[SOFTWARE](#)[SERVICES](#)[CASE STUDIES](#)[TRAINING](#)[BLOG](#)[ABOUT](#)[CONTACT](#)

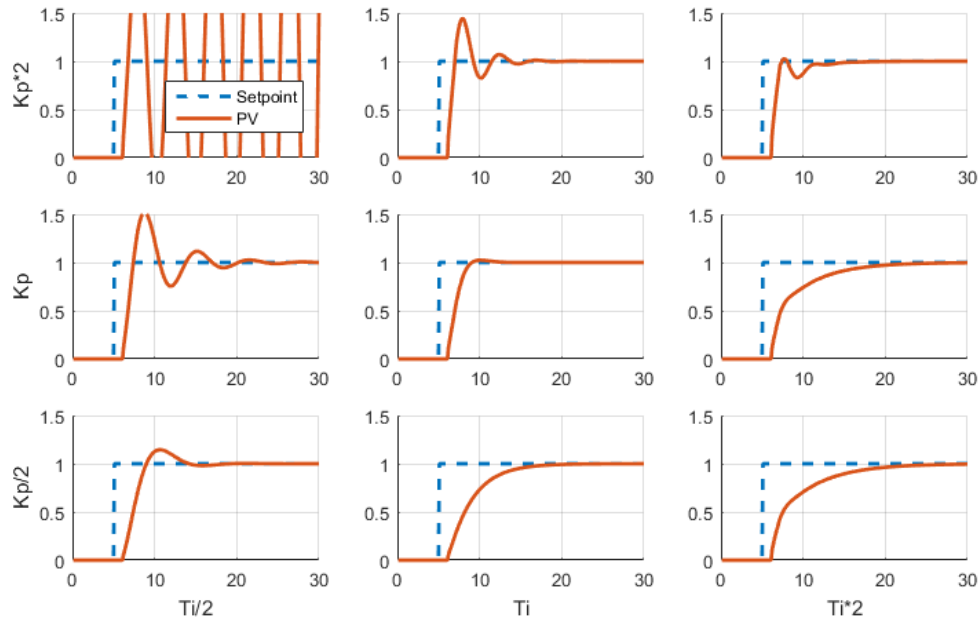
## The PID Controller

The PID controller (an abbreviation of Proportional Integral Differential) is the most widely applied feedback control formula/algorithm. It is applied in a huge variety of 'things' to automate them, such as planes, drones, cars, coffeemakers, wind turbines, furnaces, and manufacturing units. It is fair to say that the PID controller is the work horse for automation. The PID formula has three parameters that must be set or tuned in the right way and that is what is the PID Tuner is used for. In this article we will explain a suitable way to program a PID controller yourself (if you do not want to use standard libraries), and we present some background on different tuning methods.

### The effect of P, I and D

The PID Controller has three parameters that should be tuned. The proportional term ( $K_p$ ), the integral term ( $T_i$ ) and the differential term ( $T_d$ ). The figure below shows the effects of varying  $K_p$  and  $T_i$ .





### The discrete PID equation

The equations below present a PID formula that can be programmed in a control computer (note that we define the output of the PID controller as MV or  $u$ , and the controlled variable as PV):

$$\Delta u_k = K_p \left( e_k - e_{k-1} + \frac{T_s}{T_i} e_k + \frac{T_d}{T_s} (e_k - 2e_{k-1} + e_{k-2}) \right)$$

$$u_k = u_{k-1} + \Delta u_k$$

With:

$k$  = discrete time (0,1,2,...)

$u_k = MV$  = Manipulated Variable at time  $k$

$e_k = PV - SP$  = Error at time  $k$

$SP$  = Setpoint at time  $k$

$PV$  = Process Variable at time  $k$

$K_p$  = Proportional Gain

$T_i$  = Integraltime

$T_d$  = Derivative time

$T_s$  = Sample time

Example code for the discrete PID controller implemented in Matlab can be seen below. Also, you can directly download this code:

- [DOWNLOAD](#) (Matlab)

## PID Tuning

Various methods exist to tune a PID Controller:

- Open loop method, combined with model based (= method used in PID)



Tuner)

- Closed loop Ziegler and Nichols

## Open loop PID Tuning

### Open loop method to tune a PID controller

1. Freeze the PID output
2. Make a step change on the PID output (MV) in order of 3-10%. Step should be sufficiently big to see its effect on the controlled process variable (PV)
3. Log response of MV and PV and fit the model parameters of a First Order Time Delay model to this response.
4. Use model based tuning method, such as Table 1 for single PID loops

Table 1: PID Open Loop Tuning Rules based on [1]

	<b>Kp</b>	<b>Ti</b>	<b>Td</b>
<b>P</b>	--	--	--
<b>PI</b>	$0.35 \cdot T_p / (T_{dt} \cdot K)$	$\min(8 \cdot T_{dt}, T_p)$	--
<b>PID</b>	$0.5 \cdot T_p / (T_{dt} \cdot K)$	$\min(8 \cdot T_{dt}, T_p)$	$0.5 \cdot T_{dt}$

### Reference

[1] S. Skogestad, "Probably the best simple PID tuning rules in the world", Journal of Process Control, 2001

### Closed loop Ziegler and Nichols tuning method

1. Switch on proportional control only, i.e. choose:
  1.  $T_i = \text{'maximum value'}$
  2.  $T_d = 0$  (unless you know which value to choose here)
2. Increase proportional gain until loop oscillations hardly dampen.
3. Then note:
  1. Ultimate gain ( $K_U$ ) = proportional gain
  2.  $T_o$  = period of oscillation
4. Apply settings from Table 2



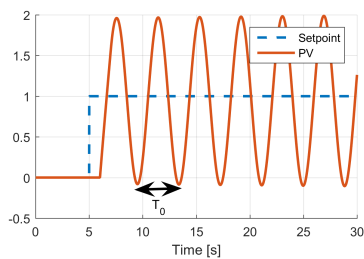


Image: Period of oscillation

Table 2: Closed Loop PI Tuning Rules (Tuning rules for PID are not reliable to our opinion).

	Kp	Ti	Td
PI	0.45*Ku	T0/1.2	0

## Additions to the PID Controller Formula

In reality the output of a PID controller will be limited, e.g. from 0-100%, and sometimes also the rate-of-change should be limited.

Limit the PID controller output between an upper and lower bound

```
if ( u < 100 OR u > 0 )
    du = 0
end
```

Limit the PID controller output rate-of-change

```
du = min( du, 3)
du = mmax( du, -3)
```

Adding a dead-band to the PID controller output

```
If |e| < 1 , then delta Uk = 0
```

## PID Tuning Tips

Tip 1. Be careful with the D action: it



amplifies noise from the sensor signal.

Tip 2. Oscillations in process variables can have three causes, and it's essential to know which one applies in order to reduce the oscillations.

- External disturbance
- Instability in loop resulting in limit cycle
- Stick-slip-like phenomena in valves or other actuators

For more tips, check out [this link \(Tuning tips\)](#) and subscribe to our newsletter

## Evaluating the performance

After you have tuned the PID controller, it is essential to check its closed-loop behavior. You can often do that by making a setpoint change, but sometimes (when the PID is in slave mode), it may be better to make a step change on the Perturbed Manipulated Variable (that is signal that you simply add to the MV, i.e. the output of the PID). Then check if the closed loop behaviour is the same as predicted by the closed-loop simulations in the PID Tuner.

## How do our control engineers do it?

Having tuned many PID controllers for companies in the field, we noticed our jobs become repetitive. That's when we decided that we needed a software tool to help us speed-up and streamline this process. The result of this is the PID Tuner - a 'toolbox' for our control engineers. You can try out a [demo of the PID Tuner](#) for free.

If you are interested in the help of one of our control engineers, or if you have any questions, send us a mail at [info@dotxcontrol.com](mailto:info@dotxcontrol.com)



info\_gw7l50np

---

### Leave a Reply

Your email address will not be published. Required fields are marked \*



Comment

Name \*

Email \*

Website

Post Comment



Oudeweg 91-95  
office B.1-1  
2031 CC Haarlem

**Phone:**  
+31 23 2050241  
**Mail:**  
info@dotxcontrol.com  
**Web:**  
www.dotxcontrol.com



Get in touch

f

in

t

@

