

French Door Cafe Robot Butler Documentation

Overview

The French Door Café has introduced a robot butler to manage customer orders efficiently. The robot navigates between the kitchen and customer tables to deliver food orders, optimising restaurant operations and reducing labour costs.

Components

1. Main Control Program (control_node.py)

This node coordinates the overall operation of the robot butler. It subscribes to topics for receiving commands and positions, and it controls the movement of the robot using the 'goto' module.

- Subscribed Topics:

- '/heard' (std_msgs/Int16): Receives commands from the voice recognition node regarding robot actions.
- '/current_pos' (std_msgs/Int16): Updates the current position of the robot published by the localization package.
- '/tables' (std_msgs/Int16MultiArray): Receives table numbers for food delivery.

- Published Topics:

- '/listento' (std_msgs/Int16): Publishes commands to the voice recognition node to listen for the robot call and table numbers.

- Functionality:

- Initializes the robot to its home position.
- Calls the voice recognition API whenever it is required.
- Logically decides the commands to move to the kitchen or deliver food to specified tables.
- Executes movement commands using the 'move_to' function from the 'goto' module.
- Finally it takes the robot to the home position.

2. Speech Recognition Program (voice_recognition.py)

This node integrates speech recognition capabilities to allow voice commands for initiating robot actions.

- Subscribed Topics:

- '/listen_to' (std_msgs/Int16): Listens for commands from the main control node.

- Published Topics:

- '/tables' (std_msgs/Int16MultiArray): Publishes table numbers for food delivery as an array.
- '/heard' (std_msgs/Int16): Sends feedback to the main control node to take further action.

- External Libraries:

- 'speech_recognition': Handles audio input for speech recognition.
- 'pyttsx3': Provides text-to-speech functionality for audio feedback.

- Functionality:

- Utilizes 'speech_recognition' to listen for spoken commands.
- Recognizes specific commands like "Butler" and "table X" to trigger corresponding actions.
- Publishes table numbers extracted from spoken commands for delivery.

3. Navigation Program ('navigation_node.py')

This node facilitates the robot's movement between predefined positions within the restaurant.

- Dictionary:

- 'positions': Maps numeric identifiers to specific coordinates representing locations within the restaurant (e.g., home position, kitchen, tables).

- Functionality:

- Provides a 'move_to' function that retrieves coordinates from the 'positions' dictionary and sends it to the Navigation Stack and Path Planning packages (Navigation Stack was not implemented as it was not asked).
- ROS service or ROS action can be implemented for a closed loop Navigation system.

Conclusion

The robot butler system deployed at French Door Cafe represents a significant advancement in restaurant automation, efficiently managing food delivery tasks with enhanced flexibility and responsiveness. Designed to replace traditional butler roles, the system seamlessly integrates multiple functionalities and adapts to dynamic operational demands.

Key Features and Achievements

1. **Task Completion:** The robot successfully executes all specified tasks, including navigation from home to the kitchen, delivering orders to multiple tables, and returning to the home position.
2. **Voice Recognition Integration:** Utilising advanced speech recognition capabilities, the system responds to verbal commands such as initiating tasks and confirming delivery destinations.
3. **Error Handling and Adaptability:** Implements robust error handling mechanisms, including timeouts for unconfirmed actions and adaptive responses to order cancellations mid-delivery.
4. **Multi-Table Management:** Efficiently manages simultaneous orders for multiple tables, optimising delivery routes and ensuring timely service to all customers.