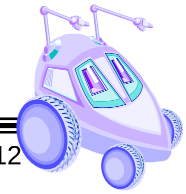


Lab 3

Wall Following: PD Control

Reading: *Introduction to AI Robotics (Sec. 4.3), Lecture 3-1*

(Demonstration due in class on **Thursday**)

(Code and Memo due in Angel drop box by midnight on **Sunday at midnight**)

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 contact, sonar and IR 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 Lab 2.

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 CEENBoT
- Use modular programming to implement subsumption architecture on the CEENBoT

Equipment: Base Robot

IR Sensors

sonar on servo turret

LCD display

obstacles, walls

Software: AVR Studio 4 (32-bit) available at

http://www.atmel.com/dyn/products/tools_card.asp?tool_id=2725

WinAVR GCC toolchain (http://winavr.sourceforge.net/)

CEENBoT API static library available at

http://www.digital-brain.info/downloads/capi324v221-v1.09.002R.zip

****

LAB PROCEDURE

****

Part 1 – Wall Following

Design a wall following behavior for the CeenBoTI using PD control. The robot should start at least 10 inches from the all and move toward the wall and maintain a distance of 4 to 6 inches from the wall as it follows the wall for at least 4 feet. The robot should follow the wall while negotiating obstacles, corners and doorways with minimal contact with walls and obstacles.

It is recommended that you start with a proportional controller using error based upon distance from the wall *[K_p * (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. 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. Figure 1 presents a sample proportional - derivative controller for wall following.

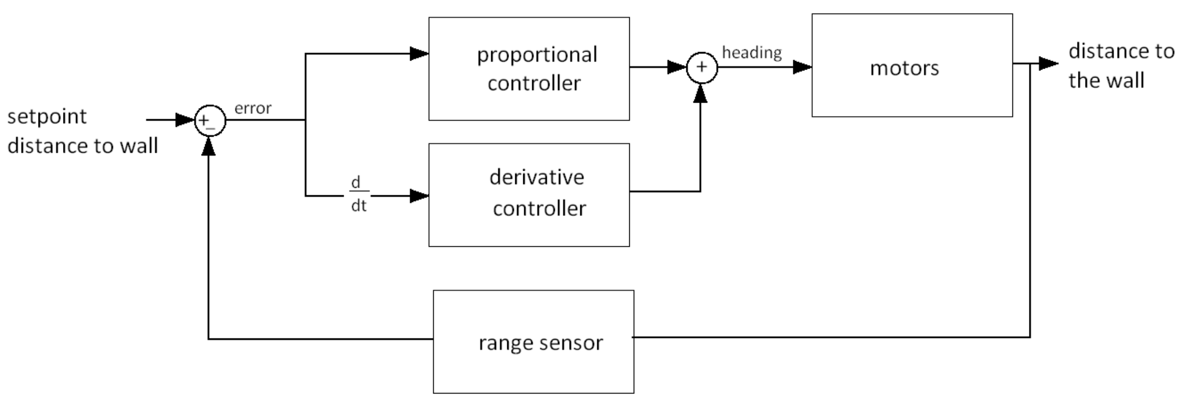


Figure 1 Wall Following PD Controller

Part 2 – Follow Center

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.

Part 3 – Layers 2 and 3 – Subsumption Architecture

Now modify the obstacle avoidance program created in Lab 2 so that there is a follow wall and stay in the middle layer. The 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 2). 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.

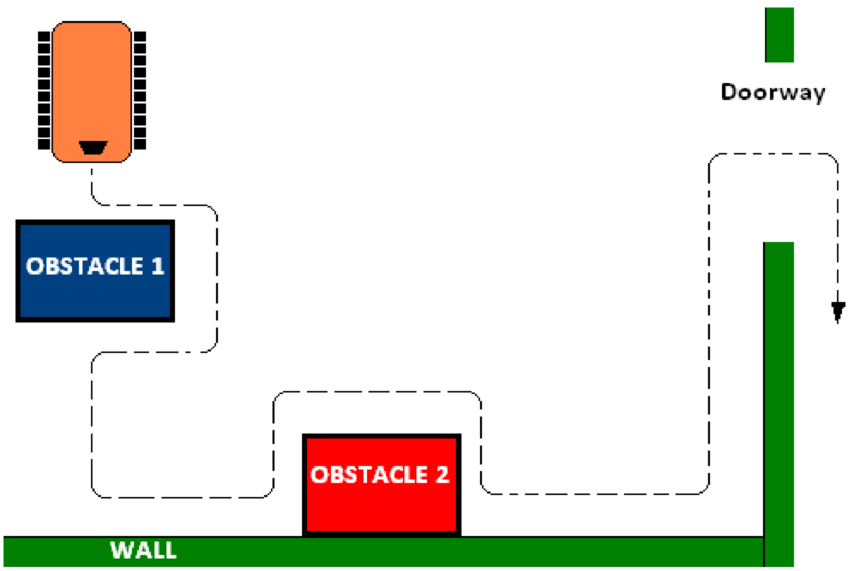


Figure 2: Wall Following Example

Demonstration:

Similar to Lab 2, the demonstration will involve showing that each behavior works separately and then that the integrated behaviors with the architecture works properly. The first test will be that the robot is able to detect a wall from 10 inches away and move toward it and follow on the robot's left or right. 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. 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. Lastly, the architecture will be evaluated by the robot starting in a wander behavior until an obstacle 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 on Thursday for the demonstration. Note that you always must re-flash the factory firmware and plug in the AC adapter in order for the robot to charge. Alternately, you can put the

robot battery in the RC car battery charger. Note that this is a fast charger and will not last as long as the outlet charge.

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. You should use the GUI, keypad, LCD and speech module as needed to illustrate robot state, input and output data.

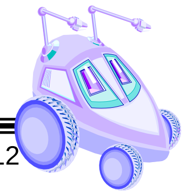
Memo:

The following list provides the basic guidelines for writing a technical memorandum.

- ✓ Format
  - Begins with Date, To , From, Subject
  - Font no larger than 12 point font
  - Spacing no larger than double space
  - Written as a paragraph not bulleted list
  - No longer than three pages of text
- ✓ Writing
  - Memo is organized in a logical order
  - Writing is direct, concise and to the point
  - Written in first person from lab partners
  - Correct grammar, no spelling errors
- ✓ Content
  - Starts with a statement of purpose
  - Discusses the strategy or pseudocode for implementing the robot paths (may include a flow chart)
  - Discusses the tests and methods performed
  - States the results including error analysis
  - Shows data tables with error analysis and required plots or graphs
  - Answers all questions posed in the lab procedure
  - Clear statement of conclusions

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?

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, 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 you properly commented code as a zipped folder of the Visual C# solution (.sln) and the lab memo in a zipped folder by **11:59 pm on Sunday** to the Angel Course Drop box. Your code should be modular with functions and classes in order to make it more readable. You should use the push buttons and LCD to indicate the robot state during program execution.