Robobo Robot - Recycling Collection and Sorting System

Overview

This project implements a reactive (subsumption) architecture for the Robobo robot to autonomously search for recyclable objects, approach them, and deliver them to appropriate recycling bins. The system performs a recycling task using a multi-threaded behavior-based approach where different behaviors can take control based on the robot's current state and sensor inputs.

Architecture

Reactive (Subsumption) Architecture

The system uses a subsumption architecture where behaviors are organized in priority levels. Higher-priority behaviors can suppress lower-priority ones when they need to take control. This allows for robust, reactive behavior that can handle multiple concurrent goals.

Threading Model

Each behavior runs as a separate thread, continuously monitoring conditions and executing actions when appropriate. The main thread coordinates all behaviors and manages the overall mission state.

Behaviors

1. SearchObject (Lowest Priority)

Purpose: Searches for objects in the environment using the robot's camera.

Activation: When no object has been found yet and not suppressed by higher-priority behaviors.

Actions:

• Tilts camera down and performs systematic search pattern

• Uses object recognition to detect objects

• When found: adjusts orientation, shares object info, shows happy emotion

2. ApproachObject (Medium Priority)

Purpose: Approaches a detected object until it's close enough to interact with.

Activation: When an object has been detected and not yet approached.

Actions:

• Tracks object using camera and adjusts direction

• Uses IR sensors to determine when close enough

• When close: stops motors and marks object as approached

3. SearchAndApproachBin (Highest Priority)

Purpose: Searches for the appropriate sorting bin and delivers the approached object.

Activation: When an object has been approached and mission not completed.

Actions:

• Classifies object (plastic, organic, paper)

• Searches for QR codes matching the object category

• **Object Loss Detection:** Continuously monitors if object is still held using camera + IR sensors

• When bin found: approaches, shows success emotion, completes mission

Key Features

1. Hybrid Object Loss Detection

• Uses both camera detection and IR sensor readings

• Only declares object lost after multiple consecutive failures

• Automatically reactivates search behavior when object is lost

2. Emotional Feedback

• Shows emotions (happy, surprised, focused, sad)

• Plays sounds (thinking, ouch, likes)

• Speaks messages ("We found something", "mission completed!")

3. Robust Search Patterns

• Systematic scanning with alternating rotations

• Position-based alignment using coordinates

• Multi-sensor fusion (camera, IR, QR codes)

Usage

Prerequisites

• Robobo robot or simulator

• Python with robobopy library

Running the System

python main.py

Expected Behavior

1. Robot searches for objects
2. Approaches found object
3. Classifies object and searches for matching bin
4. Delivers object to bin
5. Shows success emotion and completes mission

Troubleshooting

Common Issues

1. **Robot keeps searching without finding objects:**

• Check if objects are within camera range

• Verify object recognition is working

2. **Robot loses objects frequently:**

• Adjust IR sensor thresholds

• Check object size and visibility

3. **Robot doesn't find the correct bin:**

• Verify QR codes are properly labeled

• Check QR code visibility

Debug Output

The system provides console output for debugging:

• Behavior activation messages

• Object detection results

• IR sensor values

• QR code detection status

Technical Details

Threading Implementation

• Each behavior runs as a separate thread

• Main thread coordinates behavior activation

• Suppression mechanism prevents conflicts

• Shared parameters enable inter-behavior communication

Sensor Integration

• **Camera:** Object recognition and QR code detection

• **IR Sensors:** Proximity detection and object loss detection

State Management

• Shared `params` dictionary for inter-behavior communication

• Individual behavior state tracking

• Mission completion coordination