Robotics Software Engineer Take-Home Challenge

Proximity-Based Speed Control with Emergency Stop

Design and implement a modular control system that adjusts the cobot's speed based on proximity input and responds immediately to an emergency stop signal.

Functional Requirements:

- **Speed Control Logic**:
    - `FULL_SPEED` : No object within 800 mm
    - `SLOW` : Object within 800–400 mm
    - `STOP` : Object within 400 mm
- **Emergency Stop**:
    - When the emergency stop is triggered, the cobot must immediately stop, regardless of proximity input.
    - Once cleared, the system should resume normal operation.

Technical Requirements:

- **Architecture**:
    - Implement the system using multi-threading, multiprocessing, or ROS2 multi-node architecture. Use ROS2 publishers/subscribers or inter-process communication (IPC) to simulate real-world modularity.
    - Separate the following concerns into distinct threads/processes/nodes:
        - Proximity Sensor Input
        - Emergency Stop Monitoring
        - Speed Control Logic
        - (Optional) State Logger or Visualizer
- **Language**: Python, Rust or C++ (ROS2-style preferred, but not required)
- **Simulation**:
    - Simulate proximity sensor input (e.g., random or scripted values).
    - Simulate emergency stop input (e.g., keyboard input, toggle file, or timed trigger).
    - If using ROS2, use `ros2_ur5_interface` for simulation.
- **Output**:
    - Print or log the current speed state (`FULL_SPEED` , `SLOW` , `STOP`) and emergency stop status.

Bonus Points:

- Implement hysteresis to avoid rapid toggling between speed states.
- Add a state machine or event-driven architecture to manage transitions cleanly.
- Visualization.

Evaluation Criteria:

- System design and architecture
- Thoroughness of testing
- Code quality and maintainability
- Technical documentation

Submission Guidelines:

- Create a new GitHub repository
- Implement required features and test thoroughly
- Provide technical documentation
- Make your repository publicly accessible
- Prepare to discuss design decisions and implementation
- If using any code generator tools, mention it in the documentation