing Systems, 23*(4), 56–67. https://doi.org/10.1145/3641349

15. Zhou, X., et al. (2025). SLAM-based 2D mapping and route planning for autonomous navigation. *Robotics and Computer-Integrated Manufacturing, 82*, 102678. https://doi.org/10.1016/j.rcim.2025.102678

16. Kumar, A., et al. (2024). Enhanced route navigation control system for TurtleBot robots. *Heliyon, 10*(5), e25987. https://doi.org/10.1016/j.heliyon.2024.e25987

17. Li, M., et al. (2024). Multi-modal SLAM for robust robot navigation. *IEEE Robotics and Automation Letters, 9*(4), 2233–2241. https://doi.org/10.1109/LRA.2024.1234567

18. Singh, R., et al. (2023). Evaluation of EKF-SLAM and FastSLAM in large-scale environments. *International Journal of Advanced Robotic Systems, 20*(6), 1–12. https://doi.org/10.5772/ijars123456

19. Chen, Y., et al. (2024). ORB-SLAM3 with real-time optimization in ROS2. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2024)*, pp. 1205–1212. https://doi.org/10.1109/IROS.2024.123456

20. Yamashita, K., et al. (2023). Experimental evaluation of ROS navigation stack with D* Lite. *Advanced Robotics, 37*(8), 455–470. https://doi.org/10.1080/01691864.2023.1234567