Eng Phys 3BB4

PID Controllers

Dr. Adriaan Buijs

McMaster University

Winter 2021

PID project 2021 I

Under normal circumstances, the 3BB project consists of controlling a system, such as a heater, with the MSP microcontroller; the MSP runs the PID process, and communicates with MatLab to receive the set points, the PID parameters, and to present the output of the process.

As labs are performed at home, the setup is changed by replacing the hardware by a second microcontroller, an Arduino Nano, or similar.

A C-code is provided for the Arduino. The Arduino accepts a signal on one pin, (the input to the process/system), and outputs a signal on another pin; this is the output of the process/system, **but it is convoluted with the response from a sensor**.

As the hardware is simulated on the Arduino, the process/system can be anything, and it doesn't actually matter what the process is. Except if the system were a simulation of, say, a crane, which would have two outputs: the angle of the pendulum, and the position of the carriage.

PID project 2021 II

In our process, we will simulate the reactor regulating system of a CANDU reactor. (You should familiarize yourself with the general idea of nuclear reactors, and how Canadian reactors are different from everyone else's. Not for this course, but because as an Engineer you should have such knowledge.)

The input to this system is the liquid zone controller (LZC) level. A CANDU reactor has 14 LZCs distributed over the reactor core. They act as neutron absorbers, because in a CANDU reactor H_2O absorbs neutrons; the reactor itself works with heavy water, D_2O. If the LZC level is 0%, there is no water and no neutrons are being absorbed. The *reactivity* ρ is high. If the LZC is 100% full, neutrons are being absorbed, the reactivity is reduced.

PID project 2021 III

The behaviour of the CANDU reactor can be described by equations called *point kinetics equations* (because they have no spatial dependence, the reactor is considered to be a point):

$$\begin{aligned} \frac{dn}{dt} &= \frac{\rho - \beta}{\Lambda} n(t) + \lambda C(t), \\ \frac{dC}{dt} &= \frac{\beta}{\Lambda} n(t) - \lambda C(t). \end{aligned} \tag{1}$$

In these coupled equations, $n(t)$ represents the neutron population, which is proportional to the power $p(t)$; this is the output of your system. The reactivity ρ is driven by the LZCs, i.e. it is the input $x(t)$ (or $r(t)$) to your system. β is the fraction of *delayed* neutrons, originating from the *delayed neutron precursors*, which have a concentration $C(t)$. The neutron precursors decay with a decay constant λ, but are regenerated with the rate β/Λ, where Λ is the generation time.

McMaster University

PID project 2021 IV

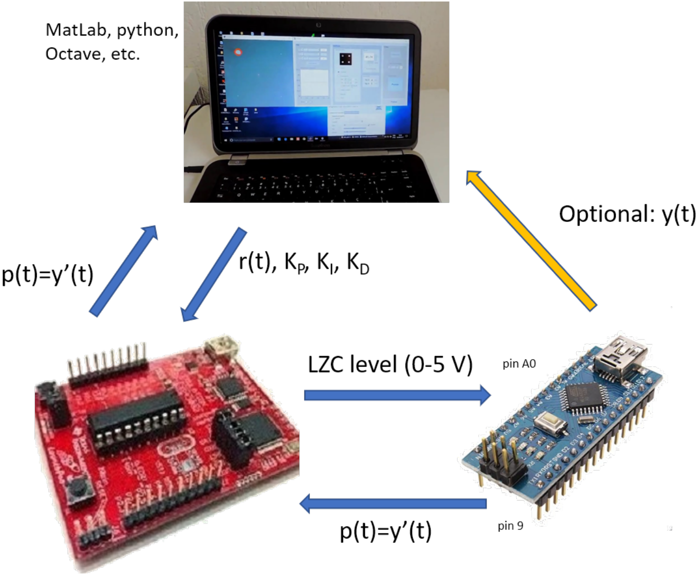
These equations look difficult, but a) you don't really need to worry about them, and b) they actually apply to a lot of things; e.g. the spread of COVID can be described by them, too.

As usual, you would approach them by transforming them with a Laplace transform to the s domain.

In this project, you will build the PID control that regulates the LZCs to keep the power (n or p) at a preset value.

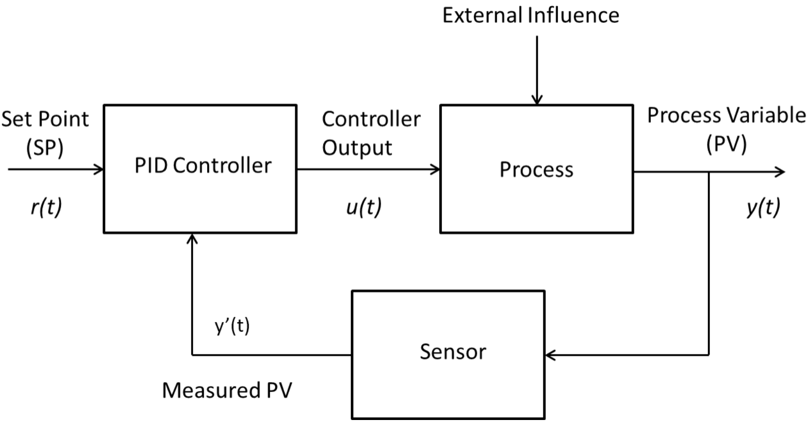
The following slides show how a process control works. Unfortunately, the variables are somewhat different from what you are used to, and may be confusing. For example, the variable $u(t)$ is used both for the input to the process, (the level of the LZC) and for the step function.

PID project 2021 V



Process Control Diagram

Diagram of a process control setup:



Youtube

Step Response I

In the following graph, $u(t)$ is a step function, and $y_{\text{step}}(t)$ is the step response, either of the process itself, or of the process followed by the measurement.

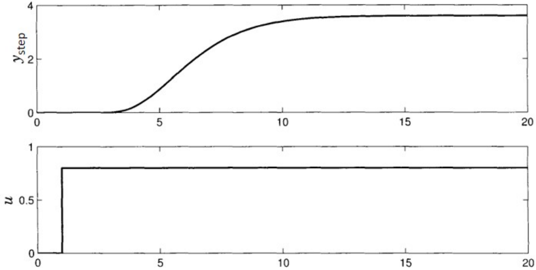


Figure 2.4 The lower curve shows an input signal in the form of a step, and the upper curve shows the response of the system to the step.

Now $h(t) = dy_{\text{step}}/dt$ is the impulse response.

Step Response II

If we know the impulse response, we can calculate the response to any function $u(t)$ by the convolution of that function with the impulse response:

$$y(t) = \int_0^\infty u(\tau)\frac{dy_{\text{step}}(t-\tau)}{dt}d\tau = \int_0^\infty u(\tau)h(t-\tau)d\tau. \tag{2}$$

Call $U(s)$ the Laplace transform of $u(t)$ (again, u is a general function):

$$U(s) = \mathcal{L}(u(t)) = \int_0^\infty u(t)e^{-st}\,dt, \tag{3}$$

with $s = \sigma + j\omega$. Similarly for $Y(s)$ and $H(s)$, and we get that

$$Y(s) = U(s)H(s), \tag{4}$$

The big advantage is that the convolution in the time domain has changed to a simple multiplication in the s domain.

Step Response III

Often, the process variable $y(t)$ that we are interested in can only be known after measurement; in that case, $h(t)$ would have to be convoluted with the measurement function, say $m(t)$. (or the product taken in the s-domain).

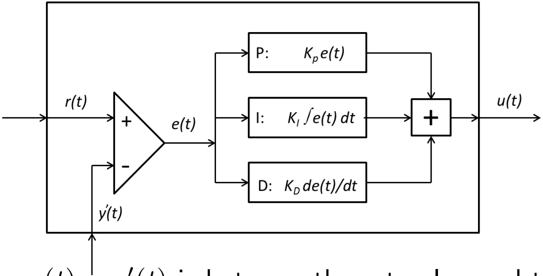
Let's call the convoluted signal $y'(t) = (y(t) * m(t))$, and in the s-domain $Y'(s) = Y(s)M(s)$.

One would measure the response of the measurement instrument separately (e.g. the thermistor response $m(t)$ can be measured (throw it in a bucket of ice-water), but it can also be obtained from the spec sheet.) And then you would try to unfold the response of the system proper.

In the following, we use this formalism to have a closer look at PID control.

PID Controller

PID Controller Diagram ($u(t)$ is not necessarily a step function):

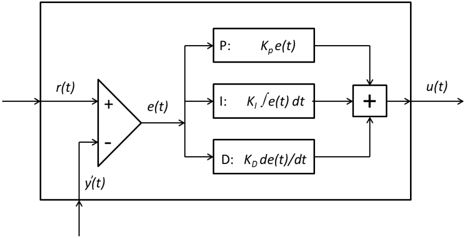


The error $e(t) = r(t) - y'(t)$ is between the set value and the *measured* Process Variable.

$$u(t) = K_{\mathrm{P}} e(t) + K_{\mathrm{I}} \int_0^t e(t')\,dt' + K_{\mathrm{D}} \frac{de(t)}{dt}, \qquad \text{or} \tag{5}$$

$$K_{\mathrm{P}} \left(e(t) + \frac{1}{T_I} \int_0^t e(t')\,dt' + T_D \frac{de(t)}{dt} \right). \tag{6}$$

PID Controller formulas I



- $e(t) = r(t) - y'(t)$ is the difference between setpoint $r(t)$ (which can vary in time) and the measured process variable: the **error**;
- $u(t) = K_{\mathrm{P}} e(t) + K_{\mathrm{I}} \int_0^t e(t')\, dt' + K_{\mathrm{D}} \frac{de(t)}{dt}$ is the response to error:
  - $K_{\mathrm{P}} e(t)$ is the **proportional**, or immediate response;
  - $K_{\mathrm{I}} \int_0^t e(t')\, dt'$ is the response to **past errors** (Integral); The lower limit does not have to be zero.
  - $K_{\mathrm{D}}\, de(t)/dt$ is the response to anticipated **future errors** (Derivative);
- Note units of K_{P}, K_{I} and K_{D}!

PID Controller formulas II

Transfer function $G(s)$ of the PID Controller:

$$G(s) = \frac{U(s)}{E(s)} = K_{\mathrm{P}} \left(1 + \frac{1}{sT_I} + sT_D \right), \tag{7}$$

and

$$E(s) = R(s) - Y'(s). \tag{8}$$

Putting everything together (including (4) and $Y'(s) = Y(s)M(s)$):

$$\frac{Y(s)}{R(s)} = \frac{H(s)G(s)}{1 + M(s)H(s)G(s)}. \tag{9}$$

Is the **closed-loop** response of the system.

The denominator may have zeroes, which determine the stability of the system.

Effects of K parameters:

Effects of increasing a parameter independently

Par	K_{P}	K_{I}	K_{D}
Rise time	decrease	decrease	minor change
Overshoot	increase	increase	decrease
Settling time	small change	increase	decrease
Steady-state error	decrease	Eliminate	No effect in theory
Stability	degrade	degrade	improve if K_{D} is small

Animated PID control (Wikipedia)

USS Gerald R. Ford

Characteristics

Characteristic	Value	Unit
Displacement	100,000	tonnes
	10^8	kg
Length	337	m
Propulsion	2	A1B nuclear reactors
Speed	56	km/h
Power	260	MW (shaftpower)

Docking I

The USS Gerald R. Ford is at distance 100 km from Newport News, where it plans to dock. This will need some PID!

- Let's assume there are two forces acting on the aircraft carrier:
  1. Force from the propulsion power: F_p,
  2. Force from the drag:

$$F_d = \frac{1}{2} v^2 \rho C_d A, \tag{10}$$

v is velocity of ship (m/s), ρ is density (kg/m^3), A is the surface in contact with the water in m^2.
$C_d \approx 0.04$ for a streamlined body. Let's assume it is.

Docking II

- Total force on ship: $F = F_p - F_d$, so

$$F = F_p - F_d = m\ddot{y}. \tag{11}$$

Therefore,

$$F_p = m\ddot{y} + F_d = m\ddot{y} + \alpha\dot{y}, \tag{12}$$

with $\alpha = \frac{1}{2} v\rho C_d A$. Note the sneaky thing with $v\dot{y}$.
Enter PID:

$$F_p = K_\mathrm{P} e(t) + K_\mathrm{D}\dot{e}(t) + K_\mathrm{I}\int_0^t e(t')\,dt', \tag{13}$$

  - $e(t)$ is the error $e(t) = r(t) - y(t)$.
  - take $r(t) = 0$ is the dock at Newport News.
  - Let's not consider the integral (it won't help much).

Docking III

- This leaves

$$-K_{\mathrm{P}}y - K_{\mathrm{D}}\dot{y} = m\ddot{y} + \alpha\dot{y}, \tag{14}$$

or

$$m\ddot{y} + (\alpha + K_{\mathrm{D}})\dot{y} + K_{\mathrm{P}}y = 0. \tag{15}$$

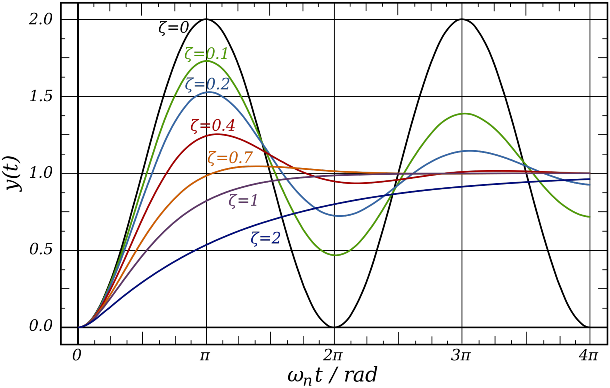
- Compare this to a damped oscillator:

$$m\frac{d^2y}{dt^2} + c\frac{dy}{dt} + ky = 0, \tag{16}$$

whose solutions can be underdamped ($\zeta < 1$), overdamped ($\zeta > 1$) or critically damped ($\zeta = 1$), depending on the damping ratio

$$\zeta = \frac{c}{2\sqrt{km}} \tag{17}$$

Docking IV



- Find critical damping by chosing K_P and K_D such that:

$$(\alpha + K_\mathrm{D}) = 2\sqrt{K_\mathrm{P} m}. \tag{18}$$

McMaster University

Docking V

- Another approach: do Laplace transformation: try $y(t) = y(0)e^{st}$, then:

$$ms^2 + (\alpha + K_\mathrm{D})s + K_\mathrm{P} = 0. \tag{19}$$

Solve for s:

$$s_{1,2} = \frac{1}{2m}\left(-(\alpha + K_\mathrm{D}) \pm \sqrt{(\alpha + K_\mathrm{D})^2 - 4mK_\mathrm{P}}\right). \tag{20}$$

- Now it is critically damped when the expression under the root is zero.
- Plot $s_{1,2}$ in the complex plane, and see how they move as a function of K_D and K_P.
- As long as $4mK_\mathrm{P} < (\alpha + K_\mathrm{D})^2$, then poles are on the negative real axis; there is no oscillatory behaviour.

Tuning a PID Controller Ziegler-Nichols Method

- Turn off the I and D components ($K_{\mathrm{I}} = K_{\mathrm{D}} = 0$).
- Find the minimum gain K_{P} that sustains oscillations with constant amplitude (K_c).
- Measure the period of oscillations (P_c) in seconds.

	K_{P}	$T_I = K_{\mathrm{I}}/K_{\mathrm{P}}$	$T_D = K_{\mathrm{P}}/K_{\mathrm{D}}$
P	$0.5 \times K_c$	–	–
PI	$0.45 \times K_c$	$P_c/1.2$	–
PID	$0.6 \times K_c$	$P_c/2$	$P_c/8$

Other Strategy (from http://ctms.engin.umich.edu/CTMS/index.php?example=Introduction&section=ControlPID)

Follow the steps shown below to obtain a desired response:

- Obtain an open-loop response and determine what needs to be improved;
- Add a proportional control to improve the rise time;
- Add a derivative control to improve the overshoot;
- Add an integral control to eliminate the steady-state error;

Adjust each of K_P, K_I, and K_D until you obtain a desired overall response.

you do not need to implement all three controllers (proportional, derivative, and integral) into a single system, if not necessary. For example, if a PI controller gives a good enough response, then you don't need to implement a derivative controller on the system. Keep the controller as simple as possible.