CS 556: Task-Sheet for Final Project

Your Name: ____________Carlos Martinez____

Your REDID: _______________827940172_______________________________________

Your Teammates names :_________Umar Umar___________________________________

Robot ID:_____Botzilla 03____

1. Project Proposal

1.1 Problem Statement

Our robot performs automatic trash detection and navigation using line sensors, sonar, and odometry to traverse an environment while avoiding obstacles and counting detected trash zones.

1.2 Sensors

- **Line Sensors (Pololu Reflectance Array):** Used to detect black and blue lines on the floor. Black lines represent trash zones, and blue lines trigger wall switching behavior.
- **Ultrasonic Sonar Sensor:** Mounted on a servo to scan distances in front, left, and right. Enables wall following and obstacle avoidance.
- **Encoders:** Used for odometry, tracking wheel rotation to calculate the robot's position (x, y) and orientation θ.

1.3 Behaviors, Automation, and Control Architecture

STATES:

- **State 1: MOVING**
  The robot performs wall-following using a PD controller. It also monitors for blue lines (to change walls) and black lines (trash detection).
  Reacts dynamically to sensor inputs, including sonar readings and line sensor thresholds.
- **State 2: IDLE**
  The robot briefly stops and performs a left rotation to avoid frontal obstacles or upon certain detection events.
  Adaptability: Triggered based on sonar-detected proximity to walls/obstacles.
- **State 3: ROTATING**
  Executes angle-based rotation to align with specific directions using a PID controller on

angular error.
Adaptability: Can dynamically rotate to any provided angle.

LOW-LEVEL BEHAVIORS:

- **Trash Escape Maneuver:** When black line is detected, the robot performs a timed maneuver to move forward a little and rotate, then resumes movement.
- **Sonar Scanning:** The servo-mounted sonar collects distance data from multiple directions, supporting behavior switching.
- **Odometry Update:** Regularly updates the robot's estimated position and distance traveled for control and debugging purposes.

TRANSITIONS:

- MOVING → IDLE IF front sonar < 11 cm
- IDLE → MOVING AFTER short turn and forward burst
- MOVING → ROTATING IF rotateTo(angle) is triggered
- MOVING → MOVING IF blue line is detected → rotate 180° + switch wall side
- MOVING → MOVING IF black line is detected → run escapeTrash protocol

Finite State Automata (FSA) Diagram AT BOTTOM

1.5 Control Architecture

Architecture Type:
Hybrid Reactive / Behavior-Based Control
Reactive components include real-time responses to sonar and line sensor inputs, while the state-based logic enables structured decision-making.

Controllers Used:

- **PD Controller (Wall Following):**
  Controls lateral distance from wall using sonar readings. Adjusts speed differential between wheels based on error between current and desired distance.
  Error Type: Positional error (cm).

2. Integration of Semester Modules

- **Lab 1:** Used navigation primitives for basic movement and motor speed control.
- **Lab 3:** Implemented calibration and reading of reflectance line sensors to detect trash.
- **Lab 4:** Applied encoder-based odometry to calculate robot's position and heading.
- **Lab 5:** Developed PD wall-following logic using sonar distance as feedback.
- **Lab 6:** Added servo scanning to gather environmental context.
- **Lab 7:** Designed and implemented a full Finite State Automaton (FSA) for control logic.
- **Lab 8/9:** Tried to apply PID control for rotation toward a target angle with fine tuning which was not fully functional at the end.

3. Challenges, Error Handling, and Reliability

Challenge 1: Blue Line Detection Failure

- Despite efforts to tune thresholds and average line sensor values, the robot consistently failed to reliably detect blue lines during final testing. The sensor values for blue were too similar to those of the floor or other surfaces in the environment, leading to false negatives.
- **Solution Attempted:** We adjusted thresholds, experimented with averaging sensor values, and added debug logging. However, due to limited testing time and sensor limitations, blue line detection was ultimately disabled in the final demonstration.

Challenge 2: Trash Detection Double Readings

- Early tests showed that the robot triggered black line detection more than once every time it entered a black square.
- **Solution:** We fine-tuned some delays. Also, after trash detection, the robot enters a temporary escape state to avoid double-counting.

Reliability:
The robot performs reliably in black square detection and wall-following tasks. It successfully responds to obstacles using sonar and executes recovery maneuvers when trash zones are encountered. However, due to unresolved issues with blue line detection, dynamic wall-switching behavior was not functional in the final version.

4. Testing and Validation

- **Testing Strategy:**
    - Logged sensor values to determine reliable thresholds for line detection.
    - Tested wall-following on both left and right walls using servo and sonar readings.
- **Performance Metrics:**
    - Accuracy of line detection.
    - Number of trash detections.
    - Distance from wall (measured using sonar vs. target 8 cm).
    - Recovery speed after detecting trash or obstacles.
- **Edge Cases:**
    - Sharp turns, missing blue line
    - Adjusted timeout durations and added manual rotation backups where needed.

5. Innovation and Creativity

- We tried implementing a bi-directional wall-following behavior that dynamically switches walls using blue-line detection to attach to the middle wall once it detected the blue square and it .
- Developed an adaptive trash escape maneuver that prevents repeated detections of the same trash area.
- Our FSA design integrates sonar-based awareness with sensor-based transitions, combining multiple sensing layers for smarter decision-making.

6. Communication and Team Collaboration

- Tasks were split based on strengths:
    - Umar focused on wall-following, PD tuning and trash detection.
    - Carlos handled edge cases and avoided obstacles.
    - We both went through the logs and adjusted multiple values for different thresholds, delays, and motor speed, We jointly tested and debugged behavior transitions.
- Communication was maintained through regular in-person check-ins, texts and calls, Also plenty of team debugging sessions during classes and office hours.

7. Use of Resources

- **TA Feedback:** Helped clarify controller integration..
- **Lab Hours and Office Hours:** Were especially valuable for debugging and testing.
- **Canvas and Lecture Notes:** Provided key references for state machine design.

1.4 Finite State Automata (FSA) Diagram