PID Control

PID control is perhaps the most common and pervasive of controller designs. It has been used in the control of ~~insulin~~ insulin delivery for the artificial pancreas, dialysis machine control, in the left ventricular assist device (LVAD) etc.

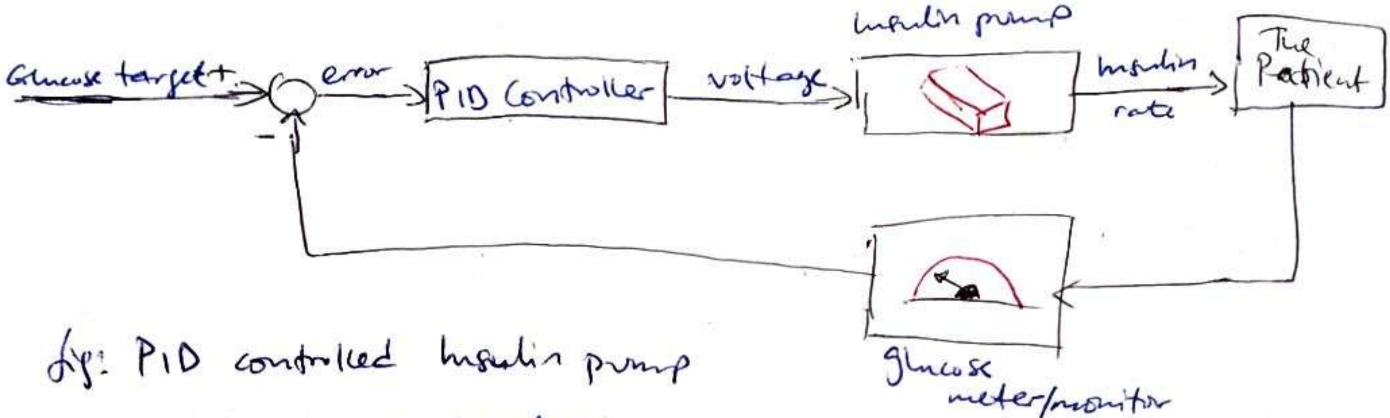


fig: PID controlled insulin pump

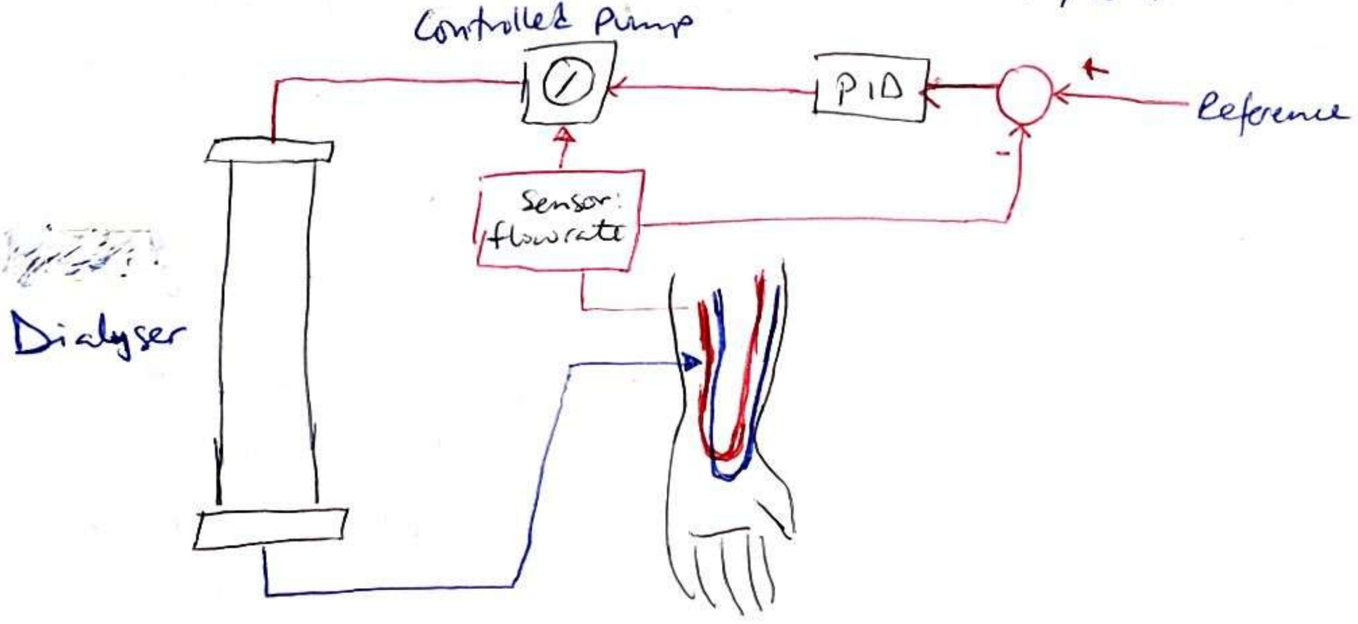


fig: A dialysis machine with a PID controlled pump

A basic form of the PID controller is:

$$u(t) = k_p e(t) + k_i \int e\,dt + k_d \dot{e}(t)$$

proportional + integral + derivative terms

where $u(t)$ is the output signal generated by the PID controller given input $e(t)$.

$$u(s) = k_p e(s) + \frac{k_i}{s} e(s) + k_d s\, e(s)$$

$$\frac{u(s)}{e(s)} = k_p + \frac{k_i}{s} + k_d s = \frac{k_p s + k_i + k_d s^2}{s}$$

The transfer function of the PID controller.

Recall that the derivative term, $k_d s$ is improper and not physically realisable hence it is, in practice more common in the filtered form: $\frac{k_d s \omega}{s + \omega}$ res.

The $k_d s$ is filtered using a first order filter $\frac{\omega}{s+\omega}$.
Other filters may be employed if desired.

The filtered PID controller has the transfer function

$$\frac{u(s)}{e(s)} = \frac{k_p s(s+\omega) + k_i(s+\omega) + k_d s}{s(s+\omega)}$$

Observe that the numerator and denominator above are both polynomials of degree 2. The PID controller with the improper derivative term has denominator polynomial less than numerator polynomial.

In order to determine suitable k_p, k_i, and k_d such that control objectives are achieved, several approaches are used. There's the purely analytical approach, trial-and-error via simulation, and the use of predetermined tuning rules.

Analytic Approach:-
Controller design, simply means determining via some means, k_p, k_i, and k_d such that the closed loop performance is acceptable. The analytic approach, in a nutshell, seeks to determine equations or formulae ~~that~~ for k_p, k_i, and k_d.

Trial-and-error:-
Here, it is common to use modelling and simulation software to model the system and investigate the response to intelligently ~~fixed~~ or arbitrarily choosen values of k_p, k_i, k_d in order to determine the best combination. This is manual tuning

Tuning rules approach:-

Some rules for determine the parameters of the PID controller ~~[illegible]~~ based on some knowledge of the system to be controlled have been determined. So the rules of tuning rules requires obtaining the required knowledge about the system that permits the determination of the required parameter values of the PID controller.

It should be noted that ~~tuning~~ tuning rules typically are oriented towards achieving certain characteristic closed-loop performance.

Popular tuning rules include Ziegler-Nichols, and Cohen-coon. There are several others, and that may be tied to specific applications.

We will demonstrate the use of tuning rules using the Ziegler-Nichols Open-Loop technique; This is also known as the ~~second~~ Ziegler-Nichols Step response method. It can be used where the output of a system exhibits a "S" shape for step inputs. That is, it is applicable when the system responses resembles that of a first order system with, probably, a small delay; or a second order system that isn't underdamped or undamped. This means that this recipe is not applicable to systems containing [a chain of] integrators or dominant complex poles.

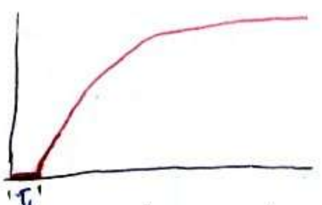


fig: A first order system with delay (or dead-time) T_d

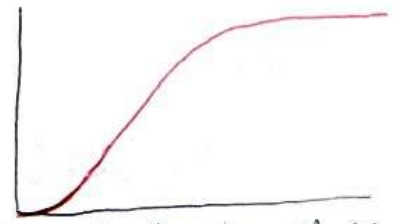
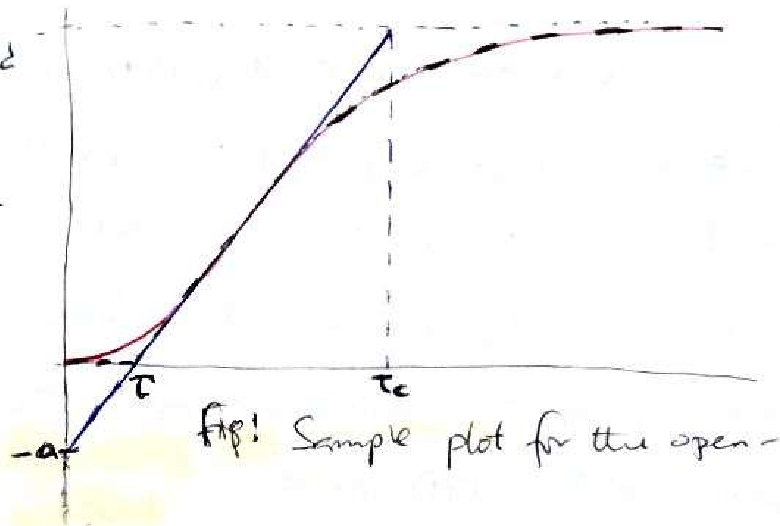


fig: A critically damped or overdamped response.

NB
The dashed response is that of a first-order plus time delay system



The parameter τ is an approximation of the time delay of the system, and $\frac{a}{\tau}$ is the steepest slope of the step response. $\tau_c - \tau$ is the assumed time constant.

Fig! Sample plot for the open-loop Ziegler-Nichols

Steps. to tune a PID controller using this method:

1. Consider a system in open loop and obtain the plot of its unit step response. It is assumed that the system has a steady state value
2. Identify values a and τ ~~on~~ the plot (as shown above). 'a' is at the intercept of the steepest tangent of the step response with the vertical axis. 'τ' comes from the intercept of the same tangent with the horizontal axis.
3. Use the values obtained to calculate the gains for each controller according to the following table:

Table: Ziegler-Nichols step response PID Design and ~~tools~~

Controller Type	k_p	T_i	T_d
P	$\frac{1}{a}$		
PI	$\frac{0.9}{a}$	3τ	
PID	$\frac{1.2}{a}$	2τ	0.5τ

Note that τ is ~~also~~ the apparent time delay. And the Tuning rule assumes a first-order plus deadtime (time delay) system.

The resulting controller is given by

$$\frac{u(s)}{e(s)} = K_p\left(1 + \frac{1}{sT_i} + T_d s\right)$$

in transfer function form

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

in time domain form.

observe that if we set

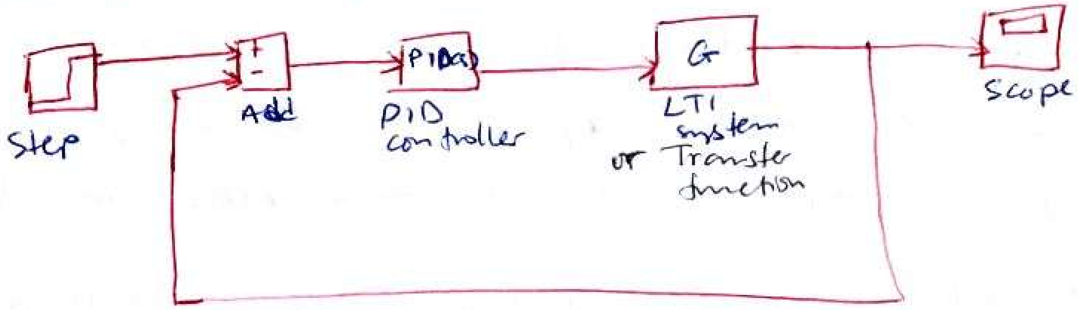
$$u = k_p e + k_i\int_0^t e(\tau)d\tau + k_d\dot{e}$$

this would imply that

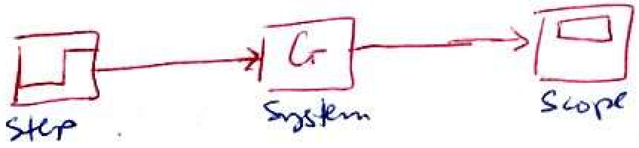
$$k_i = \frac{k_p}{T_i} \quad \text{and}$$

$$k_d = k_p T_d$$

Simulation/Test using Simulink (or Xcos on Scilab):



To obtain the open-loop step response required for the design process do, on Simulink, using a representative model G.



~~How does~~ Recall how this translates to real systems. A step is a sudden change ~~at~~ the reference input to a system, for instance increasing the level of an drug from ~~say~~ one concentration to another (within a very short interval of time). The output response will come from measurements taken from the effect of the change in input.

NB: ~~Reecall~~ Recall that the derivative term will be implemented with a filter. The closed-loop response will be slightly different depending on the choice of filter type and parameters.

» Manual ~~F~~ Tuning or Trial-and-Error Method.

Table: Contribution of each component of the PID controller to the system response when each parameter is increased

	Rise time	Overshoot	Settling time	Error
k_p	Reduces	Increases	Slight change	Reduces
k_i	Reduces	Increases	Reduces	Disappears
k_d	Slight change	Reduces	Reduces	Slight change

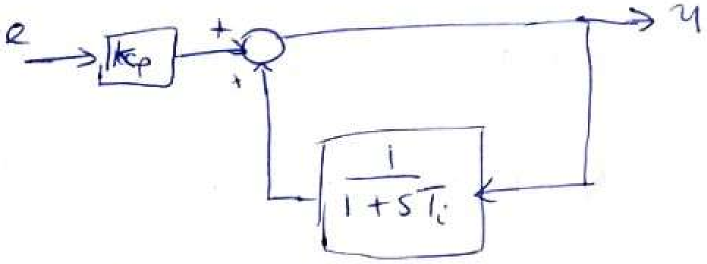
The above table will help in manual tuning of the controller. Note that trade offs exists because the effects of increasing/decreasing a parameter may conflict with the effect of another parameter. For instance, trying to reduce the rise time by increasing the value of k_p, will lead to an increment in the overshoot, which could be solved by increasing k_d, that in turn, could modify the rise time.

The selection of the parameters of a PID controller in this way is a trade-off between the response characteristics and it is the responsibility of the designer.

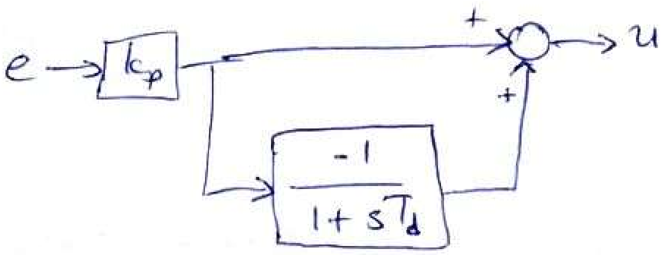
d

Block Diagram View of the Implementation of Integral and Derivative Action

(a) Integral action is here implemented using positive feedback with a first order system, sometimes called automatic reset



(b) Derivative action can be implemented by taking differences between a static system and a first-order system.



- The transfer function for the system (a) above can be determined thus:

$$u = k_p e + \left(\frac{1}{1+sT_i}\right) u$$

$$u - \frac{u}{1+sT_i} = k_p e \qquad = u\left(1 - \frac{1}{1+sT_i}\right) = u\left(\frac{sT_i}{1+sT_i}\right)$$

$$u\left(\frac{sT_i}{1+sT_i}\right) = k_p e$$

$$\frac{u}{e} = k_p\left(\frac{1+sT_i}{sT_i}\right) = k_p\left(\frac{1}{sT_i} + 1\right)$$

$$\frac{u}{e} = k_p\left(1 + \frac{1}{sT_i}\right)$$

$$= k_p + \frac{k_i}{s}, \quad \text{where } k_i = \frac{k_p}{T_i}$$

The transfer function of a PI (Proportional Integral Controller)

• The transfer function for the system (b) above can be determined thus:

$$u = k_p e + k_p e\left(\frac{-1}{1+sT_d}\right)$$

$$= k_p e\left(1 - \frac{1}{1+sT_d}\right) = k_p e\left(\frac{sT_d}{1+sT_d}\right)$$

$$\frac{u}{e} = \frac{k_p T_d s}{1+sT_d}$$

This is the transfer function of a filtered (first-order filter in this case) derivative

Example 11.1 PD action in the retina. (Åström & Murray)

(Derivative action ~~has~~ has a dampening effect on the system response, hence it reduces overshoot.)

The response of cone photoreceptors in the retina is an example where proportional and derivative action is generated by a combination of cones and horizontal cells. The cones are the primary receptors stimulated by light, which in turn stimulate the horizontal cells, and the horizontal cells gives inhibitory (negative) feedback to the cones.

The system can be modelled by ordinary differential equations by representing neuron signals as continuous variables representing the average pulse rate:

$$\frac{dx_1}{dt} = \frac{1}{T_c}(-x_1 - kx_2 + u) \; ; \quad \frac{dx_2}{dt} = \frac{1}{T_h}(x_1 - x_2)$$

where u is the light intensity and x_1 and x_2 are the average pulse rates from the cones and the horizontal cells.

* The step response of the system shows that the system has a large initial response followed by a lower, constant steady-state response typical of proportional derivative action.
  - Use parameters $k = 4$, $T_c = 0.025$ and $T_h = 0.08$ for your simulation.

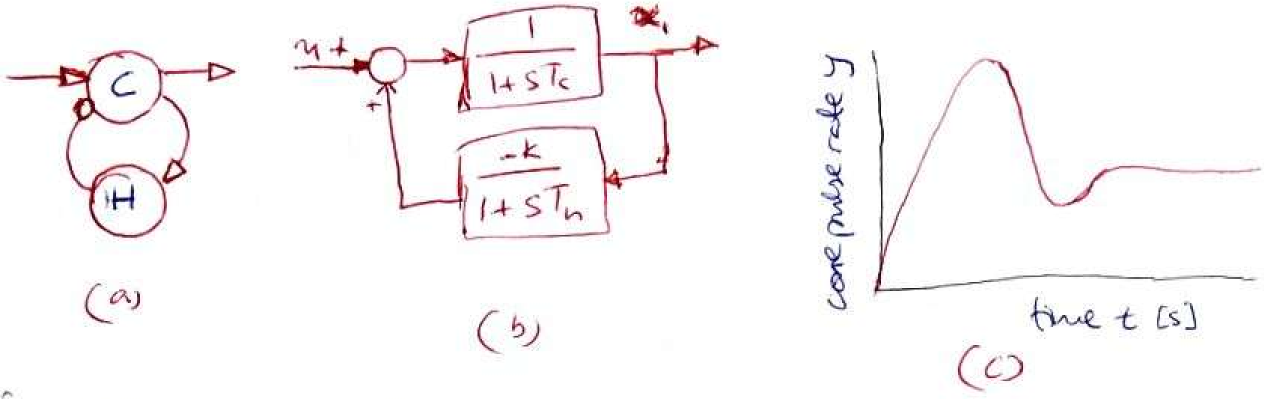


figure 11.4 (a) schematic diagram of cone photoreceptors (C) and horizontal cells (H) in the retina. Excitatory feedback is indicated by arrows and inhibitory feedback by circles.

(b) The block diagram of the system

(c) The step response of the system; showing what could be ~~approximates to~~ ~~overdamped~~ descibed as a damped oscillation.

> We could obtain the transfer function of the system:

$$\dot{x}_1 = \frac{1}{T_c}(-x_1 - k x_2 + u) \quad \text{gives} \quad sX_1(s) = \frac{1}{T_c}\left[-X_1(s) - kX_2(s) + U(s)\right] \quad \cdots (i)$$

$$\dot{x}_2 = \frac{1}{T_h}(x_1 - x_2) \quad \text{yields} \quad sX_2(s) = \frac{1}{T_h}\left[X_1(s) - X_2(s)\right] \quad \cdots (ii)$$

from (i) above, we have

$$T_c sX_1(s) + X_1(s) = -kX_2(s) + U(s)$$

$$(T_c s + 1)X_1(s) = -kX_2(s) + U(s)$$

$$X_1(s) = \frac{-kX_2(s)}{1+sT_c} + \frac{U(s)}{1+sT_c} = \frac{1}{1+sT_c}\left(-kX_2(s) + U(s)\right) \quad \cdots (iii)$$

from (ii)

$$sT_h X_2(s) = X_1(s) - X_2(s)$$

$$sT_h X_2 + X_2 = X_1 = (sT_h + 1)X_2$$

$$\therefore X_2(s) = \left(\frac{1}{1+sT_h}\right)X_1 \quad \cdots (iv)$$

* Observe how that the block diagram above (fig. 11.4 b) is constructed from eqns (iii) and (iv)

Combining (iii) and (iv)

$$x_1(s) = \frac{1}{1+sT_c}\left[\frac{-k\,x_1(s)}{1+sT_h} + u(s)\right] = \left(\frac{1}{1+sT_c}\right)\left(\frac{-k}{1+sT_h}\right)x_1(s) + \left(\frac{1}{1+sT_c}\right)u(s)$$

$$x_1 - x_1\left(\frac{1}{1+sT_c}\right)\left(\frac{-k}{1+sT_h}\right) = u\left(\frac{1}{1+sT_c}\right)$$

$$x_1\left[1 + \left(\frac{1}{1+sT_c}\right)\left(\frac{k}{1+sT_h}\right)\right] = \frac{u}{1+sT_c}$$

$$x_1\left[(1+sT_c) + \left(\frac{k}{1+sT_h}\right)\right] = u$$

$$x_1\left[(1+sT_c)(1+sT_h) + k\right] = u(1+sT_h)$$

$$\therefore \frac{x_1}{u} = \frac{1+sT_h}{(1+sT_c)(1+sT_h)+k}$$

$$= \frac{T_h s + 1}{T_cT_h s^2 + (T_c+T_h)s + 1+k} \equiv \frac{\frac{s}{T_c} + \frac{1}{T_cT_h}}{s^2 + \frac{(T_c+T_h)}{T_cT_h}s + \frac{(1+k)}{T_cT_h}}$$

- The system has a zero at $s = -\frac{1}{T_h}$

and poles at

$$s_1, s_2 = \frac{-(T_c+T_h) \pm \sqrt{(T_c+T_h)^2 - 4(T_cT_h)(1+k)}}{2T_cT_h}$$

given parameters $k=4$, $T_c = 0.025$, $T_h = 0.08$

the zero $= -\frac{1}{0.08} = -12.5$

$$\text{pole } s_{1,2} = \frac{-0.105 \pm \sqrt{0.011025 - 0.04}}{0.004} = \frac{-0.105 \pm j\sqrt{0.028975}}{0.004}$$

$$= \frac{-0.105 \pm j0.17}{0.004}$$

$$= -26.25 \pm j42.56$$

$$\frac{x_1}{u} = \frac{0.08s + 1}{0.002s^2 + 0.105s + 5}$$

$$= \frac{40s + 500}{s^2 + 52.5s + 2500}$$

Let us now rewrite the above highlighting the standard second-order system structure: $\frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}$

Hence,

$$\frac{x_1}{u} = \frac{40s + 500}{s^2 + 52.5s + 50^2} = \frac{\left(\frac{50^2}{50^2}\right)(40s + 500)}{s^2 + 52.5s + 50^2}$$

$$= \left(\frac{50^2}{s^2 + 52.5s + 50^2}\right)\left(\frac{1}{2500}\right)(40s + 500)$$

$$\therefore \frac{x_1}{u} = \left(\frac{50^2}{s^2 + 52.5s + 50^2}\right)(0.016s + 0.2)$$

The transfer function in the first set of parentheses above is in the standard second order form. We can see that the associated undamped natural frequency $\omega_n = 50$ and using

$$2\xi\omega_n = 52.5$$

$$\xi = \frac{52.5}{(2)(50)} = 0.525$$

we find that the damping coefficient is 0.525.

Note specially that we have the conventional response to the ~~first~~ transfer function in the first parentheses modified by the presence of a zero (due to the multiplication by $0.016s + 0.2$).

The zero, as found earlier is $s = -\frac{0.2}{0.016} = -12.5$

- The response will not be the same as for the standard second-order system because of the effect of the zero.

Effect of a Single zero on a Nominal Response

Given a zero described by $1+\frac{s}{z}$, where z could be either a positive or negative zero. Let $G(s) = Y(s)U(s)$ be the nominal system with no zeros ∴ the zero modified system can be described by

$$G_1(s) = \left(1+\frac{s}{z}\right)G(s) = G(s) + \frac{s}{z}G(s)$$

$$G_1(s) = Y_1(s)U(s) = Y(s)U(s) + \frac{s}{z}Y(s)U(s)$$

∴ the response with the real zero added is obtainable from

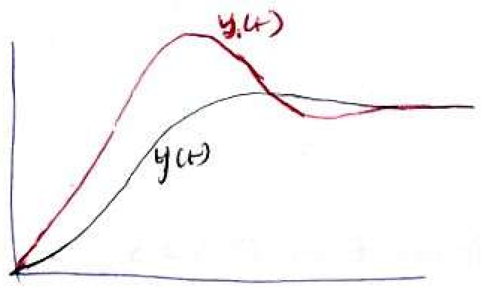
$$Y_1(s) = Y(s) + \frac{s}{z}Y(s)$$, taking inverse Laplace transforms

$$y_1(t) = y(t) + \frac{1}{z}\dot{y}(t)$$

- Observe that a term in the derivative of the nominal response that is modified by $\frac{1}{z}$ is added to the nominal response to get the real response. The contribution to the ~~full~~ response is a function of the slope of the nominal response.

Things to note: As z increases in magnitude, the effect of the zero decreases, and vice versa.
If the nominal response $y(t)$ arrives at a steady state such that $\dot{y}(t) = 0$, then the zero makes no contribution to the full response at those points.

> Effect of a Left-Half plane zero (i.e. the zero is negative) LHP

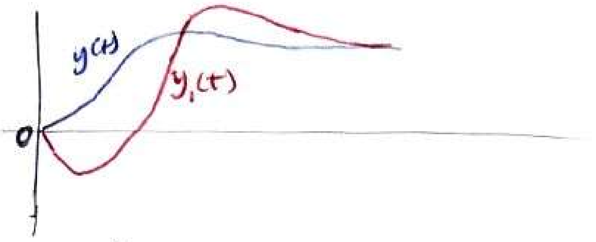


For a step input and assuming a stable system, a left-half-plane zero, in general,

- decreases the rise time
- increases the overshoot

Observe that the ~~two~~ plots cross where the slope of $y(t)$ is zero i.e. where $\dot{y}(t) = 0$. This will be the case regardless of the location of the zero (LHP or RHP.

> Effect of Right-Half-Plane (RHP) Zeros (i.e. the zero is positive)



- The effect of adding a RHP zero is to increase rise time (thus making the system slower) and it
- induces undershoot.

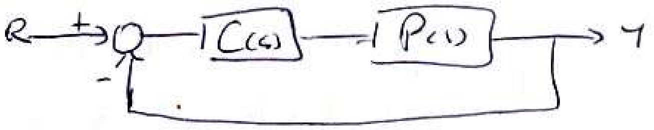
The two plots cross where the slope of $y(t)$ is zero as can be seen from $y_1(t) = y(t) + \frac{1}{2}\dot{y}(t)$ at $\dot{y}(t)=0$, $y_1(t) = y(t)$.

- For increasing numbers of RHP zeros, the size of the undershoot decreases
- It is known that the step response of a stable plant with n real RHP zeros will cross its starting value at least n times.

Assuming nominal systems with no zeros, the use of PI and PD controller will add a single real zero to the closed loop system if there are no pole/zero cancellations that make them cancel out a pole. The following examples will highlight this.

Example.

Given a closed loop system in the conventional unity feedback framework



where C(s) represents the controller and P(s) the system under control. Design a ~~[illegible]~~ simple filtered PD controller for the system $\frac{b}{s+a}$. Briefly indicate its potential for pole placement and choosing frequency response behaviour and and potential trade offs.

Let the PD controller be $C(s) = K + \frac{k_d s \omega}{s+\omega} = \frac{K(s+\omega) + k_d \omega s}{s+\omega}$

The plant $P(s) = \frac{b}{s+a}$.

$$= \frac{(K + k_d\omega)s + K\omega}{s+\omega}$$

Given the unity feedback framework given, the closed loop transfer function

$$\frac{Y(s)}{R(s)} = \frac{CP}{1+CP} = \frac{\left[\frac{(K+k_d\omega)s + K\omega}{s+\omega}\right]\left(\frac{b}{s+a}\right)}{1 + \left[\frac{(K+k_d\omega)s + K\omega}{s+\omega}\right]\left(\frac{b}{s+a}\right)}$$

$$= \frac{b(K+k_d\omega)s + bK\omega}{(s+\omega)(s+a) + b(K+k_d\omega)s + bK\omega}$$

$$= \frac{b(K+k_d\omega)s + bK\omega}{s^2 + as + \omega s + \omega a + b(K+k_d\omega)s + bK\omega}$$

$$\therefore \frac{Y(s)}{R(s)} = \frac{b(K+k_d\omega)s + bK\omega}{s^2 + s(a+\omega+bK+bk_d\omega) + \omega(a+bK)}$$

- We see immediately that this is like a nominal/standard second-order system modified by a zero
- Notice that we can choose the poles of the system arbitrarily with the ~~right~~ corresponding choice of K and k_d which are the controller gains, and a suitable value for ω (of the filter).
  - The undamped natural frequency of the nominal system can be determined by a choice of K, and then the choice ~~of~~ of damping coefficient can be determined by the choice of k_d given that ~~k_p~~ K has been chosen already. Do we see the tradeoff here?

    Note, however that we may choose k_d first, and then select a ~~k_p~~ K that suites the design specifications
- The zero is located at $s = \frac{-K\omega}{K + k_d\omega}$ (obtained by setting the numerator to zero)
  - Notice that the zero is a function of K and k_d and ω

  We see a potential tradeoff here since our choice of K, k_p and ω fixes the zero.

Example PD control

Assume a unity feedback control system with plant $\frac{b}{s+a} = \frac{2}{s+3}$

And where a PD controller is used with a filtered derivative

where the filter is given by $\frac{0.02}{s+0.02}$.

The PD control structure is $k_p + \frac{k_d N s}{s+N}$ where $\frac{N}{s+N}$ is the derivative filter.

The resulting closed loop system is second order given by

$$\frac{Y(s)}{R(s)} = \frac{b(k_p + k_d N)s + bNk_p}{s^2 + s(a + N + bk_p + bk_d N) + N(a + bk_p)}$$

$$= \frac{2(k_p + 0.02k_d)s + 2(0.02)k_p}{s^2 + s(3 + 0.02 + (2)k_p + (2)k_d(0.02)) + 0.02(3 + 2k_p]}$$

If we want the closed loop characteristic equation to have poles at $[-1, -1]$, this leads to a desired characteristic equation: $(s^2 + 1)(s+1) = s^2 + 2s + 1 = 0$

i. Determine k_p and k_d terms to achieve the desired characteristic polynomial.

ii. Where does the resulting k_p and k_d place the zero of the closed loop system

iii. Qualitatively indicate how this zero affects the nominal dynamics assuming no zeros initially.

Solution

(i) $3 + 0.02 + 2k_p + 0.04k_d = 2$

$0.06 + 0.04k_p = 1$

$$\boxed{k_p = -\frac{0.06}{0.04} = -1.5}$$

$3.02 - 3 + 0.04k_d = 2$

$0.02 + 0.04k_d = 2$

$$\boxed{k_d = \frac{2 - 0.02}{0.04} = 49.5}$$

(ii) The zero is derived thus:

$$2[-1.5 + (0.02)(49.5)]s + (0.04)(-1.5) = 0$$

$$s = \frac{0.06}{-1.02} = -0.059$$

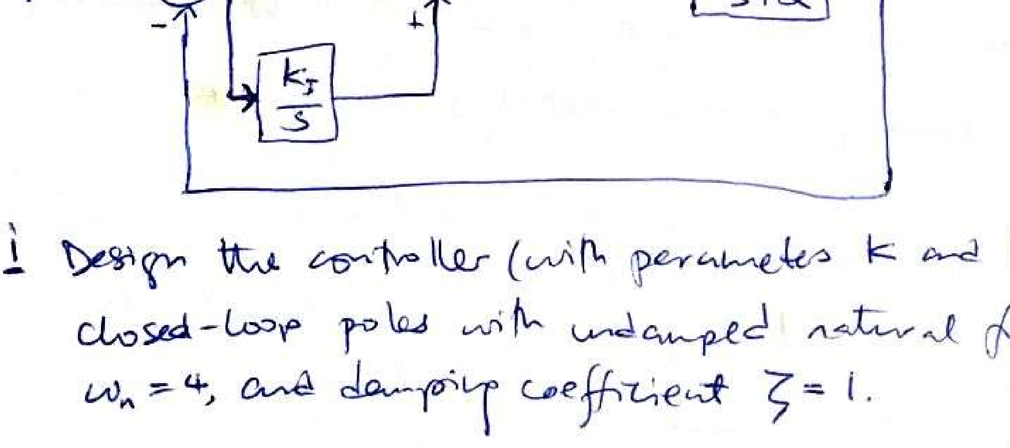
(iii) • The zero is in the left half-plane, therefore it will ~~cause~~ overshoot and reduce the rise time of the system relative to the nominal response for the second order system without a zero.

- We are certain of the overshoot because it is a critically damped characteristic equation (i.e. two real roots) that was the objective.

• Since the magnitude of the zero is relatively small, the overshoot will likely be relatively large and significant.

Example PI Control

Given the system configuration shown below where $k_\alpha = \alpha = 2$.



i Design the controller (with parameters K and k_I) to give closed-loop poles with undamped natural frequency $\omega_n = 4$, and damping coefficient $\zeta = 1$.

ii What poles are associated with the characteristic polynomial of the above system with the stated desired ω_n and ζ.

iii What is the resulting zero of the system given your answer to i above.

iv Sketch the unit step response of the standard second order system that has the same characteristic equation as for ii above (ie. assume no zeros).

v Sketch on the same plot (iv) above, the effect of the zero from iii on the response in iv.

Solution

i Desired characteristic polynomial given that $\zeta = 1$ and $\omega_n = 4$:-

$$s^2 + 2(1)(4)s + 4^2 = s^2 + 8s + 16$$

$$Y = K\left(\frac{2}{s+2}\right)\left(e + \frac{K_I e}{s}\right) = \left(\frac{2K}{s+2}\right)\left(\frac{s+K_I}{s}\right)e = \frac{2K(s+K_I)}{s(s+2)}$$

but $e = R - Y$

$$\Rightarrow Y = \left[\frac{2K(s+K_I)}{s(s+2)}\right](R - Y)$$

$$= \frac{2K(s+K_I)}{s^2+2s}R - \frac{2K(s+K_I)}{s^2+2s}Y$$

$$\Rightarrow Y + \frac{2K(s+K_I)}{s^2+2s}Y = \frac{2K(s+K_I)}{s^2+2s}R$$

multiplying through by s^2+2s

$$Y(s^2+2s) + 2K(s+K_I)Y = 2K(s+K_I)R$$

$$Y(s^2+2s+2Ks+2KK_I) = 2K(s+K_I)R$$

$$\frac{Y}{R} = \frac{2K(s+K_I)}{s^2+2s+2Ks+2KK_I} = \frac{2K(s+K_I)}{s^2+(2+2K)s+2KK_I}$$

Equating $s^2 + (2+2K)s + 2KK_I = s^2 + 8s + 16$

$$2 + 2K = 8$$

$$\therefore \boxed{K = 3}$$

$$2KK_I = 16$$

$$\therefore \boxed{K_I = \frac{8}{3} = 2.67}$$

ii $s^2 + 8s + 16$ has poles at $\frac{-8 \pm \sqrt{64-64}}{2}$

$\therefore s_1 = s_2 = -4$ same roots => a critically damped system

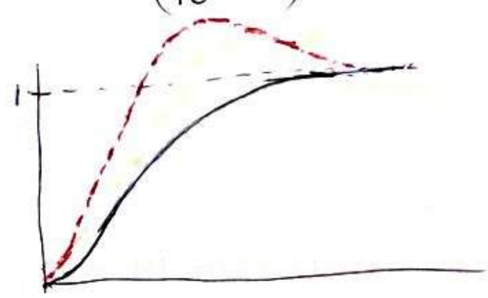
(iii) For the zero, $2k(s + k_I) = 0$

$$s + k_I = 0$$
$$s = -k_I$$
$$s = -2.67$$

iv $\frac{Y}{R} = \frac{2(3)(s+2.67)}{s^2 + 8s + 16} = \frac{2(3s+8)}{s^2+8s+16} = \frac{6s+16}{s^2+8s+16} = \frac{16(\frac{6s}{16}+1)}{s^2+8s+16}$

- The DC gain is $\frac{16}{16} = 1$; hence the final value to a unit step input is likewise 1.
- From ii above or noting that $\zeta = 1$, we know that the system is critically damped.

Hence the unit step response will have the shape below (see the solid line) assuming no influence from the zero (given by $(\frac{6s}{16}+1)$).



v See the above plot with red dashed line.

Time - Delayed Systems

Time-delayed systems are those that exhibit a [significant] reaction time, which is the most common and realistic situation for many biomedical systems. Apart from the small reaction time of devices like pumps or sensors, which is normally neglected, models of Biological processes, like ~~~~ insulin metabolism, may have significant delay.

In general, examples of time-delayed models are the reaction of patients to a particular substance or drugs, which can be expressed as a delayed first-order system:

$$G(s) = \frac{k_g e^{-Ts}}{\tau s + 1}$$

where k_g is the gain of the system, T is the delay, and τ is the time constant.

¿ Plot the system response to a unit step input for $k_g = 1$, $T = 0.5$, $\tau = 2$:-

$$G(s) = \frac{e^{-0.5s}}{2s + 1}$$

```
>> gs = tf (1, [2 1], 'IODelay', 0.5)
>> step gs
OR
>> s = tf('s')
>> gs2 = (exp(-0.5*s))/(2*s+1)
>> step gs2
```

- It is impossible to exactly represent e^{-Ts} as a transfer function, but approximations exist. One common method is called the Pade approximation. In Matlab, an approximate transfer function representation using the Pade approximation formula:

```
>> [num, den] = pade (T, N)
```

where T is the time delay and N is the order of the ~~~~ resultant system. The higher N is, the better the approximation.

* It should be obvious that we can apply the Ziegler Nichols ~~tuning rule~~ step response method tuning rules to the above first order ~~with~~ with time delay system.

Let us look at another tuning rule that may be used.

SIMC Tuning Rule (the delayed first-order system case)

We look at only first order systems with dead time (This method yields a PI controller with the first order case, ~~but~~ an a PID controller with the delay second-order system case).

- Only one parameter, τ_c, is needed to specify the required controller parameters. τ_c is the desired closed-loop time-constant.

Given a system $G(s) = \left(\frac{k}{\tau s + 1}\right) e^{-\theta s}$,

The SIMC tuned controller is given by

$$PI = k_c\left(\frac{T_i s + 1}{T_i s}\right) = \frac{k_c T_i s + k_c}{T_i s}$$

where $k_c = \left(\frac{1}{k}\right)\left(\frac{\tau}{\tau_c + \theta}\right)$

$T_i = \min\{\tau, 4(\tau_c + \theta)\}$ → choose the lesser ~~between~~ of the two terms i.e. choose τ if $\tau < 4(\tau_c + \theta)$ otherwise choose $4(\tau_c + \theta)$

Example SIMC tuning

let $G(s) = \dfrac{0.21 e^{-0.2754 s}}{1.4851 s + 1}$

Using SIMC tuning,

$$k_c = \left(\frac{1}{0.21}\right)\left(\frac{1.4851}{\tau_c + 0.2754}\right)$$

$$T_i = \min\{1.4851, 4(\tau_c + 0.2754)\}$$

In general, choosing $\tau_c = \theta$ provides good performance. However, the choice should depend on the control requirements.

Recall that τ_c is the desired or required closed-loop time constant.

Thus, setting $\tau_c = 0.2754$

we have $T_i = \min\{1.4851, 2.2032\}$

$$\therefore T_i = 1.4851$$

$$k_c = \frac{1}{0.21} \cdot \frac{1.4851}{0.2754 + 0.2754} = 12.8463$$

$\therefore$ The PI controller is given by $\dfrac{19.08s + 12.85}{1.485s}$

For the unity feedback system with controller and plant in series, the closed loop can be obtained via Matlab thus:

$G_{CL}(s) = \text{feedback}(\text{series}(PI(s), P(s)), 1)$

where $PI(s)$ represents the controller transfer function and $P(s)$ represents the system/plant under control.

¿ ~~Do~~ Ziegler-Nichols Tuning for the above system and compare the step responses.

Exercise (BCI, Ziegler-Nichols, SIMC, PI control)

A brain-computer interface (BCI) is a communication system that connects the human brain to an external device such that brain signals can be used to determine or affect the operation of the device.

The potential known as the P300 potential is a signal with a peak amplitude that is known to appear in the electroencephalography (EEG) approximately 300 ms after an unusual auditory or visual stimulus. This is used in an exogenous BCI system as they're typically designed to depend on the electrophysiological activity evoked by external stimuli.

Usually the user is exposed to a series of stimuli. Among these stimuli, there are a few that are related to the user's intention.

In this situation, the stimuli of interest, being infrequent and mixed with other much more common stimuli, cause the appearance of a P300 potential in the user's brain activity. This potential is observed mainly in the central and parietal areas of the cerebral cortex and can be registered by a simple non-invasive sensor. These features could be used to control different devices such as a wheelchair for patients with physical paralysis.

For the wheelchair case, the user ~~will~~ may set a desired speed for the system's real speed to achieve. The wheelchair speed is produced by a DC motor that can be modelled with a first-order system. The interpretation of the desired speed is obtained via the P300 signal. This signal is modelled as a delay signal. Hence, the open-loop system could be described by the following equation

$$G(s) = e^{-\theta s}\frac{k}{\tau s+1}$$

θ = time delay
τ = time constant
k = DC gain

where θ corresponds to the reaction delay to the stimuli, typically 300 ms. k and τ model the first-order model of a DC motor. (The parameters could be measured experimentally via a step response.) For this case, let $k = 0.5$ and $\tau = 0.1s$.

i A PI controller was implemented to control the system. Tune the controller using SIMC and Ziegler-Nichols tuning rules?

* We present the Matlab solution for SIMC here. The student should do the manual computations ~~for~~ SIMC and find the PI through Ziegler Nichols.

Matlab SIMC tuning

```
s=tf('s'); theta=0.3; % theta is P300 signal delay
tau=0.1; k=0.5;
num=k;
den=[tau 1];
G=tf(num,den,'InputDelay',theta)
 % SIMC PI
tau_c=theta; % set desired τc.
kc_SIMC=1/k*tau/(tau_c+theta);
Ti_SIMC= min(tau,4*(tau_c+theta));
PI_SIMC=kc_SIMC*(Ti_SIMC*s+1)/
        Ti_SIMC*s);
```

```
step(G)
hold on
Gc=feedback(G,1)
step(Gc)
Gc_SIMC=feedback(PI_SIMC*G,1);
step(Gc_SIMC)
```

This should yield:

$$PI_{SIMC} = 0.3333\left(1+\frac{1}{0.1s}\right)$$