PID Controller

A **proportional integral derivative (PID) controller** is a computer algorithm that implements real time adjustments to the “plant” [1] to compensate for any changes in the system. Some examples of its use is regulating temperatures, pressures, lifting an arm, etc. It is used to gain feedback with the implementation of a closed control loop mechanism. To understand how it works, we first need to define some terms:

Setpoint (SP)- The desired position or value that the controller tries to maintain.

Process variable (PV)- It is the actual value measured in real time.

K (Gain factor)- Each component has a K factor, and it is defined by the user. It is a constant. This calculated by the ratio of change from the output variable to the change of the input variable. It defines how sensitive the output variable will be, given the change of the input variable.

A PID controllers checks the value of the PV and subtracts it with the SP value to calculate the error. **(Error = PV-SP)(1)** This error is later applied to each component of the PID.

A PID can range from many variations/ components here we will discuss a few:

Proportional (P_{out}) = Error * Kp (2)

The proportional control essentially measures the system’s stiffness. It is a form of feedback control. This control calculates the amount of changes that need to be applied to return to the SP value and overcome the position error. This allows to minimize the fluctuation of the PV. It is called proportional because it is directly influenced by the amount of error. Even though this controller has the fastest response time than the other components, yet as the plant becomes more complicated it can produce deviations from the SP know as offset. This offset can be minimized by combining this controller with the integral or derivative controllers. *** An offset is impossible to completely eliminate, yet it can be minimized.** An example of this would be if a plant was overheating it would compare how hot it is with how hot it needs to be and send messages to the other components to manage the issue.

Integral (I_{out})=$Ki \int_0^t e(\tau)d\tau$ (3)

The integral control is another form of feedback control. This control allows for the mitigation of deviations and allows the system to return to its original settings. If the error has a positive value, it will cause the system to increase and vice versa if it the error is negative. If the plant that is being worked on, it is essential not to have any offsets then the I controller should be used. It is not a necessary component depending on what the plant needs. The downside to this controller is that it offers a slower process time than the P controller and it could destabilize the controller. This controller affects the system by responding to the error the system accumulates. An example of this controller would be if the temperature of the plant was too high it could lower the temperature, but it would have a hard time knowing when to exactly stop.

Derivative (D_{out})= Kd * $\frac{de(t)}{dt}$ (4)

The derivative controls works as a damping effect. In contrast to the other two controllers these give feedback as a form of feed forward control. It anticipates the process by analyzing the change in error. It's main function is to avoid or diminish big changes like overshoot and oscillation. D controllers downside are that it does not guide the system to a steady state. To be able to do this D controller must be combined with either P, PI, or I controllers. An example of this controller is that it will deliberately make the integral slower, so it doesn't overshoot when decreasing the temperature. It makes predictions on when the action needs to stop.

Variants:

PI Controller:

Provides quick responses and mitigates offsets. Stops the system from fluctuating and return the system to its SP. The response is quicker than the I controller but 50% slower than the P controllers.

PD Controller:

This controller combines forward and feedback control. It operates to increase the stability of the system by improving the controls using the error predictions it makes. To avoid a sudden change in the error signal the D control is taken from the error signal to output response to avoid changes in the control output.

The difference between these two controllers is that if what you are looking for is to prevent overshoot then PD controller will offer the best results, yet if the need is for speed or accuracy a PI or PID is more optimal. A PI controller works well for tracking the SP, however; it generates at a slower speed than the PID controller.

Changes in Variants

To be able to change from one variant to another:

Tunning- The process of tunning is the ability to change the parameters of the PID controller depending on the user's needs. By being able to change it's parameters one could transition from a PID to a PD or to a PI controller. There are many ways to tune a PID the most common way is by trial and error. Firstly, one could manually do this by only utilizing the P controller (with a low value) while gaining feedback (graphs) and observing when and where does it start oscillating; at that point one could implement the I controller and decrease the value of the P controller. Continue to observe and when it starts oscillating or overshooting again and implement the D controller. After this fine tuning is applied. Remember this will **not** eliminate the issue only try to mitigate it as much as possible. The goal is to make the loop stable, responsive, and minimize overshooting.

Types of Tuning:

- Hueristic tuning- Trial and error method falls in this category. It utilizes the general rules to obtain approximate or qualitative results.

- Rule-based tuning- Assumes a certain process response to obtain easy mathematical formulas to enable the PID controller tuning process.

- Model based tuning- Allows you to work according to a structured tuning process that considers both the process behavior and the control needs.

How controllers could be implemented as code

To start one must define the K value. This is done mathematically by the user even though formulas do exist to derive the value, K depends more on what the user needs the plant to do. The K value that is defined is done before hand by either a set of calculations or by trial and error. Speaking of error since this value is used directly with all the other parameters of the PID it is convenient to first establish its value using (1). After establishing the error and defining the K now it is possible to create the P control. **Note: every component is made and calibrated individually.** Using the (2) formula one can create the proportional controller. Placing the P controller in a while loop one could implement the either the I component or the D component depending on the conditions that need to be met.

[1] Plant- Mechanical equipment used for industrial activity. Name of the object we desire to control.