Robotics Dojo 2024
Autonomous Mobile Robot (AMR)

Joint Team 4

Felix K. Ronoh, Jeremiah J. Onyapidi, Hillary M. Murimi, Omariba Collins and Lukundo Okemba (ROOK Droid, Mobile Platform)
Nindo Emmanuel and Matiko Maroa (Team Echo, Navigation)

Abstract

This research presents the **development of an AMR** capable of autonomously navigating through complex maze environment. Utilizing **ROS_2**, **SLAM**, and **LiDAR**, the robot successfully demonstrates

- Accurate mapping & localization,
- Path planning
- Obstacle avoidance.

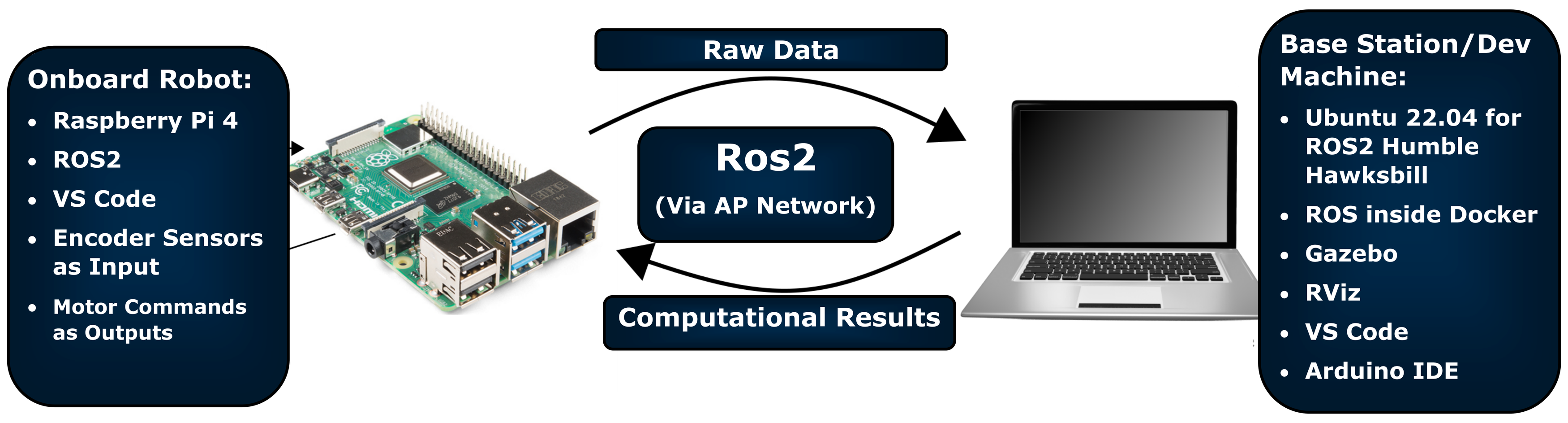
This research contributes to advancements in autonomous robotics and has potential applications in fields of **logistics**, **search and rescue,** and **industrial automation.**

1. Introduction

Objective: To deepen and share knowledge on Robotics by building an Autonomous Mobile Robot using **ROS2** for navigation.

Main Contributions:

1. Integrated real-world robotics with **ROS2, Gazebo** simulation, and hardware interfaces.
2. Developed robust **teleoperation** and **autonomous navigation**.
3. Designed and built a reliable hardware platform.



2. Design Strategy and Build Process

2.1 Robot Design

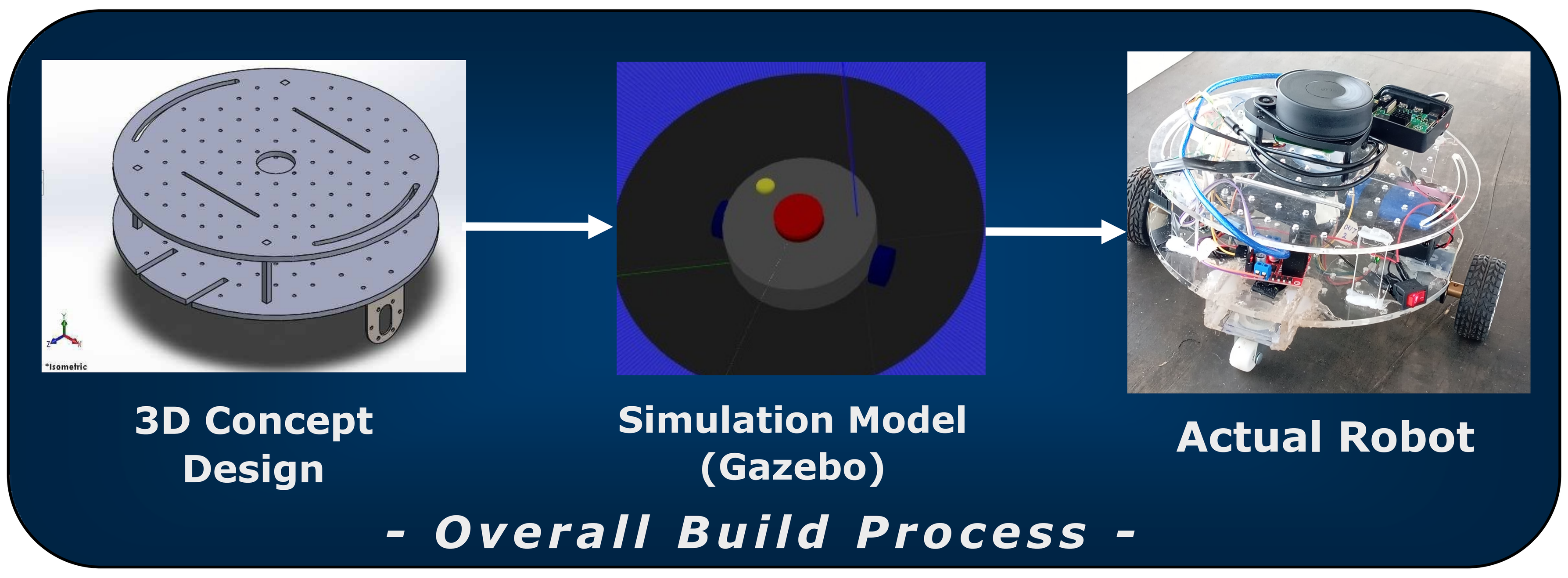
- **Modular Approach:** To facilitate easy assembly, testing, and modifications.
- **ROS 2 Framework:** Used for its flexibility, scalability, and extensive community support.
- **SLAM** and **Navigation**: Utilized the **SLAM Toolbox** and Nav2 packages for mapping and autonomous navigation.
- **LiDAR Integration:** Integrated a LiDAR sensor for accurate environment perception and obstacle avoidance.

2.2 Build Process

- **Hardware Selection:** Based on project requirements and budget constraints.
- **Chassis Construction:** Designed and built a custom chassis using **acrylic sheets** for a **lightweight** and durable structure.
- **Assembly**: Wired the components together, using proper power distribution and communication

2.3 Software Integration

- **Development** of **ROS 2 nodes** for control, navigation, and sensor processing.
- **ROS2 Setup**: Integration of ros2_control for motor control and diff_drive controller for navigation.
- **Simulation**: Utilized **Gazebo** for robot motion validation and **SLAM testing**
- **Autonomous Control**: Used **Nav2** stack for path planning and obstacle avoidance.



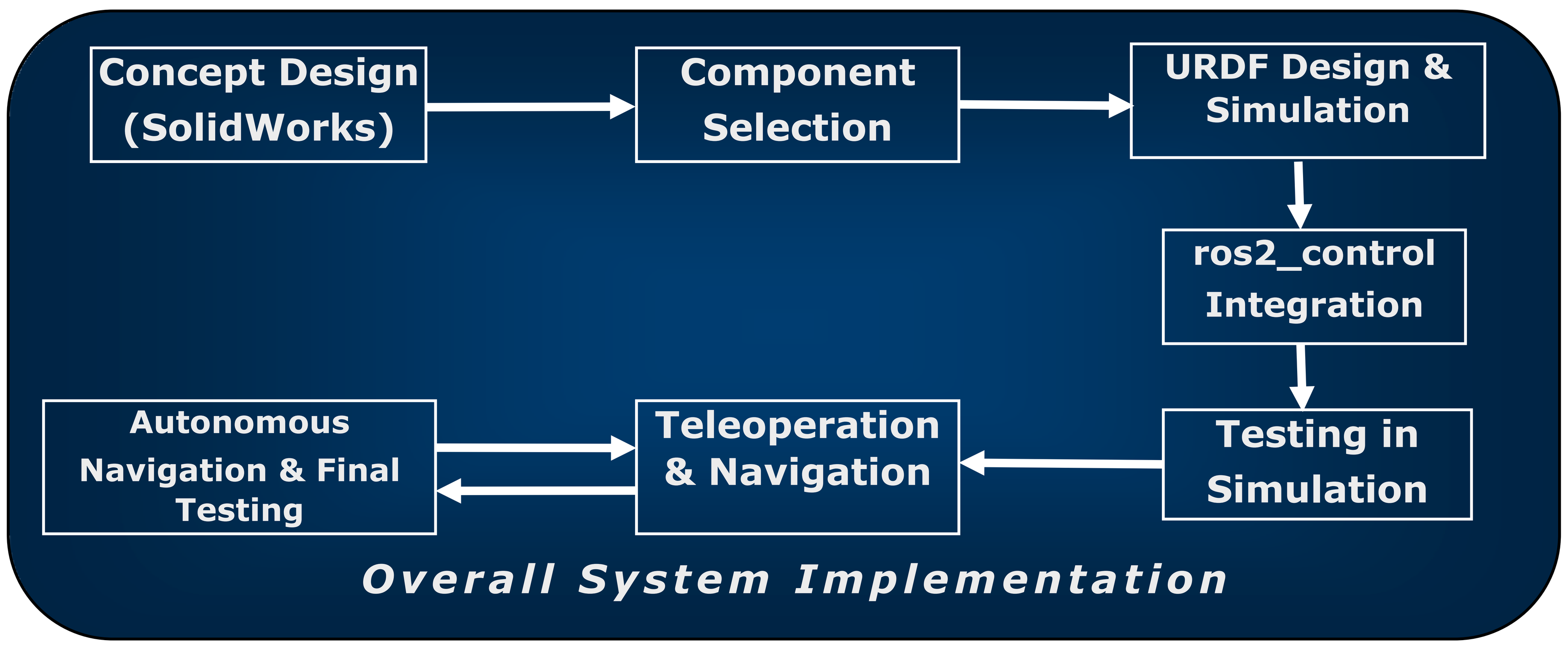
3. Control System Implementation

3.1 ROS2 Control Framework

- **Controller Manager**: Handles communication between ROS and the robot.
- **Diff Drive Controller:** Converts velocity commands into wheel speeds.
- **Hardware Interface:** Translates commands to the motor controller, receiving encoder feedback.

3.2 Teleoperation & Navigation

- **Teleoperation:** Utilized teleop_twist_keyboard for initial tests.
- **Autonomous Navigation:** Implemented SLAM for mapping and used Nav2 stack for path planning.



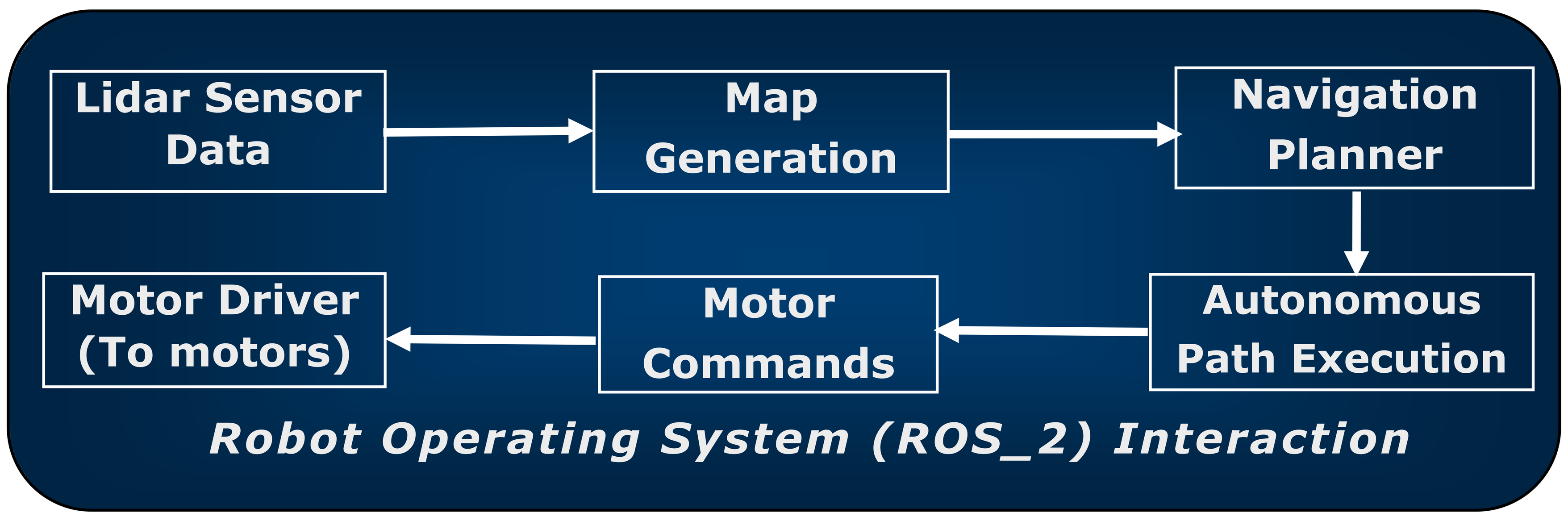
4. Testing and Perfomance Evaluation

4.1 Simulation Testing

- **Gazebo Simulation:** Verified robot motion and tested SLAM capabilities.
- **Virtual Environment:** Simulated real-world scenarios for collision detection and navigation.

4.2 Real-World Testing

- **Teleoperation Testing:** Conducted initial teleop tests using **ros2_control** and verified wheel velocities.
- **Autonomous Navigation:** Fine-tuned the Nav2 stack for smooth navigation and obstacle avoidance.



5. Results, Conclusion & Key Findings

(1) **ROS 2** proved to be a **valuable tool for robot control, communication,** and **integration** of various components.
(2) The robot demonstrated **robust obstacle avoidance** capabilities, successfully navigating around obstacles without collisions.
(3) The navigation algorithm via **SLAM efficiently planned and executed paths**, reaching the desired goal.

6. Future Works and Recommendations

- **Enhanced Environmental Perception:** Using additional sensors (cameras, depth sensors) for object recognition.
- **Advanced Navigation**: More complex navigation algorithms, such as **SLAMP** for dynamic environments.
- **Object Interaction**: Develop capabilities for interacting with objects, i.e. grasping, manipulating, or object tracking.
- **Human-Robot Interaction**: Implement features for human-robot interaction, i.e. voice commands or gesture control.

[This project was conducted as part of the Robotics Dojo Competition 2024. The authors would like to thank Dr Shohei Aoki for the mentorship and contribution to the project.]