Line-by-Line Manual for turtlerightturn() Function

This manual breaks down the MATLAB function `turtlerightturn()` by providing detailed explanations with line number references. The function is designed for a TurtleBot3 robot using ROS 2. It processes camera input to detect lane lines and performs a right-turn maneuver using PID control. Each functional block is explained for easier understanding and future modification.

1. ROS 2 Node Initialization (Lines 3-12)

Lines 3-12 set up ROS 2 nodes, publishers, and subscribers:

- Line 3: Defines the image topic from the camera.
- Lines 4-6: Create a node and subscribe to the camera image.
- Lines 7-8: Publisher and message object for processed images.
- Lines 9-10: Velocity publisher for robot motion commands.
- Line 11: Message object for velocity.
- Line 12: Stop signal subscriber for emergency halt.

2. Image Preprocessing & PID UI (Lines 13-21)

- Line 13: Sobel filter kernel for detecting vertical edges.
- Line 14: Cropping threshold to focus on road surface.
- Lines 15-17: Initialize PID controller state variables and timer.
- Line 18: Stop flag.
- Lines 19-21: Create a user interface for tuning P, I, D gains live.

3. Main Control Loop (Lines 22-104)

- Lines 23-27: Continuously checks for a stop signal.
- Lines 28-32: Receives and decodes the camera image.
- Lines 33-34: Crops the image and converts it to grayscale.
- Lines 35-37: Applies yellow mask and Sobel edge detection.
- Line 38: Thresholds grayscale edge detection to ignore noise.
- Lines 39-40: Combines yellow mask and grayscale edges.

- Lines 42-45: Uses Hough transform to detect lane lines.

- Line 47: Initializes overlay image for visualization.

- Lines 48-54: Decides whether to continue or exit the turn based on detected lines.

4. Detect and Process Rightmost Line (Lines 55-67)

- Lines 56-62: Adjust coordinates and determine which line is furthest to the right (max X value).

- Line 63: Stores the selected rightmost line.

5. Line Drawing and Averaging (Lines 68-80)

- Lines 69-70: Extract endpoints and draw the line.

- Lines 71-74: Interpolate intermediate points along the line.

- Lines 75-79: Draws a circle at each point and calculates average X position for control.

6. PID Control Computation (Lines 81-96)

- Line 81: Sets an offset to stay left of the right line.

- Line 82: Calculates Cross-Track Error (CTE).

- Line 84: Applies a deadband if CTE is small.

- Lines 87-91: Read PID values from sliders and compute PID terms.

- Line 92: Clamp the angular velocity.

- Line 93: Set forward linear velocity.

- Lines 94-95: Display computed terms and total angular command.

7. Velocity Command and Emergency Stop (Lines 97-102)

- Line 97-100: Assign computed velocities to the ROS message and publish.

- Line 101-102: If no lines detected, the robot halts by setting zero velocities.

8. Visualization and Output (Lines 103-104)

- Line 103: Displays the overlay image with lines in a MATLAB figure.

- Line 104: Publishes the image to the ROS topic for external visualization.

9. createMask Helper Function (Lines 106-113)

- Line 107: Converts RGB to HSV.

- Lines 108-110: Thresholds HSV to isolate yellow pixels.

- Lines 111-113: Applies the binary mask to RGB image for masking effect.