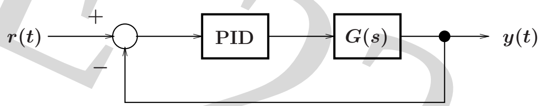
- Root locus helps us visualize what happens when we vary the gain, add poles, and add zeros.

  Is there a common approach to modifying the system behavior?



- How are systems usually designed? Dominant poles?

- Study proportional controller design.
  Then, look at proportional derivative controller design.

- P - proportional (for error tracking)
  D - derivative (for faster controller response)
  I - integral (to drive the error to zero)

- Performance specifications are given based on the step response of a second-order system.

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

ζ : damping factor
ω_n : natural frequency

- Step response.

$$Y(s) = G(s) \cdot \frac{1}{s}$$

$$y(t) = 1 - \frac{e^{-\zeta\omega_n t}}{\sqrt{1 - \zeta^2}} \sin(\omega_n \beta t + \theta),\ t \geq 0$$

where $\theta = \cos^{-1}\zeta$, for $0 < \zeta < 1$

- Specify performance parameters.
  $\Rightarrow$ determine appropriate ζ and ω_n.

- Some performance specifications.
  - T_s : settling time - time for the system output to settle within a certain % of the steady-state value.

    for $\pm 2\%$: $e^{-\zeta\omega_n T_s} < 0.02 \Rightarrow T_s \approx \frac{4}{\zeta\omega_n}$
  - T_p : time to peak.

$$T_p = \frac{\pi}{\omega_n\sqrt{1 - \zeta^2}}$$

  - M_p : peak overshoot.

$$M_p = 100 e^{\frac{-\zeta\pi}{\sqrt{1 - \zeta^2}}}$$

  - rise time, T_r and delay time, T_d (see text).

P / PD / PID Controller Design

- Giving performance specifications will determine what are the required ζ and ω_n, and the poles of the second-order system

$$s_{1,2} = -\zeta\omega_n \pm j\omega_n\sqrt{1 - \zeta^2}$$

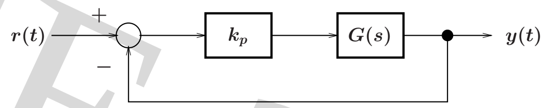
- Poles determine the response of a closed-loop system.

  To design a control system which matches the performance specifications, set

$$\text{dominant poles of the closed-loop system} = s_{1,2}$$

Proportional Controller

- Typical proportional controller.



- Design problem. Determine k_p such that the performance specifications are met.

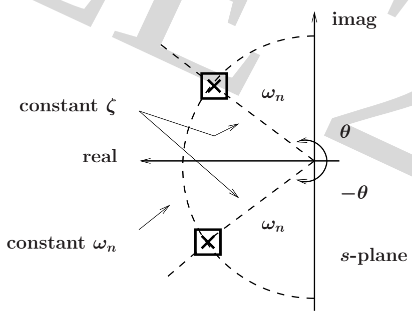
- Approach. Use root locus to find k_p such that the response due to the dominant roots (poles) meet the performance specifications.

Proportional Controller

- Second-order system.

  complex form : $s_{1,2} = -\zeta\omega_n \pm j\omega_n\sqrt{1 - \zeta^2}$

  polar form : $s_{1,2} = \omega_n \angle \pm \theta$



$$\theta = \tan^{-1}\left[\frac{\sqrt{1 - \zeta^2}}{-\zeta}\right]$$

Proportional Controller

- Example. Design a proportional controller for the following system

$$G(s) = \frac{s + 0.3}{1000s^2 + 80s + 1.5}$$

  such that the poles of the closed-loop system are located in the region similar to where the poles are of a second-order system with

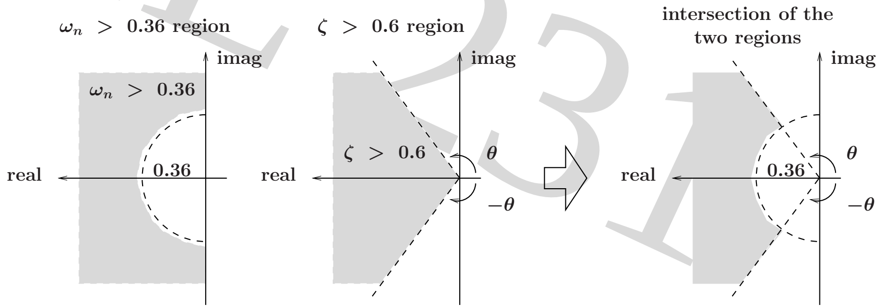
$$T_s < 18.5\ s \text{ and } M_p < 10\ \%.$$

- Using the equations for T_s and M_p, and solving for the parameters ζ and ω_n gives

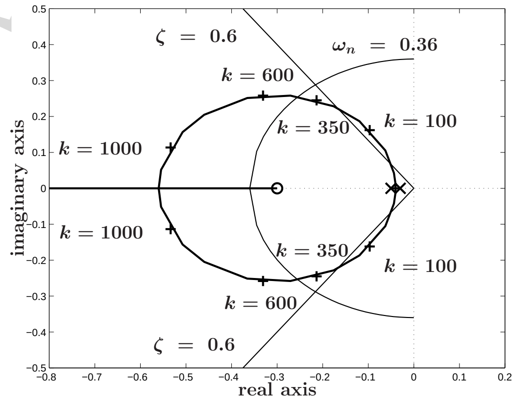
$$\zeta > 0.6 \text{ and } \omega_n > 0.36\ rad/s$$

- Different regions to consider.

$$\zeta = 0.6 \Rightarrow \theta = \tan^{-1}\left[\frac{\sqrt{1 - 0.6^2}}{-0.6}\right] = 126.87^o$$



- Use the root locus to find the gain k_p such that the dominant poles fall inside the desired region.



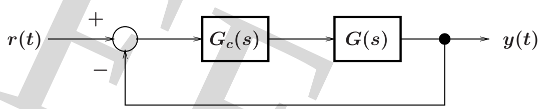
From the root locus, we can see that for about $k > 600$, the roots are inside the desired region.

Thus we can choose

$$k_p = 600$$

for our proportional controller design.

- Typical PD Controller.



$$G_c(s) = k_p + k_d s$$

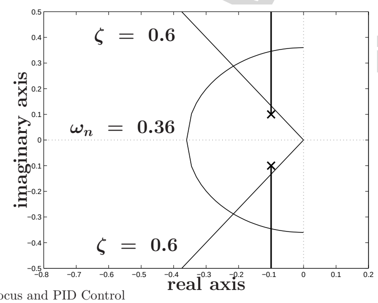
- Design problem. Determine k_p and k_d such that the performance specifications are met.

- Approach. Use root locus to find k_p and k_d such that the response due to the dominant roots meet the performance specifications.

- Why use a PD controller?

Consider the previous example but with the plant transfer function.

$$G(s) = \frac{1}{s^2 + 0.2s + 0.02}$$



From the previous example, we still have

$\zeta > 0.6$ and $\omega_n > 0.36\, rad/s$

The root locus will never enter the desired region for any value of k.

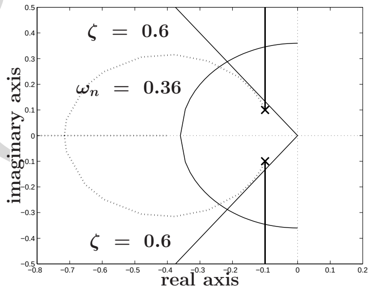
- Thus, if we try to use a P controller in our design, we will never find a k_p to satisfy the requirements.

  Proportional control is not sufficient to achieve our desired performance specifications.

  What can we do?

  If we can somehow bend the root locus to the left.

  We can introduce a zero somewhere on the negative real axis.



- Solution. Use a PD controller.

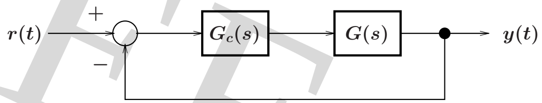
- The PD controller introduces a zero to the forward TF.

$$G_c(s) = k_p + k_d s = k_d\left(s + \frac{k_p}{k_d}\right)$$

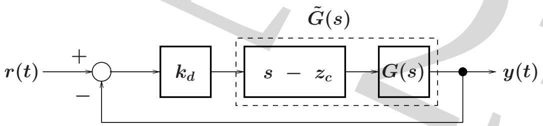
  Let $z_c = -\dfrac{k_p}{k_d}$.

- The zero z_c changes the root locus of the system.

- PD controller, a closer look.



$$G_c(s) = k_p + k_d s$$



$$\tilde{G}(s) = (s - z_c)G(s)$$

$$z_c = -\frac{k_p}{k_d}$$

- We can use the root locus for the design based on the characteristic equation : $1 + k_d\tilde{G}(s) = 0$.

- Example. Design an appropriate controller for the following system

$$G(s) = \frac{4}{s^2 + 0.2s + 0.02}$$

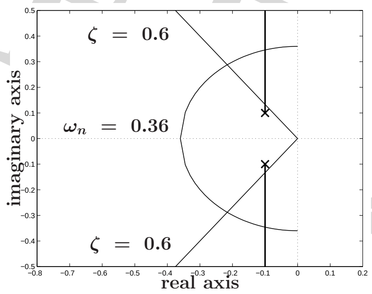
  such that the specifications in the previous example are satisfied.

- Since specifications are the same as the previous example, parameters ζ and ω_n are still

$$\zeta > 0.6 \text{ and } \omega_n > 0.36\ rad/s$$

Proportional Derivative Controller

- Looking at the root locus and the desired region,



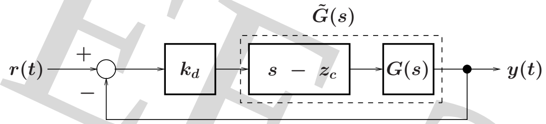
If we try to use a P controller in our design, we will never find a k_p to satisfy the requirements.

We can not satisfy the control requirements with a proportional controller.

- Try using a PD controller.

Proportional Derivative Controller

- Design the PD controller with $G_c(s) = k_p + k_d s$.



$$\tilde{G}(s) = (s - z_c)G(s)$$

$$z_c = -\frac{k_p}{k_d}$$

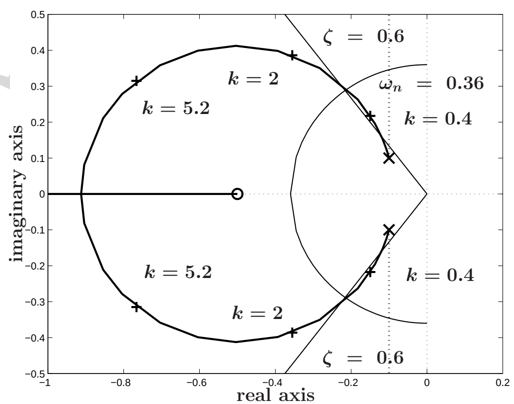
- We need to bend the root locus to the left.

From the simple rules, we can anticipate that adding a pole anywhere to the left of $s = -0.1$ will work.

$\Rightarrow$ choose $z_c = -0.5$.

Proportional Derivative Controller

- Root locus of $\tilde{G}(s)$.



From the root locus, we can see that for about $k > 0.5$, the roots are inside the desired region.

Thus we can choose

$$k_d = 2$$

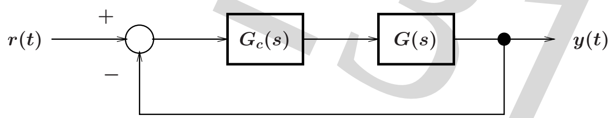
for our PD controller design.

Proportional Derivative Controller

- Then computing k_p using $z_c = -0.5$ and $k_d = 2$,

$$z_c = -\frac{k_p}{k_d} \Rightarrow k_p = 1$$

- Thus, our proportional derivative controller will be



$$G_c(s) = 1 + 2s$$

Proportional Integral Derivative Controller

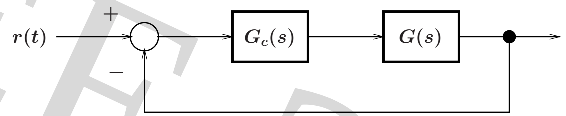
- Recall that adding an integrator increases system type.
  ⇒ drives the steady-state error to zero.

- However, the integrator also introduces a pole at the origin of the s-plane.
  ⇒ may lead to marginal stability.

- Use a PID (or PI) controller to avoid marginal stability problem but still get the benefit of having the integrator in the system.

Proportional Integral Derivative Controller

- Typical PID controller (PI controller for $k_d = 0$).



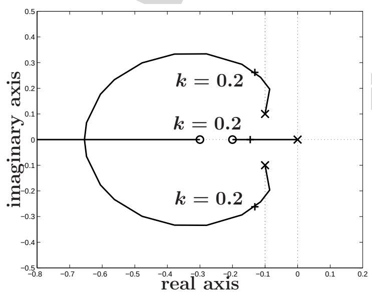
$$G_c(s) = k_p + k_d s + \frac{k_i}{s}$$

- Introduces at pole at the origin and two additional zeros.

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

Proportional Integral Derivative Controller

- Design using the PID controller.
  ⇒ choose one of the zeros of $G_c(s)$ to pull the locus from the pole at the origin towards the LHP.
  ⇒ choose the other zero of $G_c(s)$ to shape the locus of the original system as with a PD controller.



The original system has two poles at

$$p_{1,2} = -0.1 \pm j0.1$$

The zeros of $G_c(s)$ are located at

$$z_{c1,c2} = -0.2, -0.3$$

Summary

- Design systems by putting the dominant poles in the right locations.

- PID controller is the most common type of controller used in industry; the controller is suitable for most process control requirements.

  For most processes, PI or PD control is sufficient.

- The PID controller can be easily tuned by setting the constants k_p, k_d and k_i by using standard techniques or by trial and error.