ROBOLINK BASECAMP

1.2: Setting the Trim

Throughout your programming journey, your drone may start to drift and sway while flying. This is perfectly normal. In this lesson, you will learn how to trim your drone. Refer back to this lesson whenever you need to trim your drone!

This lesson works with:


CoDrone EDU


CoDrone EDU (JROTC ed.)


Python

beginner | CDE | CoDrone EDU | flight events | fundamentals | landing | Python | takeoff | trimming

Grade level:
6 - 12+

Approx. time required:
30 - 45 mins

Step 1

Checklist

By the end of this lesson, you should be able to:

- Describe how trim affects the flight of the drone
- Use the set_trim(int, int) function to improve the drone's hovering and flight

By the end of this lesson you will be able to set trim values for your drone whenever it starts to drift. In Python, you can use these functions during any lesson or in any program. When flying with the controller, use will be able to use the buttons on the remote controller to adjust the trim (see Step 8, Lesson 0.5 Flying with the Controller).

Step 2

Why does my drone drift?

Drones don't always fly in a perfectly straight line, especially if there's an air current, the battery is running low, or the drone has crashed one too many times. Sometimes, the sensors that keep the drone stable just need to be recalibrated!

You might be asking yourself, how do I know my drone is drifting? It will be most obvious when you let your drone hover in the air for a few moments after takeoff. Using Python code, you can run a quick test! In the next lessons you will learn how to fly CoDrone EDU more in depth, but for now you can copy and paste this code into a new Python file.

```
from codrone_edu.drone import *

drone = Drone()
drone.pair()
drone.takeoff()
drone.hover(3)
drone.land()
drone.close()
```

Connect your controller, power on the drone, and place your drone in a safe place on the floor. Run the code and observe the flight.

Step 3

Using the set_trim(int,int) function

If you notice your drone drifting, you can use the `set_trim()` function at the beginning of your program (underneath pairing and before takeoff) to fix its roll, its sideway movement, or pitch, its forward and backward movement. For example, if your CoDrone EDU is drifting to the right, you can set the roll, or its sideways movement, to a negative number. This is what it would look like:

```
set_trim(-5, 0)
```

This trim value will remain saved, even after powering off, until you've changed the trim either in the program or reset with the controller. NOTE: If you're setting the trim right before a takeoff, make sure to add a `time.sleep(1)` before the `takeoff()` . Otherwise, the take off might be skipped.

This is what your program would look like with the trim function:

```
from codrone_edu.drone import *
import time

drone = Drone()
drone.pair()

drone.set_trim(-5, 0). # example: drone is drifting right, so
trim to roll left a little bit

time.sleep(1)       # Add a time.sleep(1) before takeoff if
you're planning to set the trim before takeoff
drone.takeoff()
drone.hover(3)
drone.land()
drone.close()
```

Trim refers to modifying the controller inputs to counter any aerodynamic flaws in the drone itself. Since it is nearly impossible to have a drone completely balanced perfectly in its center, they will often have to be trimmed in order to hover properly. A correctly trimmed drone should hover in one place without input from the user.

In essence, what is happening is we are compensating for the imbalance in the

drone by modifying the power values of the motors. For example, if the drone is floating to the right, you can trim down the left motor values so they don't "push" the drone to the right. Ideally, all values would be exactly the same for motor power settings, but this is rarely the case. When drones crash, it can cause slight imperfections in their aerodynamics that are not visible to the human eye but can cause drift. This makes the drone difficult to control, which is why it is important to trim the drone before flying.

Step 4

Finding your current trim

Now that we know how to set our trim, how do we get our current trim value? It is important when coding the CoDrone EDU to use the `drone.get_trim()` function to see your trim levels. If you want to see your trim values printed onto the screen use `print(drone.get_trim())`

trim.py

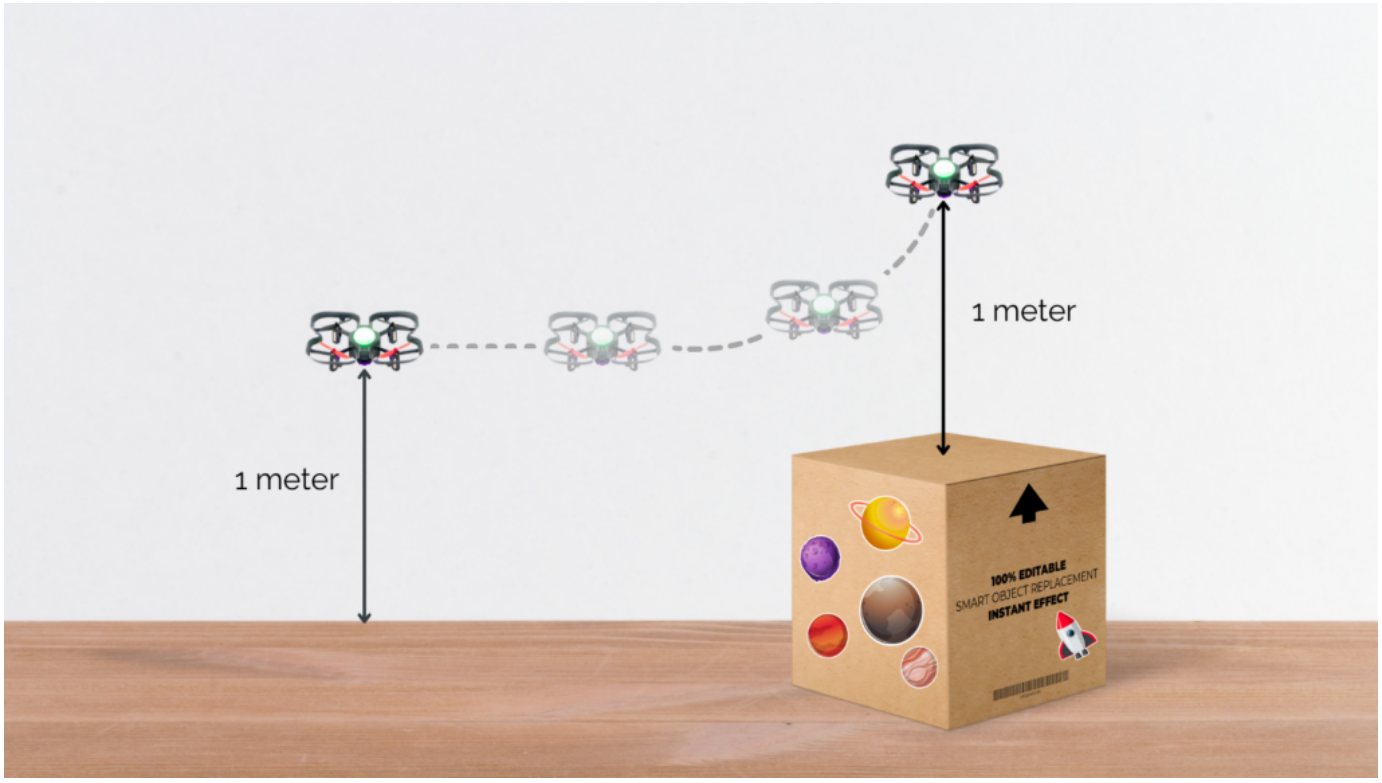
```
1  from codrone_edu.drone import *
2
3  drone = Drone()
4  drone.pair()
5  print(drone.get_trim())
6  drone.close()
7
```

Step 5

Drifting up or down

You have probably already noticed that there is no option to trim for throttle. Sometimes while flying, you may notice CoDrone EDU drift up or down even if you are not controlling the throttle, especially when programing.

If CoDrone EDU is drifting up or down, it may be because you are flying in an environment that isn't completely flat (such as a classroom with tables, chairs, or other furniture). This is because CoDrone EDU's internal sensors are trying to maintain a constant height over the surface directly below. For example, if you take off on the ground and fly over a table, CoDrone EDU may try to compensate for the change in height by throttling up.



The opposite effect can happen if you take off from a table then fly over the edge, which is why we always recommend taking off from the floor. CoDrone EDU may throttle down to adjust.

If you notice this kind of "drift", it's normal. As a challenge, you can write code to account for this height adjustment or experiment with the height sensor.

Step 6

Next steps

In this lesson you learned how to calibrate flight using the trim functions. This will come in handy when you start moving forward, back, left, and right. In the next lesson, you will learn how to program flight movements.


Lesson Complete!