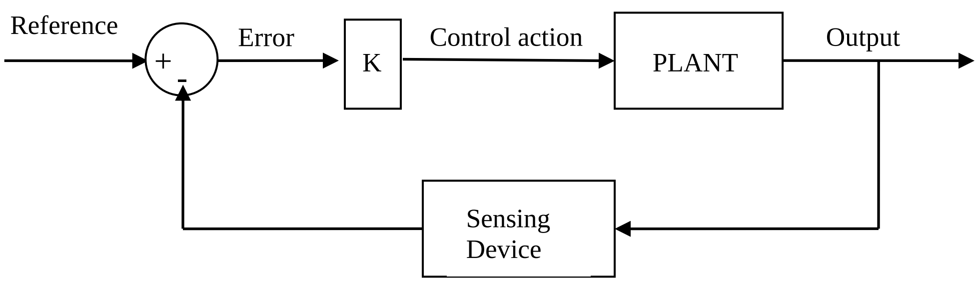
MMAN3200

PID Controllers

Until now, in MMAN3200, we mentioned that for controlling a SISO system we "close the loop" by applying negative feedback and a simple proportional controller.
(SISO: Single input - Single output)

In that approach, we simply compare the measured (or estimated) value of the system's output, and the desired value (aka "set point" or reference); that difference is then scaled by a factor K (the "gain"), for generating the control action, i.e. the signal which is actually fed to the input of the plant (the open loop system),

$$control(t) = K \cdot \left(y_{desired}(t) - y(t) \right) .$$



The resulting system is called Closed Loop System (CLS). Now, in place of directly setting the plant's input, the user simply defines the reference (which is the input of the CLS). The plant's input is hidden to the user, because the controller module is actually operating it. From the user's perspective, the system is a different one, it is the CLS, whose transfer function (TF) is what we call CLTF (Closed loop transfer function). That TF is evaluated by the following equation

$$G_{CL}(s) = \frac{K \cdot G_{OL}(s)}{1 + K \cdot G_{OL}(s)}$$

Tuning this controller would require choosing the value for the parameter K.
This basic controller would perform relatively well for treating certain class of Open Loop (OL) systems; however, for many cases it may not be adequate. Let's see some cases in which this approach (proportional controller) is a poor solution.
Consider the case of a system (we are talking about an OLS) whose dynamics has two clear dominant poles, which are real. The poles of the CLTF, due to applying a proportional controller, will depend on the value assigned to the proportional gain, K; those poles can be inferred by the associated Root Locus (we sketch this RL in class), for a range of possible values for K. (here, we obtain this RL, and discuss about it, in class.)

We can see the "right pole", initially (for low values of K) is displaced to the left, till it "reaches" the double point, then for higher gains K, the poles turn to be complex conjugate. Higher values for K would make the poles to have more relevant imaginary components. Based on this, we conclude that the best possible controller is limited to have poles close enough to that double point. (note: we say "close enough" because we cannot, usually, achieve those poles, exactly; due to diverse reasons which we briefly discuss in class).
The situation can be more critical, if we consider an OLS which is instable, e.g. having at least one pole on the right side of the s-plane. The PC may simply not achieve a stable CLS (no feasible K would result in stable CLTF poles). We consider the extreme case of a second order system whose two poles are on the unstable side of the s-plane. We can infer, from its associated Root Locus, that a proportional controller, independently of the setting we may choose for the parameter K, will never be able to control this system.

Consequently, due to that clear limitation of the proportional controller, we consider a family of controllers which introduce some dynamics. We express the dynamics of the controller by its TF in Laplace; the controller has a model **C(s)**, in place of simply being a multiplicative constant K.
In this case the error signal is filtered by the controller's module, i.e. if expressed in Laplace, the control action, applied to the open loop plant will be,

$$control[s] = C(s)\cdot\left(y_{desired}[s] - y_{actual}[s]\right).$$

(space for drawing)

We can analyze the performance of the controller, when applied to a certain plant $G(s)$, by analyzing the RL of the combined OLS, $C(s)\cdot G(s)$.

Now, for the case we had discussed before, in which the proportional controller was inadequate, we consider the case of a controller whose dynamics simply contains a zero and a gain. We fix the zero at certain negative real value, and we allow the gain K to be our adjustable parameter (as we usually do for evaluating the Root Locus). It is worth noting that we have the freedom to set two parameters of our controller, K and z; however, we decide to fix one (in this case z), and then we analyze the controller performance for a variety of values for K. In this way, we treat a 2DoF design problem by simplifying it to be a 1DoF one. The Root Locus approach allows us to easily analyze 1DoF cases.

The controller's TF is $C(s)=K\cdot(1+s/z)$, which, as we can appreciate, has a zero at s=-**z**. This is class of controller is a PD (Proportional + Derivative) controller, because it simply combines (linearly) a term which is proportional to the current error, and a second one which is proportional to the time derivative of that error,

$$control(t)=K_p\cdot e(t)+K_d\cdot\frac{d}{dt}e(t)=K\cdot\left(e(t)+\frac{1}{z}\cdot\frac{d}{dt}e(t)\right); \qquad z=\frac{K_p}{K_d}, \quad K=K_p$$

To see the effect of this type of controller, we see how the Root Locus, for the system previously mentioned, is dramatically modified.

We propose a zero, on the left side (real, negative). We can see that for certain values of the gain K, the associated CLS's poles do appear on the stable region of the s-plane; which means that, by using a PD controller with proper settings, we can stabilize this plant!

(we sketch the RL, here)

However, just by using a PD controller, although it may achieve "good" poles, it may not satisfy other performance aspects, which are not explicitly explained by the poles. A good example of it is the steady state error. Knowledge about the poles (CL or OL), does not provide enough information for inferring the stationary error.

For estimating the steady state error, of a stable system, we can exploit the Final Value Theorem, in the Laplace domain.

The theorem guarantees that if a time signal y(t) does not diverge, we can estimate its limit, for $t\to\infty$, by using its Laplace's equivalent function, in the following way:

$$\lim_{t\to\infty} y(t)=\lim_{s\to 0}\left(s\cdot y[s]\right).$$

We can exploit it, for analyzing the system response to certain typical input, e.g. a step function. We assume a step function of amplitude 1, as input (see note 1). This signal, if expressed in the Laplace domain, is $u[s]=\frac{1}{s}$. Consequently, the system's response (of a system whose TF is $A(s)$) to that step input will be $y(t)$, which in the Laplace domain is $y[s]=A(s)\cdot u[s]=G(s)\cdot\frac{1}{s}$. This signal, in the time domain, will converge to

$$\lim_{t\to\infty} y(t)=\lim_{s\to 0}\left(s\cdot y[s]\right)=\lim_{s\to 0}\left(s\cdot\frac{1}{s}\cdot A(s)\right)=A(s)\big|_{s=0}$$

(this is valid because we know that $G(s)$ is stable, so that $y(t)$ will not diverge; i.e. we are respecting that condition of the Final Value Theorem.)
For our CLS, we get the following limit,

$$\lim_{t\to\infty} y(t)=..=\lim_{s\to 0}\left(G_{CL}(s)\right)=\frac{C(s)\cdot G(s)}{1+C(s)\cdot G(s)}=....$$

In which we $G_{CL}(s)$ represents the TF of the CLS, and $G(s)$ is the OLTF.
(We complete this equation, in class.)

We can see there is certain stationary error, because $y(t)$ does not converge to 1 (does not "copy" the stationary value of the input, which is a step of amplitude = 1). We can conclude that that closed loop system, based on a PD controller, does not perform well in that sense; consequently, we will try a better controller for improving that matter.

We try a more sophisticated controller, in which we include an "integral term" (the original PD is augmented by adding that extra term), as follows,

$$control(t)=K_p\cdot e(t)+K_d\cdot\frac{d}{dt}e(t)\quad +K_i\cdot\int_{t=t_0}^{t} e(\tau)\cdot d\tau$$

which corresponds to a controller having the following TF,

$$C(s)=K_d\cdot s+K_p+K_i\cdot\frac{1}{s}=\frac{K_d\cdot s^2+K_p\cdot s+K_i}{s}=K_i\cdot\frac{\left(a\cdot s^2+b\cdot s+1\right)}{s}$$

$$\left(a=K_d/K_i,\quad b=K_p/K_i\right)$$

We express it in this way, because we will use the gain K_i as an adjustable parameter (it will take the role of the gain K in the Root Locus); the constants ***a*** and ***b***, will be chosen (and fixed) before we play with the Root Locus. We do it in this way, for converting the 3DoF analysis/design problem, in an 1DoF one; which can be treated using Root Locus.
When we decide to try a particular gain K_i we assume we will also adapt K_p and K_d for preserving the chosen coefficients ***a*** and ***b***.
This controller introduces a pole at ***p=0***, and two zeros, which are located on the left side of the plane (i.e. they always have negative real components), if the PID coefficients are all positive (usual configuration for a PID, $K_d\geq 0,\quad K_p>0,\quad K_i\geq 0$).

This new structure may slightly complicate our inference of the CLS' poles; however, it allows to eliminate the stationary error, because it introduces an OLS's pole at s=0.

Again, assuming we achieve a stable CLTF, we apply the Final Value Theorem to verify that the steady state error is eliminated (or a least well mitigated). We see the effect of that pole at s=0, in OLTF (that pole is the one which is introduced by the controller).

$$H(s) = C(s)\cdot G_{OL}(s) = \frac{N_C(s)}{s}\cdot\frac{N(s)}{D(s)}$$

$$G_{CL}(s) = \frac{H(s)}{1+H(s)} = \frac{N_C(s)\cdot N(s)}{s\cdot D(s)+N_C(s)\cdot N(s)}$$

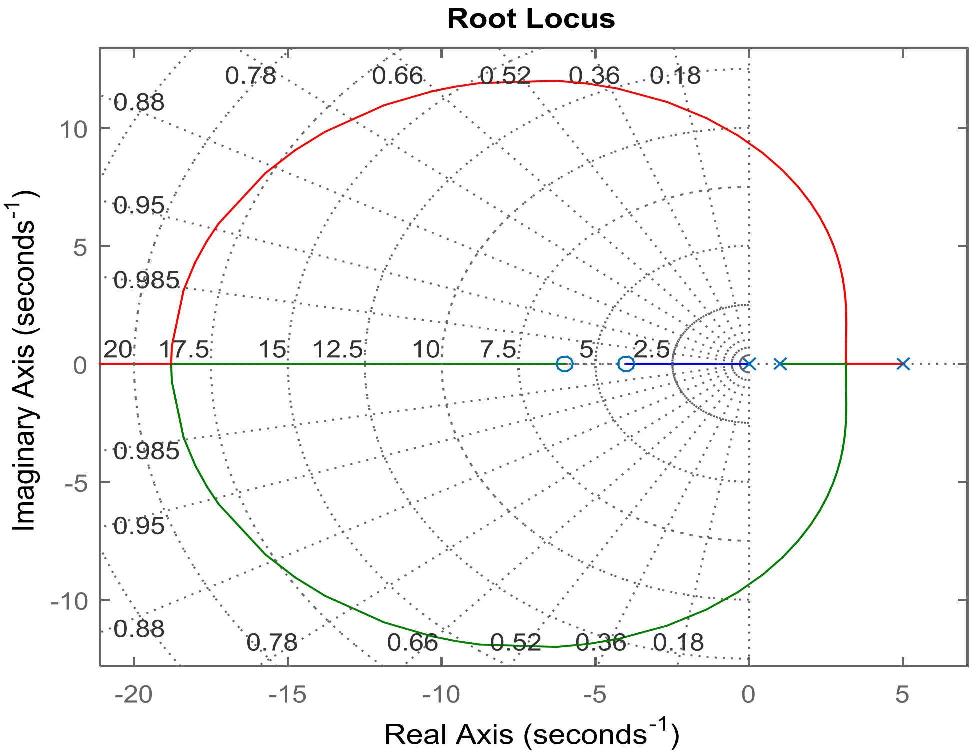
Considering the case in which the OLS $G_{OL}(s)$ does not contain a zero at s=0, then the limit is

$$\lim_{t\to\infty} y(t) = \lim_{s\to 0}\left(s\cdot y[s]\right) = \ldots = \lim_{s\to 0}\left(s\cdot\frac{1}{s}\cdot\frac{N_C(s)\cdot N(s)}{s\cdot D(s)+N_C(s)\cdot N(s)}\right) = \frac{N_C(0)\cdot N(0)}{0+N_C(0)\cdot N(0)} = 1$$

Which means we have compensated the steady state error, completely.
Again, this analysis is valid because we know that the closed loop system was stabilized by the controller (i.e. its poles are stable), so we can be sure the Final Value Theorem was properly applied.

Let see how the CLTF's poles are adjusted to satisfy that they are stable and "good". We propose the controller to introduce certain two negative real zeros. The controller's pole is always at s=0. We can sketch the Root Locus, easily by hand. However, I include, in this document, the plot from Matlab. In this example we have an OLTF which is unstable (has 2 poles which are at +1 and +5). The controller introduces two zeros (-4 and -6) and one pole at s=0.

(*Num1 = poly([-4,-6]) ; Den1 = poly([0,1,5]) ; tf1= tf(Num1,Den1); rlocus(tf1);*)



We can see, from that root locus sketch, that the originally unstable OLTF can be made stable, by using this controller and choosing a proper gain. Its three poles can even be made all real and negative, for a sufficiently high gain.

It may be the case in which the controller's zeros are complex conjugate (but still having negative real part), and not far from the real axis. The effect would be similarly adequate.
(we sketch that case, in class)

We conclude that we can still guarantee that, for certain gain K_i sufficiently high, the CLTF is stable, having adequate poles. In addition, we also know, the stationary error (for the Closed loop system trying to copy a step input) will be nil.

Next, to do in class: We will discuss about other typical cases in which a PID will be able to provide a solution which would not be achievable by applying a proportional controller.
We will also see how those controllers perform in the time domain (the domain in which the processes actually occur).

It is worth remarking an interesting fact, which we could see through the Root Locus analysis: even if there is certain discrepancy between the assumed OLTF (aka Nominal OLTF), a properly tuned PID would still perform well; i.e. the PID controller will be robust. This is a relevant matter, because the real plant is usually not perfectly known by us, because we approximate its model, or simply because we do not know, with certainty, some of its parameters.