**Lab 6: PID Control**

Prof. Yifeng Zhu

Goals:

1. Understand the PID (Proportional, Integral, and Derivative) control module
2. Understand the impact of PID parameters

Summary of the tasks:

1. Read PID controller summary
2. Study the two-stage controller

* Stage 1: Control by using accelerometer sensor readings
* Stage 2: Control by using gyroscope sensor readings

## Background: PID Controller

The focus of this lab is on the STEVAL’s PID Controller module. The PID controller model can be used in a variety of ways, and one objective of this lab is to determine which of those ways is the most appropriate.

The PID Controller is one of the most widely used feedback control implementations. The behavior of the PID Controller can be easily modified by adjusting (or excluding) any of its three parameters: Kp, Ki, and Kd. The below figure shows the PID algorithm graphically using block diagram.

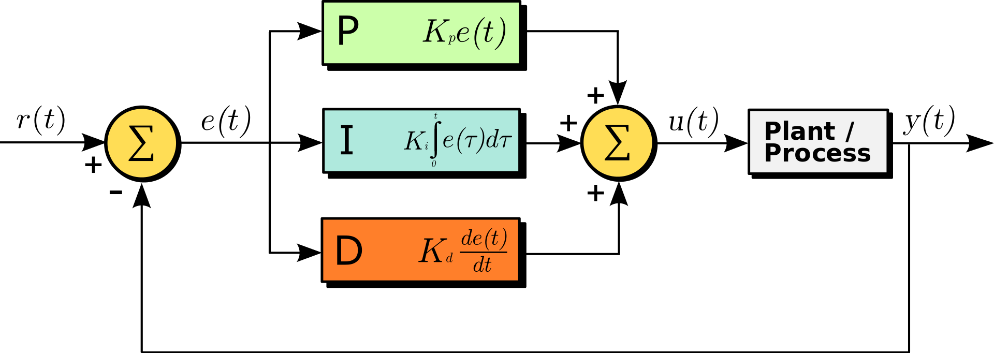


Figure 1. PID Controller Algorithm (Source: Wikipedia)

Mathematically, the control signal is defined as

where , , and are the coefficients for the proportional, integral, and derivative terms, and is defined as

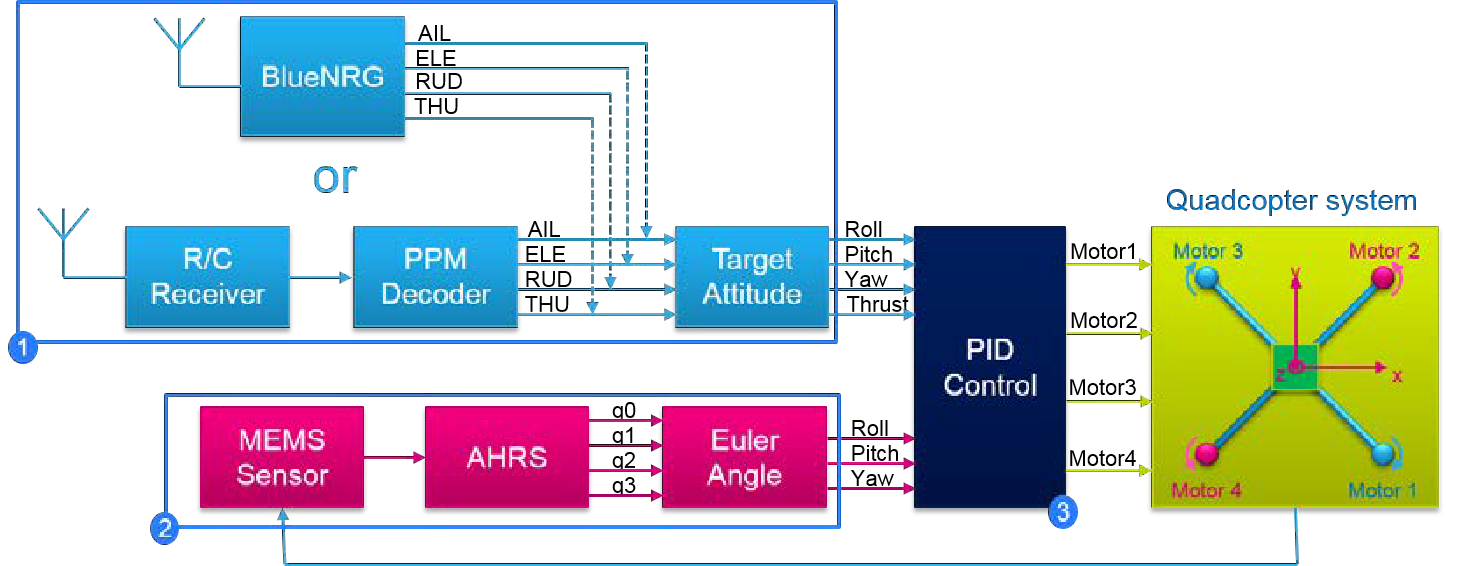
),

where is the ‘observed value’ and is the ‘desired value’.

For more information, please read this introduction: <https://en.wikipedia.org/wiki/PID_controller>

For the STEVAL, the ‘desired values’ are the roll, pitch, yaw, and thrust received from the user via the ST Drone smartphone application or the R/C receiver if available.

The ‘observed values’ are the current estimates of roll, pitch and yaw produced by the AHRS, as well as the current thrust, a value easily tracked by the drone.



As shown in the figure above, the project includes three modules: (1) remote control receiver via Bluetooth or R/C (not available by default), (2) real-position estimator, and (3) the PID controller.

The real-position estimator takes the measurements from MEMS sensors, including accelerometers (ax, ay, az) and gyroscope (gx, gy, gz), and uses the Attitude and Heading Reference System (AHRS) algorithm to calculate the quaternion elements (q0, q1, q2 and q3), and finally translates the quaternion elements to Euler angles (roll, pitch and yaw).

The PID controller takes two sets of inputs: the target attitude and the estimated Euler angles. It calculates the duty cycle of PWM outputs to control the speed of four motors.

## Quadcopter basics: flight control dynamics

|  |  |
| --- | --- |
|  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Motor 1** | **Motor 2** | **Motor 3** | **Motor 4** | **Note** |
| Pitch (x) | Slow (**Fast**) | **Fast** (Slow) | **Fast** (Slow) | Slow (**Fast**) | Move Backward (Forward) |
| Roll (y) | Slow (**Fast**) | Slow (**Fast**) | **Fast** (Slow) | **Fast** (Slow) | Move Right (Left) |
| Yaw (z) | **Fast** (Slow) | Slow (**Fast**) | **Fast** (Slow) | Slow (**Fast**) | Heading Left (Right) |
| Up/Down | **Fast** (Slow) | **Fast** (Slow) | **Fast** (Slow) | **Fast** (Slow) | Move Up (Down) |

The following diagrams will illustrate the drone movements that can be induced by the user, as well as the rotor movements corresponding to those movements.

Manipulation of the drone’s thrust, controlled via the vertical axis on the left joystick, produces an equal change in speed for all four rotors, allowing for the drone to change its vertical offset.

|  |  |
| --- | --- |
|  |  |

The drone’s yaw can be adjusted using the horizontal axis of the left joystick. When the yaw is adjusted, additional thrust is applied to the two rotors spinning in the desired direction.

|  |  |
| --- | --- |
|  |  |

The drone’s pitch and roll can be adjusted by adjusting the vertical and horizontal axes of the right joystick, respectively. An adjustment to pitch or roll results in a speed increase for one rotor and a speed decrease for the other rotor spinning in the same direction.

|  |  |
| --- | --- |
|  |  |

Now, we will discuss the actions that the ST Drone will take in response to these different user commands.

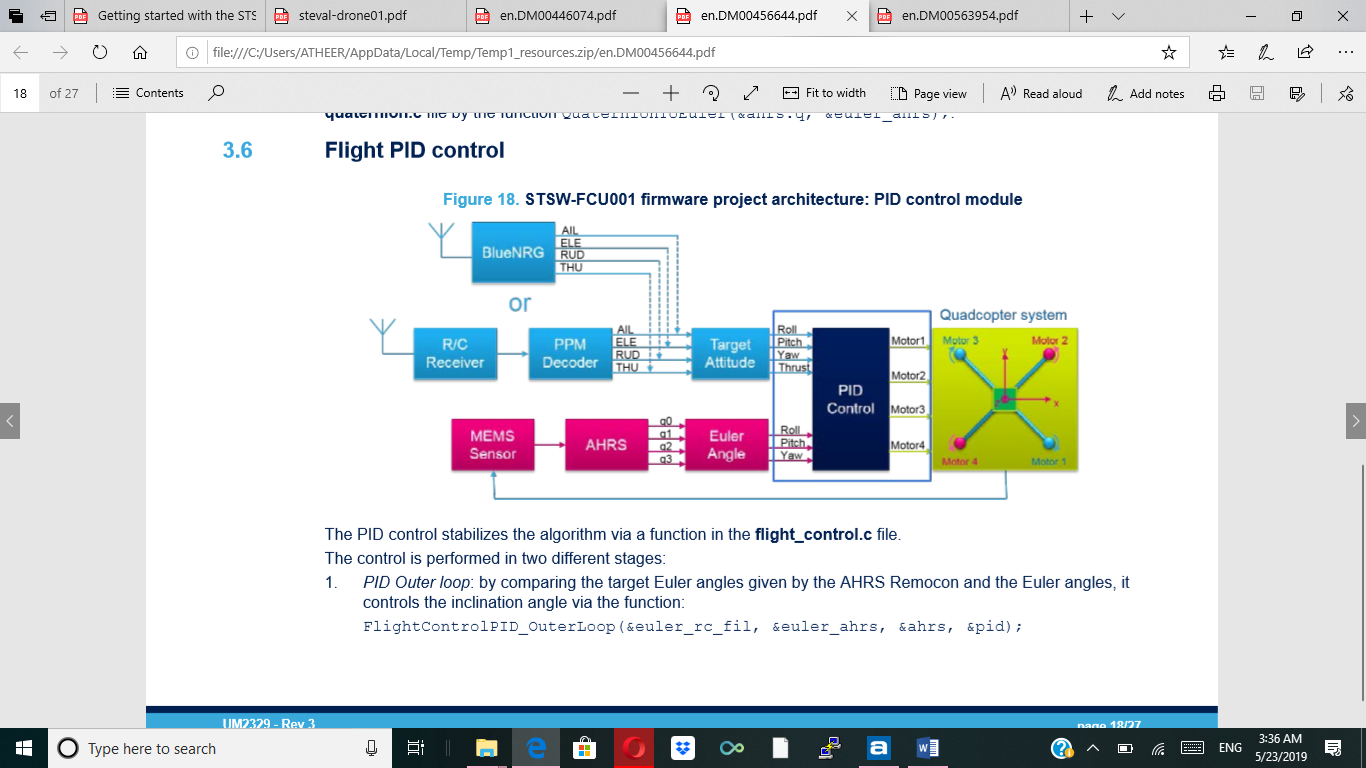


Figure 7. PID control module (UM2329-Rev 3, page 18/27)

1. As stated before, when the thrust is adjusted, the speeds of all four rotors are modified equally. This is done by the PID control nodule in two steps:

Step 1: The thrust value is calculated using the following equation:

Thrust = 0.333 × (THU\_BlueNRG/Remocon) + 633.333

Step 2: The motor PWM values are calculated through the use of the motor mixing algorithm. This produces an output PWM value for each motor between 0 and 1900. The motor mixing algorithm is shown below in Figure 8.

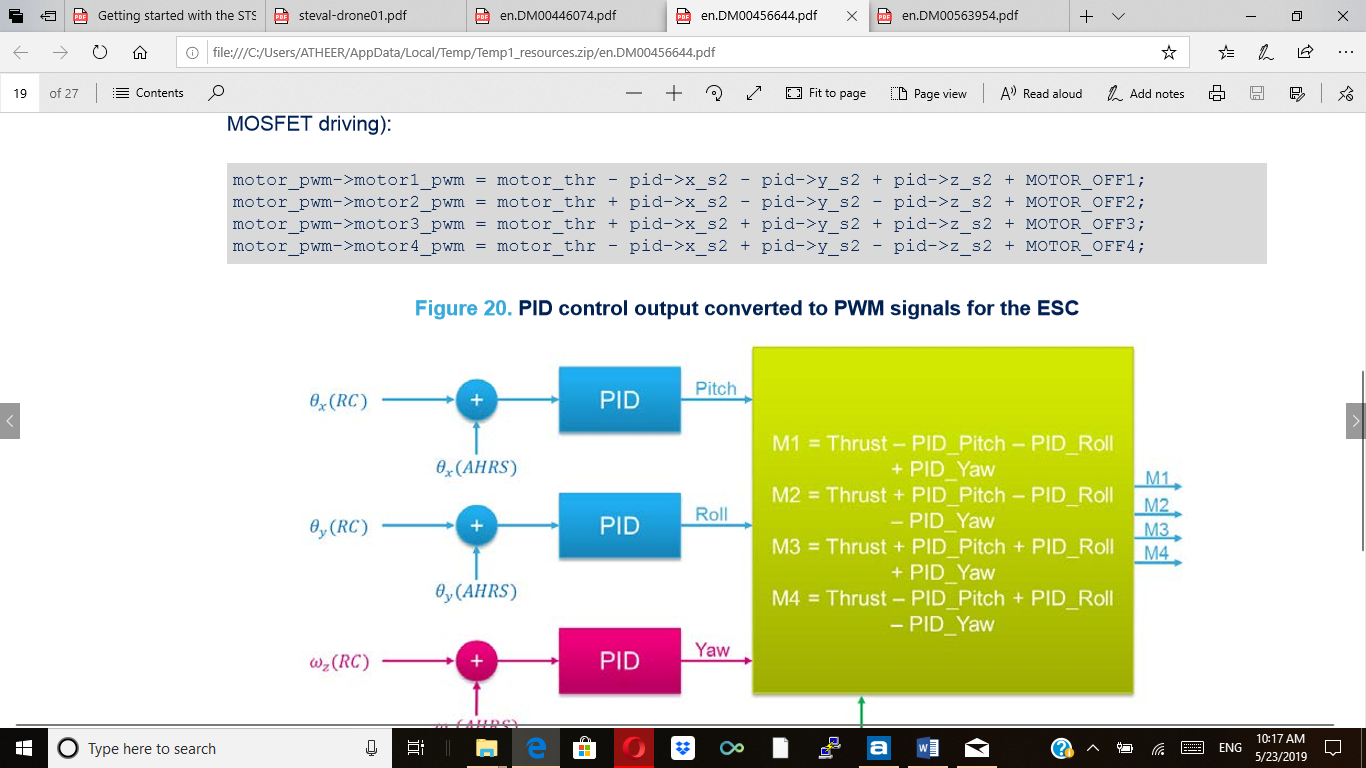
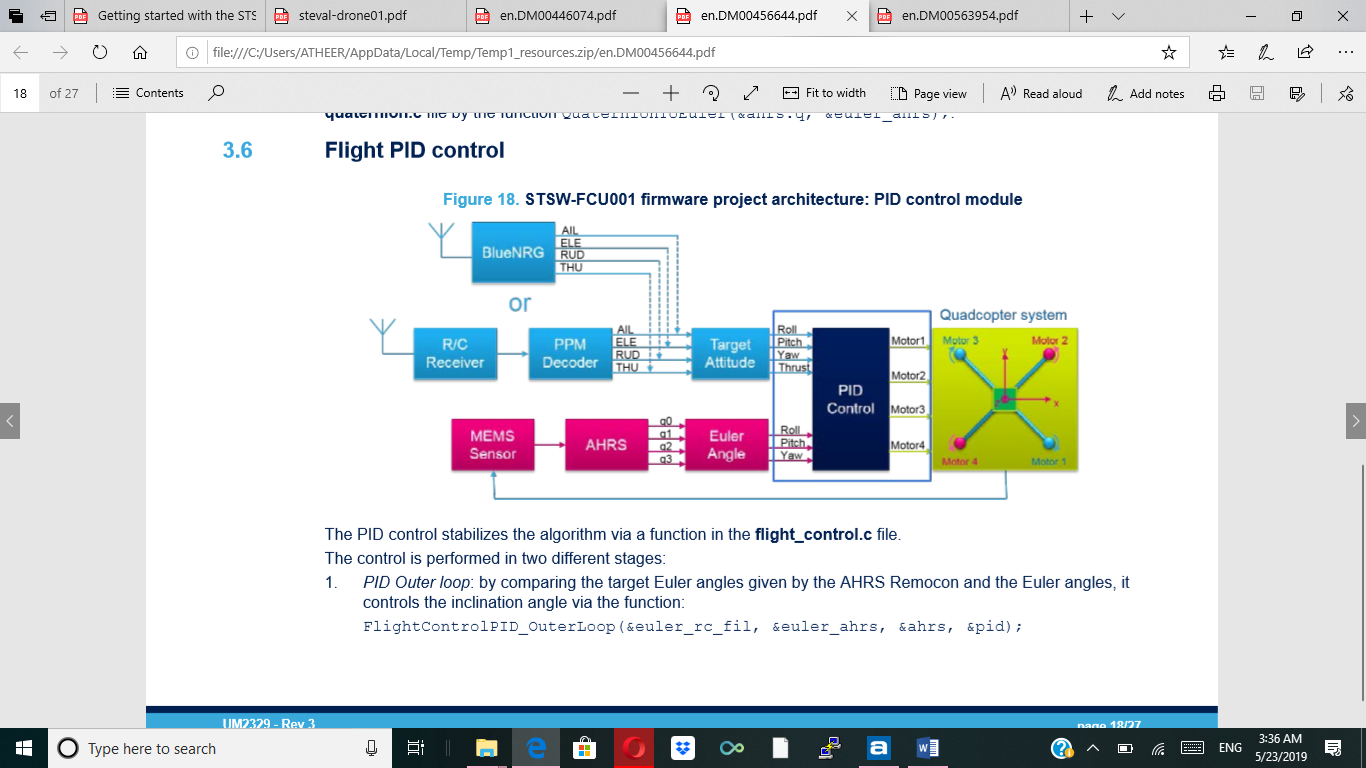


Figure 8. motor mixing algorithm



1

2

Figure 9. PID control module (UM2329-Rev 3 page 18/27)

Because we have two sets of sensor measurements, one from the accelerometer and the other from the gyroscope. We will calculate the error in two stages:

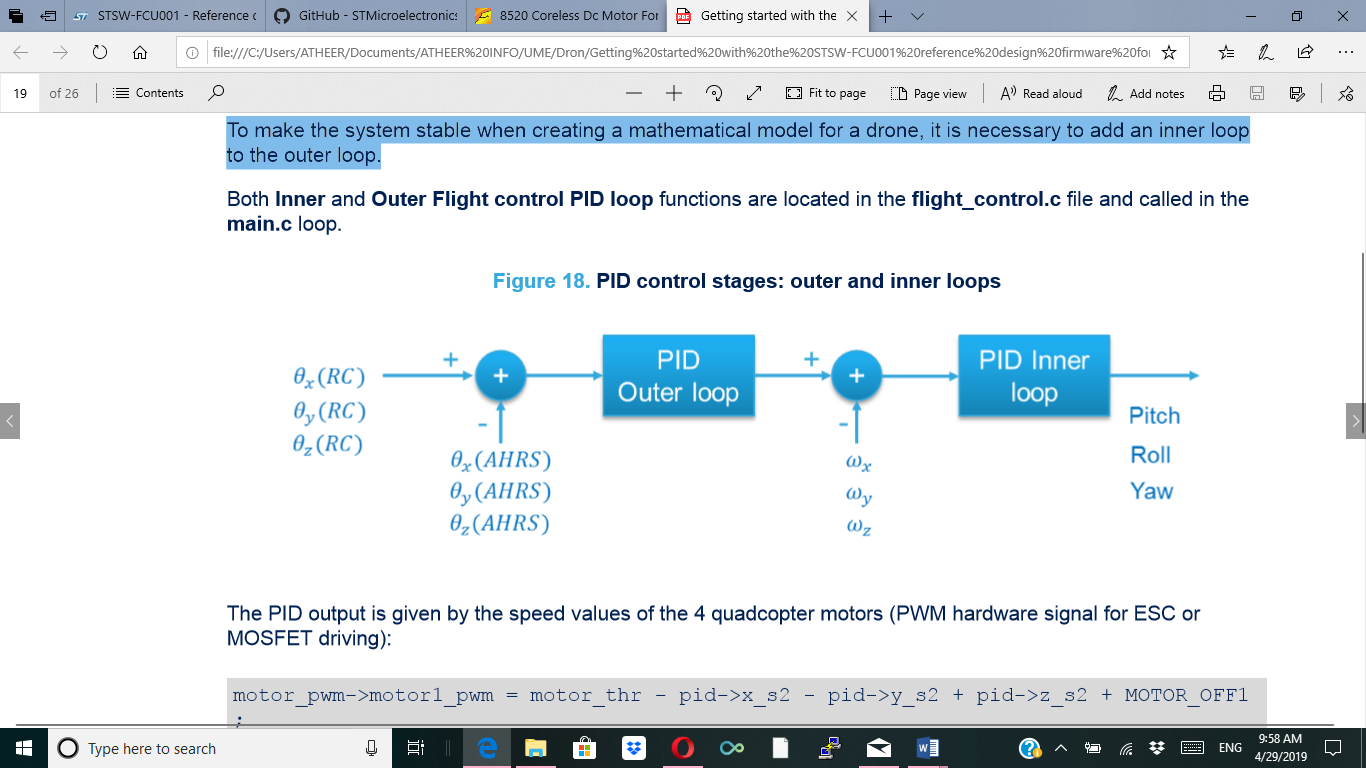
xs1

ys1

zs1

Kp2, Ki2, Kd2

Kp1



(gyroscope)

(gyroscope)

(gyroscope)

xs2

ys2

zs2

(BlueNRG/RC)

(BlueNRG/RC)

(BlueNRG/RC)

(Accelerometer)

(Accelerometer)

(Accelerometer)

error2

error1

Figure 10. PID Outer and Inner Loop

The following uses the x axis as an example to illustrate the outer and inner loop control. The calculation for the y and z axis is very similar.

First stage (PID outer loop):

Second stage (PID inner loop):

2. The proportional parameter is easier to fine tune because it has the biggest effect on stability/oscillation. For this drone I and D would deserve some fine tuning; in particular D parameter is affecting when the drone is in outdoor condition with wind, but it’s not easy to test it in the lab.

Pitch and roll usually have the same parameters unless the quadcopter have not a symmetrical structure (rare case), while usually YAW have different parameters.

Here is a short video on each of the PID parameters with a visual representation,

<https://www.youtube.com/watch?v=LtocGBngSrA&t=26s>

Here is a video showing how to fine tune the PID parameters, and how important PID is.

<https://www.youtube.com/watch?v=AN3yxIBAxTA&t=534s>

## Implementing the PID controller (flight-control.c).

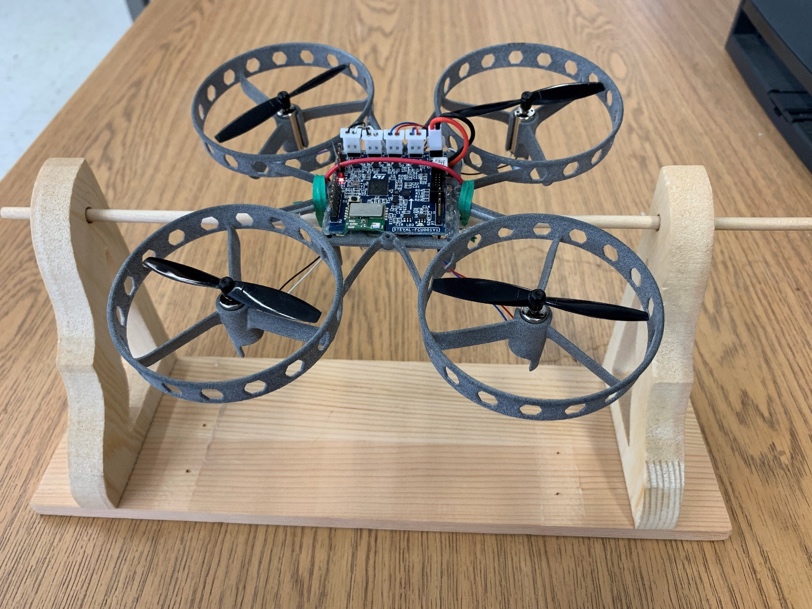
The following gives an example implementation written in pseudo code.

|  |
| --- |
| //Initial variables  Float ts, Kp1 = 3, Kp2 = 80, Ki2 = 80, Kd2 = 10, limit = 20, FILTER\_COFF = 0.025f;  Float error1, error2, deriv2, integral2 = 0;  Extern (BlueNRG/RC), (Accelerometer), (gyroscope), gTHR;  Float M1, M2, M3, M4;  // x-axis pid (Pitch)  void FlightControlPID\_OuterLoop( ) {  // P controller  error1 = (BlueNRG/RC) - (Accelerometer)  x-s1 = error1 × Kp1  }  void FlightControlPID\_innerLoop( ) {  error2 = ( x-s1) - (gyroscope)  // I controller  integral2 += error2 × ts  if (integral2 > limit)  integral2 = limit  else if (integral2 < -limit)  integral2 = -limit  // D controller  deriv2 = (error2 - pre\_error2) / ts  pre\_error2 = error2  deriv2 = pre\_deriv2 + (deriv2 - pre\_deriv2) x FILTER\_COFF  pre\_deriv2 = deriv2;  // PID controller  x\_s2 = (kp2 x error2) + (Ki2 x integral2) + (Kd2 x deriv2)  if (x\_s2 > MAX\_ADJ\_AMOUNT)  x\_s2 = MAX\_ADJ\_AMOUNT  if (x\_s2 < -MAX\_ADJ\_AMOUNT)  x\_s2 = -MAX\_ADJ\_AMOUNT  // Repeat the process for y-axis pid (Roll)  ...  // Repeat the process for z-axis pid (Yaw)  ...  Thrust = (0.3333 x gTHR) + 633.33;  // Motor PWM  M1 = Thrust – Pitch - Roll + Yaw  M2 = Thrust + Pitch - Roll - Yaw  M3 = Thrust + Pitch + Roll + Yaw  M4 = Thrust – Pitch + Roll - Yaw  } |

## Studying the impact of PID parameters

In this lab section, you will study the impact of the PID control parameters to drones in a restrained setting, as shown in the above figure.

For safety reasons, it is recommended to tie the drone to a simple framework as shown below.



1. What are the default parameters for the inner loop controller and the outer loop controller?

Outer

|  |  |  |  |
| --- | --- | --- | --- |
|  | X(Pitch) | Y(Roll) | Z(Yaw) |
| Partitional | 3 | 3 | 4 |
| Integral | 0 | 0 | 0 |

Inner

|  |  |  |  |
| --- | --- | --- | --- |
|  | X(Pitch) | Y(Roll) | Z(Yaw) |
| Partitional | 80 | 80 | 900 |
| Integral | 80 | 80 | 3 |
| Derivative | 10 | 10 | 3 |

1. What are the impacts if we increase the P parameters of the inner loop controller for the axis corresponding to the pitch and roll axis?

It increases the amount the drone wants to change its position by, and their for overshooting the target value. Also makes the drone very hard to control because it can become unstable.

1. What are the impacts if we decrease the P parameters of the outer loop controller for the axis corresponding to the pitch and roll axis?

It takes longer for the drone to get to the target value and makes it off balance and slower to change position compared to the normal values.