SE 380 Winter 2013: Notes

Shale Craig

1. Control Engineering
   (a) Control engineering is creating control functions to regulate functions for known (and unknown) values.
   (b) Systems are made up of plants and controllers.
      i. Plants are the system we want to change. They are an 3-tuple of, related by $g(t)$
         A. Disturbance function: $d(t)$.
         B. Reference function: $u(t)$ [Generally a heaviside function, which we use convolution to add up]
         C. Output function: $y(t)$.
      ii. Controllers are systems we use to change the plant. Specified by $g(t)$.
   (c) Controller Design Cycle:
      i. Get Specs
         - Closed loop stability
         - Good steady state tracking
         - Disturbance rejection
         - Transient performance
      ii. Model the plant
         - Build a mathematical model of the plant
         - typically ≥ 1 differential equations.
         - Experiments are often used to determine the numerical values.
      iii. Obtain Transfer Function $G(s) = \frac{Y(s)}{U(s)}$ of the plant
         - Classical control requires we have a transfer function for the plant
      iv. Design the Controller
         - Controller is also a transfer function
         - Corresponds to the differential equation between control signal to the output.
      v. Simulate the Controller
      vi. Implement the Controller
         - Can build a system with a transfer function that matches the one designed
         - Realistically, the controller is implemented on a computer as a difference equation.
   (d) Control design goals:
      i. Tracking $y(t)$ to track $r(t)$
      ii. Disturbance Rejection: Disturbance $(d(t))$ should not significantly affect $y(t)$.
      iii. Robustness: Obtain "good" disturbance rejection & tracking despite plant model uncertainty.
2. Mathematically Modelling Systems - An Overview
   - Controller designs target simple & effective models.

- A model is an **imperfect** set of equations used to approximately represent a physical system
- Models for design are usually simpler than accurate (tradeoff)
- Statiscical models are experimentally derived
- Analytical models are scientifically (or mathematically) derived
- Use a modular approach to break a problem into smaller sub-sections.
- There are a number of ways to model systems:
  (a) Linear ODEs:

$$-u + V_R + V_c = 0$$

$$V_r = R_i,\ V_c = y,\ i = C\frac{dV_c}{dt}$$

$$\Rightarrow -u + RC\frac{dy}{dt} + y = 0$$

  (b) Transfer functions, obtained by the $\mathcal{L}$ of the ODE

$$-U(s) + sRCY(s) + Y(s) = 0$$

$$\Rightarrow \frac{Y(s)}{U(s)} = \frac{1}{RCs+1}$$

  (c) Convolution:
  If $G(s) = \frac{1}{\tau s+1}$, then we have:

$$g(t) = \mathcal{L}^{-1}\{G(s)\} = \frac{1}{\tau}e^{\frac{-t}{RC}}$$

  Thus, the entire system can be expressed as:

$$y(t) = g(t) * u(t) = \int_0^t g(t-\tau)u(\tau)d\tau$$

  (d) Bode Plots
  More detail next chapter.
  (e) State Models Made up of more than one first order ODEs

$$\dot{x}(t) = \frac{-1}{RC}x(t) + \frac{1}{RC}u(t)$$

$$y(t) = x(t)$$

- The main point of modelling is to obtain an approximate model of the system through analysis and experiments.
- ODEs, Transfer Functions, Convolution, and State Models are equivalent representations of systems.

3. State Models:

   - Knowing the state x at time $t = t_0$, the function $x(t_0)$ encapsulates all system dynamics at time t_0
   - That is to say, $\forall t_1 > t_0$, knowing $x(t_0)$ and the applied control $u(t)(\forall t_0 \leq t \leq t_1)$, we can compute $x(t_1)$ and hence $y(t_1)$
   - If $D(\dot{y}) = \dot{y}^2$, we can't (easily) find a transfer function, since it is non-linear.
   - In mechanical systems, use positions & velocities of each mass for state.
   - In electricla systems, use capacitor voltage & inductor currents of each mass for state.

4. Linearization:

   - In the case that $D(x_2)$ is not linear, we can change the basis of the system of equations to a way we can solve.
     We can represent the state of a plant, x as a vector as follows: (u is a constant)
     $$x_1 = y,\ x_2 = \dot{y} \Rightarrow x = \begin{bmatrix} y \\ \dot{y} \end{bmatrix}$$
     We can define a system of equations to solve these:
     The state equation:
     $$\dot{x} = \begin{bmatrix} x_2 \\ \frac{u}{M} - \frac{D(x_2)}{M} \end{bmatrix}$$
     The output equation:
     $$y = x_1$$
     Define a system of equations of the form:
     $$\dot{x} = f(x, u),\ y = h(x)$$
     The $\mathcal{L}$ doesn't work for equations that involve $\dot{y}^a$ where $a > 1$.
   - In the case that $D(x_2)$ is linear, we have: $D(x_2) = dx$. We can then say that:
     $$\dot{x} = f(x, u) = \begin{bmatrix} 0 & 1 \\ 0 & \frac{-d}{M} \end{bmatrix} x + \begin{bmatrix} 0 \\ \frac{1}{M} \end{bmatrix} u$$
     If we define $A = \begin{bmatrix} 0 & 1 \\ 0 & \frac{-d}{M} \end{bmatrix}$, $B = \begin{bmatrix} 0 \\ \frac{1}{M} \end{bmatrix}$ and $C = \begin{bmatrix} 0 & 1 \end{bmatrix}$ we have:
     $$\dot{x} = f(x, u) = Ax + Bu$$
     $$y = Cx$$
     The idea is that many states can be represented by the model $\dot{x} = f(x, u)$, and $y = h(x, u)$

- In this course we only have time-invariant systems, so this always holds true:

$$u \in \mathbb{R}^m,\ x \in \mathbb{R}^n,\ y \in \mathbb{R}^p \Rightarrow$$

$$\dot{x} = Ax + Bu : \ A \in \mathbb{R}^{n,n},\ B \in \mathbb{R}^{n,m} \text{ and}$$

$$y = Cx + Du : \ C \in \mathbb{R}^{p,n},\ D \in \mathbb{R}^{p,m}$$

- Virtually all systems are non-linear. Many can be modeled as $\dot{x} = f(x, u)$, $y = h(x, u)$.
  We always linearize non-linear systems by approximating a non-linear system with a linear system.
  The general idea is that we take the Taylor Series expansion of $f(x)$ at x_0, and keep only the $n = 0$ and $n = 1$ terms because they are linear.
  We can approximate our values by saying we have $f(x) \approx f(x_0) + f'(x_0)(x - x_0)$.
  In other words, $y - y_0 = f(x) - f(x_0) \approx f'(x_0)(x - x_0)$.
  If we define $\delta y = y - y_0$, $\delta x = x - x_0$, we have:

$$\delta y = f'(x_0)(x - x_0)\delta x \rightarrow \frac{\delta y}{\delta x} = f'(x_0)(x - x_0)$$

  As $x \rightarrow x_0$, this approximation becomes more accurate.
- Vector domain:

$$f(x, u) = f(x_0, u_0) + \frac{\delta f}{\delta x}\Big|_{x=x_0, u=u_0}(x - x_0) + \frac{\delta f}{\delta u}\Big|_{u=u_0, u=u_0}(u - u_0)$$

  We usually assume that $f(x_0, u_0) = 0$ i.e. it is at the origin.
  If we define $A = \frac{\delta f}{\delta x}\Big|_{x=x_0, u=u_0}$, and $B = \frac{\delta f}{\delta u}\Big|_{u=u_0, u=u_0}$, we have (again): (i.e. the Jacobians of the matrices)

$$f(x, u) = A(x - x_0) + B(u - u_0) = A\delta x + B\delta u$$

  We know $\delta\dot{x} = f(x, u)$, so we can substitute that into the above equation.
  Similarily, we can say that:

$$\begin{aligned} \delta y &= \frac{\delta h}{\delta x}\Big|_{x=x_0, u=u_0}\delta x + \frac{\delta h}{\delta u}\Big|_{u=u_0, u=u_0}\delta u \\ &= C\delta x + D\delta u \end{aligned}$$

  So overall:

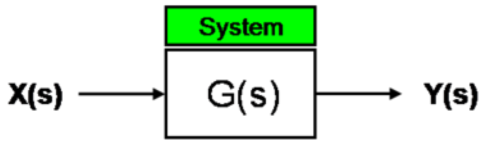
$$\begin{aligned} \delta\dot{x} &= A\delta x + B\delta u \\ \delta y &= C\delta x + D\delta u \end{aligned}$$

- Linear Model of Representation:
  We can take the laplace transform of a $f(t)$ defined for $t \geq t_0$.

  Then, $F(s) = \mathcal{L}\{f(t)\} = \int_0^\infty f(t)e^{-st}dt$.
  Note that $s = \sigma + j\omega$, so $e^{-st} = e^{-\sigma t}(\cos(\omega t) - j\sin(\omega t))$

- Radius of Convergence:
  The Radius of convergence is all points where $Re\{s\} > a$, where a is the largest pole of $G(s)$
- Transfer Functions:
  In a linear time invariant system (LTI) with the response function $g(t)$, $y(t) = g(t) * u(t)$

5. Transfer functions:



Blocks like this fill the form of:

$$y(t) = g(t) * x(t) \Leftrightarrow Y(s) = G(s)X(s)$$

(a) Getting a Transfer Function:
   i. Use theory to get equations to model the system
   ii. Find equilibrium
   iii. Linearize about equalibrium
   iv. Take the Laplace Transform of the linearized equations with zero initial conditions
   v. Solve for $Y(s)$ in terms of $U(s)$, and eventually get $G(s)$

(b) Terminology:

$$G(s) = \frac{N(s)}{D(s)}$$

   i. rational $\iff$ n and D are polynomials
   ii. proper $\iff$ $deg(D) \geq deg(N)$
   iii. strictly proper $\iff$ $deg(D) > deg(N)$
   iv. improper $\iff$ $deg(N) > deg(D)$
   v. poles are the roots of D
   vi. zeroes are the roots of N

(c) Obtaining the Transfer Function from the linear state model:

$$\dot{x} = Ax + Bu, \qquad y = CX + Du$$

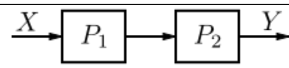
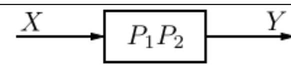
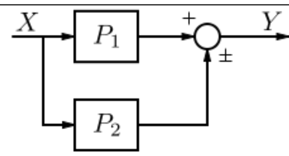
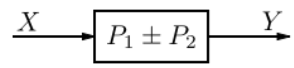
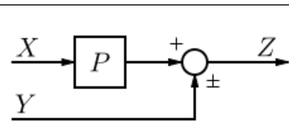
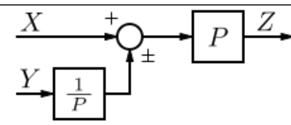
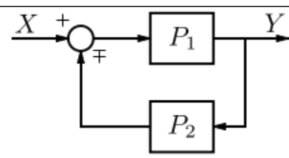
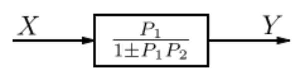
Laplace transforms with zero initial coordinates are:

$$\begin{aligned} sX(s) &= AX(s) + BU(s) \\ Y(s) &= CX(s) + DU(s) \\ (sI - A)X(s) &= BU(s) \\ X(s) &= (sI - A)^{-1}BU(s) \end{aligned}$$

By plugging this in to the function with G, we arrive at:

$$
\begin{aligned}
Y(s) &= CX(s) + DU(s) \\
\frac{Y(s)}{U(s)} = G(s) &= \frac{CX(s)}{U(s)} + \frac{DU(s)}{U(s)} \\
G(s) &= C(Is - A)^{-1}B + D \\
&\text{(“Colin is a bad dog”)}
\end{aligned}
$$

(d) Interconnections/Substitutions in Block Diagrams: (See table)

Name	Equation	From	To
Series	$Y = (P_1P_2)X$		
Parallel	$Y = P_1X \pm P_2X$		
Moving Blocks	$Z = PX \pm Y$		
Feedback	$Y = P_1(X \mp P_2Y)$		

(e) Strategies for substituting diagrams with simpler diagrams

i. Introduce variables representing the output of summing junctions.

ii. Annotate inputs to the summers.

iii. Create equations per summer

iv. Solve all equations into 1 equation in terms of $\frac{Y(s)}{U(s)}$

6. Linear System Theory

(a) If $u(t)$ is a real-valued signal then it is bounded if $\exists b \geq 0$ such that $\forall t \geq 0 : |u(t)| \leq b$

(b) We can define BIBO stability such that for all bounded input, we produce bounded output.

(c) $G(s)$ is BIBO stable, and $G(s)$ is rational and strictly proper $\iff$ the impulse response $g(t) = \mathcal{L}^{-1}\{G\}$ is absolutely integrable.

$$\int_0^\infty |g(t)|dt < \infty \Leftrightarrow \text{the system is BIBO stable}$$

(d) $G(s)$ is BIBO stable $\Leftrightarrow$ $G(s)$'s poles are complex numbers with $Re\{s\} < 0$.

(e) To prove that a given G isn't BIBO stable, we need to find a bounded input $u(t)$ that produces an unbounded $y(t)$.

7. Steady State Gain

(a) A stable transfer function with $u(t) = u_0$ (constant, so $U(s) = \frac{u_0}{s}$) will have a steady state gain of y_{ss}.

$$\frac{y_{ss}}{u_0} = \frac{\lim_{t\to\infty} y(t)}{u_0}$$

If G is stable, we can say:

$$y_{ss} = \lim_{t\to\infty} y(t) = \lim_{s\to 0} sY(s)$$

(b) Steady state gain of stable systems equal $G(0)$, since $U(s) = \frac{u_0}{s}$, and the s factors out, so:

$$\frac{y_{ss}}{u_0} = G(0)$$

8. Steady state output for $u(t) = \cos(\omega t)$:
   For $u(t) = \cos(\omega t)$, and for stable $G(s)$, we can say that: $y_{ss} = A\cos(\omega t + \theta)$

$$\begin{aligned}
u(t) &= e^{j\omega t} \\
y_{ss}(t) &= g(t) * u(t) = \left(\int_{-\infty}^{\infty} g(\tau)e^{-j\omega\tau}d\tau\right)e^{j\omega t} \\
\Rightarrow y_{ss}(t) &= G(j\omega)e^{j\omega t} \\
&= |G(j\omega)|e^{j\underline{/G(j\omega)}}e^{j\omega t} \\
&= |G(j\omega)|e^{j\left(\underline{/G(j\omega)}+\omega t\right)} \\
\Rightarrow y_{ss}(t) &= \frac{G(j\omega)e^{j\omega t}}{2} + \frac{G(-j\omega)e^{-j\omega t}}{2} \\
&= Re\{G(j\omega)e^{j\omega t}\} \\
y_{ss}(t) &= |G(j\omega)|\cos(\omega t + \underline{/G(j\omega)})
\end{aligned}$$

So, $A = |G(j\omega)|$, and $\theta = \underline{/G(j\omega)}$

y_{ss} is a sin.

We define the frequency response of G as $G(j\omega)$

9. Bode plots:

(a) Bode plots give us an idea of how a system reacts to different kinds of input.

(b) Two kinds of plots are usually analyzed at the same time:

- Magnitude plots compare the magnitude of the output (measured in dB) v.s. the frequency of the input:

$$20\log|G(j\omega)| \text{ vs } \log(\omega)$$

- Phase plots compare the angle (delay) of the output v.s. the frequency of the input:

$$\underline{/G(j\omega)} \text{ vs } \log(\omega)$$

(c) We need this data to sketch Bode plots:
- Pure Gain
- Poles
- First and Second Order Terms
- Delays

(d) Because $\log(|AB|) = \log(|A|) + \log(|B|)$ (and the same for angle), bode plots are additive.

$$\log\left(\Pi_q(\text{simple term}_q)\right) = \Sigma_q \log\left(\text{simple term}_q\right)$$

We can then break these into a number of different cases which sum together: (refer to table)
Remember: negative real poles/zeroes start out at ±180, not zero

Term	Magnitude	Phase
Constant (K)	$20\log_{10}(\lvert K\rvert)$	$K < 0?0 : \pm 180$
Pole at Origin $\left(\frac{1}{s}\right)$	-20 dB/dec, $20\log(\lvert G(j\omega)\rvert) = 0$ when $\omega = 1$	-90°
Zero at Origin (s)	20 dB/dec, $20\log(\lvert G(j\omega)\rvert) = 0$ when $\omega = 1$	90°
Real Pole $\left(\frac{\omega_0}{s+\omega_0}\right)$	Asymptote at 0dB, goes to -20dB/dec at ω_0	Asymptotes at 0°and -90° connect from $0.1\omega_0$ to $10\omega_0$
Real Zero $\left(\frac{s+\omega_0}{\omega_0}\right)$	Asymptote at 0dB, goes to +20dB/dec at ω_0	Asymptotes at 0°and +90° connect from $0.1\omega_0$ to $10\omega_0$
Underdamped Poles $\left(\frac{\omega_0^2}{s^2+2s\zeta\omega_0+\omega_0^2}\right)$, $0 < \zeta < 1$	Asymptote at 0dB, another (non-const) asymptote at -40dB/decade. Generally, they intersect at ω_0, but if $\zeta > 0.5$, there is a peak at: $\lvert G(j\omega_0)\rvert = -20\log_{10}(2\zeta)$	Low frequency asymptote at 0°and -180° connect from $\frac{\omega_0}{5^\zeta}$ to $\omega_0 5^\zeta$
Underdamped Zeros $\left(\frac{1}{\omega_0^2}(s^2 + 2s\zeta\omega_0 + \omega_0^2)\right)$, $0 < \zeta < 1$	Asymptote at 0dB, another (non-const) asymptote at 40dB/decade. Generally, they intersect at ω_0, but if $\zeta > 0.5$, there is a peak at: $\lvert G(j\omega_0)\rvert = 20\log_{10}(2\zeta)$	Low frequency asymptote at 0°and 180° connect from $\frac{\omega_0}{5^\zeta}$ to $\omega_0 5^\zeta$
Delay $\left(e^{-j\omega_0 t} = \cos(\omega_0 t) - j\sin(\omega_0 t)\right)$	"Flatlined" at 0dB	Grows linearly with time, but log plot makes it look curved.

The general strategy for drawing bode plots is:

i. Rewrite $G(s)$ in the proper form (multiplication of simpler terms)
ii. Separate the transfer function into constituent parts.
iii. Draw the bode diagram of each part.
iv. The overall bode diagram is the sum of all parts.
v. Rejoice! You are done the question.

(e) Time response of 1^{st} order ODEs:
Recall, first-order systems are the form of:

$$\tau\dot{y} + y = Ku \Rightarrow^{\mathcal{L}} G(s) = \frac{K}{1 + s\tau}$$

i. We measure the output settling to the asymptote (usually y_{ss}) in response to a step function.
ii. We define bandwidth as τ^{-1}.
iii. We measure the time response as a comparison between y_{ss} and an initial state of 0.
iv. As τ increases, the time response becomes faster.
v. We define the settling time as when the output settles to 2% of the asymptote.
vi. There is no peaking (or "overshoot") in first-order systems.
vii. As τ approaches 0, the pole $s = -\tau^{-1}$ moves to the left.
viii. Settling time $= 4\tau$
ix. TL;DR:
   A. Stable at $\tau > 0$.
   B. Pole at $s = \frac{-1}{\tau}$.
   C. Bandwidth $\approx \frac{1}{\tau}$.
   D. No overshoot or oscillation in step response.
   E. Settling time $= 4\tau$

(f) Time response of 2^{nd} order ODEs: $(\forall\zeta)$
Recall, second-order systems can be written in the form:

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

i. Poles are at $s = -\zeta\omega_n \pm \sqrt{\zeta^2 - 1}$
ii. Pole behaviour:
   A. $0 < \zeta < 1$: complex conjugate : under-damped
   B. $\zeta = 1$: 2 real repeated poles : critically damped
   C. $\zeta > 1$: 2 real distinct poles : over-damped
iii. Angle: $\theta = \cos^{-1}(\zeta)$.
iv. ζ is the damping ratio.
v. ω_n is the undamped natural frequency, and the bandwidth.
vi. As ω_n incresases (larger bandwidth), the time response is faster.

i. $0 < \zeta < 1$ (underdamped systems):
   - Impulse response:

$$\begin{aligned} g(t) &= \mathcal{L}^{-1}\{G(s)\} \\ &= \frac{k\omega_n}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n\tau} \sin(\omega_n\tau\sqrt{1-\zeta^2}) \end{aligned}$$

   - Step Response: ($u(t)$ = heaviside)

$$\begin{aligned} y(t) &= \mathcal{L}^{-1}\left\{G(s)\frac{1}{s}\right\} \\ &= \text{something very complicated} \end{aligned}$$

   - Poles are complex conjugates.
   - As ζ approaches 1, the time response oscillates less. Vice versa holds.
   - Zeros: none
   - Magnitude of poles: $\cdots = \omega_n$.

- DC gain: $G(0) = k$
- Frequency response: Bandwidth $\approx \omega_n$

ii. $\zeta > 1$ (overdamped systems):

$$G(s) = \frac{kab}{(s+a)(s+b)}$$

- Impulse response:

$$g(t) = \mathcal{L}^{-1}\{G(s)\} = \frac{kab}{b-a}(e^{-a\tau} - e^{-b\tau})$$

- Step Response: ($u(t) = \text{heaviside}$)

$$y(t) = \mathcal{L}^{-1}\left\{G(s)\frac{1}{s}\right\} = k\left(1 + \frac{1}{b-a}(ae^{-bt} - be^{-at})\right)$$

- Zeros: none
- DC gain: $G(0) = k$
- 2 real distinct poles
- As ζ approaches 1, the time response oscillates less. Vice versa holds.
- no overshoot, no oscilations
- bandwidth $= \min(a, b)$

iii. $\zeta = 1$ (critically damped systems):

$$G(s) = \frac{ka^2}{(s+a)^2}$$

- Poles: $s = -\zeta\omega_n = -\omega_n$
- Zeros: none
- slower than overdamped/underdamped systems to converge
- DC gain: $G(0) = k$
- 2 real & identical poles
- As the bandwidth (ω_n) increases, the time response is faster.
- no overshoot, no oscilations
- bandwidth $= a$

(g) Characteristics of a step response

i. Common metrics apply to any system when we evaluate it.

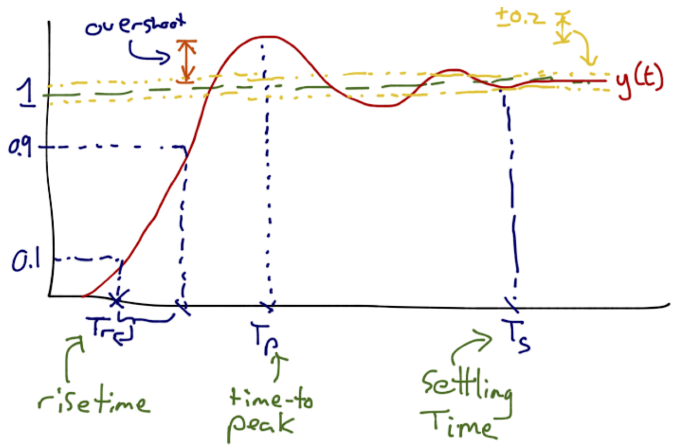
ii. Higher order systems can be approximated as 2nd order systems.

iii. Overshoot ($\%OS$):

- Only underdamped systems have overshoot.
- We find the first $\dot{y} = 0$, and plug that into $y(t)$
- Generally, we have:

$$\%OS = \frac{M_p - K}{K} = \exp\left(\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}\right)$$

iv. Settling Time (T_s)



- Amount of time it takes for a response to lie within 2% of y_{ss}
- Crude estimate: $t \geq \frac{4}{\zeta\omega_n}$

$$T_s = \frac{4}{\zeta\omega_n}$$

v. Time To Peak (T_p)
- Time it takes to reach the max value.
- Only depends on imaginary part of the poles:

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

vi. Rise Time (T_r)
- Time it takes to go from 10% to 90% of y_{ss} from 0, valid for $0.3 \leq \zeta \leq 0.8$

$$T_r \approx \frac{2.16\zeta + 0.6}{\omega_n}$$

Space intentionally left blank. [didn't take notes the day before the midterm]

10. Adding Zeros:

- Adding a minimum phase Zero:

$$\begin{aligned} G_a(s) &= G(s)(1+\tau s), \tau > 0 \\ &= \frac{k\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \end{aligned}$$

TODO: insert graph of this from class. We use $\tau > 0$ so that $\frac{-1}{\tau} < 0$ (i.e. the introduced zero is in the left-half plane)

- Step response:

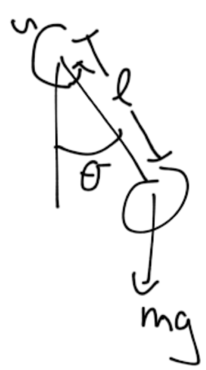
$$\begin{aligned} y(t) &= \mathcal{L}^{-1}\left\{\frac{G_a(s)}{s}\right\} \\ &= \mathcal{L}^{-1}\left\{\frac{G(s)}{s} + \frac{\tau s G(s)}{s}\right\} \\ &= (\text{step response of } G(s)) + \tau(\text{impulse response of } G(s)) \end{aligned}$$

As $\tau \to 0$, $G_a(s)$ approaches the step response of $G(s)$.
Likewise, as $\tau \to \infty$, $G_a(s)$ approaches the impulse response of $G(s)$.
This makes the phase plot more positive, and the magnitude plot "rolls off" slower. The bandwidth increases.
In all, the system becomes faster, but with the penalty of overshoot.
Knowing these attributes when designing PID controllers helps us understand that we want more 0's.

11. Feedback Control.

- ... Notes in PDF
- The most fundamental spec is stability.
  Good performance usually requires high gain but can also lead to instability.
- Two approaches to design/analysis:
  (a) Classical
  This course, freq domain, specs are based on closed-loop bandwidth, stability, margins. Design done with Bode plots.
  (b) Modern:
  State Space (ECE 488) In time domain, specs are in terms of pole locations.

  The approaches are complementary.
  (a) Classical: good SISO (single input, single output) systems, and open loop stable plants.
  (b) Modern: MIMO systems, and unstable plants
- Example:



Since we want to control position, I take the output $y = \Theta$. $y = x_1 = \Theta$
The model is (Problem Set 2): (J is the inertia of the system?)

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = \frac{-mgl\sin x_1}{J} + \frac{u}{J}$$
$$y = x_1$$

The upright position is $x_1 = \pi$. The equilibrium corresponding to this is found by solving:

$$0 = f(x_0, u_0)$$
$$y = h(x_0, u_0)$$

For $x_0 = (x_{01}, x_{02})$, we get (problem set 2):

$$x_0 = [\pi, 0]$$
$$u_0 = 0$$

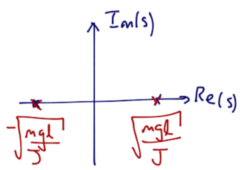
Let $dx = x - x_0$, $dy = y - y_0$, $du = u - u_0$.
The linearized model is (Problem set 3):

$$dx(dot) = \begin{bmatrix} 0 & 1 \\ \frac{mgl}{J} & 0 \end{bmatrix} dx + \begin{bmatrix} 1 \\ \frac{1}{J} \end{bmatrix} du$$
$$dy = \begin{bmatrix} 1 \\ 0 \end{bmatrix} dx$$

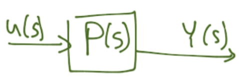
Let $Y(s) = \mathcal{L}\{dy\}$, $U(s) = \mathcal{L}\{du\}$.

$$\frac{Y(s)}{U(s)} = \frac{1/J}{s^2 - \frac{mgl}{J}} = P(s)$$
(problem set 3)

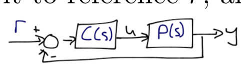
The poles are $s = \pm\sqrt{\frac{mgl}{J}}$



This implies that the plant is unstable.
The block diagram so far.



Let's try and stabilize the system by feeding back the position (using an encoder) and comparing it to reference r, and using the error $r - y$ to drive our controller



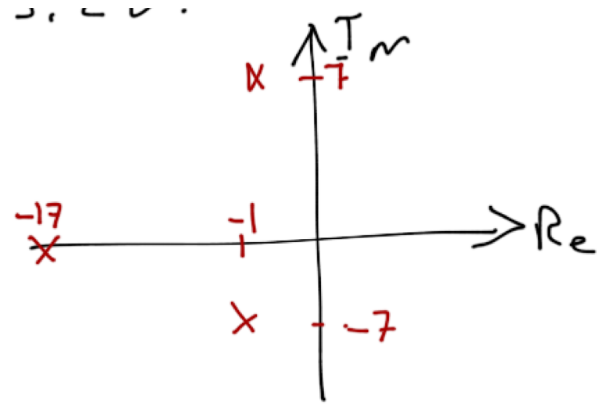
One controller that works is (for simplicity, assume $J = 1$, $mgl = 1$)

$$C(s) = 100\frac{s + 10}{s + 20}$$

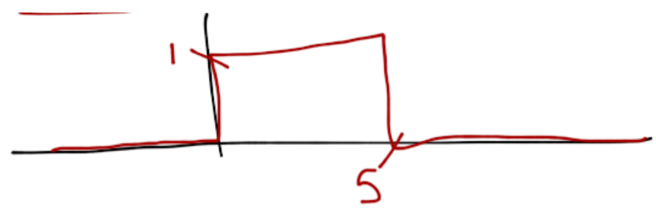
Called a lead controller
With this $C(s)$, the TF from R to Y is:

$$\frac{Y(s)}{R(s)} = \frac{100s + 1000}{s^3 + 20s^20s + 99s + 980}$$

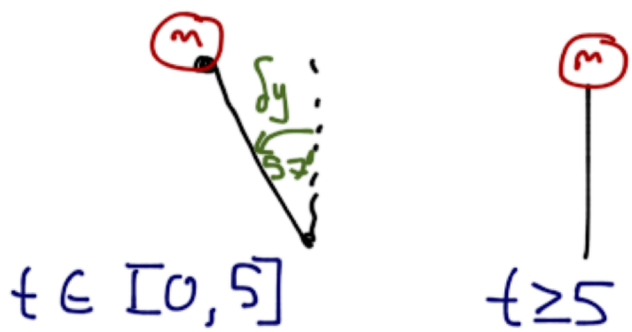
The poles of this transfer function are $\{-17.5, -1 + j7.3, -1 - j7.3\}$



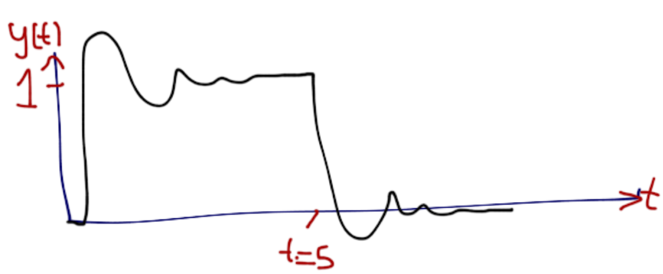
So by CH3, the system is BIBO stable. (all poles are in LHP).
Based on CH4, we expect a very oscillatory step response. The pole at $s = -17$ will have very little effect.
Let's apply $r(t) = 1(t) - 1(t-5)$:



$\rightarrow$ move pendulum to $dy = 1$ for $t \in [0, 5]$ (56 °), then back to upright position $dy = 0$ for $t > 5$.



The step response (he used Matlab):



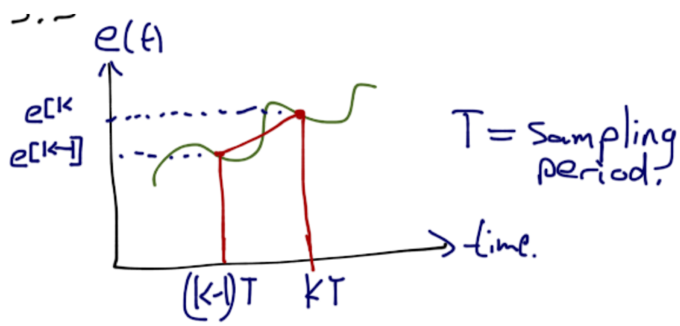
12. Closing the loop (a detailed example)

If $E(s) = R(s) - Y(s)$, then $\frac{U(s)}{E(s)} = C(s) = 100\frac{s+10}{s+20}$

$$\dot{u} + 20u = 100\dot{e} + 1000e$$

This means that this ODE needs to be implemented in software. Typically done by discretizing and approximating the ODE.

The most common way to implement the ODE is to use the "trapezoidal approximation"



$$\int_{(k-1)\tau}^{k\tau} e(\tau)d\tau \approx \frac{T}{2}\left(e[k] + e[k-1]\right)$$

This generates a "rule":

$$s = \frac{2}{T}\frac{z-1}{z+1}$$

Where s is the laplace variable, and z is the z-transform variable.

In this case we get:

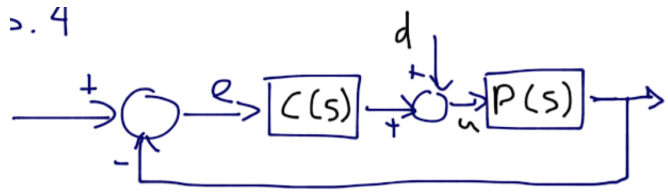
$$\begin{aligned}\frac{U[z]}{E[z]} &= C[z] \\ &= C(s)\Big|_{s=\frac{2}{T}\frac{z-1}{z+1}} \\ &= \frac{100((1+5T) + (5T-1)z^{-1})}{((1+10T) + (10T-1)z^{-1})}\end{aligned}$$

Taking the inverse z-transform, we get:

$$(1+10T)u[k] + (10T-1)u[k-1] = 100(1+5T)e[k] + 100(5T-1)e[k-1]$$

If $T > 0$ is small, the controller works well.

13. Feedback stability:



Where $P(s)$ is the plant
Where $C(s)$ is the controller
$r(t)$ is the reference
$e(t)$ is the tracking error
$d(t)$ is the disturbance
$u(t)$ is the control input
$y(t)$ is the output

6 Transfer functions associated with this system $(r, d) \to (e, u, y)$

(a) $\frac{R}{E} = \frac{1}{1+PC}$
(b) $\frac{D}{E} = \frac{-P}{1+PC}$
(c) $\frac{R}{U} = \frac{C}{1+PC}$
(d) $\frac{D}{U} = \frac{1}{1+PC}$
(e) $\frac{R}{Y} = \frac{PC}{1+PC}$
(f) $\frac{D}{Y} = \frac{P}{1+PC}$

We make the standing assumption that $C(s)$, $P(s)$ are rational, and $C(s)$ is proper, and $P(s)$ is strictrly proper, and the 2x2 matrix below has a non-zero determinant.

Equations at the output summers:

$$E = R - PU$$
$$U = D + CE$$
$$\begin{bmatrix} 1 & P \\ -C & 1 \end{bmatrix} \begin{bmatrix} E \\ U \end{bmatrix} = \begin{bmatrix} R \\ D \end{bmatrix}$$

In light of our assumptions, we have the following:

$$\begin{aligned} \begin{bmatrix} E \\ U \end{bmatrix} &= \begin{bmatrix} 1 & P \\ -C & 1 \end{bmatrix}^{-1} \begin{bmatrix} R \\ D \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{1+PC} & \frac{-P}{1+PC} \\ \frac{C}{1+PC} & \frac{1}{1+PC} \end{bmatrix} \begin{bmatrix} R \\ D \end{bmatrix} \end{aligned}$$

The output is:

$$\begin{aligned} Y &= PU \\ &= \frac{PC}{1+PC} R + \frac{D}{1+PC} D \end{aligned}$$

We can define the system as **feedback stable** if for every bounded r and d, we have bounded (u, e, y).

Since whenever r and e are bounded, $y = r - e$ is also bounded, it suffices to look at the 4 Transfor functions from $(r, d) \to (u, e)$.

Example:

$$P(s) = \frac{1}{s^2 - 1}$$
$$C(s) = \frac{s-1}{s+1}$$

The 4 transfer functions are:

$$\begin{bmatrix} \frac{(s+1)^2}{s^2+2s+2} & \frac{s+1}{(s+1)(s^2+2s+2)} \\ \frac{(s+1)(s-1)}{s^2+2s+2} & \frac{(s+1)^2}{s^2+2s+2} \end{bmatrix}$$

Three of the transfer functions are BIBO stable, the one from D to E is <u>n</u>ot. This is in spite of the fact that a bounded r produces a bounded y. Notice that $C(s)$ cancels an unstable pole in $P(s)$.

We'll see that this doesn't work.

14. Testing for feedback stability:
    Write $P(s) = \frac{N_p}{D_p}$, and $C(s) = \frac{N_c}{D_c}$
    Assume:

    - (N_p, D_p) are <u>coprime</u> (i.e. no common factors)
    - (N_c, D_c) are <u>coprime</u> too.

    The <u>characteristic polynomial</u> (he writes ch.p) of the feedback system is:

    $$\Pi(s) = N_p N_c + D_p D_c$$

    He calls this "unity feedback".

    The characteristic polynomial is the denominator of the 4 Transfer functions $(r, d) \to (e, u)$

    $$\begin{bmatrix} \frac{1}{1+PC} & \frac{-P}{1+PC} \\ \frac{C}{1+PC} & \frac{1}{1+PC} \end{bmatrix} = \frac{1}{\Pi(s)} \begin{bmatrix} D_p D_c & -N_p D_c \\ N_c D_p & D_p D_c \end{bmatrix}$$

    -

      Example 1. $P(s) = \frac{1}{s^2-1}$, $C(s) = \frac{s-1}{s+1}$

      $$\begin{aligned} \Pi(s) &= (s-1) + (s^2-1)(s+1) \\ &= (s-1)(s^2+2s+2) \end{aligned}$$

      $s - 1$ is unstable, thus this is unstable.

      Theorem 1. *The feedback system is stable if and only if the characteristic polynomialhas no roots in $\{s \in \mathbb{C}.Re(s) \geq 0\}$*

      Proof. of theorem:

– (←)
  If $\Pi(s) = N_p N_c + D_p D_c$ has no roots with $\text{Re}(s) \geq 0$, then the 4 transfer functions $(r, d) \to (e, u)$ are BIBO stable and hence the feedback system is stable.
– (→)
  Assume the feedback system is stable, i.e. the 4 TFs $(r, d) \to (e, u)$ are BIBO stable. To conclude that $\Pi(s)$ has no roots in $\text{Re}(s) \geq 0$, we must conclude show that $\Pi(s)$ has no common roots with the numerators $D_p D_c$, $N_c D_p$, $N_p D_c$
  He leaves this proof to us (as an excercise to the reader).

Definition 1. *Pole-Zero Cancellation:*
The plan $P(s)$, and controller $C(s)$ have a pole-zero cancellation at $\lambda \in \mathbb{C}$. If $N_p(\lambda) = D_c(\lambda) = 0$, then the controller pole cancels the plant zero. If $N_c(\lambda) = D_p(\lambda) = 0$, then the controller zero cancels the plant pole.

Definition 2. *Unstable pole-zero cancellation*
It's called an unstable pole-zero cancellation if Re$(\lambda) \geq 0$.

Corollary 2. *If there is an unstable pole-zero cancellation, then the feedback system is unstable.*

Proof. Suppoe there is an unstable cancellation at $\lambda \in \mathbb{C}$, with $\text{Re}(s) \geq 0$. Then:

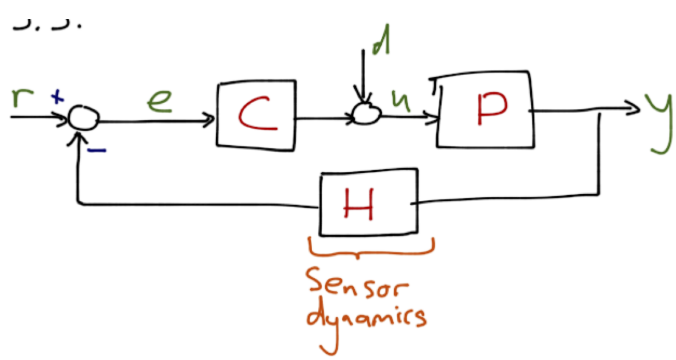
$$\begin{aligned} \Pi(\lambda) &= N_c(\lambda) N_p(\lambda) + D_c(\lambda) D_p(\lambda) \\ &= 0 + 0 \\ &= 0 \end{aligned}$$

Hence Π has a root with $\text{Re}(s) \geq 0$, so by previous theorem, the feedback system is unstable.

Theorem 3. *The feedback system is stable iff:*
– *The TF $1 + PC$ has no zeros with Re$(s) \geq 0$*
– *There are no unstable pole-zero cancellations*

Example 2. $P(s) = \frac{1}{s^2 - 1}$, $C(s) = \frac{s-1}{s+1}$
There are unstable pole-zero cancellations, so this must not be stable.

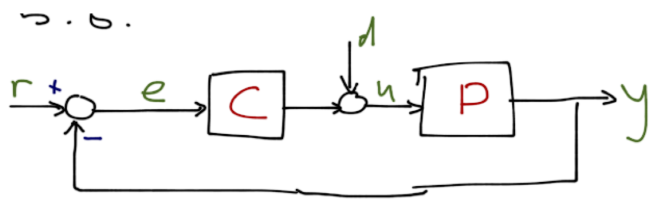
Remark 1. *Sometimes, we model the sensor as follows:*



We write:

$$P = \frac{N_p}{D_p}$$
$$C = \frac{N_c}{D_c}$$
$$H = \frac{N_h}{D_h}$$

15. 5.12: Comparison of open-loop and closed-loop poles/zeros:



We write:

$$P = \frac{N_p}{D_p}$$
$$C = \frac{N_c}{D_c}$$

The open-loop TF is $PC = \frac{N_p N_c}{D_p D_c}$.

The closed-loop TF is $\frac{PC}{1+PC} = \frac{N_p N_c}{D_p D_c + N_p N_c}$.
Observations:

- The zeroes of closed-loop system equals of the zeros of an open-loop system (except the ones that were cancelled).
- If P has a non-minimum phase zero, so does the closed-loop system.
- Usually, we design our poles to be in a good region.

16. 5.2 The Routh-Hurwitz Stability Criterion
    (a) In practice, we often check feedback stability by numerically computing roots of $\Pi(s)$
    (b) If some of the coefficients of $\Pi(s)$ are not fixed, e.g. controller gains we can't compute numerical solutions
    (c) The right hand test provides a way to check if the roots of a polynomial are in $\mathbb{C}^-$. This test is useful for design.
    (d) Example of a 1^{st} order:

$$\Pi(s) = a_1 s + a_0$$
$$a_1 \neq 0$$

The root is $s = \frac{-a_0}{a_1}$.
So $\Pi(s)$ is stable $\Leftrightarrow$ $\text{sign}(a_1) = \text{sign}(a_0)$.

(e) Example of a 2^{nd} order:

$$\begin{aligned}\Pi(s) &= a_2 s^2 + a_1 s + a_0 \\ a_2 &\neq 0\end{aligned}$$

The roots are $s = \frac{-a_1 \pm \sqrt{a_1^2 - 4a_2 a_0}}{2a_2}$. We know that $\text{sign}(a_1) = \text{sign}(a_0)$.
It's no longer obvious what the conditions are on a_2, a_1, a_0 should be.

(f) Algorithm:
Consider the general characteristic polynomial:

$$\begin{aligned}\Pi(s) &= \sum_{i=0}^{n} a_i s^i \\ &= a_0 + a_2 s^2 + a_3 s^3 + \ldots + a_n s^n \\ a_i &\in \mathbb{R}\end{aligned}$$

A necessary condition for $\Pi(s)$ to have all it's roots in $\mathbb{C}^-$ is that all the a_i's have the same sign.
To tee this, factor $\Pi(s)$ as follows:

$$\Pi(s) = a_n(s - \lambda_1) \ldots (s - \lambda_r)(s - \mu_1)(s - \bar{\mu_1}) \ldots (s - \mu_s)(s - \bar{\mu_s})$$

Where $\{\lambda_1, \ldots, \lambda_r\}$ are the real roots, and $\{\mu_1, \bar{\mu_1}, \ldots, \mu_r, \bar{\mu_r}\}$ are the complex conjugate roots.
$n = r + 2s$.
If $\lambda_i < 0$, then $-\lambda_i > 0$
If $\text{Re}(\mu_i) = \text{Re}(\bar{\mu_i})$, then we know that $(s - \mu_i)(s - \bar{\mu_i}) = s^2 - (\mu_i + \bar{\mu_i})s + \mu_i \bar{\mu_i}$.
We know that $(\mu_i + \bar{\mu_i}) > 0$, and $\mu_i \bar{\mu_i} > 0$.
If we re-expand the polynomial, we see that all coefficients have $\text{sign}(a_n)$.

(g) General way to set up a Routh-Hurwitz table:
Zig-zag the first two rows that coefficients are placed in.

$$\Pi(s) = \sum_{i}^{n} a_i s^i$$

s^n	a_n	a_{n-2}	a_{n-4}	$\ldots$	a_0
s^{n-1}	a_{n-1}	a_{n-3}	a_{n-5}	$\ldots$	0

Calculate elements using the negative determinant of the far-left column directly above and the column one to the right and above, divided by the one directly above at the far left:

s^i	$a_{(i,0)}$	$a_{(i,1)}$	$a_{(i,2)}$	$\ldots$
s^{i-1}	$a_{(i-1,0)}$	$a_{(i-1,1)}$	$a_{(i-1,2)}$	$\ldots$
s^{i-2}	$a_{(i-2,0)}$	$a_{i-2,1}$	$a_{i-2,2}$	$\ldots$

We have:

$$a_{(j,k)} = \frac{-\begin{vmatrix} a_{(j-2,0)} & a_{(j-2,k+1)} \\ a_{(j-1,0)} & a_{(j-1,k+1)} \end{vmatrix}}{a_{(j-1,0)}}$$

For example:

$$\Pi(s) = a_3 s^3 + a_2 s^2 + a_1 s + a_0$$

s^3	a_3	a_1	0
s^2	a_2	a_0	0
s^1	$\frac{a_2 a_1 - a_3 a_0}{a_2}$	0	0
s^0	a_0	0	0

We need all roots to be in the LHP to be stable, so we want to pick the values as follow: $a_3 > 0$, $a_2 > 0$, $\frac{a_2 a_1 - a_3 a_0}{a_2} > 0$, $a_0 > 0$

(h) C, P example:

$$\begin{aligned}
P &= \frac{2}{s^2 + s + 1} \\
C &= K_p + \frac{K_i}{s} \\
&= \frac{K_p s + K_i}{s} \\
\Pi(s) &= D_c D_p + N_c N_p & &= (s)(s^2 + s + 1) + 2(K_p s + K_i) \\
&= s^3 + s^2 + (1 + 2K_p)s + 2K_i
\end{aligned}$$

s^3	1	$1 + 2K_p$	0
s^3	1	$2K_i$	0
s^1	$1 + 2K_p - 2K_i$	0	0
s^0	$2K_i$	0	0

$$\begin{aligned}
1 &> 0 \\
1 &> 0 \\
2K_i &> 0 \\
1 + 2K_p - 2K_i &> 0 \\
K_p &> K_i - \frac{1}{2}
\end{aligned}$$

(i) Example:

$$P = \frac{-8.84}{s^2 - 19.6}$$

Prove that this cannot be stabilized using a Proportional controller.

(j) Examples

i. $s^4 + 3s^2 - 2s + 1$ (bad root)

ii. $s^3 + 5s^2 + 9s + 1$ (don't know)

iii. $s^3 + 4s + 6$ (bad root)

In the end, if we are given $\Pi(s)$, we can construct the Routh table:

s^n	a_n	a_{n-2}	a_{n-4}	...
s^{n-1}	a_{n-1}	a_{n-3}	a_{n-5}	...
s^{n-2}	b_1	b_2	b_3	...
s^{n-3}	c_1	c_2	c_3	...
...				...
s^1	l_1	l_2		
s^0	m_1			...

The third row is computed from the first two rows.

$$b_1 = \frac{a_{n-1}a_{n-2} - a_n a_{n-3}}{a_{n-1}}$$
$$b_2 = \frac{a_{n-1}a_{n-4} - a_n a_{n-5}}{a_{n-1}}$$
$$etc$$

The fourth row is computed from the previous two rows:

$$c_1 = \frac{b_1 a_{n-3} - a_{n-1} b_2}{b_1}$$
$$c_2 = \frac{b_1 a_{n-3} - a_{n-1} b_3}{b_1}$$
$$etc$$

- Continue along each row until you get zero consistently.
- terminate algorithm when you get a zero in the 1st column.

(k) How to "divine" the algorithm:

- $\Pi(s)$ has all roots in $\mathbb{C}^-$ $\Leftrightarrow$ all elements in the first column have the same sign.
  i. if the algorithm terminates early, $\Pi(s)$ has a bad root.
- If there are no zeroes in the first column:
  i. The number of sign changes in the first column = the number of bad roots.
  ii. There are no bads roots on imaginary axis

(l) Example of a 2^{nd} order equation $a_2 s^2 + a^1 s + a^0$

s^2	a_2	a_0	0
s^1	a_1	0	0
s^0	$\frac{a_1 a_0 - 0 a_2}{a_1} = a_0$	0	0

All roots of $\Pi(s)$ in $\mathbb{C}^-$ $\Leftrightarrow$ $\text{sign}(a_2) = \text{sign}(a_1) = \text{sign}(a_0)$

(m) Example with $\Pi(s) = rs^4 + s^3 + 3s^2 + 5s + 10$

(n) Example of early termination:

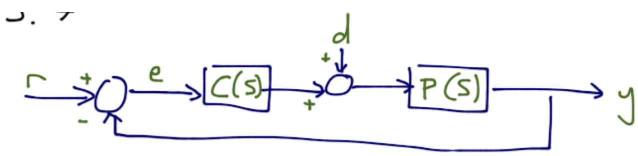
$$\Pi(s) = s^3 + s^2 + s + 1$$

Π has a bad root because the algorithm terminates early. Thus, the feedback system is unstable.

s^4	2	3	10
s^3	1	5	0
s^2	-7	10	0
s^1	$45/7$	0	0
s^0	10	0	0

s^3	1	1	0
s^2	1	1	0
s^1	$0 \to$ early termination		

(o) Example with a feedback loop:



$$P(s) = \frac{1}{s^4 + 6s^3 + 11s^2 + 6s}$$
$$C(s) = K$$

Where K is the real gain.

From section 5.1.1, we know that the feedback is stable iff all roots of Π are in the left half complex plane.

$$\begin{aligned}\Pi(s) &= N_p N_c + D_p D_c \\ &= s^4 + 6s^3 + 11s^2 + 6s + K\end{aligned}$$

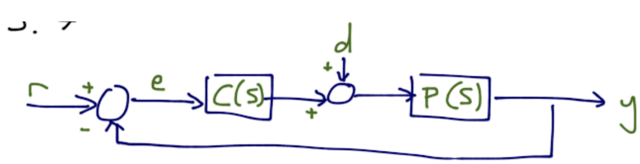
We know that k is gerater than 0 since... [missed this point.]

s^4	1	11	k
s^3	6	6	0
s^2	$\frac{66-6}{10} = 10$	$\frac{6k-0}{6} = k$	0
s^1	$\frac{60-6K}{10} = \frac{3(10-k)}{5}$	0	0
s^0	k	0	0

For feedback stability, we need all signs to be these same in the first column. $s^1 \to k < 10$, $s^0 \to k > 0$.

Thus, we know we should have $0 < k < 10$.

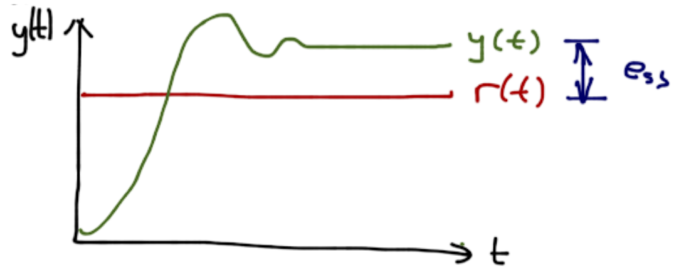
17. 5.3 Steady state performance.



- Closed-loop stability is essential

- Good performance is desirable, and can be measured as follows.
  - Transient response (T_s, $\%OS$, etc)
    Easy to measure for second order systems, but is hard to do in general.
  - Steady state.
    * steady-state gain
    * tracking error
    * disturbance rejection

    All of these are calculated using the final value theorem (FVT)



- Example: Cruise control regulates a car's speed to a desired (constant) value. The principle behind its operation is The Final value theorem. Car model:

$$\dot{x} = -x + u$$
$$\Rightarrow \frac{Y(s)}{U(s)} = P(s) = \frac{1}{s+1}$$
$$y = x$$

  If we take the integrator as our controller, we have:

$$u(t) = \int_0^{\infty} e(\tau) d\tau$$
$$\Rightarrow C(s) = \frac{1}{s}$$

  Since $\Pi(s) = s^2 + s + 1$, the feedback is stable (both poles are in the LHP)
  Suppose the driver suddently changes the desired speed to r_0 (constant). We can model this as a step chage ($r(t) = r_0$, $t \geq 0$)

- Steady-state tracking error:
  TF from $R \to E = \frac{1}{1+PC} = \frac{D_pD_c}{N_pN_c+D_pD_c}$
  Plugging in, we get:

$$E(s) = \frac{s(s+1)}{s^2+s+1}\frac{r_0}{s}$$

  Since Π is stable, the FVT applies:

$$e_{ss} \lim_{t\to\infty} e(t) = \lim_{s\to 0} sE(s)$$
$$= \lim_{s\to 0} s\frac{s(s+1)}{s^2+s+1}\frac{r_0}{s}$$
$$= 0$$

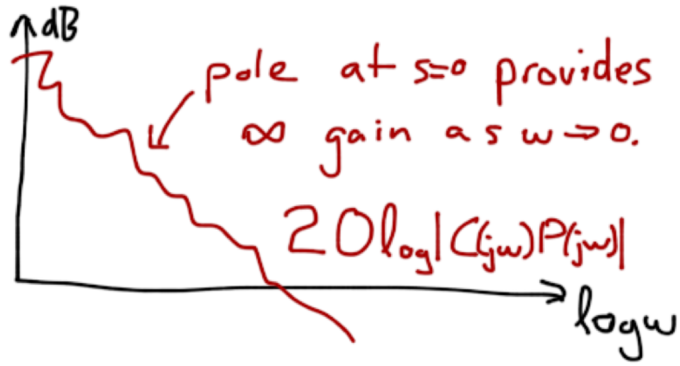
  So the system provides <u>perfect asymptotic tracking</u> of any step reference input.

- Why does this work?
  (a) In the frequency domain,

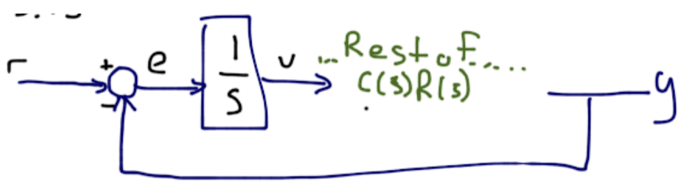
$$\frac{E(s)}{R(s)} = \frac{1}{1 + C(s)P(s)}$$

As $\omega \to 0$, we get:

$$\frac{E(j\omega)}{R(j\omega)} \to \frac{1}{1+\infty}$$



  (b) In the time domain



If CLS is stable, all signals approach a constant as $t \to \infty$.

$$v(t) = \int_0^t e(\tau)d\tau$$
$$\dot{v} = e$$

Since $v(t)$ approaches a constant, e must go to zero

  (c) Internal Model Principle
  The product $C(s)P(s)$ has a model of $R(s)$ (i.e. an integrator) embedded in it. Closing the loop creates a zero in the TF from R to E to exactly cancel the unstable part of $R(s)$. Not an illegal cancellation, since we are cancelling a pole in the signal, not in the system.
  Only the unstable parts of R are the ones that we need to do tracking for, the other ones all go to a constant.

In general, if $C(s)$ provides closed-loop stability, then FVT applies, and:

$$\begin{aligned} e_{ss} &= \lim_{t\to\infty} e(t) \\ &= \lim s \to 0 sE(s) \\ &= \lim s \to 0 s\frac{1}{1+C(s)P(s)}\frac{r_0}{s} \\ &= \frac{r_0}{1 + C(0)P(0)} \end{aligned}$$

Therefore, we can say these are identical statements:

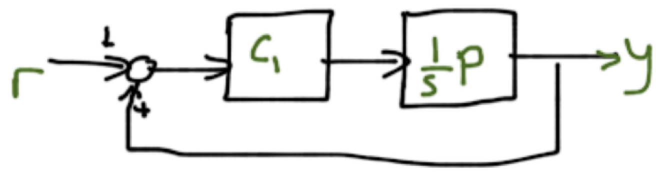
- $e_{ss} = 0$
- $P(0)C(0) = \infty$
- $P(s)C(s)$ has a pole at $s = 0$
- $P(s)C(s)$ has at least one integrator.

So the integral control is fundamental for perfect step tracking.

- If $P(s)$ does not have a pole at $s = 0$ and we want perfect step tracking, the controller must be the one that provides the integration. Thus, it is common to pick:

$$C(s) = \frac{1}{s} C_1(s)$$

so that $C(s)P(s)$ has an integrator. Then $C_1(s)$ is designed to stabilize the overall system.



The integrator decreases the phase plot of $P(s)$ which makes it harder to design C_1 to stabilize the whole system. There's no free lunch (as usual).

- If $C(0)P(0)$ is finite, then the system $C(s)P(s)$ is called a type-0 system, and $K_d := C(0)P(0)$ is the position error constant.

$$e_{ss} = \frac{r_0}{1 + K_p}$$

- Example: Nonconstant Reference:

$$\begin{aligned} C(s) &= \frac{1}{s} \\ P(s) &= \frac{2s+1}{s(s+1)} \\ E(s) &= \frac{1}{1 + C(s)P(s)} R(s) \\ &= \frac{D_p D_c}{N_p N_c + D_p D_c} \frac{r_0}{s^2} \\ &= \frac{s+1}{s^3 + s^2 + 2s + 1} \end{aligned}$$

s^3	1	2	0
s^2	1	1	0
s^1	1	0	0
s^0	1	0	0

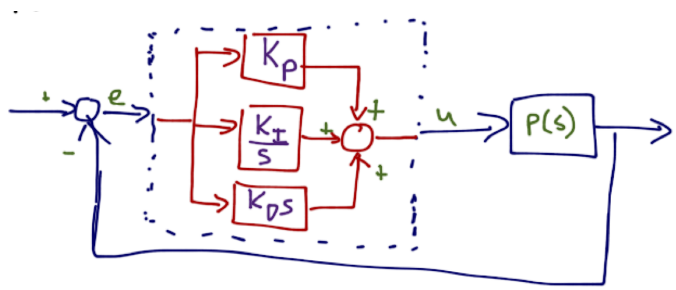
Since the first column is the same sign, then $\Pi(s)$ is stable, and the feedback is stable.

18. 5.4 Intro to PID control.

- Majority of controllers in industry are Proportional-Integral-Derivative control.
- They are popular because:

- they are easy to tune
- structure doesn't depent on plant model
- often provides decent performance

Diagram of PID controllers:



From this, we get:

$$
\begin{aligned}
C(s) &= K_p + \frac{K_i}{s} + K_d s \\
&= \frac{K_d s^2 + K_p s + K_i}{s} \\
u(t) &= K_p e(t) + K_i \int_0^{\infty} (e(\tau) d\tau) + K_d \frac{de}{dt}
\end{aligned}
$$

The gains K_d, K_i, K_D can be "designed" in the time or frequency domains. We see this in chapter 7.

(a) Refinements to basic PIDs:

i. The basic PID is improper so it can't be implemented in practice.

A. To fix this, the derivative is usually to make it proper as follows:

$$C(s) = K_p + \frac{K_i}{s} + K_D \frac{s}{(\tau s + 1)}$$

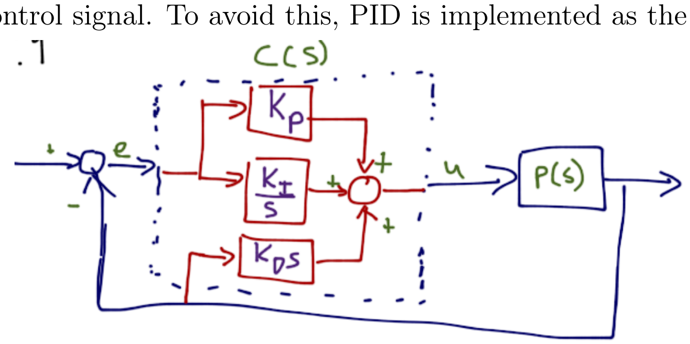
Where $\tau > 0$ and is small.

B. Another way to implement $C(s)$ is by adding a pole to the "whole thing":

$$C(s) = \frac{K_D s^2 + K_p s + K_i}{s(\tau s + 1)}$$

$\tau > 0$ is small. (Is a different tau than before.)

ii. The reference signal is often discontinuous (i.e. it steps). Differentiating steps result in nasty impulses in the control signal. To avoid this, PID is implemented as the following:



Note: (TODO: move this to the correct location) $e_{ss} = r - y$.

19. Tracking ramps:

$$\begin{aligned} E(s) &= \frac{1}{P(s)C(s)} \frac{r_0}{s^2} \\ &= \frac{D_p D_c}{N_p N_c + D_p D_c} \frac{r_0}{s^2} \end{aligned}$$

Think of partial fraction expansion. The signal $e(t)$ will blow up (like a ramp) unless one or both of the poles at $s = 0$ are cancelled.

- If $P(0)C(0)$ is finite, then $|e_{ss}| = \infty$
- If $P(s)C(s)$ has one or more poles at $s = 0$, and $C(s)$ provides CLS, then:

$$\begin{aligned} e_{ss} &= \lim_{t \to \infty} e(t) \\ &= \lim_{s \to 0} \frac{1}{P(s)C(s)} \frac{r_0}{s^2} \\ &= \frac{r_0}{sC(s)P(s)}|_{s=0} \end{aligned}$$

Therefore,

$$\begin{aligned} e_{ss} = 0 &\Leftrightarrow sC(s)P(s)|_{s=0} = \infty \\ &\Leftrightarrow \text{There are } \underline{\text{at least}} \text{ two integrators in PC} \end{aligned}$$

If $C(s)P(s)$ has one pole at $s = 0$ it's called a <u>type-1</u> system, and the <u>velocity error constant is</u>:

$$\begin{aligned} K_v &= sC(s)P(s)|_{s=0} \\ e_{ss} &= \frac{r_0}{K_v} \end{aligned}$$

(a) Internal model principle:
If $P(s)$ is strictly proper, and $C(s)$ is proper and provides closed-lopp stability. If $C(s)P(s)$ contains a model of the unstable part of $R(s)$ then we get perfect asymptotic tracking.

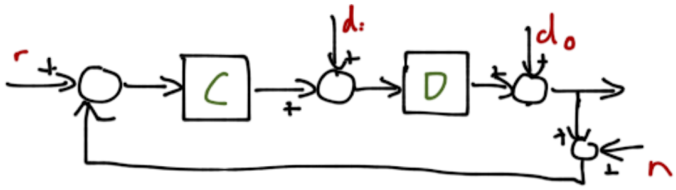
(b) Example:

$$P(s) = \frac{1}{s+1}$$

Reference command: $r(t) = r_0 sin(t) \to R(s) = \frac{r_0}{s^2+1}$ Then $P(s)C(s)$ needs a model of the unstable parts of $R(s)$.
The unstable part of R corresponds to poles at $s = \pm j$. Since P does not have a pole at $s = j$ or $s = -j$, the controller must provide the internal model.
This suggests $C(s) = \frac{1}{s^2+1} C_1(s)$, i.e. embed the model in C and design C_1 to achieve feedback stability.
In this case, $C_1(s) = s$ works.
Let's check:

$$\begin{aligned} \frac{E(s)}{R(s)} &= \frac{(s^2+1)(s+1)}{s^3 + s^2 + 2s + 1} \\ e_{ss} &= \lim_{s \to 0} s \frac{(s^2+1)(s+1)}{s^3 + s^2 + 2s + 1} \frac{r_0}{s^2+1} \\ &= 0 \end{aligned}$$

(c) Steady-state disturbance rejection: [5.3.2]



Our Goal is to minimize the effect of the disturbances and noise on the output y.
As before, we can use the final value theorem to analyze the effect of noise etc.

(d) Example:
Find conditions on P, C so that the steady state effect of an output step disturbance is zero.
Use linearity to set $r = d_i = n = 0$

$$\frac{Y(s)}{D_0(s)} = \frac{1}{1 + P(s)C(s)}$$
$$d_0(t) = d_{const}, t \geq 0.$$

If $C(s)$ stabilizes the loop, then

$$\begin{aligned} y_{ss} &= \lim_{t\to\infty} y(t) \\ &= \lim_{s\to 0} s\frac{1}{1+PC}\frac{d_0}{s} \\ &= \frac{d_0}{1 + C(0)P(0)} \end{aligned}$$

Therefore $y_{ss} = 0 \Leftrightarrow$ PC has a pole at $s = 0$.

(e) Example (broomstick):
It's not always possible to track perfectly and get stability.
If you have an inverted pendulum on your hand as so: (i.e. a broomstick)

Horizontal position: $u + L\sin\theta$
Vertical position: $L\cos\theta$
Horizontal equation of motion:

$$m\frac{d^2}{dt^2}(\text{Horizontal position}) = F\sin\theta$$

Vertical equation of motion:

$$m\frac{d^2}{dt^2}(\text{Vertical position}) = F\cos\theta - mg$$

We can take the linear approximation by assuming that if θ is small mass only moves in the horizontal direction: Horizontal:

$$m\dot{u} + mL\ddot{\theta} = F\theta$$

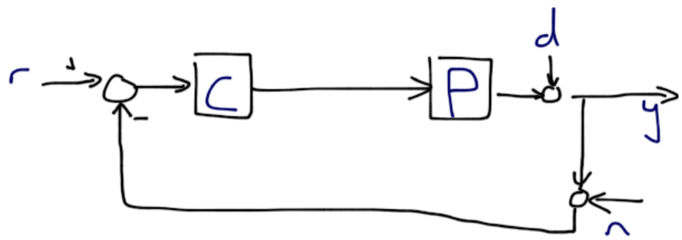
Vertical:

$$\begin{aligned} 0 &= F - mg \\ \rightarrow \dot{u} + L\ddot{\theta} &= g\theta \\ \rightarrow s^2U(s) + s^2 L\Theta(s) &= g\Theta(s) \\ \frac{\Theta(s)}{U(s)} &= \frac{-s^2}{Ls^2 - g} \\ &= P(s) \end{aligned}$$

$P(s)$ has zeros at $s = 0$, so we can't pick $C(s) = \frac{1}{s}C_1(s)$, so we can't embed an internal model if $R(s) = \frac{r_0}{s}$ without adding an unstable pole-zero cacellation.
Perfect tracking and internal stability are not possible.

20. Loopshaping - High-gain control revisited: [5.12]

   (a) Key Idea:
   the choice of high-gain versus low-gain is a frequency-dependent choice:

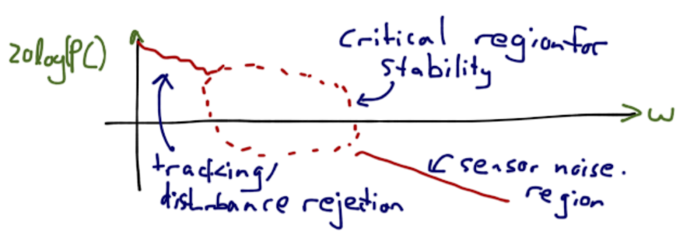


   So,

$$Y(s) = \frac{PC}{1+PC}R(s) + \frac{1}{1+PC}D(s) - \frac{PC}{1+PC}N(s)$$

   - To minimize the effect of d on y at frequency ω, $|P(j\omega)C(j\omega)|$ should be large
   - To have the TF from r to y close to one at a frequency ω, we need $|P(j\omega)C(j\omega)|$ to be large (again)
   - To minimize the effect of noise at frequency ω we need $|P(j\omega)C(j\omega)|$ to be small.

   (b) Rule of thumb (overview):
   Use high gain at frequencies where tracking/disturbance rejection is required. Use low gain at frequencies where noise is active.
   At intermediate frequencies, we will do whatever it takes to stabilize the closed-loop system.

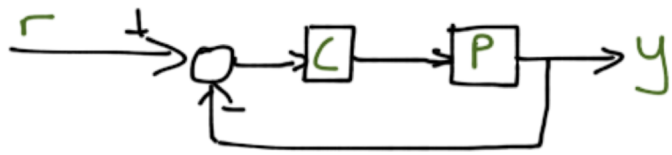


21. Root Locus Methods [Chapter 6]

(a) Loop shaping gives us intuition about our gain choices.

(b) Root Locus gives us a picture of how closed-loop poles $\Pi(s) = D(s) + KN(s)$ moves as a single parameter varied.

(c) Root Locus Methods



$$\Pi(s) = N_c N_p + D_c D_p$$

- Closed loop poles (roots of Π) determine stability directly and indirectly performance.
- If Π has uncertain plant parameters or controller gains, we can use Routh-Hurwitz to determine stability but not much else.
- Root-locus gives much more info:
  - Shows how poles move on the s-plane as a single parameter is varied
  - Ease of use makes it an efficient analysis tool
  - Can be used (for limited) design
  - In matlab, we use the `rltool` command.

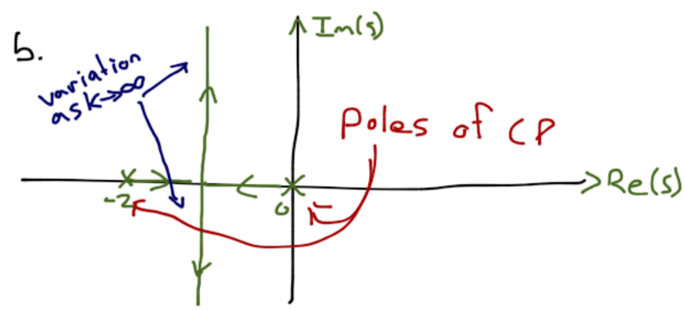
(d) Example:

$$C(s) = K$$

$$P(s) = \frac{1}{s(s+2)}$$

$$\Pi(s) = s^2 + 2s + K$$

$$\text{Closed-loop poles: } s = -1 \pm \sqrt{1-K}$$



i. $K < 0$, unstable

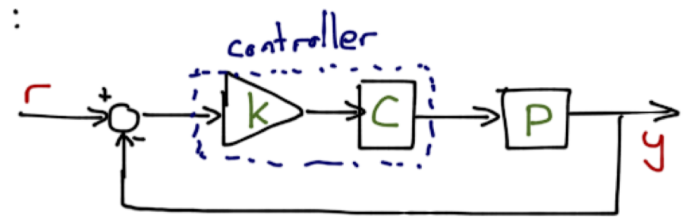
ii. $K = 0$ poles at $s = 0$, $s = -2$, unstable

iii. $K \in (0, 1)$ two real distinct & stable poles

iv. $K = 1$ two real repeated & stable poles

v. $K > 1$ complex conjugate roots, with real part $= -1$

(e) Basic construction of the Root-locus: [Chapter 6.1]
Consider a unity-feedback system with a gain pulled out of the controller:



$$\frac{Y(s)}{R(s)} = \frac{KC(s)P(s)}{1 + KC(s)P(s)}$$
$$= \frac{KN(s)}{KN(s) + D(s)}$$
$$\Pi(s) = D(s) + KN(s)$$

We write $C(s)P(s) = \frac{N(s)}{D(s)}$, $n := \deg(D(s))$, $m := \deg(N(s))$

(f) Root-locus assumptions:

We are given two polynomials $N(s)$ and $D(s)$, and want to show graphically how the roots of Π vary with the scalar K.

We make the assumption that:

- $n \geq m$ (i.e. $C(s)D(s)$ is proper)
- K varies from $0 \to \infty$
- $N(s)$ and $D(s)$ are monic (leading coefficients are 1).

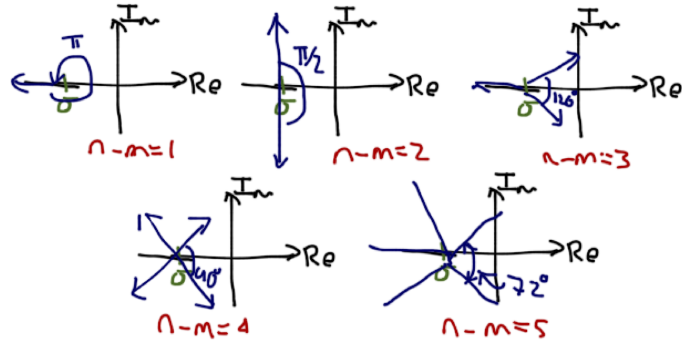
(g) Construction rules:

- The Root-locus is symmetric about the real axis.
  Since Π has real coefficients
- The Root-locus consists of n branches
  Since Π is of n^{th} order, it has n roots (which become branches, apparently)
- The Root-locus is a continuous function of K.
- The root locus "start" (i.e. $K = 0$) at the roots of $D(s)$
  When $K = 0$, $\Pi(s) = KN(s) + D(s) = D(s)$, i.e. the open-loop poles.
- As $K \to \infty$, m of the branches will terminate at the roots of $N(s)$
  The roots of $N(s)$ are the zeros of $C(s)P(s)$ This is not immediately obvious.
  The remaining $n - m$ branches go to infinity along asymptotes.
  The i^{th} asymptote has the angle ϕ_i:

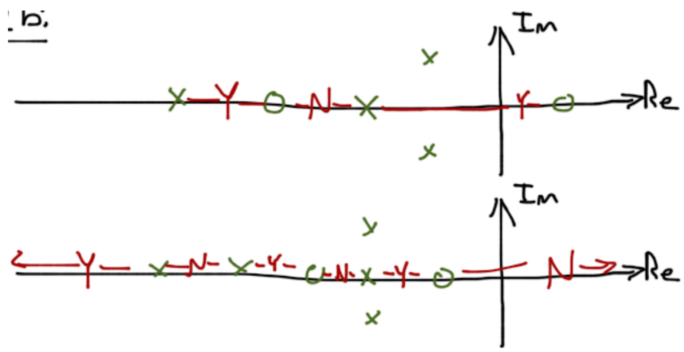
$$\phi_i = \frac{(2 + i)\pi}{n - m},\ i \in \mathbb{N}$$

and intersects the real axis at a centroid:

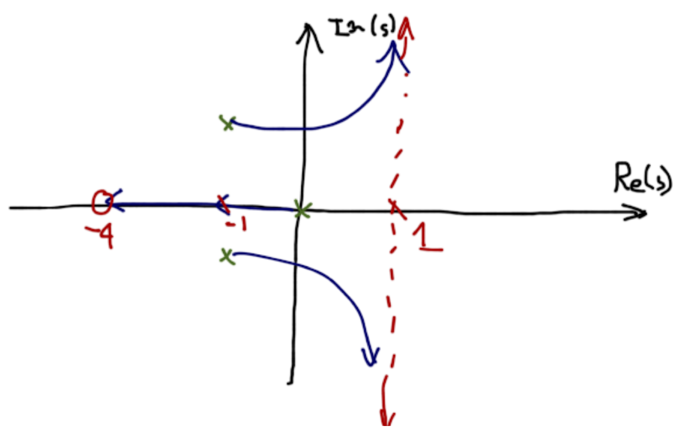
$$\sigma = \frac{\Sigma(\text{roots of } D(s)) - \Sigma(\text{roots of} N(s))}{n - m}$$



- ("no-yes-no" rule) A point s on the real axis is part of the root locus if and only if the total number of poles and zeros of $\frac{N(s)}{D(s)}$ to the right of s is odd.



- Example:



$$N(s) = s + 4$$
$$\rightarrow m = 1$$
$$D(s) = s(s^2 + 2s + 5)$$
$$\rightarrow n = 3$$
$$\Pi(s) = KN(s) + D(s)$$

i. Plot zeros (roots of N) and poles (roots of D)
ii. Use "no-yes-no" rele to fill in the real axis
iii. Find asymptotes

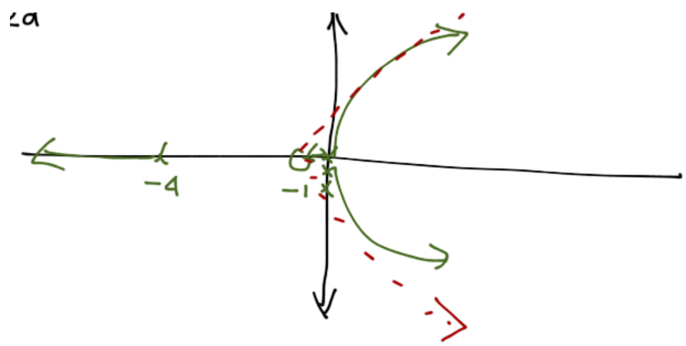
$$\sigma = \frac{\Sigma(\text{roots of } D(s)) - \Sigma(\text{roots of} N(s))}{n-m}$$
$$= \frac{0 + (-1+2j) + (-1-2j) + (-4)}{3-1}$$

iv. Sketch <u>rough</u> root locus ($rule5$) implies the pole at $s = 0$ moves towards zero at $s = -4$

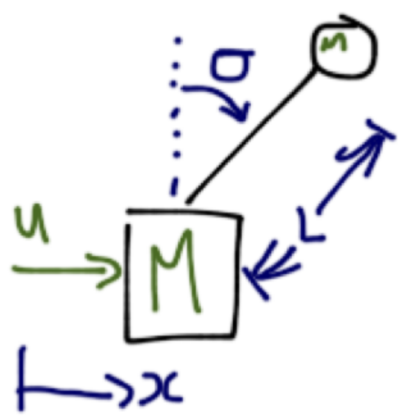
v. We could use RH to determine when the RL crosses imaginary axis

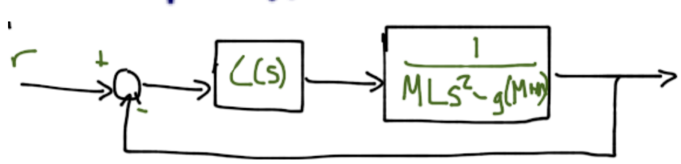
- Example:

$$\Pi(s) = s^3(s+4) + K(s+1)$$
$$D(s) = s^3(s+4)$$
$$\rightarrow n = 4$$
$$N(s) = K(s+1)$$
$$\rightarrow m = 1$$
$$\rightarrow \sigma = -1$$



- Example:





Determine how closed-loop poles vary as a function of k for a P, PD, PI, and PID controller.

- P Controller:

$$
\begin{aligned}
C(s) &= K \\
\Pi(s) &= MLs^2 - g(M+m) + K \\
D(s) &= MLs^2 - g(M+m) \\
n &= 2 \\
N(s) &= 1 \\
m &= 0
\end{aligned}
$$

P cannot stabilize

- PD controller:

$$
\begin{aligned}
C(s) &= K_p + K_D s \\
&= K(1+\tau s) \\
\Pi(s) &= MLs^2 - g(M+m) + K(1+\tau s) \\
D(s) &= MLs^2 - g(M+m) \\
N(s) &= (1+\tau s)
\end{aligned}
$$

- Summary:
  - Root locus gives us a picture of how closed-loop poles move wrt a single changing variable.
  - Basic rules (1-6) operate on the asymptotes and centroids.
  - We went through an example using Root-locus to decide on a controller type.

(h) Example:
TODO: insert Diagram 6-13a.

$$P(s) = \frac{1}{M\ell s^2 - g(m+M)}$$

This is connnected in unity-feedback

$$C(s) = P, PD, PI, PID$$

i. P cannot stabilize (from last class.) TODO: coagulate

ii. PD:

$$
\begin{aligned}
C(s) &= K_p K_D s \\
&= K(1+\tau s)
\end{aligned}
$$

TODO: insert Diagram 6-13b

iii. PI:

$$
\begin{aligned}
C(s) &= K_P + \frac{K_I}{s} \\
&= K(1+\frac{a}{s}) \\
&= K(\frac{s+a}{s}) \\
\Pi(s) &= s(M\ell s^2 - g(M+m)) + K(s+a) \\
\rightarrow D(s) &= s(M\ell s^2 - g(M+m)) \\
\rightarrow N(s) &= K(s+a)
\end{aligned}
$$

TODO: insert Diagram 6-13c

iv. PID

$$\begin{aligned}C(s) &= K_P + \frac{K_I}{s} + K_D s\\ &= K(a + \frac{b}{s} + s)\\ &= K\frac{s^2 + as + b}{s}\\ \Pi(s) &= s(MLs^2 - g(m+M)) + K(s^2 + as + b)\\ \rightarrow D(s) &= (MLs^2 - g(m+M))\\ \rightarrow R(s) &= (s^2 + as + b)\end{aligned}$$

TODO: insert Diagram 6-13d

(i) Rules 7 $\rightarrow$ 9: Refer to notes. He doesn't care about these. [Chapter 6.2]

22. Non-Standard Problems: [6.3]

We made 3 major assumptions in the basic root locus constructions:

- Unity Feedback
- We can "pull" a gain out of $C(s)$
- $\Pi(s)D(s) + KN(s)$, $\deg(D) \geq \deg(N)$

Here's how you tackle these problems:

(a) Sensor dynamics:
TODO: insert diagram 6-13e
Let:

$$\begin{aligned}\frac{N(s)}{D(s)} &= C(s)P(s)H(s)\\ \rightarrow \Pi(s) &= KN(s) + D(s)\end{aligned}$$

In other words, the RL is exactly the same, except:

$$\begin{aligned}N(s) &= N_c N_p N_h\\ D(s) &= D_c D_p D_h\end{aligned}$$

(b) Non-fixed parameter can't be factored from controller:
Even if K can't be factored from $C(s)$, the characteristic polynomial will still have the form $\Pi(s) = KN(s) + D(s)$ but in this case, $D(s) \neq D_c D_p$, $N(s) \neq N_c N_p$

- Example:
TODO: insert diagram 6-13f
PD Controller $C(s) = 10(1 + \tau s)$, can't factor τ out.

$$\begin{aligned}\Pi(s) &= s(s+2) + 10(1 + \tau s)\\ &= (s^2 + 2s + 10) + (10\tau)s\\ \rightarrow D(s) &= (s^2 + 2s + 10)\\ \rightarrow K &= (10\tau)\\ \rightarrow N(s) &= s\end{aligned}$$

TODO: insert diagram 6-13g

(c) Deg(D) < Deg(N)

$$\begin{aligned}\Pi(s) &= D(s) + KN(s)\\ &= 0\\ \iff N(s) + \frac{D(s)}{K} &= 0\end{aligned}$$

Define the following:

$$\begin{aligned}\hat{D} &= N\\ \hat{N} &= D\\ \hat{K} &= \frac{1}{K}\end{aligned}$$

We can draw the root locus for:

$$\begin{aligned}\hat{\Pi}(s) &= \hat{D} + \hat{K}\hat{N}\\ \hat{K} : 0 &\to \infty\end{aligned}$$

We will need to reverse everything, including the root locus, etc.

- Example:

$$\begin{aligned}C(s) &= \frac{s+3}{\tau s+1}\\ P(s) &= \frac{1}{s(s+1)}\Pi(s) = (s+3) + (\tau s+1)(s(s+1))\\ &= expand\hat{D} & &= s^2(s+1)\\ \hat{N} &= s^2+2s+3\\ \hat{K} &= 1/\tau\hat{n} = 3\\ \hat{m} &= 2\end{aligned}$$

TODO: insert diagram 6-13h

(d) Summary:

i. Use RL to pick a controller:

ii. Non-standard RL problems

A. $\deg(D) < \deg(N)$

$$\Pi(s) = D(s) + KN(s)$$

B. Example:

$$\begin{aligned}\Pi(s) &= s^2+2s+3+\tau s^2(s+1)\\ \to_{\tau\to\infty} & \tau s^2(s+1)\end{aligned}$$

TODO: insert diagram 6-14a

23. Introduction to control design [Ch. 7]

- "classical" conrol design
  (a) lead
  (b) lag
  (c) lead-lag
- specs are in terms of bandwidth and stability margins
- Our real interest is how the system behaves in physical time in using freq domain. We'll employ the duality between frequency and time.
- Design Philosophy:
  (a) Take time domain specs and convert into frequency domain.

$$\begin{bmatrix} \%\text{ OS} \\ T_s \\ \text{tracking} \end{bmatrix} \to \begin{bmatrix} \text{BW} \\ \text{gain margin} \\ \text{phase margin} \end{bmatrix}$$

  (b) Adjust gains to meet steady-state specs
  (c) Design a dynamic controller (TF) to meet other specs without affecting the gains too much.
  (d) Simulate a controller
  (e) If necessary, adjust specs and restart.

(a) Inro to stability margins [7.1]:

If a system is stable, how stable is it? This depends entirely on our plant model, how we get it, and what uncertainty there is about the model.

In the frequency domain, uncertainty is naturally measured in terms of gain and phase as functions of frequency.

To understand stability margins, we need the Nyquist stability critereon. Instead, for now, we'll find stability margins from the Bode plot.

- Gain margin: (G_M)
  TODO: insert diagram 6-14b
  Think $k = 1$ as our normal model (gain margin G_M).

$$G_M = \sup\{\bar{k} \geq 1 : \text{closed-loop stability for } k \in [1, \bar{k})\}$$

- Phase margin: (P_M)
  TODO: insert diagram 6-14c
  Think of $\phi = 0$ as the nominal model (phase model P_M)

$$P_M = \sup\{\bar{\phi} \geq 0 : \text{closed-loop stability for any } \phi \in [0, \bar{\phi})\}$$

- Collectively, P_M and G_M are called the stability margins.
  Large stability margins not only imply robustness, but also good transient behaviour.
  TODO: insert diagram 6-14d
  Let $L(s)$ be as follows:

$$L(s) = C(s)P(s)H(s)$$

  Draw the bode plot of $L(j\omega)$:
  TODO: insert 6-14e

- ω_{P_M}
  - Freq at which we measure P_M
  - called the gaincrossover frequency
  - ω at which $20\log|L(j\omega)| = 0$

$$P_M = 180 - \underline{/L(j\omega_{P_M})}$$

- ω_{G_M}
  - freq at which we measure G_M
  - called the phase crossover frequency
  - ω at which $\underline{/L(j\omega)} = -180$

$$G_M = -20\log|L(j\omega_{G_M})|$$

(b) Design Problem [7.2]

- Given a plant $P(s)$ and a set of specs (common ones are closed-loop stability, transient stability, steady-state, G_M, P_M, bandwidth), Design $C(s)$ so that closed-loop system meets the specs.
  We focus on the following criteria:
  i. Closed-loop stability
  ii. Tracking error
  iii. % OS (Converted to P_M)
  iv. Bandwidth
  We focus on the following controllers:
  i. Lag:

$$C(s) = K_c \frac{s+z}{s+p}$$
$$z > p > 0$$

  TODO: insert diagram 6-14f
  ii. Lead:

$$C(s) = K_c \frac{s+z}{s+p}$$
$$p > z > 0$$

  TODO: insert diagram 6-14g
  iii. Lead-lag:

$$C(s) = K_c \frac{s+z_1}{s+p_1} \frac{s+z_2}{s+p_2}$$

  Lucas W. posted a link to intuitively explain the Lead-Lag controller. Wiki Article
  We focus on these because they are simple but useful. Also the are closely related to PID controllers.
  The approch we'll discuss works well for "nice plants"
  i. stable or at worst one pole at $s = 0$
  ii. only one crossover frequency.

i. Relationship between crossover frequency and bandwidth: [7.2.1]

$$\frac{Y(s)}{R(s)} = \frac{C(s)P(s)}{1+C(s)P(s)}$$
$$= T(s)$$

TODO: Insert diagram 7-1a
Usually:

$$\omega_{P_M} < \omega_{BW} < \omega_{GM}$$

For purposes in this course, we use:

$$\omega_{P_M} \approx \omega_{BW}$$

ii. Damping ratio and phase margin:
We claimed that small stability margins may mean por transient response. This relationship can be quantified <u>exactly</u> for 2nd order systems:
TODO: insert diagram 7-1b
i.e.

$$\frac{Y}{R} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

We can compute the P_M exactly:

$$P_M = atan\left[2\zeta\sqrt{\sqrt{1+4\zeta} - 2\zeta^2}\right]$$

TODO: insert diagram 7-1c

$$\Rightarrow P_M = 100\zeta$$

Where P_M is measured in degrees, and $0 \le \zeta < 0.7$

- Example:
  If we want our CLS to behave like a second order system with $\zeta = 0.65$, so we should make sure $P_M = 65$

(c) Lag compensation: [7.3]
TODO: insert diagram 7-1d

$$C(s) = KC_1(s)$$
$$= K\frac{\alpha Ts + 1}{Ts + 1}$$
$$0 < \alpha < 1$$
$$T > 0$$
$$K > 0$$

We observe the following:

- Log controller has DC gain K

- Has a pole at $s = \frac{-1}{T}$ and zero at $s = \frac{-1}{\alpha T}$

TODO: insert diagram 7-1E

The key benefit is high frequency gain reduction without affecting low frequency gain or high frequency phase as follows:

TODO: insert diagram 7-1F

Lag is used for two purposes:

i. Boost the low frequency gain (and improve steady-state performance) without much effect on G_M, P_M, nor high frequency behavior.
ii. To increase P_M (at the cost of reducing the bandwidth)
    TODO: insert diagram 7-1G

- Example:

$$P(s) = \frac{1}{s(s+2)}$$

  Specs:

  i. $r(t) = r_0 t$, $t \geq 0$, $|e_{ss}| \leq 0.05 r_0$
  ii. $P_M = 45$

  i. Choose K to meet the first spec:

$$\begin{aligned}
\lim_{t\to\infty} e(t) &= \lim_{s\to 0} sE(s) \\
&= \lim_{s\to 0} s \left. \frac{r_0}{sC(s)P(s)} \right|_{s=0} \\
&= \frac{r_0}{K/2} \\
&\leq 0.05 r_0 \\
\rightarrow K &\geq \frac{2}{0.05} \\
K &\geq 40
\end{aligned}$$

  We could've picked a bigger K, but we may as well do equal to 40.
  Pick $K = 40$

  ii. Draw bode of $KP(s) = \frac{40}{s(s+2)}$
  From the plot, we get $\omega_{PM} = 6$ rad/s, $P_M = 18$
  ω_{PM} is the gain crossover frequency.
  To increase P_M while preserving $K = 40$, we use a lag controller.
  We'll actually aim for $P_M = 50$ to compensate for Bode approximation.
  From Bode plot of $KP(s)$, we get:

$$\begin{aligned}
P_M^{\text{desired}} &= 45 + \delta \\
&= 50 \\
&= 180 + \angle KP(j\omega)
\end{aligned}$$

  We choose $\delta = 5$ °because that's about how much the bode plot is off from 90°.
  This works for when $\omega = 1.7$ rad/s
  So we want to make the new gain crossover frequency equal to 1.7 rad/s without changing the phase at $\omega = 1.7$ rad/s
  Tl;dr: we are trying to decrease the magnitude so it crosses zero at log(1.7) without changing the phase. In this example, that is a 19 dB decrease.

iii. Getting α
From the Bode plot, we choose $20\log|KP| = 19dB$ at $\omega = 17$ rad/s
So we want to reduce the gain by 19dB at $\omega = 1.7$ without changing the phase.

$$\begin{aligned} 20\log\left|\frac{\alpha Tj\omega+1}{Tj\omega+1}\right|_{\omega=1.7} &= -19 \\ &= 20\log\alpha \\ \rightarrow \alpha &= 0.111 \end{aligned}$$

iv. Getting T:
$\angle\frac{j\omega}{10}+1$ is the point where the phase angle meets the axis.
Set $\frac{10}{\alpha T} \le 1.7$ rad/s.
In this case, we pick $T = 52.7$ (equality).

v. Transfer function:

$$\begin{aligned} C(s) &= K\frac{\alpha Ts+1}{Ts+1} \\ K &= 40 \\ \alpha &= 0.111 \\ T &= 52.7 \end{aligned}$$

From the bode plot of $C(s)P(s)$, we get $P_M = 44.6°$.

- Lag controller design algorithm:
  i. Use the FVT to fix K
  ii. Draw the bode plot of $KP(s)$ (for future reference)
  iii. If P_M spec is met, we can end our algorithm (a P controller works)
  iv. Find ω such that:

$$P_M^{\text{desired}} + \delta = 180 + \angle KP(j\omega)$$

  P_M^{desired} is usually given in the question.
  δ is the buffer, usually is 5 °.

  v. Shift the gain down at the frequency calculated before ($\angle KP(j\omega)$) to get a new gain crossover frequency. Calculate α as follows:

$$\alpha = \frac{1}{KP(j\omega)}$$

  vi. Ensure phase isn't affected from the change we've done by setting T as follows:

$$\begin{aligned} \frac{10}{\alpha T} &\le \omega \\ T &\ge \frac{10}{\alpha\omega} \end{aligned}$$

  vii. Check the bode plot of CP to make sure the spec is met.

i. Lag vs PI [7.3.1]
Recall the ideal PI is expressed:

$$\begin{aligned} C(s) &= K_p + \frac{K_i}{s} \\ &= K_I\left(\frac{\frac{K_P}{K_I}s+1}{s}\right) \end{aligned}$$

We can view this as a lag controller where the pole is at $s = 0$. The gains can be determined using the approach for a lag controller.

(d) Lead Compensator [7.4]
TODO: insert diagram 7.2b

$$\begin{aligned} C(s) &= KC_1(s) \\ &= K\frac{\alpha Ts+1}{Ts+1} \\ \alpha &> 1 \\ K &> 0 \\ T &> 0 \end{aligned}$$

Observations:

- DC gain is K.
- Pole at $s\frac{-1}{T}$
- Zero at $s = \frac{-1}{\alpha T}$
  TODO: insert diagram 7.2c
- Bode plot of $C_1(s)$:
  TODO: insert diagram 7.2d The key benefit is to add phase to our system in order to increase the P_M.
  We will be adding the frequency at the middle of the peaks ($log(\omega_m)$).

ω_m = geometric mean of the pole and zero.
A lead controller is used when:

- We want to increase the gain crossover frequency (to meet the bandwidth [BW] spec.)
- We can increase the phase margin [PM] by adding ϕ_{max} to the appropriate frequency.
- i.e. we make the system have faster response at the cost of increasing the control effort.

i. Lead design equations [7.4.1]:
ω_m = frequency at which ϕ_{max} is added, i.e. the geometric mean of $\frac{1}{\alpha T}$, $\frac{1}{T}$ on the log scale.

$$\begin{aligned} \log(\omega_m) &= \frac{1}{2}\left(\log\left(\frac{1}{\alpha T}\right) + \log\left(\frac{1}{T}\right)\right) \\ \omega_m &= \frac{1}{T\sqrt{\alpha}} \\ \phi_{max} &= \sin^{-1}\left(\frac{\alpha-1}{\alpha+1}\right) \end{aligned}$$

A. Magnitude of $C_1(s)$ at ω_m:

$$\begin{aligned} \log|C_1(j\omega_m)| &= \frac{1}{2}\left(\log(1) + \log(\alpha)\right) \\ &= \log(\sqrt{(}\alpha)) \\ |C_1(j\omega)| &= \sqrt{\alpha} \end{aligned}$$

B. Maximum phase addition:

$$\begin{aligned}\phi_{Max} &= \angle C_1(j\omega_m)\\ &= \frac{\angle 1+\sqrt{\alpha}j}{\angle 1+\frac{1}{\sqrt{\alpha}}j}\end{aligned}$$

TODO: insert diagram 7-2e

$$\begin{aligned}\frac{\sin(\theta)}{\sqrt{(1+\frac{1}{\alpha})}} &= \frac{\sin(\phi_{Max})}{\sqrt{\alpha}-\frac{1}{\sqrt{\alpha}}}\\ \sin(\theta) &= \frac{1}{\sqrt{1+\alpha}}\\ \rightarrow \alpha &= \frac{1+\sin\phi_{max}}{1-\sin\phi_{max}}\end{aligned}$$

In general, it is not usually practical to pick $\alpha > 15$, so if ϕ_{max} needs to be > 60 °, we use two lead controllers in series.

- Re-do previous example:

$$\begin{aligned}P(s) &= \frac{1}{s(s+2)}\\ r(t) &= r_0 t\end{aligned}$$

Our spec is as follows:

A. $|e_{ss}| < 5\%$

B. $P_M = 45$ °

We start to solve the previous example:

A. Pick $K = 40$ to meet steady state spec.
Draw the bode plot of $KP(s)$ (he didn't draw it, but use matlab)
We get: $\omega_{PM} = 6$ rad/s, $P_M = 18°$.

B. We need at least $45 - 18 = 27$ °of phase added.
Since a lead controller also adds gain at ω_m this well shift ω_{PM}. Since this usually decreases P_M, we add some margin:

$$\begin{aligned}\phi_{max} &= 27 + 27(0.1)\\ &= 30\end{aligned}$$

We use the $0.1 = 10\%$ approximation because we're engineers, and approximate the buffer using that.
Calculate α

$$\begin{aligned}\alpha &= \frac{1+\sin\phi_{max}}{1-\sin\phi_{max}}\\ &= 3\end{aligned}$$

C. Ensure that the gain crossover frequency is exactly ω_m to have the max phase addition.
The lead controller increases the gain at ω_m by as follows:

$$20\log\sqrt{\alpha} = 4.77\text{dB at } \omega = \omega_m$$

From the bode plot of $KP(s)$, we see:

$$20 \log |KP(j\omega)| = 4.77\text{dB at } \omega = 8.4\text{rad/s}$$

We set $\omega_m = 8.4$ rad/s, so we can calculate T.

$$\begin{aligned} \frac{1}{T\sqrt{\alpha}} &= \omega_m \\ \rightarrow T &= 0.00687 \end{aligned}$$

Yields a P_M of 44°

- How is it that both a lead & lag controller can be used to meet the same specs? We only had specs on low frequency performance and P_M. We had no specs on BW.
- Lead Controller Design Algorithm:

A. Use FVT to get K or pick K to get desired bandwidth

B. Draw Bode plot of $KP(s)$ (We'll use this to do lookups later)

C. Find ω_{PM} and PM

D. Determine the amount of phase to add:

$$\phi_{max} = P_M^{\text{desired}} - P_M + \delta$$

δ is the buffer.

E. Find alpha

$$\alpha = \frac{1 + \sin \phi_{max}}{1 - \sin \phi_{max}}$$

F. Solve for ω_m
Solve for the ω_m that matches the following:

$$\begin{aligned} 20 \log |KP(j\omega)| &= -20 \log \sqrt{\alpha} \\ \omega_m &= \frac{1}{T\sqrt{\alpha}} \end{aligned}$$

G. Verify the graph in Matlab.

ii. Lead vs PD [7.4.2]:

$$\begin{aligned} PD_{\text{ideal}} &= K_P + K_D s \\ &= K_P \left(\frac{K_D}{K_P} s + 1 \right) \end{aligned}$$

We can view PD_{ideal} as a lead controller $C(s)$ with a tiny T and a large α.

$$C(s) = K \frac{\alpha T s + 1}{T s + 1}$$

In fact, since a PD controller can't be implemented exactly, the "practical" PD controller is actually just a lead controller.
TODO: insert 7-3a (bode plot of ideal PD)

(e) Lead-Lag Compensation [7.5]

Sometimes not all specs can be made by a lead or lag. We combine them. Lag is used to reduce the high frequency gain, lead is used at mid-frequency to boost P_M.

Lead-lag alleviates the weakness of individual lead and lag controllers, and is able to handle a richer class of specs.

TODO: insert 7-3a (Lead-lag controller)

$$\begin{aligned} C(s) &= KC_1(s)C_2(s) \\ &= K\left(\frac{\alpha_1 T_1 s + 1}{T_1 s + 1}\right)\left(\frac{\alpha_2 T_2 s + 1}{T_2 s + 1}\right) \end{aligned}$$

Where C_1 is the lead, and C_2 is the lag. $\alpha_1 > 1$, $0 < \alpha_2 < 1$

Typical block diagram for lead-lag controller: TODO: insert 7-3b

The typical pole/zero map of a lead-lag controller: TODO: insert 7-4a

The corresponding bode plot of $C_1(s)C_2(s)$: TODO: insert 7-4b

In the case that we don't have an $\omega_{BW}^{desired}$ spec: (Case 1)

- Specs:
  i. Closed-loop stability
  ii. $|e_{ss}| \leq e_{ss}^{max}$
  iii. $P_M \geq P_M^{min}$
- Algorithm:
  i. Use FVT to pick K

$$\begin{aligned} e_{ss} &= f(K) \\ &= e_{ss}^{max} \\ \rightarrow K &= \text{something} \end{aligned}$$

  ii. Divide the P_M spec into two roughly equal parts:

$$P_M^{min} = P_{M_1} + P_{M_2}$$

  In the example, we pick nice, round values of $P_{M_2} = 20$, $P_{M_1} = 25$

  iii. Design the lag controller to get $P_M = P_{M_2}$
    A. Bode plot of $KP(s)$
    B. Frequency at which $P_M = P_{M_2} + \delta$ ($\delta = 5$ °)

$$\omega = \text{some number from the bode plot}$$

    C. Pick α_2:

$$\alpha_2 = \frac{1}{|KP(j\omega)|}$$

    D. Pick T_2:

$$\begin{aligned} \frac{10}{\alpha_2 T_2} &\leq \omega \\ \rightarrow T_2 &\geq \frac{10}{\alpha\omega} \end{aligned}$$

    He usually picks $T_2 = \frac{10}{\alpha\omega}$

iv. Design the lead controller (for the partially compensated system from the previous step) to get $P_M \geq P_M^{min}$

A. Draw the bode plot of $KC_2(s)P(s)$.
Note: He mentioned that he won't make us do 3 bode plots in a question, because it takes too long. This algo takes 3 bode plots.

B. From the plot, find ω_{PM}. From that, find P_M.

C. Pick ϕ_{max}

$$\begin{aligned}\phi_{max} &= P_M - P_{M_1} + \delta \\ &= 30\end{aligned}$$

He iterated a few times, and found a δ.
Note: We're expected to understand when to put a buffer, but not to solve it.

D. Find α_1

$$\alpha_1 = \frac{1 + \sin \phi_{max}}{1 - \sin \phi_{max}}$$

E. Find the frequency where $20 \log |KC_2P(j\omega_m)| = -20 \log \sqrt{\alpha_1}$: Use the most recent bode plot, this value is ω_m

F. Use ω_m to find T_1:

$$\begin{aligned}\omega_m &= \frac{1}{T_1\sqrt{\alpha_1}} \\ T_1 &= \frac{1}{\omega_m\sqrt{\alpha_1}}\end{aligned}$$

G. Overall, we get:

$$C_1 = \frac{1 + \alpha_1 T_1 s}{1 + T_1 s}$$

We set the lag first because the lead controller "flattens" the phase which means the crossover frequency will be small (low bandwidth makes a slower system)
In the case that we have an $\omega_{BW}^{desired}$ spec: (Case 2)

- Specs:
  i. Closed-loop stability
  ii. $|e_{ss}| \leq e_{ss}^{max}$
  iii. $P_M \geq P_M^{min}$
  iv. $\omega_{BW} = \omega_{BW}^{desired}$
- Algorithm:
  i. Use FVT to pick K
  ii. Draw the bode plot of $KP(s)$ (lead controller)
  iii. Determine the amount of phase to add at $\omega_{BW}^{desired}$ so P_M spec is met if $\omega_{PM} = \omega_{BW}^{desired}$. This additional phase is what the lead controller adds.

$$\alpha_1 = \frac{1 + \sin \phi_{max}}{1 - \sin \phi_{max}}$$

iv. Make sure ϕ_{max} is added at $\omega_{BW}^{desired}$

$$\begin{aligned}\omega_M &= \frac{1}{T_1\sqrt{\alpha}} \\ &= \omega_{BW}^{desired}\end{aligned}$$

v. Draw a bode plot of $KC_1(s)P(s)$. (Lag controller)
We don't want the phase reduction of the lag controller, so we want the crossover frequency to match the actual bw.

vi. Use the lag controller to reduce the gain and ensure that $\omega_{BW}^{desired} = \omega_{PM}$. (Determine α_2)

$$\alpha_2 = ?$$

vii. Ensure that by reducing the gain, we haven't affected the phase addition we did earlier. Choose T_2:

$$\frac{10}{T_2\alpha_2} \leq \omega_{BW}^{desired}$$

i. Lead-lag vs PID [7.5.1]
Ideal PID:

$$\begin{aligned}C(s) &= K_P + \frac{K_I}{s} + K_D s \\ &= \frac{K_D s^2 + K_P s + 1}{s}\end{aligned}$$

.1 Tutorial Notes

1. Unitiy feedback system with plant:

$$P(s) = \frac{1}{s+1}$$
$$C(s) = K$$

Find the minimum $K > 0$ such that $e_{ss} \leq 0.01$ for:

(a) $r(t) = 1(t)$

(b) $r(t) = cos(\omega t)$, $0 < \omega < 4$

Solve for stability first.

$$\begin{aligned}\Pi(s) &= N_c N_p + D_c D_p \\ &= s + (K+1) \rightarrow K + 1 \qquad > 0 \\ K &> -1\end{aligned}$$

Solve for the e_{ss} next:

$$\begin{aligned}e_{ss} &= \frac{1}{1 + P(0)C(0)} \\ &= \frac{1}{1+K} \\ &\leq 0.01 \\ K &\geq 99\end{aligned}$$

Solve for the $E(s)$ (error) next:

$$\begin{aligned}E(s) &= \frac{1}{1 + P(s)C(s)} R(s) \\ &= \frac{s+1}{s+1+k} \frac{s}{s^2+\omega^2} \\ \rightarrow \frac{E}{R} &= G(s) \\ &= \frac{s+1}{s+1+k} \\ \rightarrow R(s) &= \frac{s}{s^2+\omega^2}\end{aligned}$$

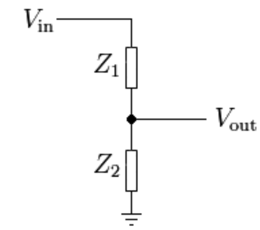
Solve for e_{ss} for the cos now:

$$\begin{aligned}e_{ss} &= |G(j\omega)| \cos \omega t + \angle G(j\omega) \\ |G(j\omega)| &= \frac{|j\omega + 1|}{|j\omega + 1 + K} \\ &= \frac{\sqrt{\omega^2 + 1}}{\sqrt{\omega^2 + 1 + 2K + K^2}} \\ &\leq 0.01\end{aligned}$$

...

.2 Circuits Review

- Voltage Divider Rule:

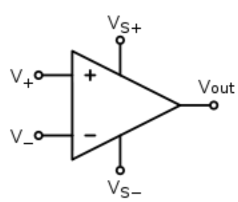


$$V_{out} = \frac{Z_2}{Z_1 + Z_2} V_{in}$$

- Capacitors, Inductors, and Resistors

Name	Voltage ($v(t)$)	Current ($i(t)$)	Laplace	Z-domain
Capacitor	$v(t)$	$C\frac{dv}{dt}$	$sV(s) = I(s)$	$Z_c = \frac{1}{sC}$
Inductor	$L\frac{di}{dt}$	$i(t)$	$V(s) = sI(s)$	$Z_L = sL$
Resistor	IR	$\frac{V}{R}$	$V(s) = RI(s)$	$Z_r = R$

- Ideal Op-Amps:



$$V_+ = V_-$$

$$I_{out} = \text{arbitrary?}$$

$$I_+ = I_- = 0$$

.3 Math Review

- Suprema:
  The maxima of an infinitely long set of elements.
- Convolution

$$f(t) * g(t) = \int_0^t g(t-\tau)f(\tau)d\tau = \int_0^t f(t-\tau)g(\tau)d\tau$$

$$= \mathcal{L}^{-1}\{G(s)F(s)\} = \mathcal{L}^{-1}\{F(s)G(s)\}$$

- Matrix Multiplication:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} ax+by \\ cx+dy \end{bmatrix}$$

- Matrix Inverse:

$$M = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$\Rightarrow M^{-1} = \frac{1}{det(M)} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

- Taylor series of $f(n)$:

  Note: $f^{(n)}(a)$ is the n^{th} derivative of f at the point a.

$$f(x) = \Sigma_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n$$

- Jacobians:

  If we have a function $F : \mathbb{R}^n \to \mathbb{R}^m$, then we have the Jacobian matrix J of F:

$$\frac{\partial F}{\partial X} = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \cdots & \frac{\partial F_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial F_m}{\partial x_1} & \cdots & \frac{\partial F_m}{\partial x_n} \end{bmatrix}$$

  i.e.

$$\frac{\partial f}{\partial y} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} \end{bmatrix}$$

- Laplace ($\mathcal{L}$):

  1. $\mathcal{L}\{f+g\} = \mathcal{L}\{f\} + \mathcal{L}\{g\}$
  2. $\mathcal{L}\{af\} = a\mathcal{L}\{f\}$
  3. $\mathcal{L}\{\frac{df}{dt}\} = s\mathcal{L}\{f\} - f'(0)$
  4. $\mathcal{L}\{f * g\} = \mathcal{L}\{f\}\mathcal{L}\{g\}$
  5. $\mathcal{L}\{\int_0^t f(\tau)d\tau\} = \frac{1}{s}\mathcal{L}\{f\}$
  6. If $\lim_{t\to\infty} f(t)$ exists and is finite, then $\lim_{t\to\infty} f(t) = \lim_{s\to 0} sF(s)$

- Imaginary numbers and phasors: ($z \in \mathbb{C}$)

  1. $z = x + jy \Rightarrow |z| = \sqrt{x^2+y^2}$

2. $z = |z|e^{j\angle z}$
3. $z = |z|\cos(\angle z) + j|z|\sin(\angle z)$
4. $cos(\omega t) = \frac{e^{j\omega t}+e^{-j\omega t}}{2}$
5. $sin(\omega t) = \frac{e^{j\omega t}-e^{-j\omega t}}{2j}$
6. $\angle\frac{ab}{c} = \angle a + \angle b - \angle c$

- Log Laws:

  1. $\log(\left|\frac{ab}{c}\right|) = \log(|a|) + \log(|b|) - \log(|c|)$

- Identifying 1st vs 2nd Order ODEs:
  (k, τ, ω_n are constants)

  - 1st order:

$$\tau\dot{y} + y = Ku \Rightarrow^{\mathcal{L}} G(s) = \frac{K}{1+s\tau}$$

or

$$\dot{x} = \frac{-x}{\tau} + \frac{Ku}{\tau}$$
$$y = x$$

  - 2nd order:

$$\ddot{y} + 2\zeta\omega_n\dot{y} + \omega_n^2 y = K\omega_n^2 u \Rightarrow^{\mathcal{L}} G(s) = \frac{K\omega_n^2}{s^2 + 2s\omega_n\zeta + \omega_n^2}$$

or

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = -\omega_n^2 - 2\zeta\omega_n x_2 + K\omega_n^2 u$$
$$y = x_1$$

- Z-transform:
  Z^{-1} corresponds to going back in time.

.4 Physics Review

- Newton's 2nd Law

$$M\ddot{y} = F$$

$$\Sigma F = M\ddot{y}$$