Team 1 Lab 3 Oral Briefing

Clementine Mitchell, Martina Stadler, Nick Villanueva, Samir Wadhwania, Jake Liguori, Jose Gomez

Overview

- ROS Nodes

- Wall Detection

- Regression
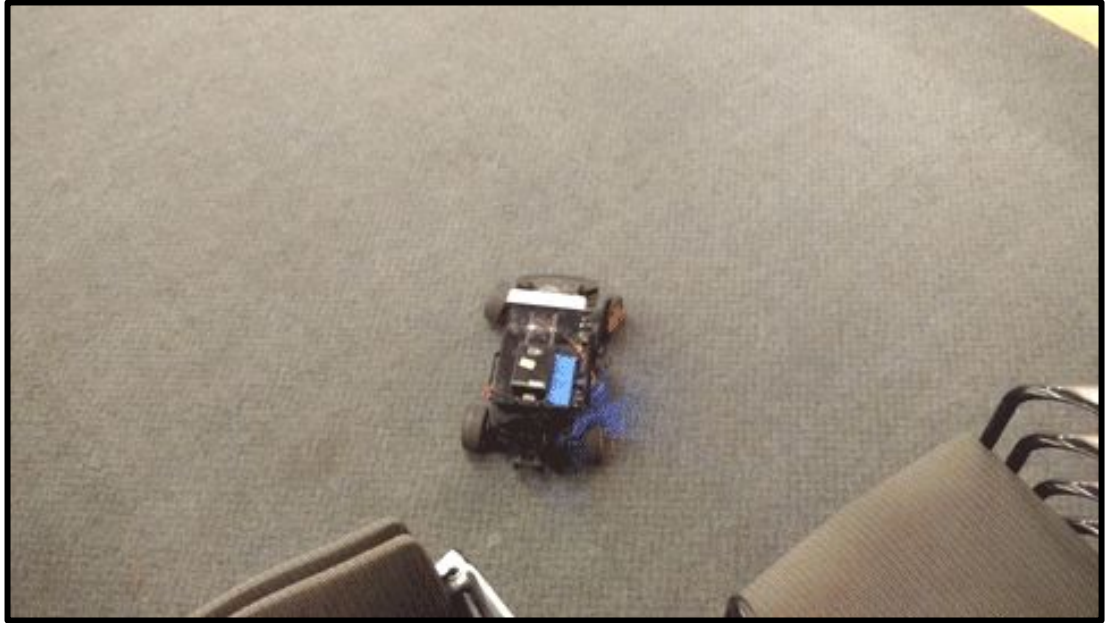
- Control

- Lessons Learned

Figure 1: Racecar driving and stopping autonomously

Why Subscribe and Publish Nodes Before Algorithms Ready?

- Early Bug Detection

- Modularity

- Eliminate Lag Time

Node Architecture

- **Question** - how to build a system for easy integration in future projects?
- Created three nodes to handle mission objectives:
    - Regression Node
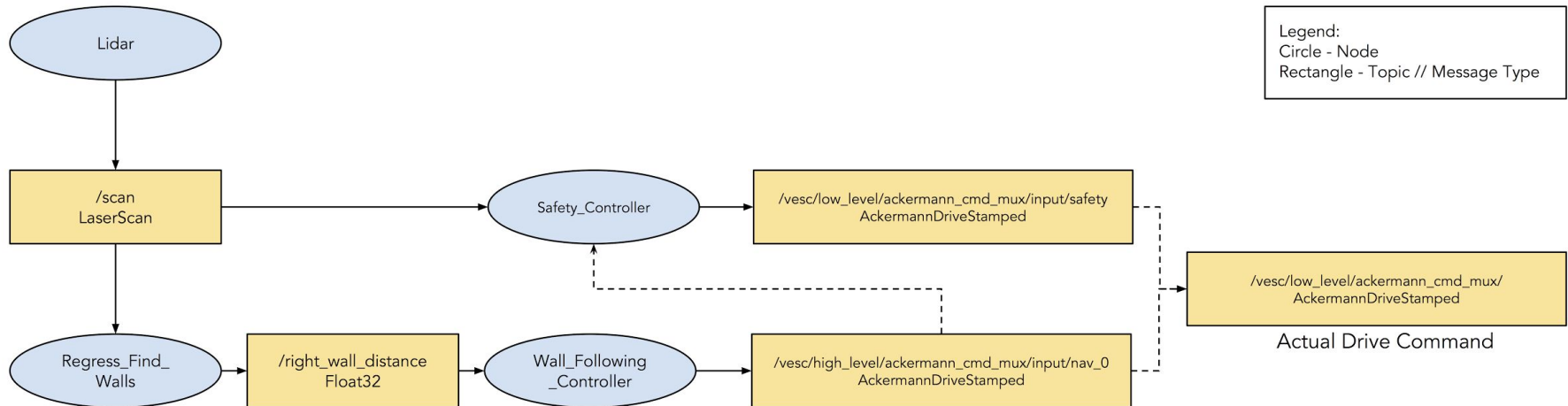   - Safety Node
    - Wall Following Controller

Figure 2: Nodes added to existing system

Raw LiDAR data was processed to detect walls.

1. Convert Raw LiDAR data to Cartesian coordinates.

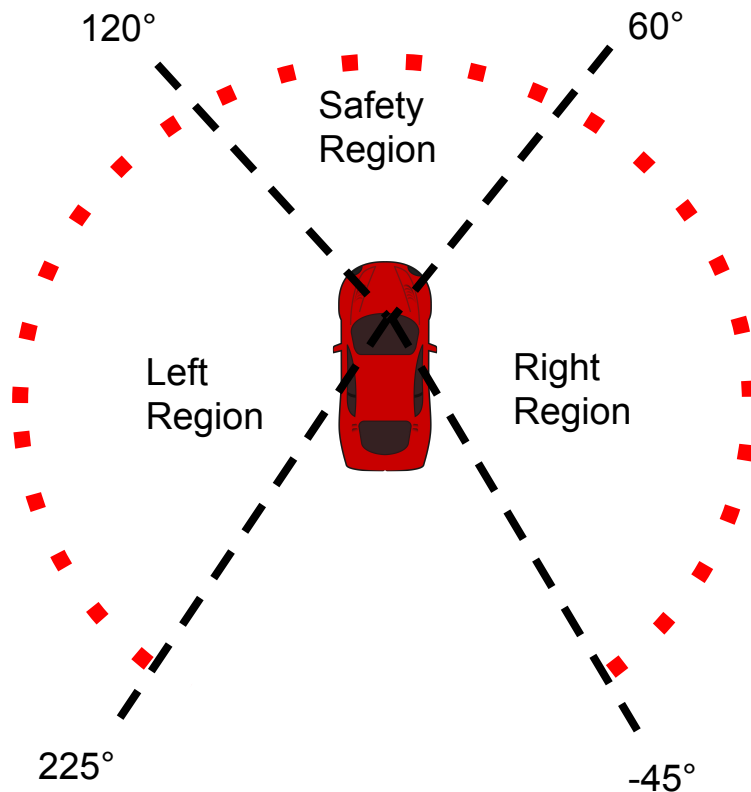2. Separate Cartesian points into left and right region.

120°   60°

Safety
Region

Left
Region

Right
Region

225°   -45°

Figure 3: Simple LiDAR partition model

5

Raw LiDAR data was processed to detect walls.

3. Filter out Infinite values, which correspond to no objects detected.

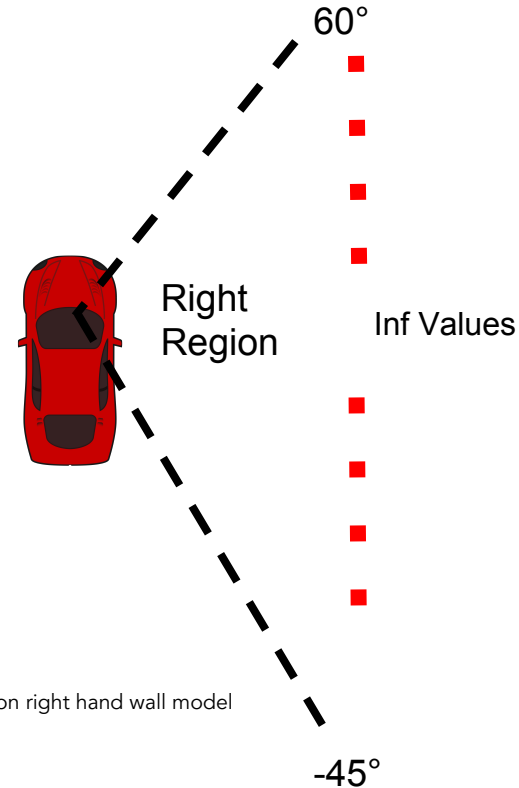4. Identify walls using infinite values as flags.

60°

Right
Region

Inf Values

Figure 4: LiDAR partition right hand wall model

-45°

6

Raw LiDAR data was processed to detect walls.

5. Store points corresponding to individual walls inside dictionary
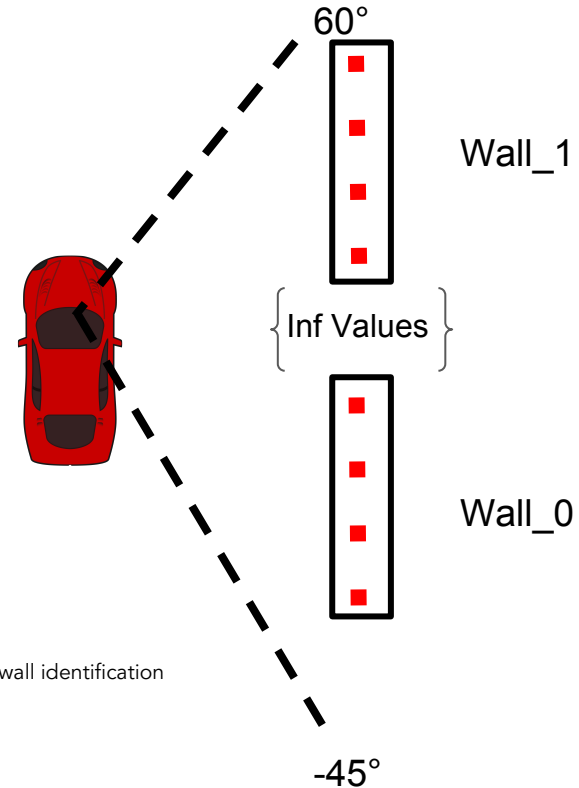
6. Pass information for regression.

60°

Wall_1

{ Inf Values }

Wall_0

Figure 5: LiDAR wall identification

-45°

Fitting lines to wall data to calculate distance and orientation.
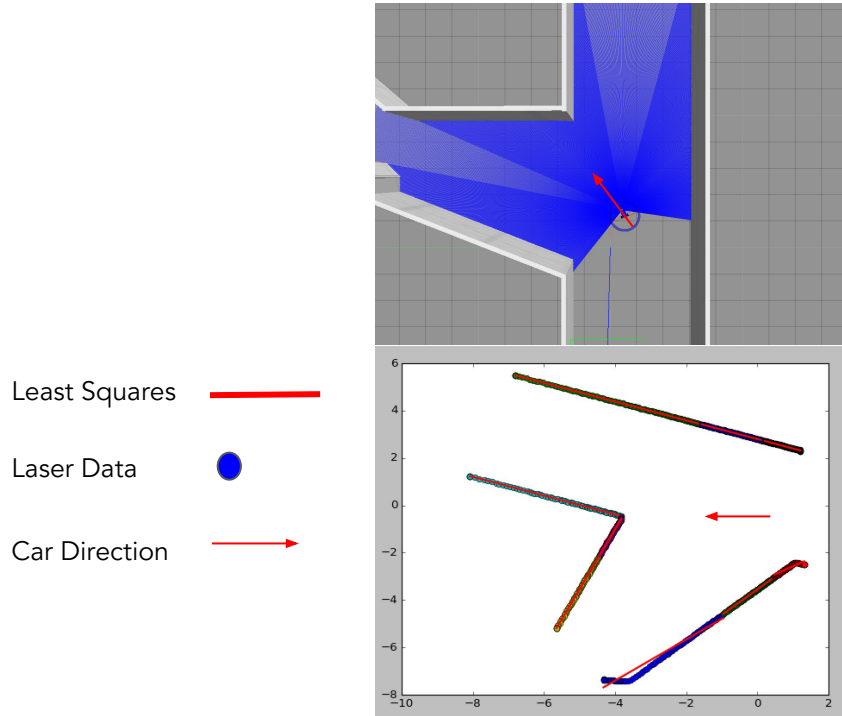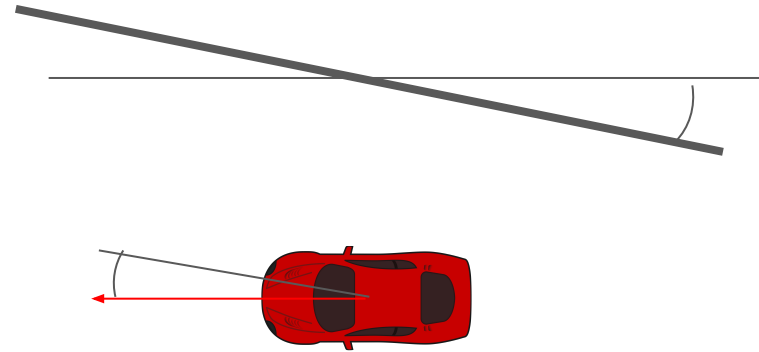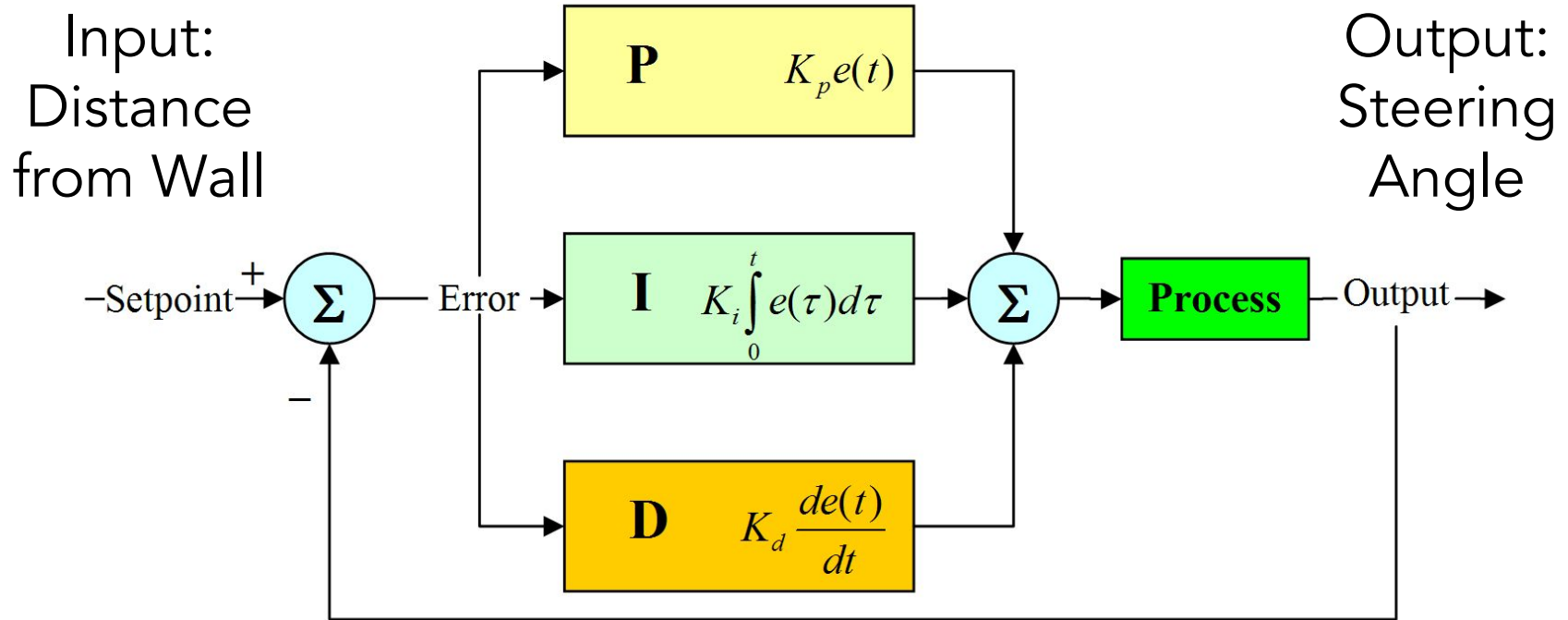

Figure 6: Wall Detection

Least Squares

Laser Data

Car Direction

Orientation = arctan(m)

Figure 7: Orientation Calculation

A PID controller was used to control the vehicle.

Input:
Distance
from Wall

Output:
Steering
Angle

$$\textbf{P} \qquad K_p e(t)$$

$$\textbf{I} \qquad K_i \int_0^t e(\tau)d\tau$$

$$\textbf{D} \qquad K_d \frac{de(t)}{dt}$$

−Setpoint → + Σ − → Error → Σ → **Process** → Output →

Source: https://radhesh.files.wordpress.com/2008/05/pid.jpg

Figure 8: PID design

9

A low pass filter was used for obstacle detection.

Relevant
laser scan
data - ranges → Low pass filter → Is there a point within 0.5 meters of the racecar? → Yes → **STOP**
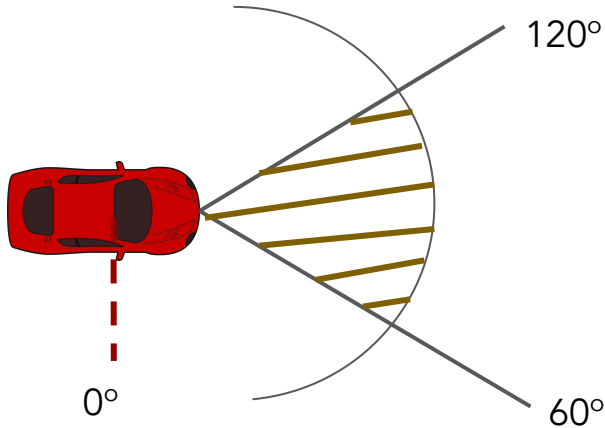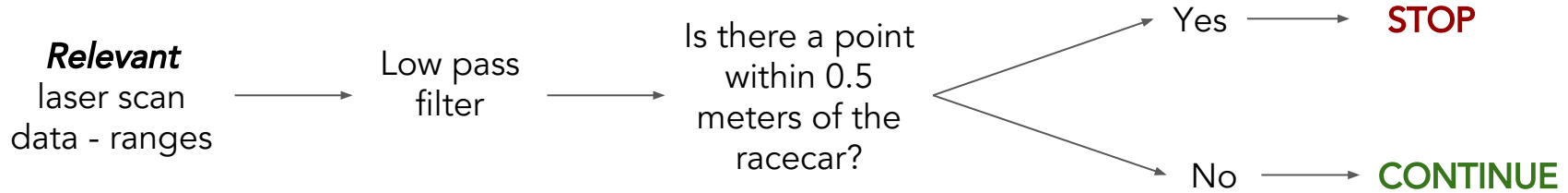
→ No → **CONTINUE**

Figure 9: Region used in safety controller

Figure 10: Racecar stopping using safety controller

Lessons learned:

Technical:

- Update the workspace manually if git is having trouble merging and updating.
- Try different installation methods.
- Check that controls are oriented correctly.

Teamwork:

- Meet as a group earlier in the process and more frequently.
- Allow more time for debugging - things don't necessarily work the first time!

Questions?