Booth Number: 309

Autonomous Object-Picking Robot

Christoffer Cox, James Dickson, Kathy Vo

Faculty Mentor: John Lusher

TEXAS A&M UNIVERSITY Department of Electrical & Computer Engineering

Problem Definition

With millions of packages processed daily through warehouses, the need for safe and efficient sorting methods is vital. Implementing an autonomous robot that is able to differentiate between objects and take specific ones to a designated location will help increase labor efficiency while decreasing workplace injuries. Currently, warehouse injury rates are at 5.5 per 100 employees, more than double the all-industry average.

Methodology

We built an ROS2-based pipeline using embedded hardware, with the main components being a Raspberry Pi 5, a Kinect v2, and 6DOF robotic arm.

Functions:

- The ability to navigate through indoor environments with no human control
- The ability to differentiate between objects depending on the color
- The ability to grasp and transport designated objects to a given location

System Diagram

- Below, you can see the system diagram and how all the parts communicate with each other

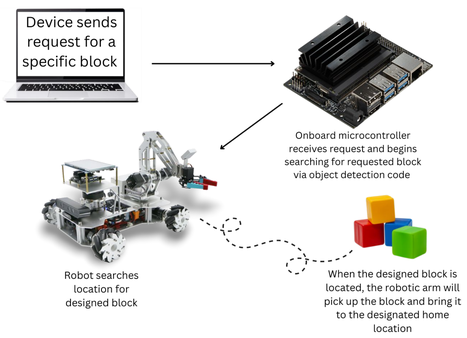

Figure 1.System Diagram

Software:

- Python was used to code throughout all the subsystems. ROS2 Jazzy was used for robotic control. Gazebo Harmonic was used for simulations pairs well with Jazzy. The YOLO v11 object-detection model is trained on our custom dataset. The robotic arm microcontroller and the Raspberry Pi communicated over a USB-to-UART serial interface.

Simulations:

- By simulating the robot's movement in Gazebo Harmonic, we observed if the object-detection code could navigate through obstacles and detect how close it is to objects.

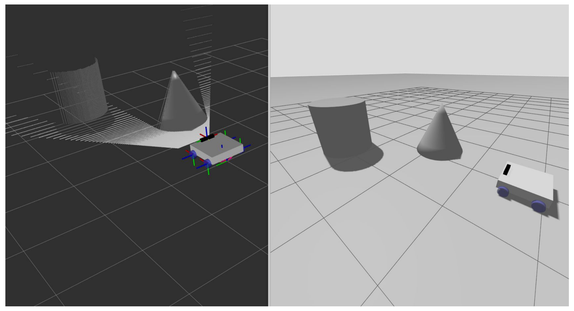

Figure 2. Gazebo Simulations with obstacles

Engineering Analysis

The main engineering problem behind this project was integrating the ROS virtual system with the physical robot.

- Software Compatibility: Integrating different ROS2 packages on an ARM Raspberry Pi 5 resulted in multiple compatibility issues, requiring a large amount of time developing custom code to connect libraries.
- The Raspberry Pi 5, a late change to the project, proved to be not as powerful as expected. We ran into issues with hardware limitations. For instance, the kinect dropped 25 percent of its depth packets
- Transferring the simulation data to the hardware allowed rapid prototyping. This streamlined testing, which was extremely important in this project.

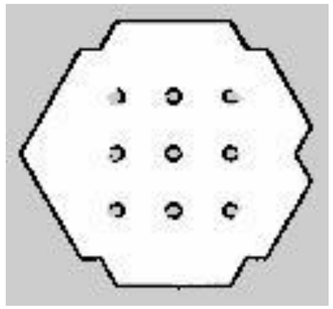

Figure 3. SLAM Simulation

Outcomes

After merging the navigation, object detection, and arm control subsystems into a cohesive system, the robot body successfully detected a colored block and measured its distance from it. It then applied the differential-drive equation to compute and reach the desired position. The sabertooth motor driver would take the instructions and control the left and right motors. Additionally, the robot body also stops motion if the block is moved out of frame. After the robot was positioned directly in front of the object, the robot arm was able to pick up the block by sending commands, ranging from 500-2700, from the Raspberry Pi to the servo's MCU.

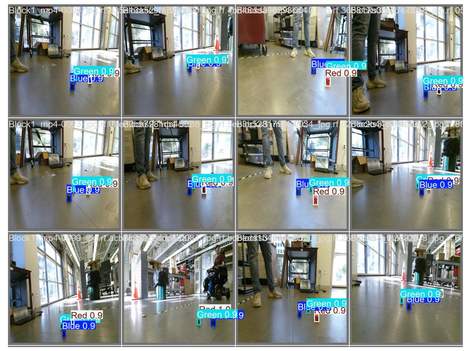

Figure 4. Object Detection Results

Impact

The autonomous object-picking robot can work safely in hazardous environments, cutting down on human injuries and illness. Its ability to operate without supervision boosts productivity and efficiency, which in turn supports economic growth.

- **Cost Savings:** Over time, fewer workplace accidents and reduced labor costs
- **Enhanced Safety:** The robot reduces workers' exposure to injury and illness.
- **Operational:** Students can build on this project like adding new sensors or modules o reprogram the robot for new objects, tasks, or workflows.

References

1. LSC-6 Bluetooth Servo Controls https://drive.google.com/drive/folders/15teEZyf8i5MFzClf_jEyv4guxBQljwf8

2.Standard for Robot Data Map Representation for Navigation https://ieeexplore.ieee.org/document/7300355

Acknowledgements

Special thanks to Swarnabha Roy for being our sponsor!