

AUTONOMOUS CAR

Rahul N

14D170029

Rohith Prakash

14D170022

Micky

140010059

PROBLEM STATEMENT

- Goal 1 : Make a stationary obstacle avoiding autonomous path following robot
- Goal 2 : Make a incoming (moving) object sensor and give path to it
- Goal 3 : Overtake a slow object

Code Parts and names

Line Follower Code

- Senses stationary object



Stationary Obstacle Code

- Rotates by 90 degrees
- Maneuvers around the object
- Realigns to white line
- Back to line follower code (1)

Senses incoming object



Slave Code

- Senses incoming object
- Gives path to it
- Realigns to white line
- Back to line follower code (1)

Senses slowly moving object



Master Code

- Analyses object speed
- Maneuvers around the object
- Realigns to white line
- Back to line follower code (1)

WHICH CODE TO EXECUTE ?

Video – Avoiding Stationary object



Step 1 : Bot senses the obstacle

Step 2 : Bot stops the line follower code

Step 3 : Bot executes right angle turning code

Step 4 : Bot executes obstacle maneuvering code

Step 5 : Bot senses the line and aligns itself and re-executes line follower code

Video – Overtaking slow object



Step 1 : Bot senses the slow moving object

Step 2 : Bot stops the line follower code and judges the object speed

Step 3 : Bot starts the master code and overtakes the object

Step 4 : Bot re-aligns itself to white line and re-executes line follower code

Video – Giving way to incoming object



Step 1 : Bot senses the incoming object

Step 2 : Bot stops the line follower code

Step 3 : Bot starts the slave code

Step 4 : Bot senses the object when it crosses

Step 5 : Bot re-aligns itself to white line and re-executes line follower code

Challenges

1) Error in ADC Readings

- **Problem :** ADC readings are not reliable as they keep varying – high Run time errors
- **Solution 1 :** Found the exact range where values are accurate(250-350) and run the program in that range
- **Solution 2 :** Not relying much on ADC values. We have backup plans

2) Varying Voltage

- **Problem :** Bot runs on a velocity controlled function and delay. If the voltage changes, the velocity changes and the delay wont suffice for the turn
- **Solution1 :** Minimized such cases and are relying more on ADC
- **Solution 2 :** Not using inbuilt battery

Challenges

3) Turning the bot at obstacle occurrence

- **Problem :** Needed the bot to turn at the obstacle occurrence by breaking the line follower code as per proximity sensor readings
- **Solution 1 :** Bypassed the problem by turning the bot by 90 degrees at obstacle occurrence
- **Solution 2 :** We are bringing the bot closer to the obstacle step by step
- **Solution 3 :** Include an extra sharp IR sensor at the two sides of the bot and get readings. (Not done)

4) Line follower does not work effectively at high speeds

- **Problem :** Bot crosses over the line at sharp turns and at the line rejoining junction after crossing the obstacle
- **Solution :** Reduced bot speed by including velocity control at obstacle occurrences

Future Work



- Extend the project usability to GPS assisted bot, which uses GPS to define a path between the bot location and a remote location
- Use the idea to make a smart car key which signals the car to reach its owner destination avoiding all the objects and moving cars in its path
- Signaling the bot (city lights). –Stop the bot with red signal and start with green
- Car to car interaction

Future Work

Make a Google car

THANK YOU