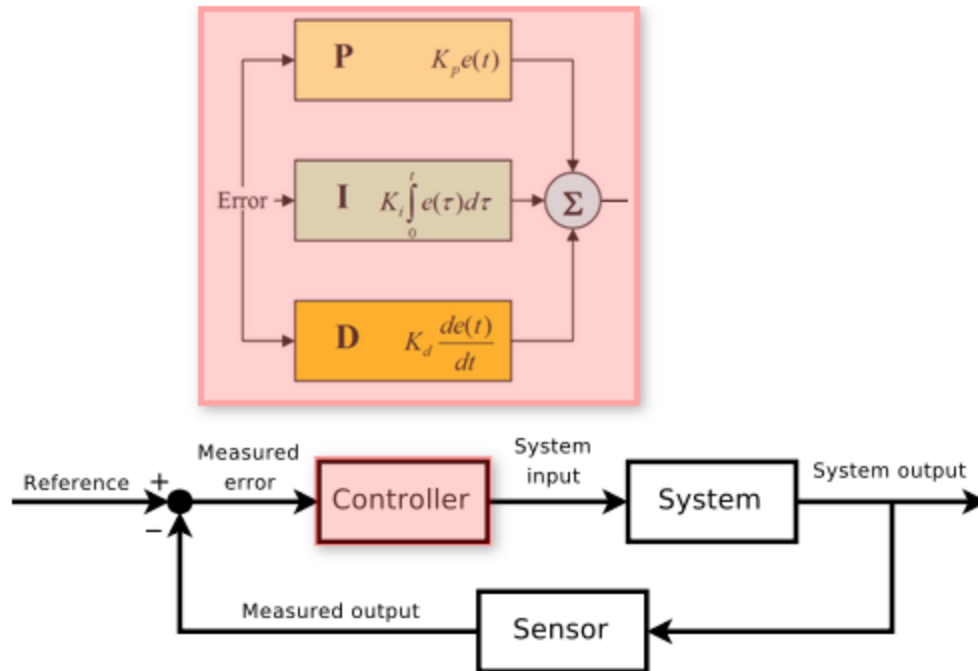# PID Tutorial

Contents

What is PID?

PID is an algorithm used in a control feedback loop to regulate a process such as the motion of a motor or the flow through a valve. Using PID control will make your robot design more stable, robust, and has the potential to improve response characteristics.

PID Algorithm

Proportional-Integral-Derivative (PID) control is the most common control algorithm used in industry and has been universally accepted in industrial control. The popularity of PID controllers can be attributed to a number of reasons. Their robust performance in a wide range of operating conditions allows them to be used in a multitude of applications, while their functional simplicity, enables engineers to operate them in a simple, straightforward manner.

As the name suggests, a PID algorithm consists of three basic coefficients: proportional, integral and derivative. These gains are varied to achieve an optimal system response.

P $\quad K_p e(t)$

Error → I $\quad K_i \int_0^t e(\tau)d\tau$ → Σ

D $\quad K_d \dfrac{de(t)}{dt}$

Reference + / Measured error → Controller → System input → System → System output

Measured output ← Sensor

The basic structure of a system with PID control implemented is illustrated above. The System output (also called the Process variable) is read by a sensor of some sort, returning the Measured output of the system, which is then compared to a desired reference value (also called the Setpoint). The comparison of the Reference and the Measured output results in an error value which is used in calculating proportional, integral, and derivative responses. These three responses are then summed to obtain the output of the controller. The output of the controller is used as an input to the system you wish to control, changing some aspect of the system. For example, if you are controlling a motor, the controller would provide more or less current. If you are controlling the flow of a fluid, the controller would cause a valve to open or close. The system output is then measured again and the whole process repeats. One completion of this process is known as a single iteration of the control loop.

A PID controller works by continuously measuring the output of your system and providing corrective input calculated from the PID control algorithm. The PID controller is both scalable and tunable. It can be simplified and scaled down by only using the proportional gain, or a combination of the proportional gain and either the integral or derivative gain (P, PI, PD, or PID controllers). The system can also be tuned by adjusting the P, I, and D gains of the controller for your specific system.

Effective Use of PID in Robotics Systems

PID algorithms are used in a variety of control systems in industry. In robotics and for FRC specifically, you will commonly see PID implemented for motion control, either for drive train motors or for other servo actuators. Before you decide to implement PID control on your robot, be sure to consider the following factors to determine if PID is warranted:

1) Do you need PID in the first place?

- If your system doesn't exhibit error outside of your acceptable performance parameters...
- If all possible inputs to the system cannot result in undesired or possibly dangerous behavior...
- If you don't need to improve the response of your system...

...then you don't really need PID control, and your time is probably better spent on other areas of your robot.

2) Is the system you are trying to control linear?

PID control is only effective for linear systems. If you have a system with non-linear response, the PID gains that are effective in tuning your system for one section of the response curve will not be valid for other parts of the response curve, resulting in erratic and uncontrolled behavior of your system. You will have to implement gain scheduling (applying different PID gains for each approximately linear section of the response curve) or PID in combination with feed-forward control to implement an effective control system.

Almost all motors will have documentation containing a Torque/Speed Curve. Use this curve to determine the linearity of your motor. A straight line with a constant slope indicates a linear motor. An exponential or polynomial curve indicates a non-linear motor. If you can't find information or specifications on your motor or other device, then you will need to perform a characterization of your system to determine its linearity.

## 3) Do you have adequate real-world feedback?

You must have some method to measure the output of your system to provide feedback in calculating the system error. For motion systems this usually means an encoder or a gear tooth sensor.

Once you have determined that you have a linear system, and that you have some means of providing the necessary feedback, the next step is learning how to implement a PID controller.

Using PID in LabVIEW

The LabVIEW FRC PID palette consists of four VIs:

1. PID
2. PID Setpoint Profile
3. PID Control Input Filter
4. PID Lead-Lag
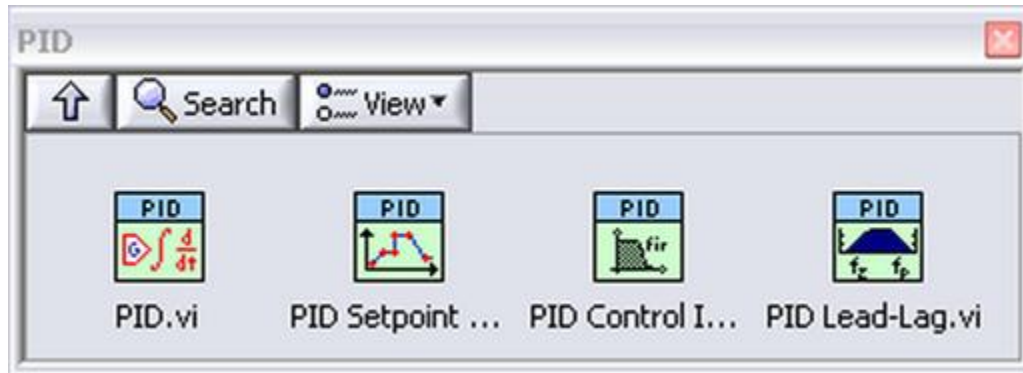
The only VI you need to construct a regular PID controller is the PID VI. The other VIs in this palette implement more advanced functions and can be useful for some specific applications:

PID Setpoint Profile.vi
You can use the Setpoint Profile VI to generate a Setpoint profile. Setpoint profiling can be useful when the system that you are controlling needs to step through a pre-defined series of states. For example, a robotic arm intended to complete a precise sequence of moves to pick up and place an object onto a raised platform. You can use Setpoint profiling to define this sequence of movements so that your robot's behavior is repeatable and predictable; the arm will be in a specified position at a specified time. This method can be quicker and more precise than human control. It's handy for autonomous tasks, as you can program logic to trigger a certain profile based on input from other systems on your robot.

PID Control Input Filter.vi
The Input Filter VI implements a low-pass filter and can be used to filter out high-frequency noise from your process variable signal. Noise on the process variable line can result in undesirable fluctuations in the PID output.

PID Lead-Lag.vi
The PID Lead-Lag VI uses a positional algorithm that approximates a true exponential lead/lag. Lead-Lag control is often used in feed-forward control schemes as a dynamic compensator. This is in direct contrast to the PID algorithm, which is used in feedback systems. In the event that you have a feed-forward controller, use the Lead-Lag VI in place of the PID VI.

Tuning a Controller

A PID controller needs to be tuned in order to function properly. This process is done by adjusting the PID gains to appropriate values for your specific system .The performance of your control system is defined by a set of measurements made when applying a specific input step function as the Setpoint command variable (transitioning from 0 to 100% of the output value instantaneously) and then measuring the response of the System output. These measurements are shown in the graph of a system's response to a step input below:
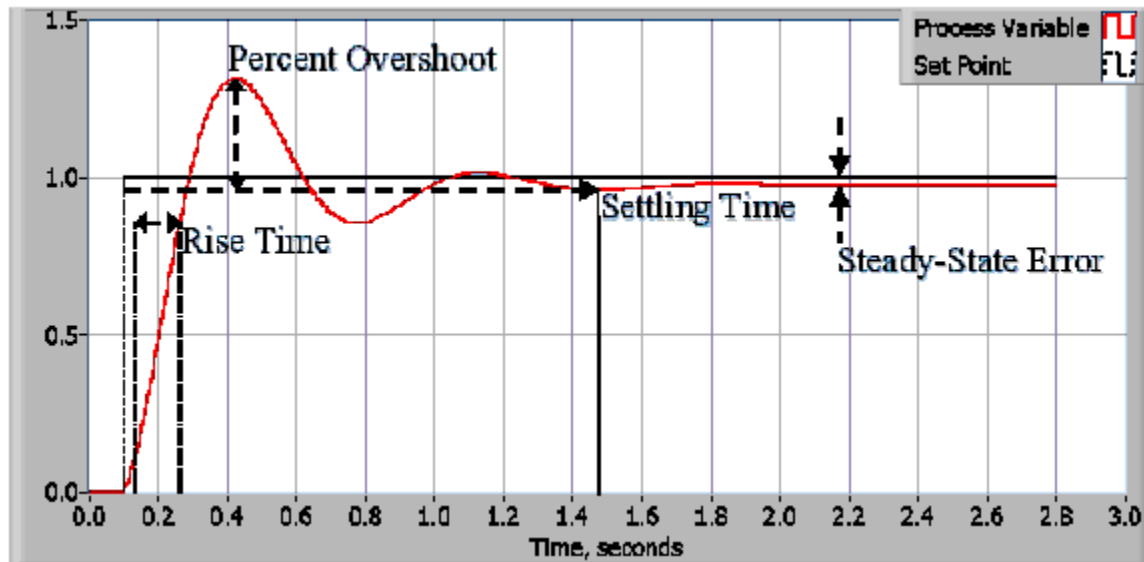
When tuning your controller, you may desire an over-damped, critically damped, or under-damped system. In most robotics applications overshoot is unacceptable and may cause damage to the system. Most robotics systems are over-damped so that they never overshoot their Setpoint. The goal of tuning such systems, then, is decreasing the rise-time and steady-state error to achieve the best possible performance.
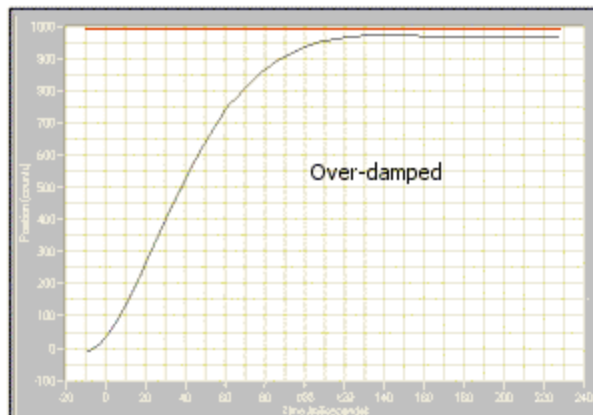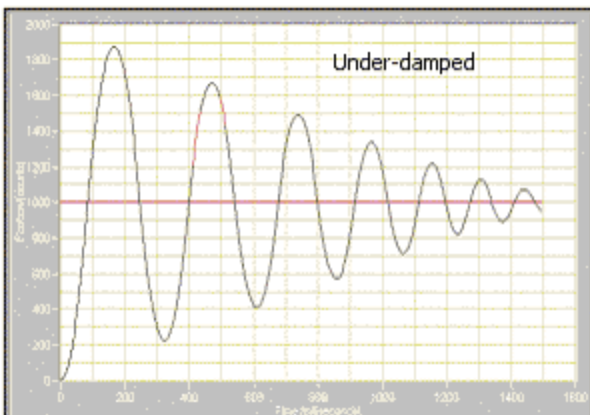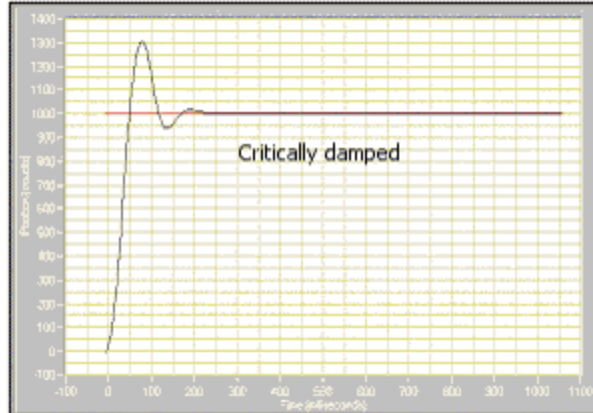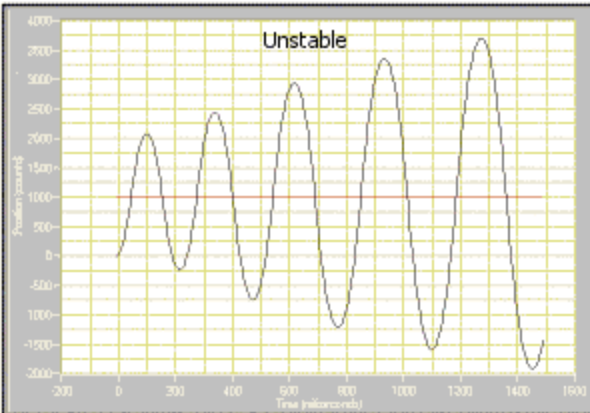
- An **unstable** system produces an oscillatory, exponentially diverging step response. This kind of system never settles down; in fact, the oscillations tend to worsen over time.

- An **under-damped** system produces a slight oscillatory response that eventually dampens out. An under-damped system is characterized by a large overshoot, a long settling time, and short peak and rise times.

- A **critically damped** system response provides a balanced medium between over- and under-damping. This type of system response balances response time and damping effects. A critically damped system is characterized by low overshoot, shorter rise and settling times when compared to over-damped systems, and longer peak times when compared to under-damped systems.

- An **over-damped** system produces a smoother, slower step response. An over-damped system is characterized by no overshoot, and long rise and settling times.

Conclusion

Congratulations, you now have a basic working knowledge of PID theory as it relates to FRC and robotic systems. Leverage the power of LabVIEW and use PID control to build a better robot! To learn more about PID control theory, as well as some advanced topics like gain scheduling and lead-lag controllers, check out the additional resources below, along with the LabVIEW shipping examples (LabVIEW Help menu»Find Examples). If you have suggestions on how to improve this tutorial, please add them to the comments section below so that we can continue to improve our documentation and training resources.

Additional Resources

PID Theory Explained
http://zone.ni.com/devzone/cda/tut/p/id/3782

PID controller
http://en.wikipedia.org/wiki/PID_control