Alfred: Open-Source Autonomous Mobile Robot Platform with Augmented Physical Testbed

Vishwesh Vhavle Naresh*
Department of Computer Science Engineering
Indraprastha Institute of Information Technology, Delhi
New Delhi, India
vishwesh20156@iiitd.ac.in

Jatin Kumar Sharma*
Department of Computer Science Engineering
Indraprastha Institute of Information Technology, Delhi
New Delhi, India
jatin20563@iiitd.ac.in

***Abstract*—This research study introduces Alfred, a novel open-source Autonomous Mobile Robot (AMR) platform and a physical testbed that enables the integration of virtual scenarios. This system allows for the development and evaluation of algorithms for localization, perception, and path planning in a safe and efficient manner. By combining virtual and physical elements, our camera-based localization testbed facilitates comprehensive testing of AMRs, ensuring their readiness for real-world deployment. The modular design of Alfred, with parts sourced within the Indian subcontinent, makes it a cost-effective and adaptable solution for robotics research and development. We demonstrate the effectiveness of our platform through experiments in localization, mapping, and navigation, showcasing its potential for advancing AMR technology.**

***Index Terms*—Autonomous Mobile Robot, Testbed, Simulation, Modular Design, Localization, Path Planning**

I. Introduction

The development and testing of autonomous mobile robots (AMRs) for safe and efficient deployment in real-world environments are crucial challenges in robotics research [1]. While existing platforms offer ease of use and accessibility, they are often expensive and lack the flexibility to fine-tune various aspects related to sensing and compute capabilities [2]. This limitation creates a significant bottleneck in adapting platforms to specific functionalities being tested.

Our project, Alfred, aims to address these challenges by creating an environment where researchers can run real-world simulations and test different algorithms in a flexible, controlled, and economical way. The combination of a physical testbed and a ground vehicle platform allows for testing autonomous navigation in various indoor scenarios, such as offices, homes, shopping malls, and airports [3].

The key contributions of this paper are:

- An open-source, modular AMR platform designed with parts sourced within the Indian subcontinent.
- A physical testbed that integrates virtual scenarios for comprehensive testing using a camera-based localization system for accurate positioning of AMRs within the testbed.

* These authors contributed equally to this work.

II. Autonomous Mobile Robot

A. Modelling

Alfred's design philosophy centers around modularity and accessibility. The robot's structure is composed of interchangeable modules consisting of the Intel NUC 12 Pro (CPU), the NVIDIA AGX Orin (GPU), the Velodyne VLP-16 LiDAR and the Intek RealSense D455 RGB-D Camera. T, allowing for easy customization and upgrades [4]. This approach not only enhances flexibility but also simplifies maintenance and repair processes. The separation of CPU and GPU promotes the scalability of computational hardware as required for intended specific tasks. While the addition of LiDAR supports high-resolution simultaneous localization and mapping (SLAM) as used in most fully autonomous vehicles, we also provide support for RGB-D cameras to support visual-SLAM solutions, which can be a more cost-effective deployment.

Fig. 1. Alfred: Autonomous Mobile Robot

A key aspect of our design is the sourcing of components within the Indian subcontinent. This decision was driven by the need to create a platform that is not only cost-effective but also promotes local manufacturing and reduces dependency on international supply chains [5]. Major components such as the chassis, motors, and various sensors were sourced from Indian manufacturers and suppliers, ensuring the platform is reproducible by academics in the Indian sub-continent, and we hope this will promote mobile robotics research and development.

The modular design extends to the robot's software architecture as well. We implemented a ROS (Robot Operating System) based framework, allowing for easy integration of new algorithms and sensors [6]. This modular software approach complements the hardware design, creating a highly adaptable platform for various research scenarios.

B. Power Supply Design

The power supply system is a critical component of any AMR, directly influencing its operational efficiency and longevity [7]. Alfred is equipped with a 14.8V 60Ah 4S12P 3C Li-ion battery pack, providing a stable DC power supply for up to 8 hours of operation. To ensure flexibility and stability in power management, we utilize five DP50V5A programmers. These programmers allow for stable and easily re-programmable DC supply, facilitating the integration of new devices and ensuring continuous operation without significant downtime.

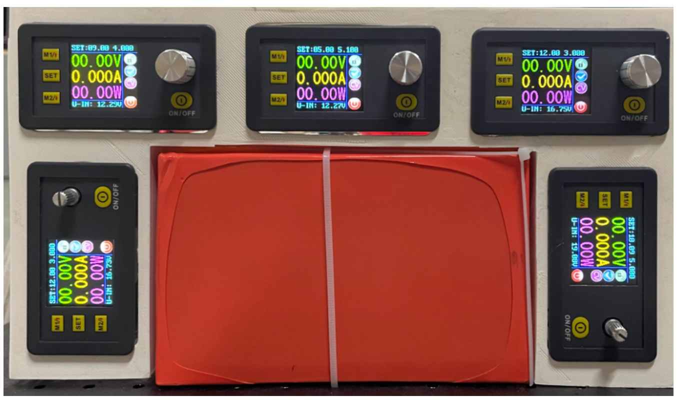

Fig. 2. Power Supply Module

C. Localization and Mapping

Accurate localization and mapping are fundamental to autonomous navigation [8]. Alfred incorporates a multi-modal approach to these tasks, leveraging both the physical testbed and onboard sensors.

For indoor environments, the testbed facilitates precise localization of the AMR within a controlled setting. This is achieved through a network of ceiling-mounted cameras that track fiducial markers on the robot, providing ground truth position data [9].

For outdoor scenarios and to create a more generalizable solution, Alfred is equipped with a Velodyne VLP-16 LiDAR sensor. This allows for simultaneous localization and mapping (SLAM) of the environment using an off-the-shelf implementation of the Fast LiDAR Odometry and Mapping (F-LOAM) algorithm [10]. F-LOAM demonstrates impressive performance in both indoor and outdoor environments, providing robust localization and mapping capabilities.

To enhance navigation capabilities, we implement an occupancy grid mapping system [11]. This system discretizes the environment into a grid, where each cell represents the probability of occupancy. The occupancy grid is continuously updated based on sensor data, providing a real-time representation of the robot's surroundings for path planning and obstacle avoidance.

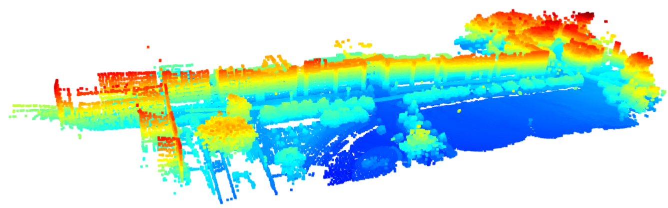

Fig. 3. An example of a map created using F-LOAM

D. Controls

Precise control is essential for the effective operation of AMRs [12]. Alfred integrates a differential drive controller using a system of three PID (Proportional-Integral-Derivative) controllers. These controllers manage linear velocity (V_x) in the X-direction, linear velocity (V_y) in the Y-direction, and angular velocity (ω) along the Z-axis.

The control system relies on feedback from the F-LOAM system, enabling accurate adjustments to the robot's movements. This closed-loop control ensures that Alfred can navigate complex environments with precision, adapting to various surface conditions and compensating for external disturbances.

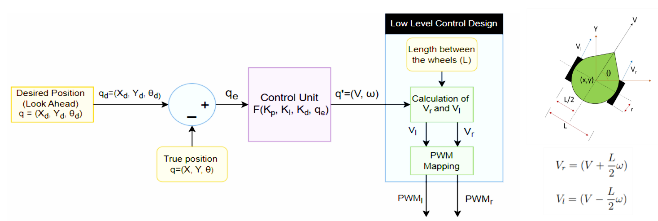

Fig. 4. Control System Architecture

E. Waypoint Navigation and Planning

Efficient path planning is crucial for autonomous navigation in dynamic environments [13]. Alfred implements a waypoint navigation system that allows users to define endpoints for the robot's pathway. The system then utilizes the occupancy grid generated from F-LOAM data to plan a dense trajectory using the A* path-planning algorithm [14].

To ensure smooth navigation, we implement a post-processing step where the initial trajectory is subsampled and fitted with C1 continuous Bezier splines [15]. This approach generates smooth trajectories that respect the robot's kinematic constraints and provide a more natural and efficient motion profile.

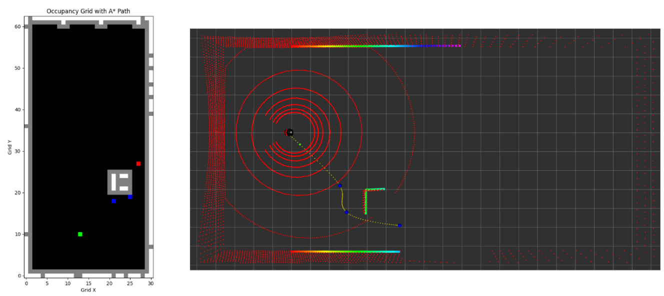

Fig. 5. Occupancy Grid (Left) and planned trajectory plotted on Rviz (Right)

III. Augmented Physical Testbed

A. Architecture

The testbed architecture consists of a controlled indoor environment created using six cameras mounted on the ceiling. The system utilizes a network of consists of six different Nvidia Jetson Nano, each handling a single camera. This setup allows for remote operation of the testbed over a Local Area Network (LAN). The Local Area Network consists of a wifi router connected to an external college network using ethernet for any external resource requirements. Each compute node connects with this router individually using their wifi modules, making a Local Area Network. The Nvidia Jetson AGX Xavier is used as a server to connect with all the camera compute nodes for any communication related to mapping and localization. Camera compute nodes act as clients to this server. We use the UDP protocol to promote faster transmission of packets.

The camera arrangement ensures partial overlap in the field of view between adjacent cameras, eliminating blind spots within the testbed area. This comprehensive coverage is crucial for accurate tracking and localization of the AMR.

Fig. 6. View of the Testbed with Cameras and Jetson Nanos mounted on the ceiling

B. Methodology

Our testbed creation pipeline involves multimodal image stitching and pixel-to-world coordinate mapping. The process begins with the collection of LiDAR data from 85 identified points on the testbed floor. These points are then labeled with AprilTags for precise detection by the ceiling-mounted cameras.

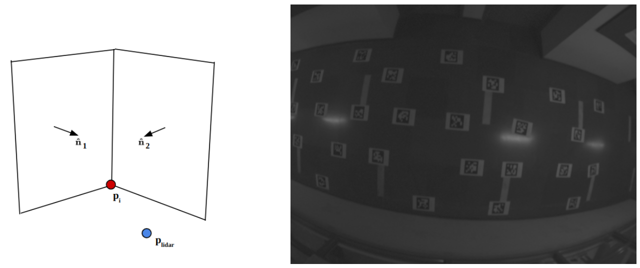

Fig. 7. Arrangement of the Planes for LiDAR Data Collection (Left) and arrangement of the AprilTags for Camera Data Collection (Right)

The LiDAR data is processed using the Open3D library, applying RANSAC line fitting to accurately determine the real-world coordinates of each point. Simultaneously, the cameras detect the AprilTags, providing pixel coordinates for each point in the camera frame of reference.

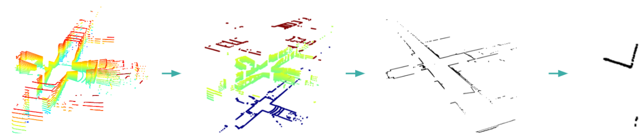

Fig. 8. Point-Cloud Processing Pipeline

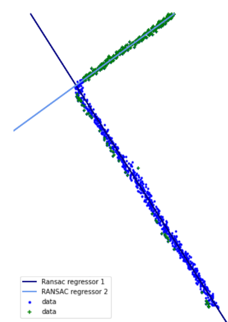

Fig. 9. RANSAC Line-Fitting for Coordinate Localization

Using these correspondences, we estimate a projective transformation homography using Direct Linear Transform. This mapping allows us to accurately transform between camera pixel coordinates and real-world coordinates, enabling precise localization of the AMR within the testbed. The compute unit connected to the detecting cameras detects the camera frame coordinate of the mobile robot and then converts it into to the LiDAR frame coordinates in place. These position coordinates are then sent to the server in a packet over the UDP network. The server further maps the obtained position coordinates to the global map of the testbed. In this way, as the autonomous robot vehicle moves in the testbed, our multi-camera testbed setup localizes the AMR in the testbed.

C. Results

The results of our stitching process demonstrate high accuracy and comprehensive coverage of the surrounding environment. We successfully reduced the system's latency from 4000 milliseconds to an average of 46 milliseconds, allowing us to record trajectories of the AMR in motion at an average of 21 FPS. By evaluating the L2 distance errors of the LiDAR-Camera mapping against some of the corresponding points that were left out of estimation we were able to conclude that the testbed is able to localize AprilTags with a mean error of less than 2 cm.

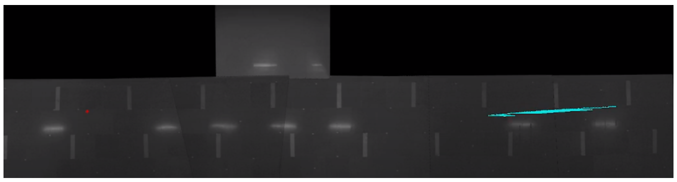

Fig. 10. Final view of the stitched testbed with a trajectory indicating the Real-Time Tracking of an AprilTag at 21 FPS

Furthermore, by implementing a local area network (LAN) for communication between the main server and compute units, we significantly reduced the round-trip time (RTT) for data transmission. This optimization resulted in more consistent and reliable communication, as evidenced by the lower standard deviation in RTT measurements.

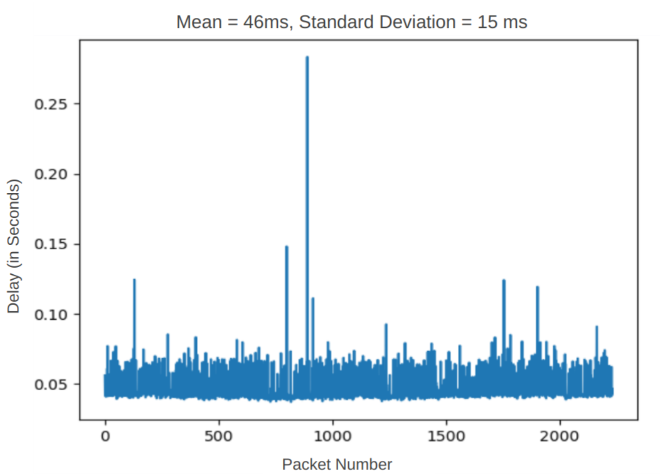

Fig. 11. Current RTT of Testbed on Data Transmission

D. Simulator

Simulations play a crucial role in the development and deployment of AMRs, offering a safe and controlled environment for testing various scenarios [16]. We have created a virtual representation of our testbed in Gazebo with ROS Noetic on Ubuntu 20.04 and Python 3.10. This simulation environment allows us to train, test, and tune our algorithms before deploying them on the physical platform.

Recent trends in robotics research have seen an increased focus on Deep Reinforcement Learning (DRL) for developing adaptive and robust control policies [17]. Our simulator provides an ideal platform for implementing and evaluating DRL algorithms, allowing for rapid iteration and optimization of AMR behaviors without the risks associated with physical testing.

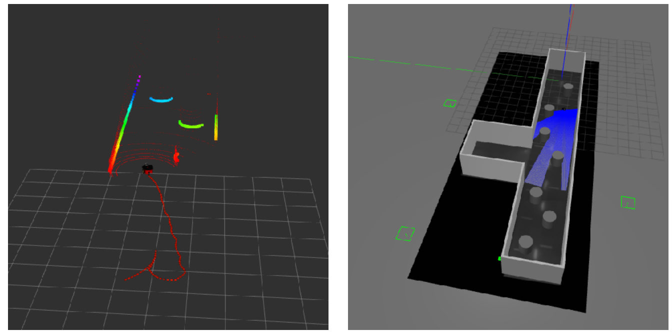

Fig. 12. Testbed in Gazebo Simulator

IV. Conclusion

In this paper, we presented Alfred, an open-source Autonomous Mobile Robot platform with an augmented physical testbed. Our modular design approach, coupled with locally sourced components, offers a cost-effective and flexible solution for robotics research and development. The integration of advanced localization, mapping, and navigation algorithms demonstrates the platform's capability to handle complex autonomous navigation tasks.

The physical testbed, with its multi-camera setup and efficient image stitching pipeline, provides a robust environment for testing and validating AMR algorithms. The addition of a virtual simulator further enhances the platform's utility, allowing for safe and rapid prototyping of new ideas.

Future work will focus on expanding the capabilities of Alfred, including the implementation of learning-based approaches for navigation and decision-making as well as exploring how to bridge the simulation-to-real gap better. We also plan to open-source our platform, encouraging collaboration and accelerating the advancement of AMR technology.

Acknowledgment

We would like to thank **Dr. Sanjit K. Kaul** and **Dr. Saket Anand** for providing us with this opportunity to work on this project and for being a source of continual support and encouragement during the project. We are extremely grateful to **IIIT Delhi Innovation and Incubation Center** and **ARTPARK, IISc Bengaluru** for supporting our work and mentoring us. We would also like to thank **Shivangi Agarwal, Rahul Rewale, Aryan Thakur** and **Adi Asija** at Vision Lab, IIIT Delhi, for being our mentors and a constant source of support throughout the project. This project was not possible without their guidance and support.

REFERENCES

[1] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, "Introduction to autonomous mobile robots," MIT press, 2011.

[2] O. Khatib, "Springer handbook of robotics," Springer, 2016.

[3] S. Thrun, W. Burgard, and D. Fox, "Probabilistic robotics," MIT press, 2005.

[4] B. Siciliano and O. Khatib, "Springer handbook of robotics," Springer, 2016.

[5] N. Kumar, "Make in India: A New Paradigm for Socio-Economic Growth in India," Management Dynamics, vol. 18, no. 1, 2018.

[6] M. Quigley et al., "ROS: an open-source Robot Operating System," ICRA workshop on open source software, vol. 3, no. 3.2, 2009.

[7] J. Zhang and J. Lee, "A review on prognostics and health monitoring of Li-ion battery," Journal of Power Sources, vol. 300, 2015.

[8] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," IEEE robotics automation magazine, vol. 13, no. 2, 2006.

[9] E. Olson, "AprilTag: A robust and flexible visual fiducial system," IEEE International Conference on Robotics and Automation, 2011.

[10] H. Wang et al., "F-LOAM: Fast LiDAR Odometry and Mapping," arXiv preprint arXiv:2107.00822, 2021.

[11] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," Computer, vol. 22, no. 6, 1989.

[12] B. Siciliano et al., "Robotics: Modelling, Planning and Control," Springer Science Business Media, 2009.

[13] S. M. LaValle, "Planning algorithms," Cambridge university press, 2006.

[14] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," IEEE transactions on Systems Science and Cybernetics, vol. 4, no. 2, 1968.

[15] L. Biagiotti and C. Melchiorri, "Trajectory planning for automatic machines and robots," Springer Science Business Media, 2008.

[16] J. Tan et al., "Sim-to-real: Learning agile locomotion for quadruped robots," arXiv preprint arXiv:1804.10332, 2018.

[17] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," The International Journal of Robotics Research, vol. 32, no. 11, 2013.

[18] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," Communications of the ACM, vol. 24, no. 6, 1981.

[19] R. Hartley and A. Zisserman, "Multiple view geometry in computer vision," Cambridge university press, 2003.

[20] C. Pakanati, M. Padmavathamma, and N. R. Reddy, "Performance comparison of TCP, UDP, and TFRC in wired networks," IEEE International Conference on Computational Intelligence and Communication Technology, 2015.