BU

Senior Design

ENG EC 464

To: Professor Pisano

Team: 10

Date: 03/5/2025

Subject: Second Prototype Testing Report

Second Prototype Test Plan

Draper Convoy

CONVOY

By

Team 10

Team Members

Sadman Kabir: kabirs@bu.edu
Benjamin Hsu: bhsu25@bu.edu
Nikhil Krishna: nikhilk@bu.edu
Hugo Silmberg: hsilmber@bu.edu
Michael Moran: mmoran3@bu.edu

Required Materials

Hardware:

- Desktop Computer running Ubuntu
- Physical Turtlebot4 Lite:

    OAK-D-LITE (Camera)
    RPLIDAR-A1 (Lidar)
    Raspberry Pi 4B

- Banana Stick

Software:

- ROS2 installed on the desktop computer
    - Using ROS2 Jazzy distribution
    - Allows data to be accessed through TurtleBot sensors and communicate with other software
- Person detection node
    - YOLO object detection algorithm
        - Pretrained model from Ultralytics
    - Identifies human beings detected within camera view and outputs information about their position relative to the robot
    - Highly accurate, even from long distances and figure partially cut off
- Motor control node
    - Accepts data from the person detection node and outputs information to control the TurtleBot’s motors

Set-Up

1. Remove Turtlebot from the dock and make sure it starts up correctly. Restart the robot if it does not start up right the first time.

2. On the desktop, set up a new terminal environment and launch our nodes using the launch_robot.sh bash script:

```
cd ~/ros2_ws
chmod +x launch_robot.sh
./launch_robot.sh
```

3. Once all nodes are fully started, simply press P in the motor control node terminal to start human tracking.

Testing Procedure

1. Hold the banana within camera’s field of view. The Turtlebot should rotate to face the banana. Wave the banana in front of it. This tests the robot’s ability to keep the target in sight.
2. Start backing away from the robot. It should start picking up speed smoothly and follow the banana.
3. Stand still with the banana and let the robot reach it. The robot should stop within 1m of the banana. This tests its ability to maintain distance.
4. Now walk the banana around and behind the Turtlebot. It should now spin in the direction it saw the robot go until it acquires vision on the banana again. This tests the ability to seek out the target if it goes out of the camera’s field of view.

Discussion and Results

The measurements taken in this prototype testing were behavior-based criteria for our TurtleBot. Since we have recently received our TurtleBot 4 robots, we were able to test on physical hardware and targets this time around. Our system is designed to identify human targets, but with the lab space being very crowded during the test we opted to use a banana as the designated tracking target. The behaviors we were looking for in this test were the following:

Functionality
Identifies a banana target
Bounding box around banana w/ confidence level, angle, and centroid
Seek banana target
Stop when close enough to the banana target
Auto relocate -> continue seeking bananas

Our test successfully identified a banana in the camera's view demonstrated by a bounding box that was placed around the banana with the confidence of the identification, along with angle and distance information. The only slight issue with the detection system was the position of the banana was very important for the detection, making it necessary to see the curvature of the banana for most detections. This issue is not present when we are searching for humans with our system, as the robot is able to identify a person just by seeing a hand or foot within view. When the robot came close enough to the banana target, it stopped successfully and continued to detect the banana to check for movement. Our auto relocate behavior also worked as expected with the robot spinning to search for the lost target, and when the target was detected again the robot continued its seeking behavior.

Conclusions from this test about our project's current success and necessary improvements were insightful. Our target detection and seeking behavior is working as expected but needs improvements in some areas. We are currently working to implement protections to allow the robot to view multiple humans and only track the designated target without getting confused (this will eliminate the need for a banana target). Our motor control behavior is also working correctly as of now, receiving inputs from the detection and tracking systems to output correct data to the robot's motors, allowing for follow-the-leader behavior. This will also need some improvements to allow for smoother movement and more predictable behavior from the movement of the robot. One aspect that is crucial for our project and was missing from the second prototype test was the ability to detect and avoid obstacles in a dynamic environment. We are currently working hard on implementing this behavior through the ROS2 navigation framework Navigation 2 which should allow for obstacle avoidance and real-time path planning behavior.