Can you create an effective wall-following robot using a simple IR sensor?

Shivam Gaind

Objective

- Our RedBot was designed to trace the path of a wall using an IR sensor
- We used values form our IR sensor to keep the RedBot at a constant distance from the wall

System Design

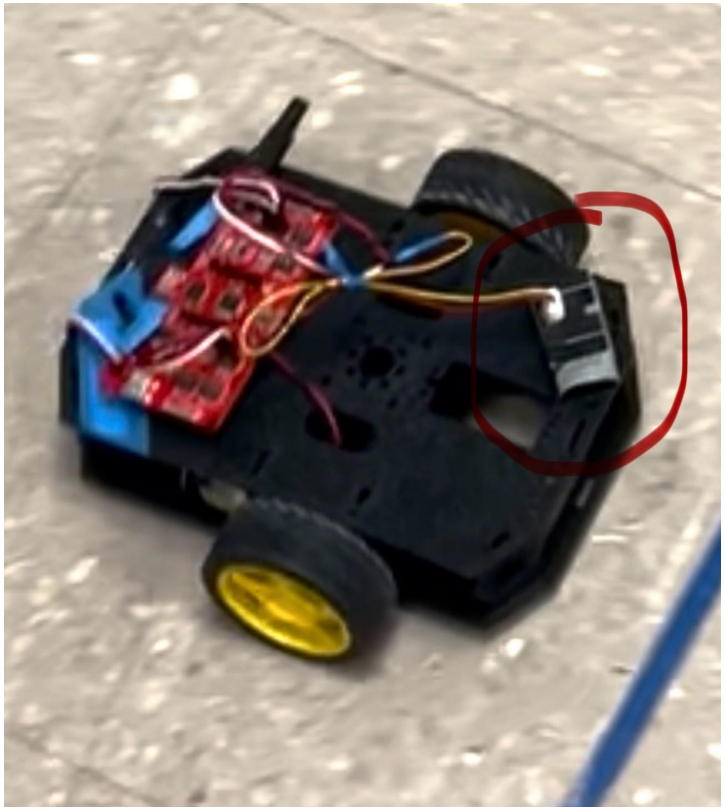
- We used proportional control to make sure the RedBot maintains the desired distance from the wall and to correct the Redbot if was too close or too far from the wall

- We tuned the proportional gain (Kp) through trial and error.
  - At a low Kp value, the robot did not respond as fast and was sluggish
  - As Kp value became larger, RedBot became more responsive.
  - Using a very large Kp value caused our RedBot to become unstable.
  - Through trial and error, we found that Kp = 0.3 worked best for our RedBot.

```
error = refrence - sensorVal; // our error is our refrence value minus the sensor value, ex. if 350 was sensor value we get error value of -50
Proportionalgain = Kp*(error); // we found proportioal gain by multiping Kp by the error, the value of Kp was found through trial and error
lms= Basespeed - Proportionalgain;  // left motor speed is Basespeed - Proportional gain
rms= Basespeed + Proportionalgain;   // right motor speed is Basespeed + Proportional gain
```

Sensor Placement

- We placed the sensor at an angle; this allowed us to get a view of both walls along the RedBot and ahead of the RedBot.
- Originally, the sensor was mounted at less of an angle and facing more toward the wall.
- This made it difficult for our RedBot to detect turns.
- Through trial and error, we found that this angled placement was the best option for measuring values using our IR sensor.



Challenges

- We had a challenge where our algorithm would cause the RedBot to try and travel at a speed that exceeded the max value for speed, which was 255.

- To solve this issue, we implemented an if statement for the right motor and left motor.

```
if(lms > 255);  // if left motor speed exceeds 255 its set to 255 because that is the max speed
{
  lms =255;
}
   if(rms > 255);  // if right motor speed exceeds 255 its set to 255 because that is the max speed
{
  rms =255;
}
```

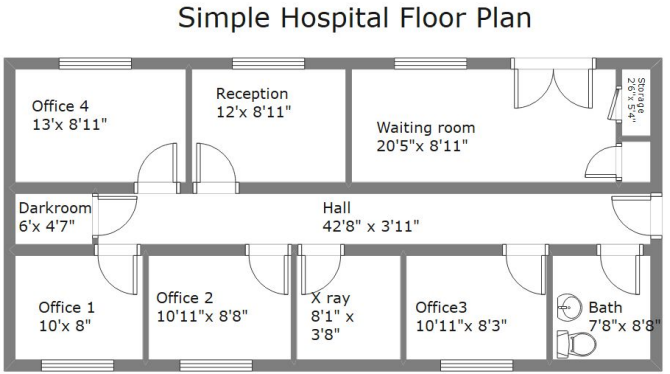
Challenges

- Another challenge we faced was trouble shooting our RedBot.
- Originally our IR sensor was not working, so we had to try and test multiple IR sensors to find one that would work.
- We did this using the serial monitor.
- We displayed values for the IR sensor in the serial monitor along with other parameters like the left motor speed, right motor speed, error, and proportional gain.

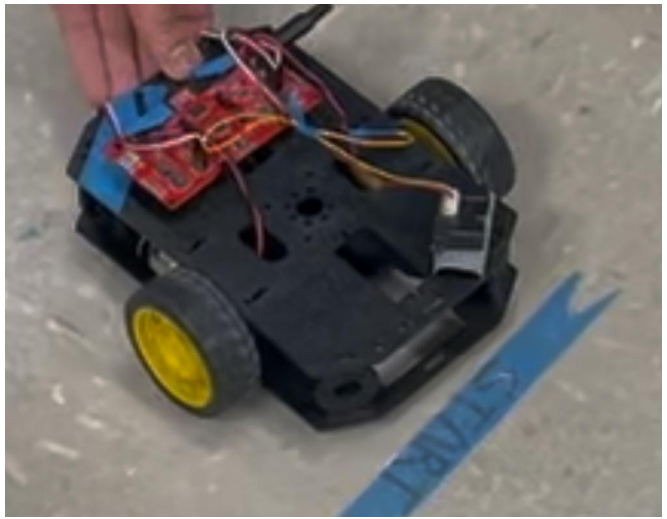
```
Serial.print(sensorVal); // prints the sensor value to the serial monitor
Serial.print(",");
 Serial.print(error); // prints the error value to the serial monitor
Serial.print(",");
Serial.print(Proportionalgain); // prints the proportional gain value to the serial monitor
Serial.print(",");
Serial.print(lms); // prints the left motor speed value to the serial monitor
Serial.print(",");
Serial.println(rms); // prints the right motor speed value to the serial monitor
```

Real World Application

- Our RedBot can be used to make deliveries within hospitals.
- By adding an attachment for the RedBot to contain items, we can use it to trace the walls within a hospital to find the appropriate room that it needs to make a delivery at.
- Doing so can help reduce the number of tasks nurses within a hospital are required to do and can help them focus more on patients rather than delivering materials.



Video



Conclusion

- Yes, it’s possible to make a wall following robot using an IR sensor
- With the right Kp value and sensor placement, the robot stayed at a steady distance from the wall
- Faced issues like sensor errors and speed limits, but resolved them
- Shows real potential in places like hospitals

thanks for watching