**Lane-Assist Robot Lab**

**Simulating Real-World Vehicle Safety Systems**

**Overview**

In this lab, you’ll explore how a robot can use its sensors to **detect and avoid lane markings**—simulating the behavior of modern **lane-assist systems** in vehicles. By adjusting key parameters like speed and sensor sensitivity, you’ll fine-tune its ability to stay "in the lane" while avoiding crossing black tape boundaries.

**Objective**

* Understand how IR sensors can be used to simulate lane detection
* Tune parameters to improve the robot’s boundary avoidance behavior
* Explore **how speed affects** the robot's ability to stay within a lane
* Design your own test track to challenge the robot’s lane-assist behavior

**Part 1: Setup and Calibration**

1. **Place** the robot on a clean surface with black electrical tape forming a square, lane, or path
2. **Press A** to begin calibration. The robot will spin briefly to capture values for white floor and black tape
3. After calibration, **press A again** to begin forward movement. The robot will attempt to stay within boundaries by **steering away** from dark lines

**Part 2: Parameter Tuning Experiment**

Try different parameter combinations and observe the result. **Your goal is to find the fastest speed that still avoids tape reliably.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Trial #** | **Speed** | **Border Threshold** | **Turn Strength** | **Possible Prediction of Results**  **(may not though)** | **Your Results/observation**  May be different than previous column |
| 1 | 2500 | 500 | 0.2 | Too sensitive, many false positives |  |
| 2 | 2000 | 600 | 0.4 | May wobble or turn too early |  |
| 3 | 1500 | 800 | 0.6 | More stable but slower |  |
| 4 | 2000 | 850 | 0.4 | Likely to avoid and recover cleanly |  |
| 5 | 2000 | 900 | 0.4 | May miss tight turns, less responsive |  |
| 6 | ⭐ **Your Ideal** |  |  | Most successful settings |  |

**Part 3: Creative Lane Course Design**

Design a lane-marked track using black tape. Include:

* **Straight paths**
* **Tight and gentle curves**
* Optional: add broken lines to simulate a passing lane or an intersection that splits the lane in two paths

**Goal:**

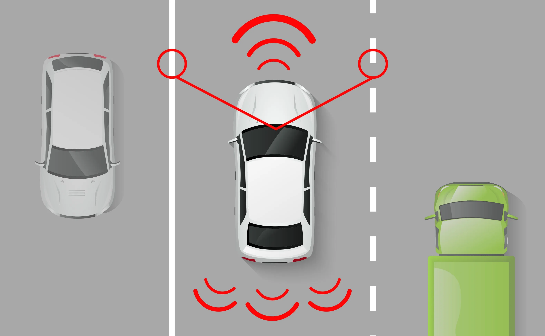
Find the **maximum speed** the robot can handle while still successfully staying within the lane!

**Goal:**

Find the **maximum speed** the robot can handle while still successfully staying within the lane!

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test #** | **Curve Type** | **Turn Radius or Angle** | **Speed Tested** | **Did Robot Avoid Line?** | **Notes (e.g., overcorrect, failed)** |
| 1 | Gentle curve | ~30 cm radius |  | ✅ / ❌ |  |
| 2 | Moderate curve | ~20 cm radius |  | ✅ / ❌ |  |
| 3 | Tight curve | ~10 cm radius |  | ✅ / ❌ |  |
| 4 | Sharp corner | ~90° turn |  | ✅ / ❌ |  |

**Real-World Connection**

Modern cars use **lane departure warning systems** and **lane-keeping assist** to help drivers stay in their lanes. These systems use cameras and sensors to detect road lines and steer the car back if it drifts. This lab demonstrates a similar concept using IR sensors and basic motor control.

Instead of line sensors and black tape, real cars use:

* **Forward-facing cameras** (usually mounted near the rearview mirror) that look at the road ahead.
* **Computer vision algorithms** to detect white or yellow lane markings and determine if the vehicle is drifting.
* **Electric steering controls** to gently nudge the car back into the lane — this is called **Lane Keeping Assist (LKA)**.
* Some systems also include **Lane Departure Warnings (LDW)** that beep or vibrate the steering wheel if the driver starts to leave the lane unintentionally.

Advanced vehicles combine this with **radar or lidar** to track nearby vehicles and keep the car centered in the lane, even in bad weather.

➡️ Just like your robot had to tune its sensor thresholds and adjust its steering to stay between the lines, real cars need careful calibration and advanced software to help drivers stay safe on the road.