

# 04\_03\_wall\_following\_teacher\_

May 2, 2025

Part of the InnovatED STEM and DroneBlocks Land, Air, and Sea Robotics Curriculum  
Licensed for educational use in schools only.

Redistribution, commercial use, or resale is strictly prohibited.

© 2025 InnovatED STEM & DroneBlocks. All rights reserved.

## 1 Wall Following with MultiRanger – Student Version

In this notebook, you'll learn how to follow a wall using the **Crazyflie drone** and its **MultiRanger sensors**.

We'll break down the code step-by-step to help you understand the logic behind autonomous wall following.

### 1.1 Setup

Import and initialize the drone simulator. ## Command Reference Table | Command | Description | Parameters (units/type) | |-----|-----|  
| | | **takeoff()** | Drone takes off and hovers | height (m/float), speed (m/s/float) | | **land()** | Lands the drone | speed (m/s/float) | | **forward()** | Move forward | distance (m/float), speed (m/s/float) | | **left()** / **right()** | Move sideways (strafe) | distance (m/float), speed (m/s/float) | | **get\_distances()** | Returns dictionary of sensor distances | None | | **get\_distances()['left']** | Distance to the wall on the left (m/float) | None | | **get\_distances()['right']** | Distance to the wall on the right (m/float) | None |

```
[ ]: import time
from crazyflie_sim import CrazyflieSimulator

drone = CrazyflieSimulator(real=False)
```

### 1.2 Step 1: Take Off

Let's take off to 1.0 meter altitude at 0.3 m/s.

```
[ ]: drone.takeoff(1.0, 0.3)
time.sleep(2)
```

### 1.3 Step 2: Wall Following Logic (Right Wall)

**Goal:** Stay about 0.5m from the right wall - If too far from the wall (> 0.6m), turn right - If too close to the wall (< 0.4m), turn left - Otherwise, move forward

### 1.3.1 Step 3: One Loop Iteration

Read distances and react based on right wall.

```
[ ]: distances = drone.get_distances()
right_dist = distances['right']

if right_dist > 0.6:
    print("Too far from wall - adjusting right")
    drone.rotate(-10, 0.5)
elif right_dist < 0.4:
    print("Too close to wall - adjusting left")
    drone.rotate(10, 0.5)
else:
    print("Maintaining distance")

drone.forward(0.1, 0.2)
time.sleep(1)
```

### 1.3.2 Step 4: Loop it!

Now let's run that logic for 10 steps.

```
[ ]: for _ in range(10):
    distances = drone.get_distances()
    right_dist = distances['right']

    if right_dist > 0.6:
        print("Too far from wall - adjusting right")
        drone.rotate(-10, 0.5)
    elif right_dist < 0.4:
        print("Too close to wall - adjusting left")
        drone.rotate(10, 0.5)
    else:
        print("Maintaining distance")

    drone.forward(0.1, 0.2)
    time.sleep(1)
```

## 1.4 Exercise 1: Follow the Left Wall Instead

Modify the loop to follow the **left wall** instead of the right. - If **left** > 0.6: rotate **left** - If **left** < 0.4: rotate **right** - Then move forward.

```
[ ]: # for _ in range(10):
#     distances = drone.get_distances()
#     left_dist = distances['left']
#     if left_dist > 0.6:
#         drone.rotate( , ) # turn left
```

```
# elif left_dist < 0.4:
#     drone.rotate( , ) # turn right
#     drone.forward( , )
#     time.sleep( )
```

## 1.5 Exercise 2: Stop if a Wall is Ahead

If the drone sees a wall **closer than 0.5 meters in front**, it should stop and print a warning.

```
[ ]: # for _ in range(20):
#     d = drone.get_distances()
#     if d['front'] < 0.5:
#         print("Wall ahead! Stopping.")
#         break
#     # Adjust to right wall as needed...
#     drone.forward( , )
#     time.sleep( )
```

## 1.6 Land and Close

Always finish your flight cleanly.

```
[ ]: drone.land(0.3)
      drone.close()
```