PID Tuning Methods
An Automatic PID Tuning Study with MathCad

Neil Kuyvenhoven
Calvin College
ENGR. 315
December 19, 2002

Abstract

There are several methods for tuning a PID controller. This paper takes a qualitative look at three common methods, with comparisons of accuracy and effectiveness. These three methods include a guided Trial and Error method, the Ziegler-Nichols method, and the Cohen-Coon method. For an exceptionally responsive system the Trial and Error method is often used after the Ziegler-Nichols or Cohen-Coon so as to enhance the rough results of these two methods. Using these methods in cooperation will result in a finely tuned control system. A study is completed using MathCad to implement automating PID tuning. Due to the nature of MathCad, the process is not fully automated due to some limitations of MathCad, rather the process solves the solution for manually inputting PID coefficients, much like a MatLab process.

Introduction

Models will never emulate their actual physical counterpart perfectly because the mathematical formulae applied is completely predictable; whereas, the physical system being modeled will change over time and due to unaccounted for disturbances. Control systems, specifically PID control systems attempt to reduce the error due to unknown disturbances by designing for typical disturbances that ideally will include any unsuspected disturbances in the physical system being modeled.

PID systems are very unique to each application. As one set of settings may be ideal for one system these same settings can throw another system horribly off. For this reason, multiple methods for tuning the PID coefficients have been made. As some methods are better than others for given applications, each method has its advantages and disadvantages. This paper will outline and compare the three methods known as Trial and Error, Ziegler-Nichols, and Cohen-Coon.

PID Basics

Before explaining the methods for tuning PID control systems, the effects of changing the different components must be examined. For the analysis, the system in figure one will be used, here the plant is shown to be G(S). PID controllers consist of three components; the proportional, integral, and derivative controls.

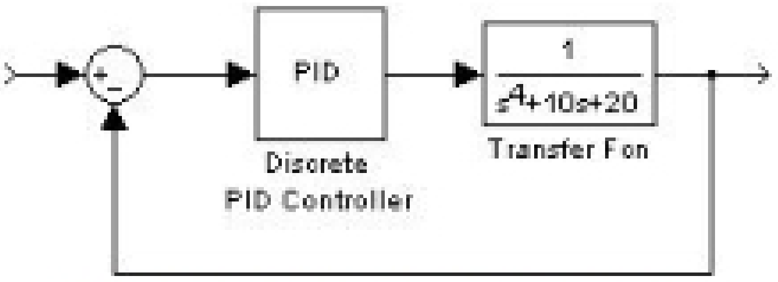
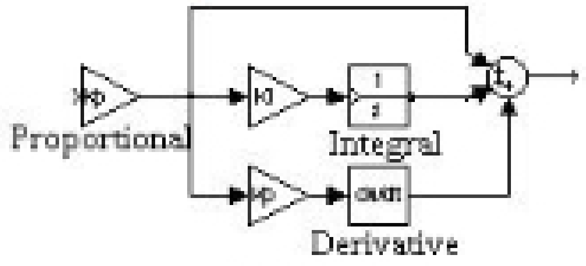
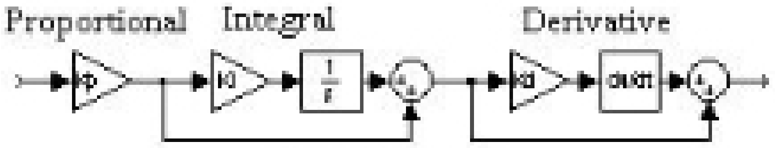


Figure 1 – System with a PID controller

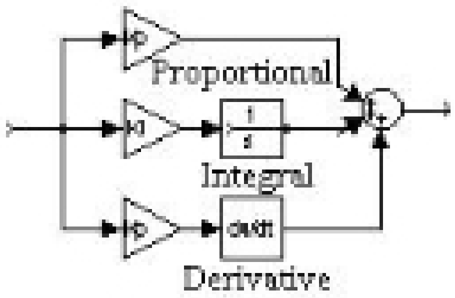
Each of these components has very distinct effects on the system. Table one outlines the effects of the PID components on a particular system. Though there is no set standard for the way a PID controller is set up, there is however three main types. These include the ideal, parallel and series controllers. These types can be seen in figure 1.5 along with there corresponding frequency domain equations as equations one, two and three.



IDEAL PID



SERIES PID



PARALLEL PID

Figure 2 – Ideal, series, and parallel PID configurations

$$H(s) = K_p\left(1 + \frac{1}{K_I \cdot s} + K_d \cdot s\right) \qquad \text{Eq. 1}$$

$$H(s) = K_p \cdot \left(s + \frac{1}{K_I \cdot s}\right)(1 + K_d \cdot s) \qquad \text{Eq. 2}$$

$$H(s) = K_p + \frac{1}{K_I \cdot s} + K_d \cdot s \qquad \text{Eq. 3}$$

The difference between these types of PID controllers is only seen when attempting to tune them. The outcome can be made similar with different values, but as is realized from the equations, one set of parameters for the controller will result in drastically different effects on the the different types of controllers. This difference is most significant on the derivative control in the series and parallel controllers. With the series controller, the derivative control is operating on the partially fixed error as it has already gone through the proportional and integral control. The is juxtaposed by the parallel controller where the parameters are order independent. The effect of the derivative control is not amplified by proportional control which results in a less fine tuned system. The series system begins by amplifying the error which promotes a faster response from the integral and derivative control. The ideal system creates a sensitive response in the proportional control as small changes have big effects on the integral and derivative portions separately. This is different from the series model due to the lack of direct interaction between the integral and series control.

This paper uses the parallel PID control model for analysis.

To increase the rise time of the system, the proportional, integral, and derivative components are tweaked so as to improve the rise time, steady state error, and overshoot respectively. Despite the fact that changing one component effects all the characteristics as seen in table one, the system can still be brought to stability with the three components together; although sometimes fewer than all three are needed.

Table 1 – Effects of PID

	Rise Time	Steady State	Overshoot
Proportional	Decrease	Decrease	Increase
Integral	Decrease	Eliminate	Increase
Derivative	None	None	Decrease

Method 1: Trial and Error

The Trail and Error method requires a closed loop system, it steps through the system from proportional to integral to derivative. This method is a divide and conquer approach, first it puts the system into a rough solution from which small tweaks are performed to perfect the response. To begin, each coefficient of the PID controller is set to zero. The proportional component is now considered by increasing its value until a steady oscillation is obtained as in figure two. Scaling the current proportional value down by a factor of two will give the resulting proportional value. Applying this proportional value will dismiss the steady oscillations. Next the integral coefficient is increased until steady oscillations are again obtained. The present value of the integral coefficient is scaled up by a factor of three and applied to the integral as the final value. This once again sets oscillations off, which brings up the derivative control, this value is increased until for a final time the oscillations are at a constant period and amplitude. The coefficient of the derivative is then scaled down by a factor of three and applied as the final value for the derivative control. The resulting output may still have some noise associated with it, this must now be tuned by hand with small educated tweak of the different coefficients.

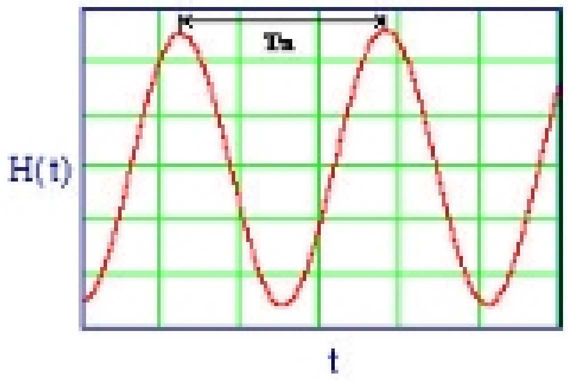


Figure 2 – Steady oscillation illustrating the ultimate period

Method 2: Ziegler-Nichols

The Ziegler Nichols method takes two approaches depending on the system at hand. First, with the closed method or begins in much the same way as the Trial and Error method as a steady oscillation is desired with only a proportional influence present. The proportional value at which the oscillations become constant is coined the term 'ultimate gain'. The period of oscillations at the ultimate gain is termed 'ultimate period'. The ultimate gain can be found in an simpler way with the root locus of the open loop transfer function. The ultimate gain and ultimate period as noted in figure 2, are applied to the Ziegler-Nichols formulae as noted in table 2. This method works provided the closed loop transfer function is known and there is an ultimate gain, the point where the root locus value has zero for the real portion.

Table 2 – Tuning parameters for Ziegler Nichols closed loop ultimate gain method

	KP	KI	KD
P	0.5*Ku		
PI	0.45*Ku	1.2/Tu	
PID	0.6*Ku	2/Tu	Tu/8

The second Ziegler-Nichols method applies to the open loop transfer function. It is simpler to calculate because the guess work is taken out as opposed to the closed loop method where the accuracy of 'steady oscillations' becomes an estimation at best. The tuning parameters for the Ziegler-Nichols open loop method is shown in table three.

Table 3 – Tuning parameters for Ziegler-Nichols open loop ultimate gain method

	KP	*TI*	*TD*
P	$1/(RL)$		
PI	$0.9/(RL)$	$3L$	
PID	$1.2/(RL)$	$2L$	$0.5L$