

PID - The Basic Technique for Feedback Control

Paula Raica

Department of Automation

71-73 Dorobantilor Str., room C21, tel: 0264 - 401267

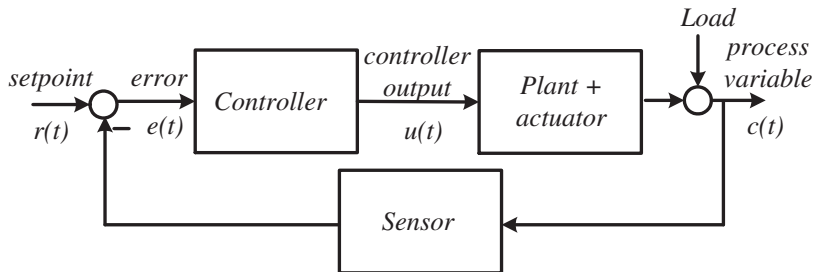
26-28 Baritiu Str., room C14, tel: 0264 - 202368

email: Paula.Raica@aut.utcluj.ro

Technical University of Cluj-Napoca

Introduction

A feedback controller is designed to generate an *output* that causes some corrective effort to be applied to a *process* so as to drive a measurable *process variable* towards a desired value known as the *setpoint*.

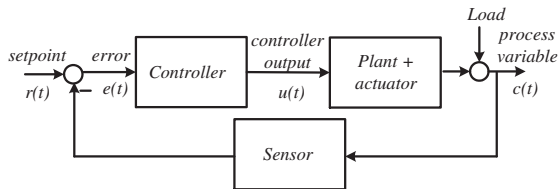


Example

Temperature control in a room:

- Process - the room,
- Process output - temperature
- Setpoint - desired room temperature,
- Transducer/Sensor - thermocouple
- Controller - thermostat
- Controller output - the activation signal to the air conditioner
- Actuator - the air conditioner
- Load(s) - random heat sources (such as sunshine and warm bodies)

PID Control



- PID (Proportional-Integral-Derivative) is the control algorithm most often used in industrial control (95 %).

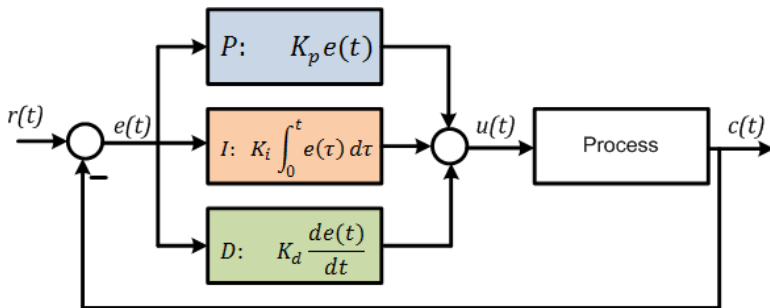
$$u(t) = K_P e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt}$$

- $u(t)$ - the current controller output
- $e(t)$ - error = setpoint - measured output
- K_P - proportional tuning constant
- K_I - integral tuning constant
- K_D - derivative tuning constant

PID controller

Controller output as a function of time:

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt}$$



PID controller

The transfer function of an (ideal) PID controller

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt} \quad | \mathcal{L}$$

$$U(s) = K_P E(s) + K_I \frac{1}{s} E(s) + K_D s E(s) = (K_P + K_I \frac{1}{s} + K_D s) E(s)$$

$$G_{PID}(s) = \frac{U(s)}{E(s)} = K_P + K_I \frac{1}{s} + K_D s$$

proportional controller(P): $G_P(s) = K_P$

proportional-integral controller (PI): $G_{PI}(s) = K_P + \frac{K_I}{s}$

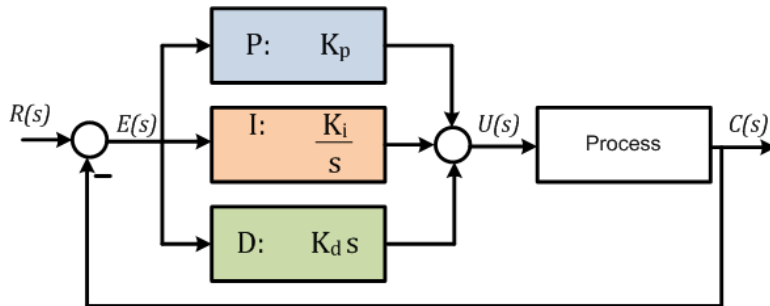
proportional-derivative controller (PD): $G_{PD}(s) = K_P + K_D \cdot s$

proportional-integral-derivative controller (PID): $G_{PID}(s) = K_P + \frac{K_I}{s} + K_D s$

PID Controller

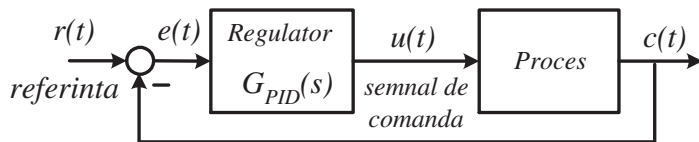
The transfer function of an (ideal) PID controller:

$$G_{PID}(s) = \frac{U(s)}{E(s)} = K_P + K_I \frac{1}{s} + K_D s$$



PID actions

Consider a closed-loop system with a plant having the transfer function $G(s)$ and a PID controller.



For each term P, I and D, analyze:

- the error signal $e(t)$ and the control signal $u(t)$
- the process output $c(t)$

P action

$K_D = 0$, $K_I = 0$. The P controller output is:

$$u(t) = K_P e(t) \text{ and the transfer function: } G_P(s) = K_P$$

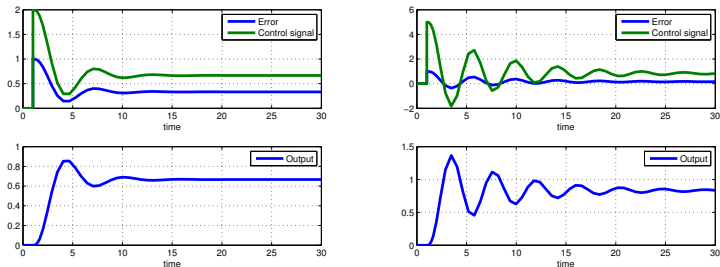


Figure: (left) $K_P = 2$, (right) $K_P = 5$

Notice that:

- the steady-state error decreases with increasing K_P
- process output is more oscillatory with increasing K_P

I action

$K_D = 0$. The PI controller output is:

$$u(t) = K_P e(t) + \int_0^t e(\tau) d\tau \quad \text{and the transfer function: } G_{PI}(s) = K_P + \frac{K_I}{s}$$

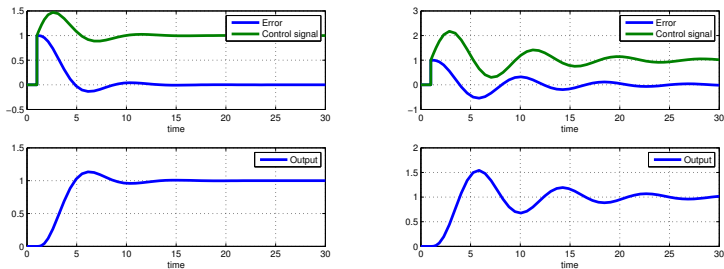


Figure: PI control. (left) $K_P = 1, K_I = 0.5$, (right) $K_P = 1, K_I = 1$

Notice that:

- the response is faster (smaller rise time) but more oscillatory with increasing integral gain K_I
- the steady-state error is zero for any value of the integral gain K_I
- the main function of the integral action is to make sure that the error is zero at steady-state.
- with integral action, a small positive error will always increase the control signal, and a negative error will give a decreasing control signal, no matter how small the error is.

D action

$K_I = 0$. The output of a PD controller is:

$$u(t) = K_P e(t) + K_D \frac{de(t)}{dt} \quad \text{and the transfer function:}$$

$$G_{PI}(s) = G_{PI}(s) = K_P + K_D s$$

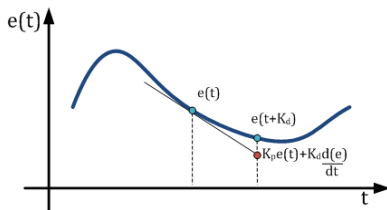


Figure: Derivative as prediction obtained by linear extrapolation

The action of a PD controller may be interpreted as if the control is proportional to the *predicted* process output: extrapolate the error by the tangent to the error curve.

D action

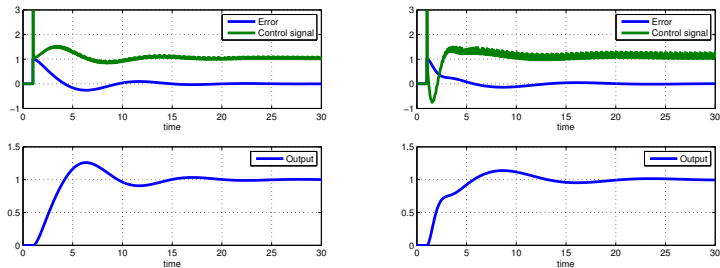


Figure: PID control. (left) $K_P = 1, K_I = 1, K_D = 1$, (right) $K_P = 1, K_I = 1, K_D = 3$

Notice that:

- the steady-state error is zero because of the integral term in the controller
- the overshoot decreases with K_D
- because of the sudden change in error, the derivative term has a large value at the initial time, thus producing a very large initial control signal

PID controller - other forms

- The general transfer function of a PID controller is:

$$G_{PID}(s) = \frac{U(s)}{E(s)} = K_P + K_I \frac{1}{s} + K_D s \quad (1)$$

- The relation above can be rearranged to give::

$$\begin{aligned} G_{PID}(s) &= K_P \left(1 + \frac{K_I}{K_P s} + \frac{K_D}{K_P} s \right) \\ &= K_P \left(1 + \frac{1}{T_i s} + T_d s \right) \end{aligned} \quad (2)$$

- where the controller parameters are:

- K_P the controller gain
- $T_i = \frac{K_P}{K_I}$ the integral time
- $T_d = \frac{K_D}{K_P}$ the derivative time

PID controller - filtering the derivative

- We have assumed that a D behavior can be realized by a PID controller. This is an ideal assumption and in reality the ideal D element cannot be realized.
- In **real** PID controllers a lag is included in the D behavior, as follows:

$$G_D(s) = \frac{T_d s}{\frac{T_d}{N} s + 1}$$

where N has a large value.

- The transfer function of the *real* PID controller is:

$$G_{PID} = K_P \left(1 + \frac{1}{T_i s} + \frac{T_d s}{\frac{T_d}{N} s + 1} \right)$$

Tuning PID controllers

Tuning = setting the K_P , K_I , and K_D so that the weighted sum of the P, I and D terms produces a controller output that drives the process variable in the direction required to eliminate the error.

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt}$$

Sluggish process:

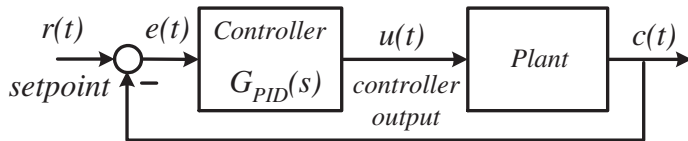
- If an error is introduced abruptly, the controller's initial reaction will be determined primarily by the D term.
- The proportional term will then come in to play: keeps the controller's output going until the error is eliminated.
- The integral term will also begin to contribute to the controller's output as the error accumulates over time. The process variable may overshoot.

Tuning PID controllers

Three approaches:

- 1 'Trial-and-error' (experienced engineers). Example: decrease of $K_I \Rightarrow$ reduces the overshoot but slows the rate of change of the error
- 2 The analytical approach. If a mathematical model of the process is known, there are various methods for controller tuning.
- 3 A compromise between purely heuristic trial-and-error techniques and the analytical techniques. It was originally proposed in 1942 by John G. Ziegler and Nathaniel B. Nichols of Taylor Instruments.

Tuning PID controllers



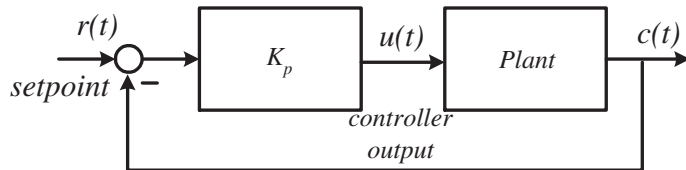
In practice the output of a PID controller is given by:

$$u(t) = K_p \left[e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right]$$

$$G_{PID}(s) = \frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s} + T_d s \right)$$

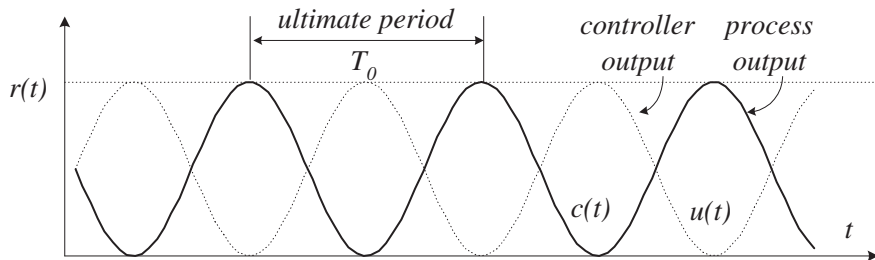
where K_p = proportional gain, T_i = integral time, T_d = derivative time

The Ultimate Sensitivity Method



- Set $T_i = \infty$, and $T_d = 0$.
- Increase K_p from 0 up to a critical value K_0 where the output, $c(t)$, first exhibits sustained oscillations.

The Ultimate Sensitivity Method



Determine experimentally:

- K_0 - the ultimate gain
- T_0 - critical period.

The Ultimate Sensitivity Method

Type of controller	K_p	T_i	T_d
P	$0.5K_0$	∞	0
PI	$0.45K_0$	$1/1.2T_0$	0
PID	$0.6K_0$	$0.5T_0$	$0.125T_0$

PID - Ziegler-Nichols methods

The PID controller tuned by the ultimate sensitivity method of Ziegler-Nichols gives:

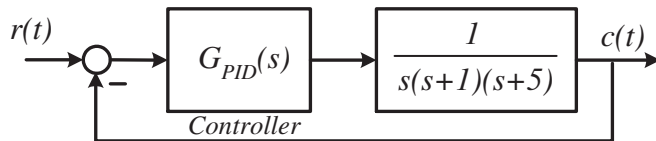
$$\begin{aligned} G_{PID}(s) &= K_p \left(1 + \frac{1}{T_i s} + T_d s \right) = 0.6 K_0 \left(1 + \frac{1}{0.5 T_0 s} + 0.125 T_0 s \right) \\ &= 0.075 K_0 T_0 \frac{(s + 4/T_0)^2}{s} \end{aligned}$$

Thus, the PID controller has a pole at the origin and double zeros at $s = -4/T_0$.

Derivative cautions. If the process output rate of change is caused by noise, the D term may cause over-corrections. (ex. in pressure and level control)

PID - Ziegler-Nichols methods

Ziegler-Nichols tuning rules can be applied to plants where the model is known.



$T_i = \infty$, $T_d = 0$, closed-loop transfer function:

$$\frac{C(s)}{R(s)} = \frac{K_p}{s(s+1)(s+5) + K_p}$$

The value of K_p that makes the system marginally stable so that sustained oscillation occurs can be obtained by use of Routh's stability criterion.

PID - Ziegler-Nichols methods

The characteristic equation:

$$s^3 + 6s^2 + 5s + K_p = 0$$

s^3	:	1	5
s^2	:	6	K_p
s^1	:	$\frac{30 - K_p}{6}$	0
s^0	:	K_p	

$$s^3 + 6s^2 + 5s + 30 = 0$$
$$(s + 6)(s^2 + 5) = (s + 6)(s^2 + \omega_n^2) = 0$$
$$\Rightarrow \omega_n = \sqrt{5} = 2\pi \cdot f = 2\pi \cdot \frac{1}{T_0},$$

$$\Rightarrow K_0 = K_p = 30$$

$$T_0 = \frac{2\pi}{\omega_n} = \frac{2\pi}{\sqrt{5}} = 2.81$$

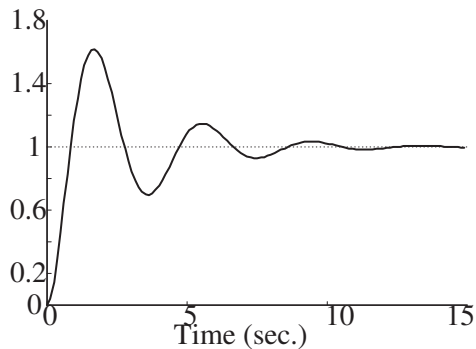
$$K_p = 0.6K_0 = 18, \quad T_i = 0.5T_0 = 1.405, \quad T_d = 0.125T_0 = 0.35$$

PID - Ziegler-Nichols methods

The transfer function of the PID controller is:

$$G_{PID}(s) = 18 \left(1 + \frac{1}{1.405s} + 0.35s \right) = \frac{6.32(s + 1.42)^2}{s}$$

The unit step response of this system (computer simulation):



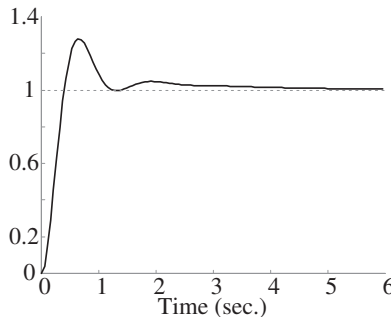
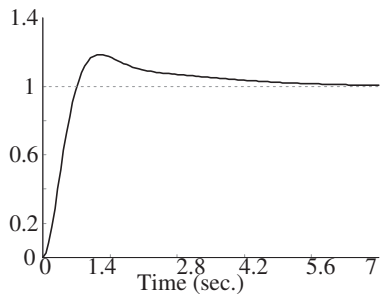
PID - Ziegler-Nichols methods

Move the zero at -0.65 :

$$G_{PID}(s) = \frac{13.84(s + 0.65)^2}{s}$$

Increase K_p to 39.42:

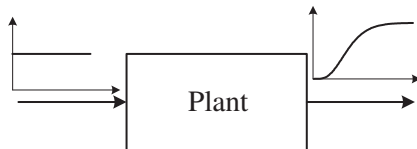
$$G_{PID}(s) = \frac{30.322(s + 0.65)^2}{s}$$



PID - Ziegler-Nichols methods

Transient response method or *reaction curve (open-loop) method*

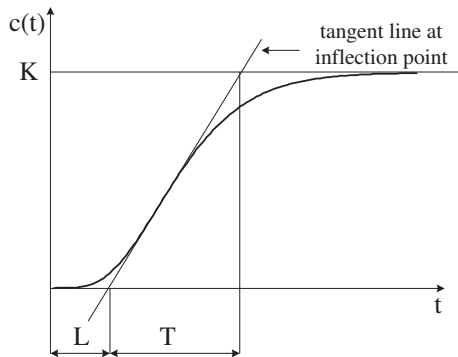
Make a step change to the process and record the process output:



⇒ open-loop gain K , the loop apparent deadtime L , the loop time constant T .

PID - Ziegler-Nichols methods

Transient response method



The transfer function of the plant = first order with time delay:

$$\frac{C(s)}{U(s)} = \frac{Ke^{-Ls}}{Ts + 1}$$

PID - Ziegler-Nichols methods

Transient response method

Type of controller	K_p	T_i	T_d
P	T/L	∞	0
PI	$0.9T/L$	$L/0.3$	0
PID	$1.2T/L$	$2L$	$0.5L$

PID controller:

$$G_{PID}(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) = 0.6T \frac{(s + 1/L)^2}{s}$$

a pole at the origin and double zeros at $s = -1/L$.

PID - Ziegler-Nichols methods

Zeigler-Nichols tuning methods tend to produce systems whose transient response is rather oscillatory and so will need to be tuned further prior to putting the system into closed-loop operation.