MECH 466 Summary

Reese Critchlow

Second-Order Systems

For a system for which the transfer function takes on the form:

$$G(s) = \frac{X(s)}{F(s)} = \frac{1}{ms^2 + Ds + k},$$

or in the time domain:

$$m\ddot{x} + D\dot{x} + kx = f(t)$$

we can extract key parameters.

Solving Second Order ODEs

In the time domain, we can we can decompose the problem into a homogenous and a particular solution:

$$x(t) = x_p(t) + x_c(t)$$

and then just solve accordingly. However, in most cases, we will be analyzing the step response of systems. Generally, this comes in the following form:

$$\ddot{x} + 2\zeta\omega_n\dot{x} + \omega_n^2 = \frac{k}{m}u(t).$$

Or, in the frequency domain:

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

given that $\omega_n = \sqrt{\dfrac{k}{m}}$. All of these systems have (rougly) the following form of solution in the time domain:

$$x(t) = Ae^{s_1 t} + Be^{s_2 t}$$

Hence, from this, we get four following response types:

1. Undamped: $\zeta = 0$

   $s^2 + w_n^2 = 0 \implies s_{1,2} = \pm j\omega_n$

   $\implies x(t) = 1 - \cos(\omega_n t).$

2. Overdamped: $\zeta > 1$

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

   $\implies x(t) = 1 + Ae^{s_1 t} + Be^{s_2 t}$

3. Critical Damping: $\zeta = 1$

   $s_{1,2} = -\zeta\omega_n$

   $\implies x(t) = 1 + (A + Bt)e^{-\zeta\omega_n t}$

4. Underdamped: $\zeta < 1$

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

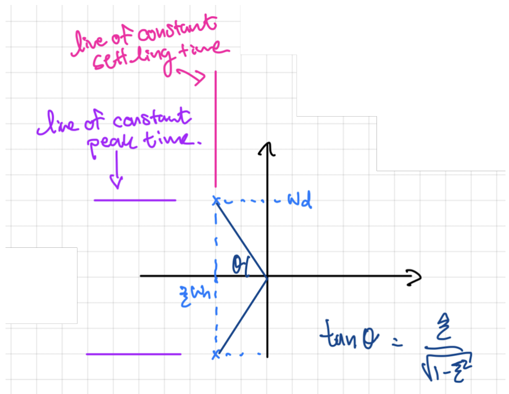
   $\implies x(t) = 1 - \dfrac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}}\cos(\omega_d t - \phi)$

   where: $\phi = \arctan\left(\dfrac{\zeta}{\sqrt{1-\zeta^2}}\right)$ and $\omega_d = \omega_n\sqrt{1-\zeta^2}$.

From all of these parameters, we can extract the following metrics:

- Peak Time: $\tau_p = \dfrac{\pi}{\omega_d}$
- Settling Time: $\tau_p = \dfrac{4}{\zeta\omega_n}$
- Percentage Overshoot: $\%\text{OS} = 100e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}}$

From this, we can extract the following information about our system on pole graphs:



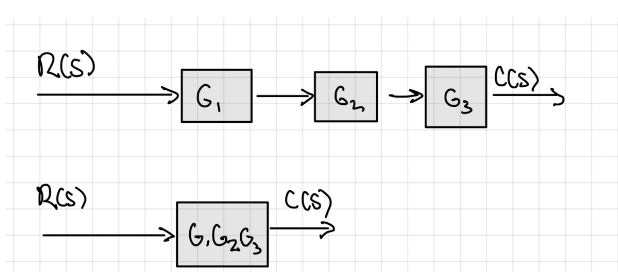
Poles

Assorted Rules about Poles:

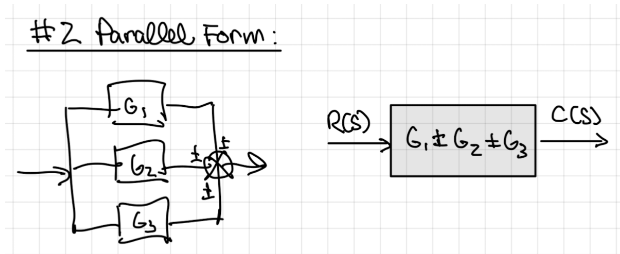
1. A pole in the input function generates the forced response.
2. A pole in the transfer function generates the natural response.
3. The further to the left a pole is on the negative real axis, the faster the transient response will decay to zero.
4. The zeroes and poles generate the amplitude for both the forced and natural responses.
5. The position of the poles determines the systems peak time and settling time.
6. Repeated poles on the imaginary axis guarantee stability.

Reduction of Systems

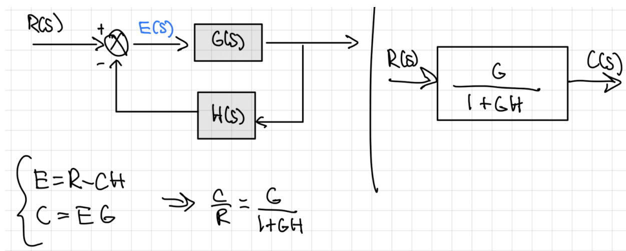
1. Cascade:



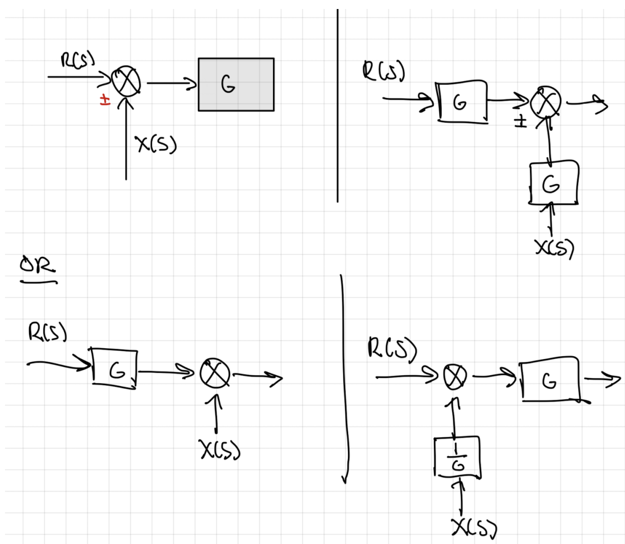
2. Parallel:



3. Feedback:



4. Moving Blocks:



Routh-Hurwitz Criterion

For a system with the following transfer function:

$$H(s) = \frac{N(s)}{a_4 s^4 + a_3 s^3 + a_2 s^2 + a_1 s + a_0}$$

s^4	a_4	a_2	a_0
s^3	a_3	a_1	0
s^2	$-\frac{\begin{vmatrix} a_4 & a_2 \\ a_3 & a_1 \end{vmatrix}}{a_3} = b_1$	$-\frac{\begin{vmatrix} a_4 & a_0 \\ a_3 & 0 \end{vmatrix}}{a_3} = b_2$	0
s^1	$-\frac{\begin{vmatrix} a_3 & a_1 \\ b_1 & b_2 \end{vmatrix}}{b_1} = c_1$	0	0
s^0	$-\frac{\begin{vmatrix} b_1 & b_2 \\ c_1 & 0 \end{vmatrix}}{c_1} = d_1$	0	0

Routh-Hurwitz Edge Cases:

1. Zero in first column with non-zero numbers to the right.
   (a) Substitute the 0 for some ϵ.
   (b) At the end, take the limit $\epsilon \to 0$.
   (c) If there are sign changes, the system is unstable and the number of ORHP poles are equal to the number of sign changes.
   (d) If there are no sign changes, the system is marginally stable and there are at least two poles on the imaginary axis.
2. Zero for all entries in a row.
   (a) Using the powers from the column labels, take the derivative of the last non-zero row. This is called the *auxillary polynomial*.
   (b) Just yoss the resultant coefficients into the next row.
   (c) As you were.
   (d) If there is a zero row and no sign changes, then all the poles in the auxillary polynomial are on the imaginary axis and are symmetric. They also correspond to the roots of the auxillary polynomial.
   (e) Hence, the system is marginally stable.

Useful Methods of System Analysis

Lagrangian Mechanics

$$\frac{d}{dt}\left(\frac{dT}{d\dot{q}}\right) + \frac{dU}{dq} + \frac{dR}{d\dot{q}} = \text{external forces}$$

Motors

Motors have the following transfer function:

$$\frac{\Theta(s)}{E(s)} = \frac{\kappa}{s(s+a)}.$$

Which can be found using the following equations:

$$\frac{T_{\text{stall}}}{e_a} = \frac{k_t}{R} \qquad k_b = \frac{e_a}{\omega_{\text{no-load}}} \qquad k = \frac{k_t}{RJ} \qquad a = \frac{1}{J}\left(\frac{k_t}{R}k_b + D\right)$$

— midterm cutoff —

Steady-State Analysis

Final Value Theorem: For some function $f(t)$ with Lagrange transform $\mathcal{L}\{f(t)\} = F(s)$, then it follows that:

$$\lim_{t\to\infty} f(t) = \lim_{s\to 0} sF(s).$$

Generally, steady-state analysis is done on closed-loop systems with unity feedback (Figure 1 with $H(s) = 1$). Hence, we can derive the error in the following manner:

$$E(s) = \frac{R(s)}{1 + G(s)H(s)}$$

And hence, to find the steady state error, take the limit as $\lim_{s\to 0} sE(s)$.

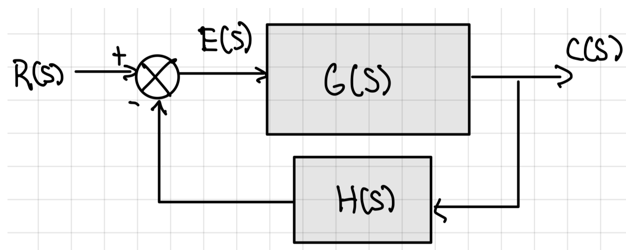


Figure 1: Generic System

Given this, for different inputs, there are conditions on the number of "pure integrators" in a system (i.e. the degree of the denominator). The condition is that:

For an input to a system $R(s) = \frac{1}{s^n}$, then the denominator of the system must have degree $\geq n$.

Static Error Constants: For a system with unity feedback ($H(s) = 1$), then we can define the "static error constants" for it. They are as follows:

- Position Constant: $K_p = \lim_{s\to 0} G(s)$
- Velocity Cosntant: $K_v = \lim_{s\to 0} sG(s)$
- Acceleration Cosntant: $K_a = \lim_{s\to 0} s^2 G(s)$

As these constants increase, the value of the steady-state error decreases. One can apply these generally to systems with $H(s) \neq 1$ by doing the analysis with $G(s) \to G(s)H(s)$.

Root Locus

The Root Locus is a graphical presentation of how the locations of the closed-loop poles for a given system change as a system parameter is varied. It is a powerful method of analysis and design for stability and tranisent response.

Take for example some system with a gain coefficient K. As one changes K, one can analyze how the poles of the system change. By plotting those poles as K changes, one can analyze the behaviour of the system under changing K. By plotting the root locus, one can answer several questions about how changing K impacts the system, such as:

- How does the overshoot OS change?
- How does ω_n and ω_d change?

Hacking: It wouldn't be a mech course if there wasn't some strange way of trying to shortcut doing the full procedure. We can determine whether a point is on the root locus of a system, and what K it corresponds to in the following way for some system $T(s)$:

$$T(s) = \frac{KG}{1 + KGH}$$

1. The poles occur for $1+KGH = 0$, and thus, $KGH = -1$. This corresponds to a phase of $(2k+1)\cdot 180°$, $k \in \mathbb{Z}$. Thus, it follows that:

$$|KGH| = 1 \qquad \angle KGH = (2k+1)\cdot 180° K = \frac{1}{|GH|}$$

2. In order to verify that it is on the root locus, check whether:

$$\angle KGH \in \{(2k+1)\cdot 180° : k \in \mathbb{Z}\}$$

3. Then, if so, find the K for which the point corresponds to:

$$K = \frac{1}{|GH|}$$

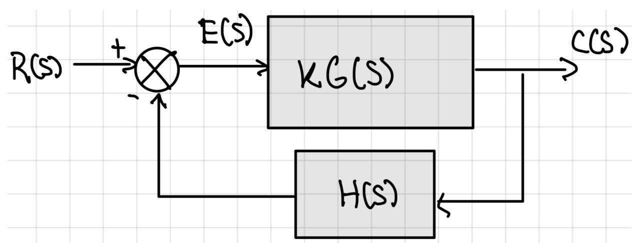


Figure 2: Generic System with Gain K

Bode Plots/Frequency Response

We call the transfer function of a system with $s = j\omega$ the *frequency response function* (FRF). It can help to determine how a system responds to different frequency inputs. The FRF has two main components that are analyzed:

- Gain: $M = |G(j\omega)| = \sqrt{\text{Re}^2 + \text{Im}^2}$
- Phase: $\phi = \angle G(j\omega) = \arctan\left(\frac{\text{Im}}{\text{Re}}\right)$

These can both be plotted in a Bode Plot:

1. Plot 1 (Gain):
   - x axis: input frequency w, log scale
   - y axis: gain in decibels: $M' = 20\log(M)$
2. Plot 2 (Phase):
   - x axis: input frequency w, log scale
   - y axis: phase ϕ, normal scale, degrees.

Properties of Bode Plots:

- Multiplication $\rightarrow$ Summation: if $G(s) = N(s)P(s)$ then:
  - Gain Plot is the sum of the $N(s)$ and $P(s)$ gain plots.
  - Phase plot is the sum of the $N(s)$ and $P(s)$ gain plots.
- Inverse: if $G(s) = 1/N(s)$ where $N(s)$ is some known function:
  - Gain Plot is the negative $N(s)$ gain plot.
  - Phase Plot is the negative $N(s)$ gain plot.

Nyquist Plots

The Nyquist plot examines how the open-loop transfer functions transforms a contour spanning the right-half plane in the s domain into the "F" domain. The open-loop transfer function is given by $G(s)H(s)$ for some generic system like in Figure 1. This is useful computing the *Nyquist Criterion*.

Nyquist Criterion: Let:

- Z be the number of closed-loop poles in the right-half plane.
- P be the number of open-loop poles in the right-half plane.
- N be the number of counter-clockwise revolutions around the point $(-1, 0)$ in the Nyquist plot.

Then, if $Z = 0$, then the system is stable.

Plotting the Nyquist Plot:

1. Find the following four points:
   (a) $\omega = 0$
   (b) $\omega = \infty$
   (c) Imaginary Intercepts (Re $= 0$)
   (d) Real Intercepts (Im $= 0$)
2. Draw the points, and them connect them in the following order, assuming no duplicate points:
   (a) $\omega = 0$
   (b) Imaginary Intercepts (Re $= 0$)
   (c) Real Intercepts (Im $= 0$)
   (d) $\omega = \infty$
3. Mirror it across the real axis.

Then, using the open-loop transfer function, one can calcualte P, and then deduce N from the Nyquist Plot. If the loop passes through -1, then it is generally decided that the system is "marginally stable".

Controllers

A control system consists of certain subsystems assembled for the purpose of obtaining a desired output, with desired performance, given a specific input. One can move the position of zeroes and poles of a system to change its performance. One can also *compensate* a system by adding extra zeroes and poles. This is called controller/compensator design.

In a PID system, we can do the following:

- Multiply the error by k_p before feeding it back into the system.

- Take the cumulative total error over a period and multiply it by a constant k_i.
- Take the rate of change in error and multiply it by a constant k_d.

Mathematically, this looks like:

$$\mathcal{L}\left\{k_p e(t) + k_i \int e(t)dt + k_d \frac{de(t)}{dt}\right\}$$
$$\left[k_p + \frac{k_i}{s} + k_d s\right] E(s)$$

Generally, it is known that:

- Differentiation improves transient response (overshoot, settling time).
- Integration improves steady-state error.

Hence, one can induce this with different types of controllers:

- PI Controller: Proportional control and integral control

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

  - Generates a pole at the origin and a zero at $-\dfrac{k_i}{k_p}$.
  - It improves the steady state error.

- PD Controller: Proportional and derivative control

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

  - Generates a zero at $-\dfrac{k_p}{k_d}$
  - It improves the transient response (OS, settling time)

Another class of controllers are *Lead/Lag Controllers.* They have the general form of:

$$\frac{s + z_c}{s + p_c}$$

- Lag Controllers:
  - Zero and pole close to the origin: $z_c, p_c \approx 0$.
  - Zero smaller than pole: $|z_c| < |p_c|$.
  - Improves steady-state error.
- Lead Controllers:
  - Zero and pole far from the origin $|z_c|, |p_c| \gg 0$.
  - Zero larger than pole: $|z_c| > |p_c|$.
  - Improves transient response.

Design Process:

- An engineer always designs first for Transient Response, and then Steady-State error after.
- The process is meant to be iterative, i.e. T, SS, T, SS,...

- If one first designs a PD controller, then a PI controller, it is called a PID controller.
- If one first designs a Lead and then a Lag compensator, the resulting controller is called a Lead-Lag compensator.

We can also rewrite everything to be in terms of Lag/Lead compensators:

Controller	**Function**
PI	$K\frac{s+z_c}{s}$
Lag	$K\frac{s+z_c}{s+p_c}$
PD	$K(s+z_c)$
Lead	$K\frac{s+z_c}{s+p_c}$
PID	$K\frac{(s+z_{\text{lag}})(s+z_{\text{lead}})}{s}$
Lag-Lead	$K\frac{(s+z_{\text{lag}})(s+z_{\text{lead}})}{(s+p_{\text{lag}})(s+p_{\text{lead}})}$

We can also do this all in the form of op-amps:

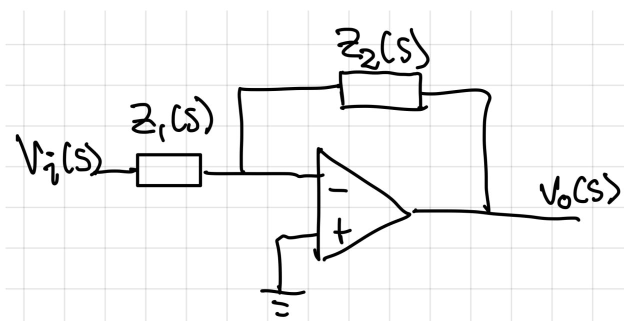
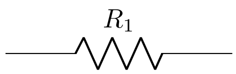
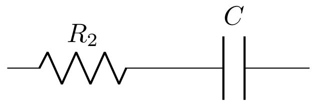
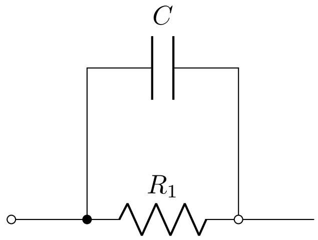
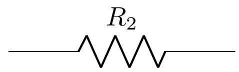
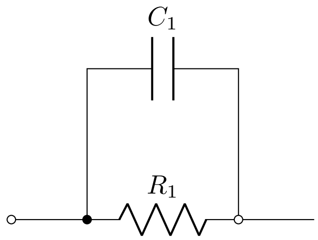
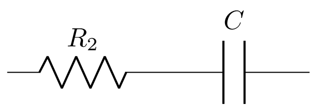


Figure 3: Generic Opamp

And so to create controllers out of impedances Z, we can do the following:

Controller Type	Z1 Element	Z2 Element	Function
PI			$-\frac{R_2}{R_1}\frac{\left(s+\frac{1}{R_2C}\right)}{s}$
PD			$-R_2C\left(s+\frac{1}{R_1C}\right)$
PID			$-\left[\left(\frac{R_2}{R_1}+\frac{C_1}{C_2}\right)+R_2C_1s+\frac{1}{R_1C_2s}\right]$

and also this (I'm lazy)

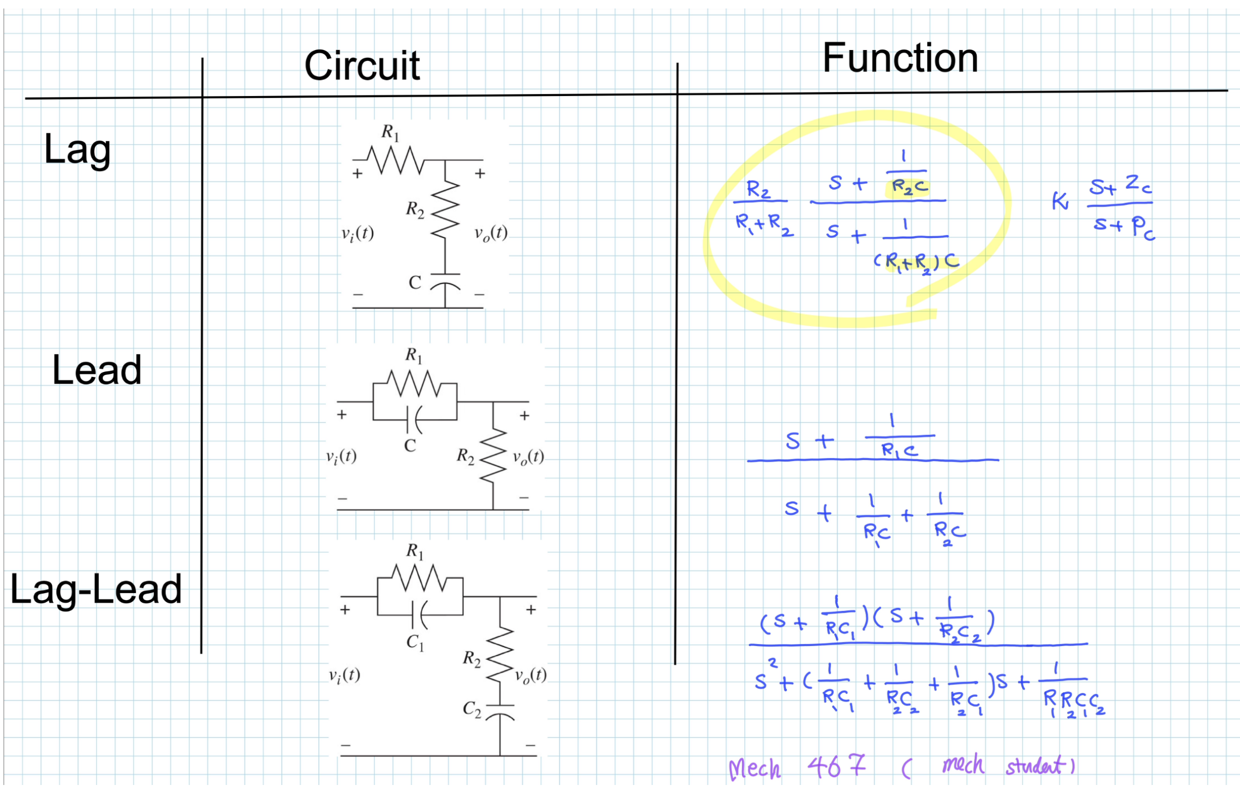


Figure 4: Lazy