Memo-ECE425-LAB4

Date: 1/30/2022

To: Professor Berry

Team: Leonardo - Nathan Lee, Shantao Cao

The purpose of this Lab was to implement Braitenberg Vehicles using photosensors and build on top of last lab to use a hybrid controller for beacon docking. For this Lab, we used LEONARDO’s four IR sensors and added two photo resistive sensors. For Braitenberg Vehicles, we were able to execute the four vehicles by reacting to light. LEO was also able to execute a hybrid controller, and dock towards a beacon of light from wall following and return to its original location before docking.

**Prelab: Hybrid Controller**

Before working with LEO, we designed a hybrid controller (Appendix A) for executing beacon docking with all previous behaviors in last lab (wall following, wandering, etc.). All the behaviors from last lab was set to be in the reactive layer of the hybrid controller because it is simple states and functions to execute based on the environment. There is no though processing or recalling that LEO would need in any of the behavior states. The planning and middle layer is set when LEO detects a beacon of light and execute his docking control. This is in the planning and middle layer because once LEO has approached the light, he would need to think and plan out his return pathing to where he initially found the light.

**Photoresistor Testing**

After setting up a hybrid controller for LEO, we set up and tested the photo-resistive sensors. Since the sensors were photo-resistive, we set up a voltage divider and have the Arduino read the voltage drop across the sensor. This means that there will be a larger voltage drop when less light is present. The photo sensors were attach to the front of LEO and both sensors pointed directly ahead of it. Table 1 shows our initial calibrations of the photosensors. The photosensors data was much more constant than any other sensors on LEO. Further testing and calibration of the photosensors were done to calibrate distances to the readings (Appendix B). Upon both calibrating both left and right sensors to the environment, there shows little differences between the left and right sensors. When light is detected, the small difference between the left and right sensors (about 50 mV difference) is negligible compared to the difference between environment and the light beacon (about 250 mV difference). This lets LEO easily find which direction the beacon of light is coming from and could detect the angle of attack from where the beacon is coming from with little turning movements.

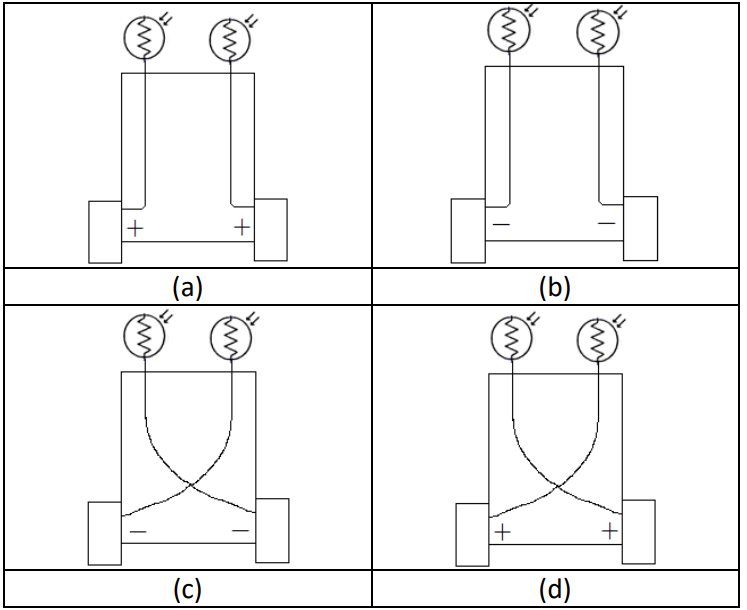
|  |  |  |
| --- | --- | --- |
|  | **Left (mV)** | **Right(mV)** |
| Ambient on table | 496 | 417 |
| Ambient under table | 220 | 209 |
| Sensor covered | <40 | <40 |
| In front of a light | >750 | >770 |

**Table 1:** Ambient and flashlight readings of the environment.

**Braitenberg Vehicles**

To test the photosensors in action, we set up the four reactive Braitenberg Vehicles. Figure 1 shows the four types of vehicles: Love, Fear, Exploration and Aggression. For the vehicles, the left or right sensor is coupled to their corresponding or opposite wheel based on the type of vehicle. For example, the Braitenberg Vehicle for Love would have the left sensor attached to the left wheel. And when light is present on the left side, the left wheel should decrease speed (pointing the robot towards the light). LEO was able to execute all four vehicles with some minor issues over looked. Since our light sensors were both facing directly ahead of LEO, if the light was centered or too far from LEO, it had difficulty differentiating which side the light source would be coming from and executed an unusual behavior. Putting the light source much closer and in front of one side of the photosensor demonstrated the four vehicle behaviors more clearly.

At this stage, we have also discovered that using the main loop in Arduino works much better for sensor pings and using Timer One Interrupt for behavior execution. This is because the Serial Print uses timer zero (main loop) and using the with the movements/behavior would change the timings of each step. Setting the movement in Timer One avoids any timing interrupt from Serial and makes the movement much cleaner. We discovered this a little too late and kept using the main loop for movement and Timer One for sensor pings because the code was in that structure last lab.



**Figure 1**: Braitenberg Vehicles uses sensors to react to their environment. The four types of vehicles are fear (a), love (b), exploration (c), and aggression (d)

**State Machine**

After setting up the last reactive control with the photosensors, we then implemented the photosensor’s love vehicle with a state machine. The state machine had random wander, Braitenberg Love and obstacle avoidance. The subsumption architecture for the state machine has random wander as the default state, which can then be suppressed by Braitenberg Love (if light is present) or obstacle avoidance (if there is an obstacle). Then the Braitenberg Love would then be suppressed by obstacle avoidance as well (if there is an obstacle). LEO has successfully executed the state machine with the addition of Braitenberg Love reactive control.

**Hybrid Control Docking**

The final stage is implementing light docking into last lab’s behaviors to create a hybrid controller. We took a simple approach to the docking method and had LEO only take linear movements. Since both photosensors are facing LEO’s front, it could easily turn and locate the direction of the light beacon. Therefore, LEO would first detect that light is present and move into the Middle and Deliberative Layer of hybrid control. LEO will then turn slowly to locate the direction of the beacon. Once the direction is found, LEO would approach the beacon and count how far it has gone. Once the front IR sensor triggers and sees in obstacle in front, LEO has successful reached the light beacon. LEO would then turn around and return to its original position (prior to detecting light) based off previously recorded data. LEO would finally turn back to its original orientation and continue its previous state.

This method proved to be successful, the only issue is when the light beacon triggers the hybrid control too early and LEO cannot detect the light. This happens when the light beacon is too far (more than 15 inches) and LEO confuses the ambient environmental light with the beacon light. LEO turns to the light beacon by finding the voltage difference between the left and right sensors. When the ambient light reads the same as the beacon light, LEO tends to turn towards a random direction when triggered. This issue is solved through turning the light beacon on after LEO is within 15 inches or less from the beacon. Other possible solutions would be increasing the threshold for docking or refining the calibration of ambient light. Another issue with docking, is that we have mounted both photosensors on front of LEO and have both pointing forwards. This really limits LEO’s field of view and only allows it to see light well in a 60-degree cone in front of it. The light beacon had to be angled to fit LEO’s field of view during demonstration and execution. A future improvement would be trying to broaden LEO’s field of view and angling the photosensors away from each other. This would then create other issues but allows LEO to react with a more diverse environment.

Overall, LEO successfully executed tasks using a hybrid controller and was able to utilize photosensors in doing so. There were a lot of limitations due to methods and field of view that could use future improvements. In doing so, there would be more calibrations and testing required to tune the controller to its environment.

**Appendix A: PreLab Hybrid Control Diagrams**

Diagram

Description automatically generated*The State Machine Diagram for light detection*

Diagram

Description automatically generated

*The Hybrid Control Diagram implementing Docking*

**Appendix B: Additional Photo-resistive Sensor Calibration**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Left (mV) | | | Right (mV) | | |
| Environment | Distance(in) | -45 degrees | 0 degree | 45 degrees | -45 degrees | 0 degree | 45 degrees |
| On the table | 3 | 634 | 570 | 551 | 508 | 569 | 777 |
| On the table | 6 | 761 | 599 | 523 | 463 | 696 | 762 |
| On the table | 9 | 700 | 554 | 514 | 487 | 631 | 730 |
| On the table | 12 | 627 | 544 | 508 | 460 | 620 | 695 |
| On the table | 15 | 607 | 550 | 500 | 480 | 590 | 640 |
| Under the table | 3 | 686 | 390 | 239 | 184 | 425 | 678 |
| Under the table | 6 | 542 | 378 | 281 | 250 | 416 | 651 |
| Under the table | 9 | 500 | 369 | 262 | 274 | 434 | 570 |
| Under the table | 12 | 479 | 328 | 231 | 258 | 322 | 523 |
| Under the table | 15 | 468 | 317 | 236 | 218 | 319 | 435 |

*Calibration was done to 15 inches because anything more proved to be the same as ambient environmental light. At 3 inches, the light source was too close and not on the same level as the photosensors and created a much lower reading than expected.*