Simulation of Material Handling System using Mobile Robot

System Overview
This is a comprehensive web-based robot control system that integrates multiple navigation algorithms with ROS2 for autonomous material handling in industrial environments.

1. Control Algorithms Module

1.1 Proportional Control

Purpose: Basic proportional control for robot navigation **Logic**:

- Calculates linear velocity proportional to distance error
- Calculates angular velocity proportional to heading error
- Simple but effective for basic navigation tasks

Parameters:

- kp_linear: 0.8 (linear gain)
- kp_angular: 2.5 (angular gain)
- max_linear: 0.6 m/s (maximum linear velocity)
- max_angular: 1.2 rad/s (maximum angular velocity)

Algorithm:

---

linearVel = min(kp_linear × distance_error, max_linear)
angularVel = constrain(kp_angular × heading_error, ±max_angular)

1.2 PID Control

Purpose: Advanced control with integral and derivative terms for better performance

Logic:

- Proportional term: immediate response to current error
- Integral term: eliminates steady-state error
- Derivative term: reduces oscillations and overshoot

Parameters:

- Linear PID: Kp=1.0, Ki=0.15, Kd=0.08
- Angular PID: Kp=3.5, Ki=0.25, Kd=0.12
- Integral windup protection: ±1.0 limit

Algorithm:

// Linear PID
linearVel = Kp×error + Ki×∫error×dt + Kd×(error-prev_error)/dt

// Angular PID
angularVel = Kp×heading_error + Ki×∫heading_error×dt + Kd×(heading_error-prev_heading_error)/dt
---

1.3 Pure Pursuit Control

Purpose: Path-following algorithm commonly used in autonomous vehicles **Logic**:

- Calculates curvature to a lookahead point
- Transforms target to robot's local coordinate frame
- Uses geometric relationship between robot and target

Parameters:

- lookahead_distance: 0.6m (lookahead distance)
- kp_linear: 0.9 (linear velocity gain)
- max_linear: 0.65 m/s
- max_angular: 1.3 rad/s

Algorithm:

// Transform to robot frame
local_x = cos(θ)×dx + sin(θ)×dy
local_y = -sin(θ)×dx + cos(θ)×dy

// Calculate curvature
curvature = 2×local_y / (lookahead_distance^2)
angularVel = curvature × linearVel

1.4 State Machine Control

Purpose: Hierarchical control with distinct navigation states **Logic**:

- Separate states for different behaviors
- Alignment phase followed by forward motion
- Robust handling of complex navigation scenarios

States:

- IDLE: Waiting for commands
- NAVIGATING: Moving toward target
- PICKING: Performing pickup operation
- DROPPING: Performing dropoff operation
- ERROR: Handling error conditions

Parameters:

- arrival_tolerance: 0.18m (arrival detection threshold)
- heading_tolerance: 0.2 rad (heading alignment threshold)
- max_linear: 0.5 m/s

- max_angular: 1.0 rad/s

2. ROS2 Integration Module

2.1 Connection Architecture

ROS Bridge: WebSocket connection to ROS2 system
- URL: ws://localhost:9090
- Handles connection, error, and close events
- Automatic system pause on connection loss

2.2 Topic Communication

Publisher - Command Velocity (/cmd_vel)

Message Type: geometry_msgs/Twist **Purpose**: Send velocity commands to robot

Structure:

```
{
  linear: { x: linear_velocity, y: 0, z: 0 },
  angular: { x: 0, y: 0, z: angular_velocity }
}
```

Subscriber - Odometry (/odom)

Message Type: nav_msgs/Odometry **Purpose**: Receive robot pose and velocity feedback

Data Extracted:
- Position: message.pose.pose.position.{x,y}
- Orientation: Quaternion → Euler conversion
- Velocity: message.twist.twist.linear

2.3 Safety Features
- **Connection Monitoring**: Continuous status checking
- **Emergency Stop**: Immediate velocity zeroing
- **Timeout Handling**: Automatic system pause on communication loss
- **Manual Override**: Direct velocity control when needed

3. Job Management System

3.1 Job Structure

```
{
  id: timestamp,
```

```
  pickup: location_id,
  dropoff: location_id,
  quantity: number,
  status: 'QUEUED'|'ACTIVE'|'COMPLETED'|'CANCELLED',
  stage: 'PICKUP'|'TRANSPORT'|'DROPOFF'
}
```

3.2 Job Execution Flow

1. **Queue Management**: FIFO job processing
2. **Pickup Phase**: Navigate to pickup location
3. **Material Transfer**: Simulate item pickup (2.5s delay)
4. **Transport Phase**: Navigate to dropoff location
5. **Dropoff Phase**: Simulate item dropoff (2.5s delay)
6. **Completion**: Update statistics and start next job

3.3 Material Tracking

- **Inventory Management**: Track items at each location
- **Validation**: Ensure sufficient items for pickup
- **Real-time Updates**: Live inventory display

4. Navigation System

4.1 Location Management

Predefined Locations:

- Home Base: (0.0, 0.0)
- Storage A: (0.0, 2.0)
- Storage B: (2.0, 2.0)
- Workstation: (2.0, 0.0)
- Loading Dock: (1.0, 0.0)
- Inspection: (1.0, 2.0)

4.2 Navigation Control Loop

Update Rate: 10 Hz (100ms intervals) **Process**:

1. Calculate distance and heading error
2. Apply selected control algorithm
3. Smooth velocity commands
4. Publish to ROS2
5. Monitor arrival conditions

4.3 Velocity Smoothing

Purpose: Prevent abrupt velocity changes **Algorithm**:

---

maxChange = maxAcceleration × dt

currentVel += constrain(targetVel - currentVel, ±maxChange)

---

5. Performance Monitoring

5.1 Real-time Metrics

- **Update Rate**: Communication frequency (Hz)
- **Position Tracking**: Current robot coordinates
- **Error Metrics**: Distance and heading errors
- **Velocity Monitoring**: Linear and angular velocities

5.2 Excel Data Collection

Data Points Collected:

- Timestamp and relative time
- Robot position (x, y)
- Velocity and heading
- Control mode and system state
- Job information and stage
- Control errors and commands

Export Features:

- **Raw Data Sheet**: All collected data points
- **Statistics Sheet**: Performance summary
- **Control Analysis**: Algorithm comparison
- **Job Timeline**: Event chronology
- **Excel Formulas**: Analysis templates

6. System States and Transitions

6.1 System States

- **IDLE**: System ready, no active jobs
- **RUNNING**: Actively executing jobs
- **PAUSED**: Temporarily stopped, resumable

6.2 State Transitions

IDLE → RUNNING: Start system with jobs queued
RUNNING → PAUSED: Manual pause or error
PAUSED → RUNNING: Resume operation
ANY → IDLE: Stop system or emergency stop

7. Error Handling and Safety

7.1 Connection Management

- **Auto-pause**: System pauses on ROS disconnect
- **Reconnection**: Automatic status updates
- **Timeout Protection**: Prevents hung operations

7.2 Emergency Controls

- **Emergency Stop**: Immediate halt of all motion
- **Robot Reset**: Clear all control states
- **Queue Clear**: Remove all pending jobs
- **Return Home**: Navigate to safe position

8. User Interface Architecture

8.1 Control Panels

- **System Control**: Algorithm selection and parameters
- **Location Management**: Facility layout and status
- **Job Creation**: Transport task definition
- **Advanced Controls**: Emergency and manual operations

8.2 Status Display

- **Real-time Monitoring**: System state and metrics
- **Job Queue**: Active and pending tasks
- **Performance Indicators**: Control effectiveness
- **System Log**: Event history and diagnostics

9. Technical Specifications

9.1 Performance Characteristics

- **Update Rate**: 10 Hz control loop
- **Arrival Tolerance**: 0.15-0.25m (algorithm dependent)
- **Maximum Velocity**: 0.5-0.7 m/s (algorithm dependent)
- **Data Collection**: Up to 10 samples/second

9.2 Communication Protocol

- **WebSocket**: Real-time bidirectional communication
- **Message Format**: JSON-serialized ROS2 messages
- **Error Recovery**: Automatic reconnection attempts
- **Bandwidth**: Optimized for real-time control

10. Integration Requirements

10.1 ROS2 Dependencies

- **ros2_bridge**: WebSocket bridge package
- **geometry_msgs**: Twist message support
- **nav_msgs**: Odometry message support
- **Robot Platform**: Differential drive robot

10.2 Web Browser Requirements

- **Modern Browser**: WebSocket support
- **JavaScript**: ES6+ features
- **Excel Export**: SheetJS library support

- **Real-time Display**: Canvas/SVG rendering

This system demonstrates a comprehensive integration of multiple control algorithms with ROS2, providing a robust platform for autonomous material handling research and development.