**Virtual Initio Programming: Wall Following**

**AIM:** After completing this worksheet you should be able to integrate your new knowledge of Python programming in order program a simple wall following algorithm in Python.

**You Need:** To complete this worksheet you need to have a virtual Initio simulator (see WS1), and to be able to use files to store Programs (WS5). You also need to know the commands to operate the Initio motors and sensors (WS3 & WS4). You should be able to use If statements (WS7), while loops (WS8 & WS10), variables, numbers (WS12), data types (WS14) and functions (WS15) in Python programs.

**If the simulator isn’t already running: Start the Simulator, Select the Initio Simulation and house.xml, then start IDLE (open a *new IDLE window* if you have used IDLE to start the simulator).**

**Challenge:** Create a program, using functions for turning left or right which will drive around the outside wall of the “house” in the **house.xml** world in one direction until it enters the house and then will reverse out and drive back around the house in the other direction until it enters again.

You can go ahead and attempt to write the program now – or you can follow the suggested steps below.

**Step 1:** Write a follow\_wall(side) function which moves forward if it detects an obstacle on side and turns towards side when it does not detect an obstacle to that side. You may want to get it to move forward a little and then turn when it doesn’t detect an obstacle on the desired side (this will help prevent the robot getting “stuck” turning towards and then away from a wall.

**Step 2:** Modify your function so that if it detects an obstacle in the centre (you will need to use the distance sensor for this and decide how close something need to be to count as an obstacle) it turns away from side until it doesn’t detect an obstacle in the centre *or* an obstacle on side

**Step 3:**  Write a program that will follow a wall on one side until it detects a black “floor” beneath it. Then will reverse backwards for a few seconds, turn back the way it came for a few seconds and then follow the wall on the other side.

**Step 4:** Finally add a function drive\_to\_wall() which will drive forward until the robot detects an obstacle and, at that point, will start following the wall around the house.



University of Liverpool, 2019

This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/).