**PID Tuning Conceptual Information**

PID loops, or Proportional-Integral-Derivative loops, are fundamental control mechanisms used in many systems, including flight controllers like iNav. iNav is an open-source flight control software primarily used in multirotor drones and fixed-wing aircraft. PID loops in iNav are responsible for stabilizing the aircraft by adjusting control surfaces or motor speeds based on sensor feedback.

PID tuning aims to tweak the PID loop gains so that it achieves the fastest response to command changes as fast as possible without overshooting or oscillating.

The P term controls the strength of the correction that is applied to bring the craft toward the target angle or rotation rate. If the P term is too low, the craft will be difficult to control as it won't respond quickly enough to keep itself stable. If it is set too high, the craft will rapidly oscillate/shake as it continually overshoots its target.

The I term corrects small, long-term errors. If it is set too low, the craft's attitude will slowly drift. If it is set too high, the craft will oscillate (but with slower oscillations than with P being set too high).

The D term attempts to increase system stability by monitoring the rate of change in the error. If the error is rapidly converging to zero, the D term causes the strength of the correction to be backed off to avoid overshooting the target.

Tuning PID loops in iNav involves adjusting these three terms to achieve the desired balance of responsiveness, stability, and robustness to disturbances. This tuning process often involves experimentation and fine-tuning to find the optimal gains for a particular aircraft and flight conditions. Many users rely on trial and error, flight testing, and sometimes automated tuning algorithms to optimize PID parameters.

**PID Tuning steps**

**Finding P value:**

Start by setting the Integral and Derivative values to 0. Then increase the proportional value until the controller starts to become unstable and oscillate.

A controller where the oscillations become smaller is considered a “stable” controller, as eventually it will stabilize, and when the oscillations start getting larger is an ”unstable” controller.

Once the proportional value that causes the controller to oscillate is found, take this value and divide it in half. This will be the starting P value.

**Finding I value:**

Once the proportional value is found, we can start to tune the integral. Always start with small steps when adjusting a PID controller, and give time between each adjustment to see how the controller reacts.

Increase the integral gain in small increments, and with each adjustment, change the set point to see how the controller reacts.

The goal of tuning the integral value is to achieve an adequate controller response or reaction time (after the initial response from the proportional is set).

If the controller starts to oscillate or become unstable, adjust the I value in the opposite direction until the controller becomes stable again. Once the controller is stable, and responding desirably… congrats! you now have a working PI controller.

**Finding D value:**

There are many PI controllers out there, and for certain applications, this is all that is required. But if your application could benefit from the dampening effects of the derivative, you will need to find the value that works!

Now that you have a stable PI controller, start by increasing the derivative value slowly, changing the set point, and allowing time for the controller to stabilize.

The purpose of the D value is to monitor the ramp rate of the process value and prevent it from overshooting the set point.

Then, continue to change the set point and increase the derivative until the overshoot has been dampened to an acceptable level. If the controller starts to react in a negative way (unexpected changes in the output, poor control, or oscillation) lessen the D value until the controller is stable again.

Be careful when adjusting the derivative value, as a higher value is tempting – everyone wants a smooth controller don’t they? – but too high of a derivative will start to affect the output negatively and “fight” what the P and I values are trying to accomplish.

The other concern with a high D value is if there is noise or distortion of the process value feedback, the derivative may see this as a fast change in the ramp rate, and bias the output at an undesired time.