02_01_forloops_simulator_

May 2, 2025

1 Repeating Flight Commands with `for` Loops

In this notebook, you'll learn how to use `for` loops to make your drone repeat movements or patterns more efficiently.

We'll use the `CrazyflieSimulator` to visualize the drone's path in simulation mode.

[ ]:
```
#  Imports and Setup
import time
from crazyflie_sim import CrazyflieSimulator

# Create a simulated drone instance
drone = CrazyflieSimulator(real=False)
```

1.1 Takeoff

`drone.takeoff(height, velocity)`

- `height`: target hover height in meters (ex: 1.0)
- `velocity`: speed to rise (max ~0.5 m/s)

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

1.2 Why Use Loops?

Instead of writing repetitive commands manually:

```
drone.forward(0.2, 0.2)
drone.forward(0.2, 0.2)
drone.forward(0.2, 0.2)
```

You can use a `for` loop:

```
for _ in range(3):
    drone.forward(0.2, 0.2)
    time.sleep(1)
```

1.3 Example: Fly a straight path using a `for` loop

[ ]:
```
for _ in range(5):
    drone.forward(0.2, 0.2)
    time.sleep(1)
```

1.4 Example: Zig-zag pattern using loops

[ ]:
```
for i in range(3):
    drone.left(0.2, 0.2)
    time.sleep(1)
    drone.right(0.2, 0.2)
    time.sleep(1)
```

1.5 Exercise 1: Fly forward 4 times with pauses

[ ]:
```
# for _ in range(4):
#     drone.forward( , )
#     time.sleep( )
```

1.6 Exercise 2: Try a square path using a loop

Hint: Use 4 sides and turns!

[ ]:
```
# for _ in range(4):
#     drone.forward( , )
#     drone.rotate( , )
#     time.sleep( )
```

1.7 Land and Close

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