Lab 03

Wall Following: Feedback PD Control

Reading: *Ch. 3 of the text, Lecture 3-1*

Read this entire lab procedure before coming to lab.

****

Purpose: The purpose of this lab is to implement a wall following behavior on the CEENBot by using feedback control. The IR and/or sonar sensors will be used to detect the wall and the robot should use proportional-derivative (PD) control to maintain a distance between 4 and 6 inches from the wall. The wall following behavior should then be integrated as the top layer onto the subsumption architecture implemented in the obstacle avoidance lab.

Objectives: At the conclusion of this lab, the student should be able to:

- Acquire and use data from all of the robot's range sensors
- Implement a wall following behavior with PD control on the Arduino Robot
- Use modular programming to implement subsumption architecture on the Arduino Robot

Equipment: Base Robot

Combination of IR or Sonar Sensors

Obstacles, Walls

3 LEDs

Theory:

In this lab you will use feedback control to design a wall following behavior. The initial task will be to design a bang-bang controller that is either on or off based upon whether the robot is in an acceptable band from the wall. If the robot is within the band it drives forward, if it is outside the band then it stops and makes one adjustment turn to get back within the band. This lab builds on the prior lab except the robot now tries to maintain a given distance from the obstacle versus running away from it. After the bang-bang control is working at an acceptable level, you will design a proportional controller. Proportional control will adjust motor turn angles or speed based upon the amount of error the robot is away from the wall once it is outside of the dead band. Finally, in order to adjust for the rate and moment or transient control, you will add the derivative to create a PD controller.

It is recommended that you start with a proportional controller using error based upon distance from the wall $[K_p * (\text{error input})]$. The gain on the controller should control heading and possibly motor speed. The first step would be to tune the proportional controller by selecting the gain with the best performance. It is recommended that the proportional controller should be adjusted until the robot follows the wall with regular oscillations or wavering but crudely maintains a distance to the wall.

Once the proportional control works at an acceptable level try to incorporate a derivative controller, *[K_d ∗ d (error input)/dt]*. Since the derivative of the error is the rate of change, it will be necessary to store the last value of the error and find the difference with respect to the current value and multiply by some constant. Finally, tune the derivative controller to yield the best robot performance. Devise a method to test that the wall following behavior works correctly and report the results in the lab memo. Note that reasonable gains may be between 1 and 10 and the derivative gain may be about 1/10th of the proportional gain. See the "*TuningofaPIDControllerusing Ziegler-NicholsMethod*" document in the Moodle Lab folder. Figure 1 presents a sample proportional - derivative controller for wall following.

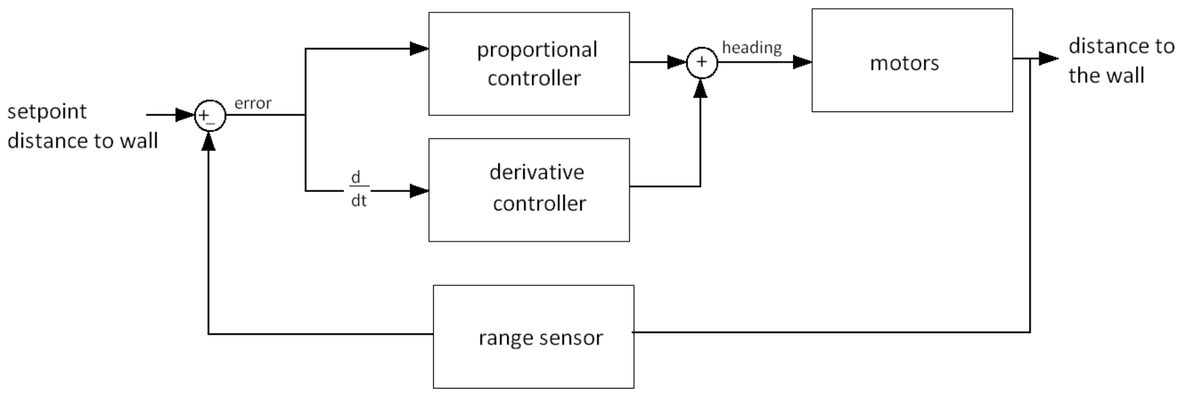

Figure 1: Wall Following PD Controller

The final code for the robot will be an integration of obstacle avoidance, random wander, wall following and hallway following. This will be implemented by using subsumption architecture and state machines as discussed in the prior lab. The key to being able to implement this more complicated behavior is to use modularity, state machines and organization and planning at the beginning. See sample state machines and subsumption architectures in Figures 2-4 for guidance on how to design the robot architecture with the required layers.

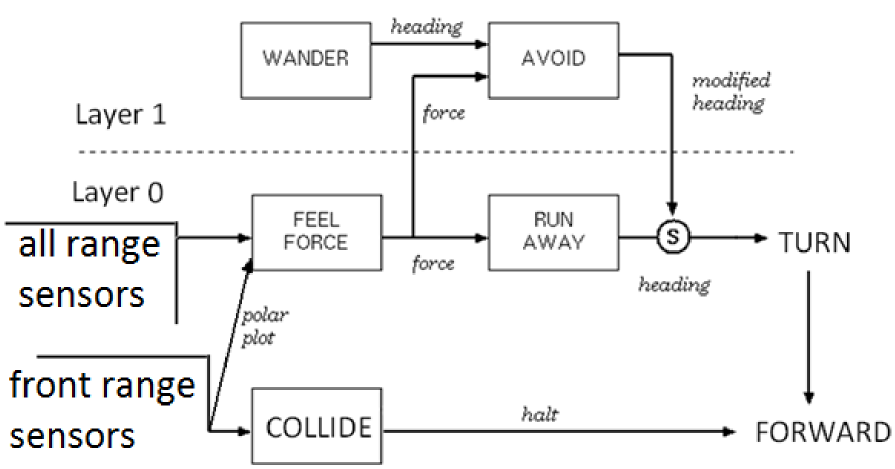

Figure 2: Subsumption Architecture (add follow wall and hallway top layers)

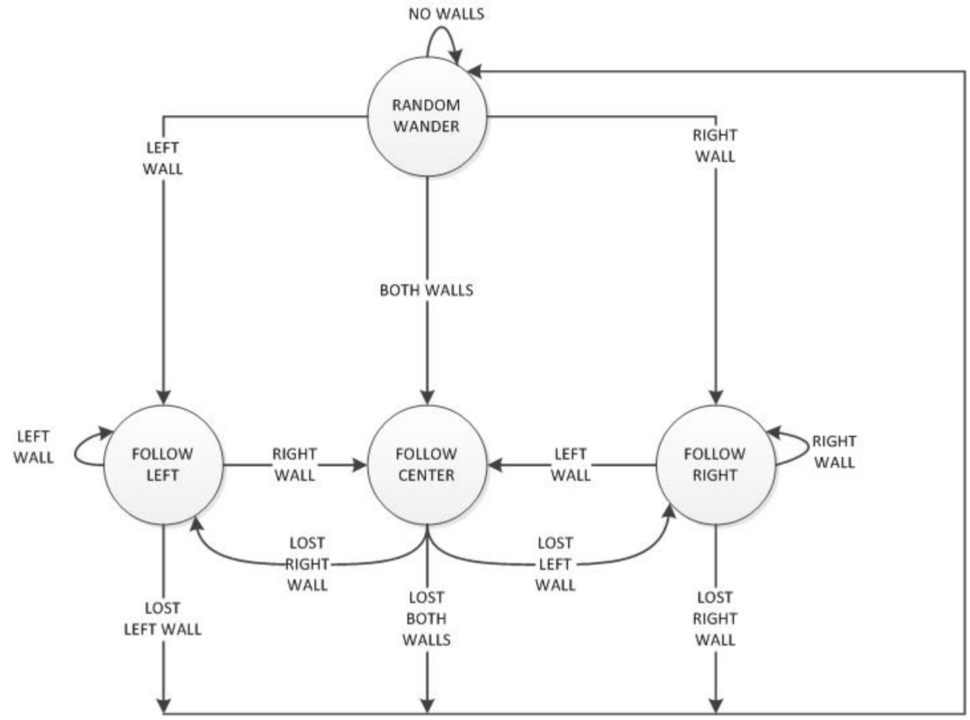

Figure 3: Wall Following State Diagram

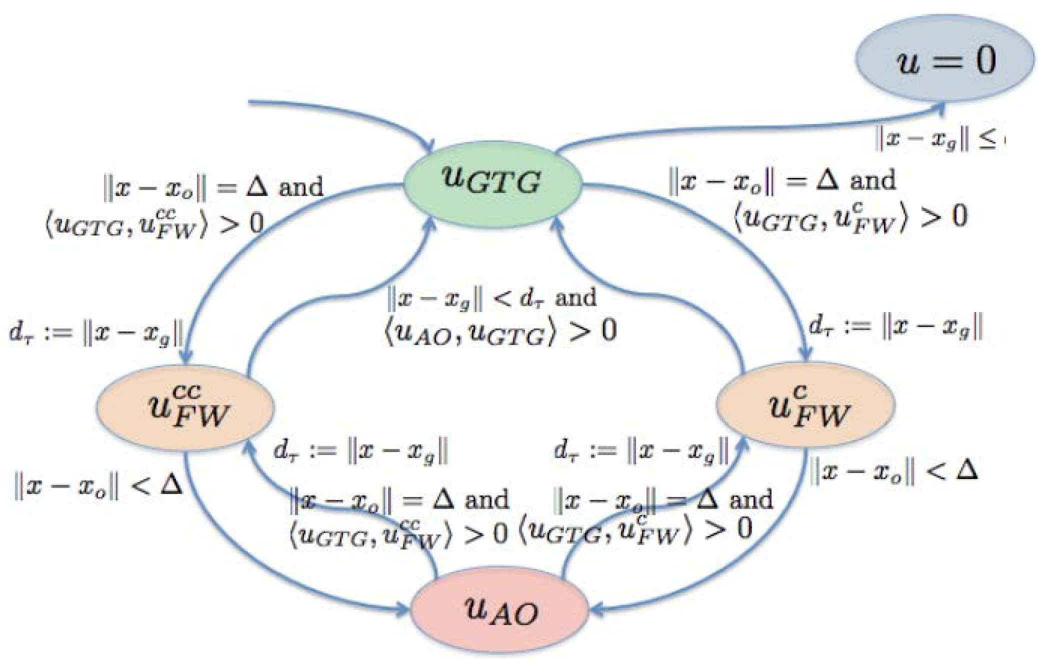

Figure 4: Go-To-Goal, Avoid-Obstacle and Follow-Wall Finite State Machine

Pre-Lab:

- Review the *Tuning of a PID Controller Using Ziegler-Nichols Method*" document to prepare for the lab.
- Create a subsumption architecture with random wander, follow wall, and follow center behaviors.
- Create the state diagram with the following states: follow left, follow right, follow center, random wander, too far, too close, inside corner, outside corner. Inputs to the states would be based upon the range sensors.

****

LAB PROCEDURE

****

Part 1 – Wall Following (Layer 2)

Design a wall following behavior for the Arduino Robot using PD control.

1. Download the sample code in the Moodle lab folder, "*RobotPD.ino*", for a guide on using the sensor interrupt and plotting the proportional and derivative error. Rename the code, "*YourRobotName-WallFollowing.ino*". Note that some students have found that it works better to drive the robot backward and to update sonar in the loop and IR with the timer.
2. Modify the code to create a bang-bang controller. The robot should follow a wall on the left and right as long as it is within 4 to 6 inches by driving forward. If the robot moves outside of the band, the controller should turn on and make the appropriate adjustment to the motors to get the robot back within the band. The controller should be either on or off. You will eventually need a state machine so design your code to keep track of the robot's states such as following on left, following on right, too far or too close from wall. The inputs to the states would be the sensor data. Once the bang-bang control is working at an acceptable level go on to the next part. Acceptable level means the robot can follow the wall for at least 2 to 3 feet by using the bang-bang control.
3. Next, modify the code to create a proportional controller based upon the robot's distance from the wall. There should still be a dead band where the robot simply moves forward if it is within 4 to 6 inches from the wall. If it is outside of the band, the motors should be adjusted based upon the direction and magnitude of the error. Once the P control is working at an acceptable level go on to the next part. An acceptable level means that the robot can follow the wall for at least 2 to 3 feet but there will be regular oscillations.
4. Next, modify the code to create the proportional-derivative control. The derivative control should put the brakes on the robot's momentum so this value should be used to subtract off the change in the motor speed or direction based upon the rate of change in the error. Once the PD control is working at an acceptable level go on to the next part. An acceptable level means that the robot can follow the wall for at least 2 to 3 feet and the oscillations are greatly reduced. Turn on a combination of LEDs to indicate left or right wall following.

5. Next, modify the code to use the front sensor to detect a front wall or corner and make the necessary adjustments or turn to navigate the corner. You should make the appropriate changes to the state machine based upon whether the robot is on the wall or in a corner. Make sure your code uses a timer or states in order to extract the robot from stuck situations.
6. Now modify the code to detect an outside corner or doorway where the robot follows the wall and makes the necessary turn to go through the doorway and hop back on the wall. The state machine should be updated so that the robot keeps track of whether it was initially following left or right when the wall was lost in order to know which way to turn. Make sure your code uses a timer or states in order to extract the robot from stuck situations.
7. Now you will create the first part of your state machine which includes the random wander behavior created in the prior lab. Start the robot at least 10 inches from the wall or obstacles and the robot should randomly wander until it detects a wall on the left or right and then switches into wall following. Turn on a combination of LEDs to indicate random wander, front, and/or outside corner.

Part 2 – Follow Center (Layer 3)

Improve the wall following behavior created in part I such that if the robot detects a wall on both sides (i.e. hallway), it will move to the center and stay in the middle until one of the walls is lost. At that point, the robot should return to the basic wall following behavior. If both walls are lost the robot should then return to wandering the environment with obstacle avoidance. How would this layer look when integrated with the rest of the architecture?

In the modified subsumption architecture, avoid obstacle is layer 0, random wander is layer 1, follow wall is layer 2 and the follow center is layer 3. The robot should wander until an obstacle is detected and attempt to navigate around it by maintaining a distance of 4 to 6 inches. If the robot encounters a wall or obstacles on both sides, it should move to the center of the two objects and move forward. You should attempt to address issues such as doors, getting unstuck from corners and turning corners (see Figure 5). Note that although the robot circumvents obstacle 1 in the figure, your architecture may cause the robot to get stuck in a loop circling the box. If this happens, what could you do to break the robot out of this endless loop? Devise a method to test and confirm that your program works correctly and present the results in the laboratory memo. Turn on a combination of LEDs to indicate follow center or follow hallway.

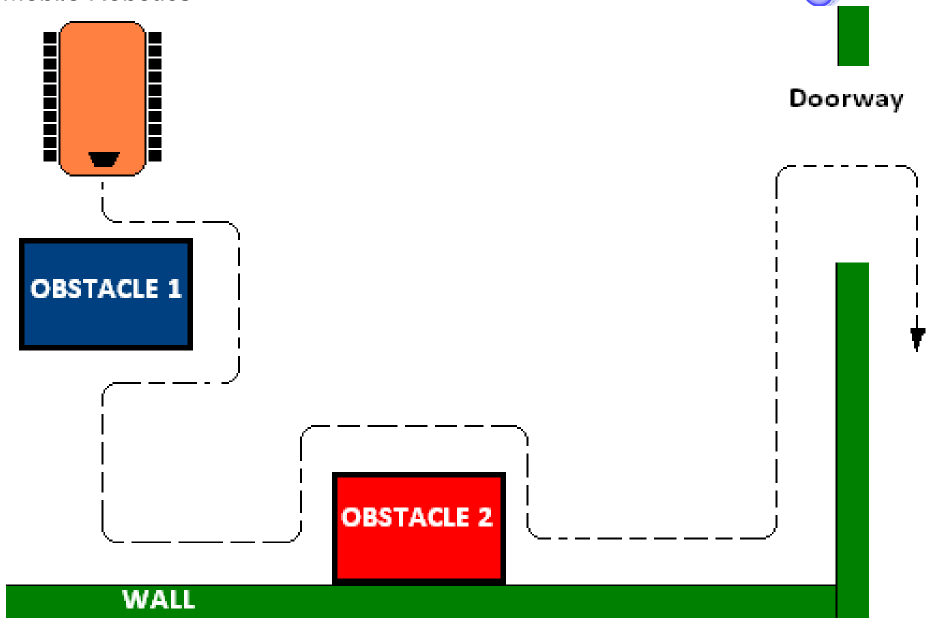

Figure 5: Wall Following Example

Demonstration:

Similar to prior labs, the demonstration will involve showing that each behavior works separately and then that the integrated behaviors with the architecture works properly.

- To demonstrate the Follow-Wall behavior, the robot should be placed next to a wall and show that it can follow the wall at a distance between 4 and 6 inches for at least 2 to 3 feet.
- The robot will also be tested on its ability to navigate an obstacle next to the wall and how it handles doorways in the wall (inner and outer corners).
- The next demonstration will be to place the robot in a hallway and show that it moves to the center and continues to follow the hallway until one or both walls are lost.
- Next, the architecture will be evaluated by the robot starting in a wander behavior until an obstacle or wall is detected, the robot should then attempt to follow the object or wall at a distance of 4 to 6 inches unless a wall is detected on the opposite side. At that point the robot should attempt to follow the center of the hallway until one or both walls is lost.

Bring your robot fully charged to class everyday.

Program:

The program should be properly commented and modular with each new behavior representing a new function call. The design of the subsumption architecture should be evident from the program layout.

Questions to Answer in the Memo:

1. What does diagram for the 3 layer subsumption architecture look like?
2. What did the robot do when it encountered a corner while wall following?
3. What did the robot do when it encountered doorways and/or corners?

4. When tuning the proportional controller and/or derivative controller, did the robot exhibit any oscillating, damping, overshoot or offset error? If so, how much?
5. What were the results of the different P and D controller gains? How did you decide which one to use?
6. How accurate was the robot at maintaining a distance between 4 and 6 inches?
7. Did the robot ever lose the wall?
8. Compare and contrast the performance of the *Wander* and *Avoid* behaviors compared to last week's lab.
9. What was the general plan to implement the feedback control and subsumption architecture on the robot?
10. How could you improve the control architecture and/or wall following/follow center behaviors?
11. What does the overall subsumption architecture diagram with all 4 layers look like?
12. What was the pseudocode and flow chart for the program design?
13. Did you use any suppression and inhibition with the integration of Layers 2 and 3?
14. How did you implement the finite state machine to integrate the various behaviors? Did you use any inhibition and suppression to create layers in this behavior?
15. How did you keep track of the robot's state and as it switched between behaviors?

Memo Guidelines:

Please use the following checklist to insure that your memo meets the basic guidelines.

- ✓ Format
  - o Begins with Date, To , From, Subject
  - o Font no larger than 12 point font
  - o Spacing no larger than double space
  - o Written as a combination of sentences or paragraphs and only bulleted list, if necessary
  - o No longer than three pages of text
- ✓ Writing
  - o Memo is organized in a logical order
  - o Writing is direct, concise and to the point
  - o Written in first person from lab partners
  - o Correct grammar, no spelling errors
- ✓ Content
  - o Starts with a statement of purpose

- Discusses the strategy or pseudocode for implementing the robot remote control (includes pseudocode, flow chart, state diagram, or control architecture in the appendix)
- Discusses the tests and methods performed
- States the results and or data tables including error analysis, if required
- Shows any required plots or graphs, if required
- Answers all questions posed in the lab procedure
- Clear statement of conclusions

Grading Rubric:

The lab is worth a total of 30 points and is graded by the following rubric.

Points	Demonstration	Code	Memo
10	Excellent work, the robot performs exactly as required	Properly commented with a header and function comments, easy to follow with modular components	Follows all guidelines and answers all questions posed
7.5	Performs most of the functionality with minor failures	Partial comments and/or not modular with objects	Does not answer some questions and/or has spelling, grammatical, content errors
5	Performs some of the functionality but with major failures or parts missing	No comments, not modular, not easy to follow	Multiple grammatical, format, content, spelling errors, questions not answered
0	Meets none of the design specifications or not submitted	Not submitted	Not submitted

Submission Requirements:

You must submit your properly commented Sketch code & memo to the Moodle DropBox by midnight on Sunday. Check the course calendar for the lab demonstration due date.