Robot Programming Methods

Krzysztof Rudnicki, 307585

November 19, 2023

Chapter 1

System Controller Design

1.1 System Initialization and Configuration

- **Initialization:** On receiving the START command, initialize the system, calibrating the manipulator's position and ensuring the Kinect sensor is operational.
- **Configuration:** Load predefined settings for object size, shapes (triangle, square, circle), and corresponding mold types.

1.2 Sensor Integration

- **Kinect Sensor:** Use the Kinect sensor to continuously monitor the conveyor. Implement image processing algorithms to detect the presence and shape of objects on the conveyor.
- **Encoders:** Integrate feedback from the encoders on the electric motors to precisely control the position and movement of the manipulator.

1.3 Object Detection and Classification

- **Image Processing:** Process the images from the Kinect sensor to identify object shapes and positions. Use shape detection algorithms to classify objects as triangles, squares, or circles.

- **Localization:** Calculate the position of each detected object relative to the manipulator's base coordinate system.

1.4 Motion Control

- **Path Planning:** For each detected object, plan a trajectory for the manipulator to pick the object from the conveyor and place it into the corresponding mold.
- **Manipulator Control:** Use the encoder feedback to control the 6 DOF manipulator, ensuring precise movement. Implement inverse kinematics algorithms for accurate positioning.
- **Gripper Control:** Control the suction gripper to pick and release objects, synchronizing its operation with the manipulator's movements.

1.5 Mold Handling

- **Mold Matching:** Match each detected object with the corresponding mold type (triangle, square, circle).
- **Insertion Sequence:** Control the manipulator to place each object into the designated mold. Ensure molds are replaced as soon as an object is inserted.

1.6 System Monitoring and Feedback

- **Real-time Monitoring:** Continuously monitor the conveyor and feeder status, adjusting the manipulator's operation accordingly.
- **Error Handling:** Implement error detection and handling mechanisms for scenarios like misaligned objects, system malfunctions, or unexpected interruptions.

1.7 System Termination

- **Stop Command:** On receiving the STOP command, safely terminate the system's operation. Ensure the manipulator is returned to a safe position and all active processes are halted.

1.8 User Interface and Communication

- **Status Indicators:** Provide real-time feedback on system status, including current operation, detected objects, and any errors or warnings.
- **Command Interface:** Implement a communication interface for receiving START and STOP commands and potentially for manual override or system diagnostics.

1.9 Software and Hardware Integration

- **Software Framework:** Choose an appropriate software framework that supports real-time control, image processing, and communication with all hardware components.
- **Hardware Compatibility:** Ensure all software components are compatible with the hardware, especially the Kinect sensor, the encoders, and the electric motors of the manipulator.

1.10 Testing and Calibration

- **Simulation Testing:** Before deploying, simulate the system's operation to identify and rectify potential issues.
- **Calibration:** Regularly calibrate the system to ensure accuracy, particularly the Kinect sensor and the manipulator's positioning.

Chapter 2

System Structure in Terms of Agents

2.1 Agents and Their Internal Structure

2.1.1 Sensing Agent

- **Internal Structure:** Consists of a Kinect sensor and encoders.
- **Sampling Rate:** 60 Hz for Kinect, 100 Hz for encoders.

2.1.2 Manipulator Agent

- **Internal Structure:** 6 DOF robotic arm with electric motors and a suction gripper.
- **Sampling Rate:** 10-100 Hz, depending on motion complexity.

2.1.3 Control Agent

- **Internal Structure:** Central processing unit integrating inputs and controlling the manipulator.
- **Sampling Rate:** Up to 100 Hz for real-time responsiveness.

2.2 General Behavior of Virtual Effectors and Receptors

- **Virtual Effectors:** Execute actions based on processed data.
- **Virtual Receptors:** Receive and process sensory inputs.

2.3 Data Structures within the Control Subsystem

- **Buffers for Sensory Data:** Storage for real-time sensor data.
- **Command Queue:** Buffer for storing control commands.
- **State Information:** Data structure for storing the current state.

2.4 Transition Functions and Terminal Conditions

- **Transition Function:** $T(s, a) = s'$ where s is the current state, a is the action, and s' is the new state.
- **Terminal Conditions:** Conditions under which a state transition occurs.

2.5 Structure of the FSM of the Control Subsystem

- **FSM Graph:** Nodes represent behaviors and arcs represent transitions.
- **Nodes:** Idle, Detecting Objects, Moving to Object, Picking Object, Moving to Mold, Placing Object, Returning to Initial Position.
- **Transitions:** Defined by predicates representing initial conditions.