SFWR ENG 3DX4 Summary

Instructor: Dr. Lawford
Course: SFWR ENG 3DX4

Math objects made using MathType.

Table of Contents

Note: the following summaries may be useful:

- SFWR ENG 2MX3
- ENGINEER 3N03
- TRON 3TA4

I may review to clarify or correct, but mostly I will omit those things.

Introduction to Systems

Systems can be represented by **block diagrams** to make it easier to marginalize the different parts of the systems.

Transducer: converts any form of energy to electrical signals

Laplace

Useful for…

Time begins when your signal begins

$$h(t) = \begin{cases} 0, & t < 0 \\ 1, & t \geq 0 \end{cases}$$

Initial conditions:

1. $c(0)$

Time domain (t): variables are <u>lower case</u>, e.g. $f(t)$
Frequency domain (s): variables are <u>upper case</u>, e.g. $F(s)$

Transfer function:

When doing the inverse Laplace, it's useful to break your fractions up so that you can

Strictly Stable: it will eventually get back to the initial position
Marginally Stable:
Unstable: it will progressively get worse

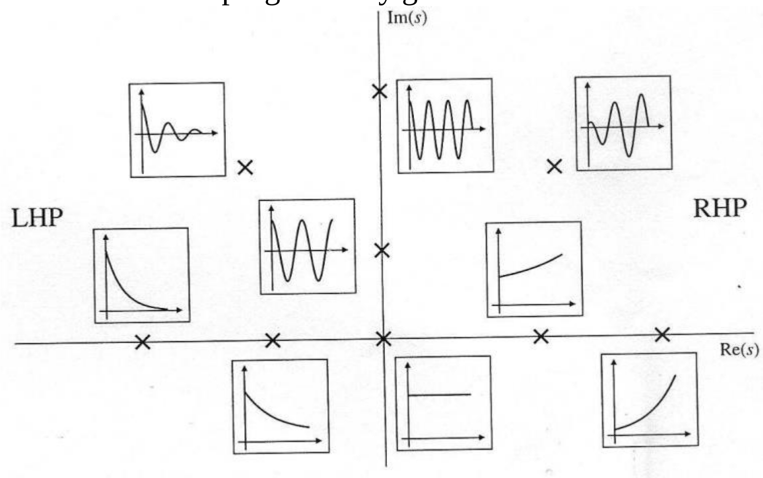


Figure 2.5 from Dorf and Bishop, *Modern Control Systems (10th Edition)*, Prentice-Hall, 2004.

Transfer Functions

Electrical

Component stuff

Impedence: $Z = \frac{V(s)}{I(s)}$

$Z_R = R$

$Z_L = sL = j\omega L$

$Z_C = \frac{1}{sC} = \frac{-j}{\omega C}$

Current

$i_R = \frac{1}{R}$

$i_L = L\,\frac{di}{dt}$

$i_C = C\,\frac{dv}{dt}$

Voltage

$v_R = Ri(t)$

$v_L = L\frac{di}{dt}$

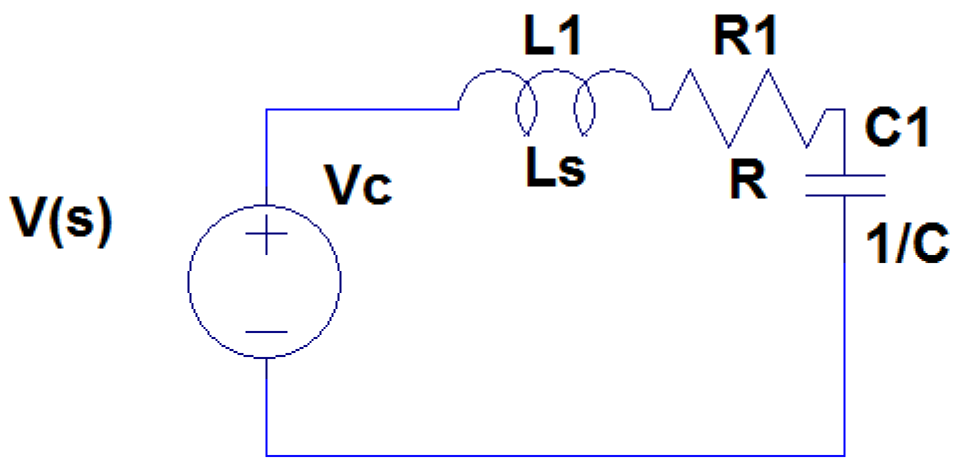
$v_C = \frac{1}{C}\int_0^1 i(\tau)d\tau$

admittance:

$Y(s) = \frac{I(s)}{V(s)} = \frac{1}{R} = G$

$V_c(s) = I(s)\frac{1}{Cs}$

$I(s) = \frac{V(s)}{L_s + R + \frac{1}{Cs}}$



Mesh Analysis

Add the voltages, where V = IZ

Noodal Anal

1. Identify nodes
2. Represent currents in terms of voltage

Cramer's Rule

$$x_1 = \frac{\det(A_1)}{\det(A)}, x_2 = \frac{\det(A_2)}{\det(A)}, \ldots, x_n = \frac{\det(A_n)}{\det(A)}$$

$$V_C(s) = \overbrace{H(s)}^{\text{transfer function}}\frac{1}{Cs}$$

OP-Amps

Mechanical

Translational systems:
Rotational Systems:

Newton's Second Law of Motion: $\Sigma f = Ma$

$$Z_m(s) = \frac{F(s)}{X(x)}$$

$$\begin{aligned} f(t) &= Ma(t) \\ &= M\frac{\mathrm{d}^2 x}{\mathrm{d}t^2} \end{aligned}$$

Translational Systems

For sure make a free-body diagram

e.g.

$$d_1 + 7v_1 + 2x_1 + 5v_1 = 2x_2 + 5v_2$$

$$d_2 + 2x_2 + 5v_2 = 2x_1 + 5v_1 + F(t)$$

$$v_1 = \frac{\mathrm{d}x_1}{\mathrm{d}t}$$

$$d_1 = \frac{\mathrm{d}v_1}{\mathrm{d}t}$$

$$\dot{x}_1 = v_1$$

$$\dot{v}_1 = d_1$$

$$\dot{x}_2 = v_2$$

$$\dot{v}_2 = d_2$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{v}_1 \\ \dot{x}_2 \\ \dot{v}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -2 & -12 & 2 & 5 \\ 0 & 0 & 0 & 1 \\ 2 & 5 & -2 & -5 \end{bmatrix} \begin{bmatrix} x_1 \\ v_1 \\ x_2 \\ v_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} f(t)$$

$$\text{Output} = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ v_1 \\ x_2 \\ v_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}$$

All inductances are in the opposite direction of the applied force

Spring

Spring is like a capacitor

Force displacement: $f(t) = Kx(t)$

Viscous Damper

Using viscous fluid to slow something down

Viscous Damper is like a resistor

Force displacement: $f(t) = f_v \frac{\mathrm{d}x(t)}{\mathrm{d}t}$

$F(s) = F_v Xs$

Mass

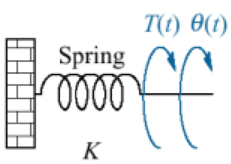
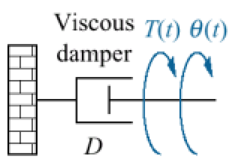
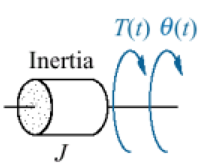
Mass is like a inductor

Force displacement: $f(t) = M\frac{\mathrm{d}^2 x(t)}{\mathrm{d}t^2}$

F(s) = MXs^2

Rotational Systems

Impedence: $Z_m(s) = \frac{T(s)}{\theta(s)}$

Component	Torque-angular velocity	Torque-angular displacement	Impedance $Z_M(s) = T(s)/\theta(s)$
	$T(t) = K\int_0^t \omega(\tau)\,d\tau$	$T(t) = K\theta(t)$	K
	$T(t) = D\omega(t)$	$T(t) = D\frac{d\theta(t)}{dt}$	Ds
	$T(t) = J\frac{d\omega(t)}{dt}$	$T(t) = J\frac{d^2\theta(t)}{dt^2}$	Js^2

2. Each θ is on an inertia block. The impedences connected to the motion at θ include the impedences directly to the left and right of the inertia block.
3. When finding the sum of impedences between 2 θ's only count the impedences on wires that don't go through other θ's, i.e. 0 if no direct connection
4. When there is a torque, but no inertial block, draw a fake inertial block

$$\begin{bmatrix}\text{Sum of}\\\text{impedances}\\\text{connected}\\\text{to the motion}\\\text{at }\theta_1\end{bmatrix}\theta_1(s) - \begin{bmatrix}\text{Sum of}\\\text{impedances}\\\text{between}\\\theta_1\text{ and }\theta_2\end{bmatrix}\theta_2(s)$$

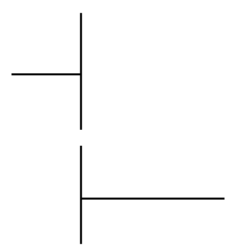
$$- \begin{bmatrix}\text{Sum of}\\\text{impedances}\\\text{between}\\\theta_1\text{ and }\theta_3\end{bmatrix}\theta_3(s) = \begin{bmatrix}\text{Sum of}\\\text{applied torques}\\\text{at }\theta_1\end{bmatrix}$$

$$- \begin{bmatrix}\text{Sum of}\\\text{impedances}\\\text{between}\\\theta_1\text{ and }\theta_2\end{bmatrix}\theta_1(s) + \begin{bmatrix}\text{Sum of}\\\text{impedances}\\\text{connected}\\\text{to the motion}\\\text{at }\theta_2\end{bmatrix}\theta_2(s)$$

$$- \begin{bmatrix}\text{Sum of}\\\text{impedances}\\\text{between}\\\theta_2\text{ and }\theta_3\end{bmatrix}\theta_3(s) = \begin{bmatrix}\text{Sum of}\\\text{applied torques}\\\text{at }\theta_2\end{bmatrix}$$

Motors and Gears

1. Pick an end of the system to use as a reference frame. Choose the easiest one and walls don't move.
2. Represent T

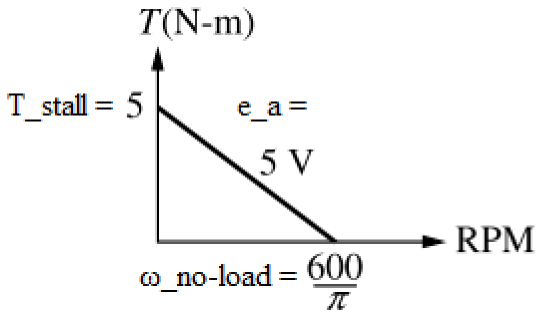


Meshing Gears are represented in the following way:
[N]: number of teeth
Let's assume var_1 = before and var_2 = after.
When gears are lined up $\frac{T_2}{T_1} = \frac{\theta_1}{\theta_2} = \frac{N_2}{N_1}$
Applied Armature Voltage [e_a]: a.k.a. input voltage
Armature Resistance [R_a]:
Motor Torque Constant [K_t]:
Back EMF Constant [K_b]:
No load speed [$\omega_{\text{no-load}}$]: when the voltage line touches the x-axis

$$\omega_{\text{no-load}} = \frac{e_a}{K_b}$$

Stall torque[T_{stall}]: when angular velocity reaches 0, i.e. y-intercept if equation is given.

$$T_{\text{stall}} = \frac{K_t}{R_a} e_a$$



[J_a]: any J on the same line, including a motor
[J_L]: load J

[J_m]: $J_m = J_a + J_L\left(\frac{N_1}{N_2}\right)^2$

[D_m]: coefficient of viscous dampening $D_m = D_a + D_L\left(\frac{N_1}{N_2}\times\frac{N_3}{N_4}\right)^2$

$$T_e = T\left(\frac{N_2}{N_1}\right)$$

$$T(s)\left(\frac{N_2 N_4}{N_1 N_3}\right) = \theta_{destination}\left(J_{eq}s^2 + D_{eq}s\right)$$

$$\frac{\theta_m(s)}{E_a(s)} = \frac{K_t/(R_a J_m)}{s\left[s + \frac{1}{J_m}\left(D_m + \frac{K_t K_b}{R_a}\right)\right]}$$

Hints:

- If you have a spring and / or a damper in series, the wire between them rotates independently

Degrees of Freedom

How to calculate

1. count the number of masses/moments of inertia blocks
2. find any hidden inertia blocks

Signals

Transducer: anything that converts energy to electrical energy
Transmitter: long distances

Unstable systems have ∞ steady state error
Steady-state error [e_∞]:

$$e_\infty = \lim_{t\to\infty} e(t)$$

Final Value Theorem

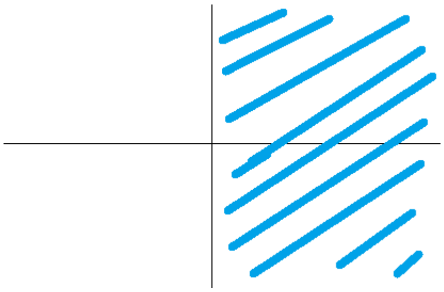
Final value theorem: finds steady state error

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

So $e_\infty = \lim_{s \to 0} sF(s)$ and you're given F(s), so just multiply by s and find the limit.

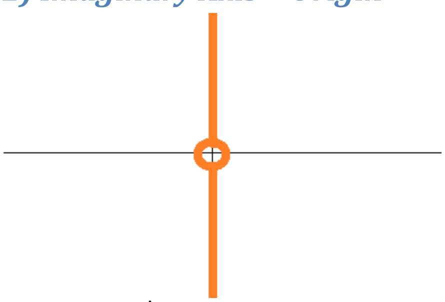
There are limitations as to where you can use this theorem. It is dependent on the location of the poles.

1) Right half plane



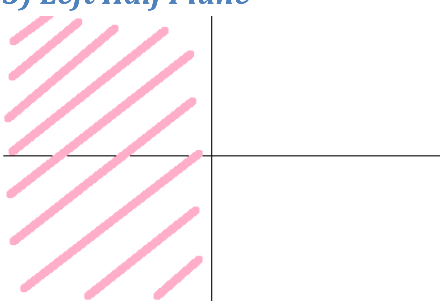
System is unstable: $e^+ \rightarrow \infty$

2) Imaginary Axis – Origin



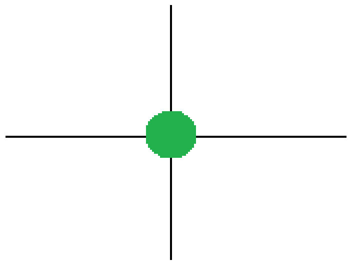
Unstable: $e^i \rightarrow$ Oscillatory system, so limit will be average, i.e. midpoint

3) Left Half Plane



Stable: e^- converges to 0, but makes transfer function 0 for every single pole

4) Origin



Stable: integrator, i.e. 1/s, so $\lim_{s\to 0}\frac{s}{s}=1$

Don't use this theorem if any poles are 1 or 2.

Transient Response

Transient Response:

Rise time [T_r]: time between 10% and 90% of final value [c_{final}]
Peak time [T_p]: time it takes to get to highest peak [$c_{\max}$]
Settling time [T_s]: how long it takes to get to the steady state within ±2%

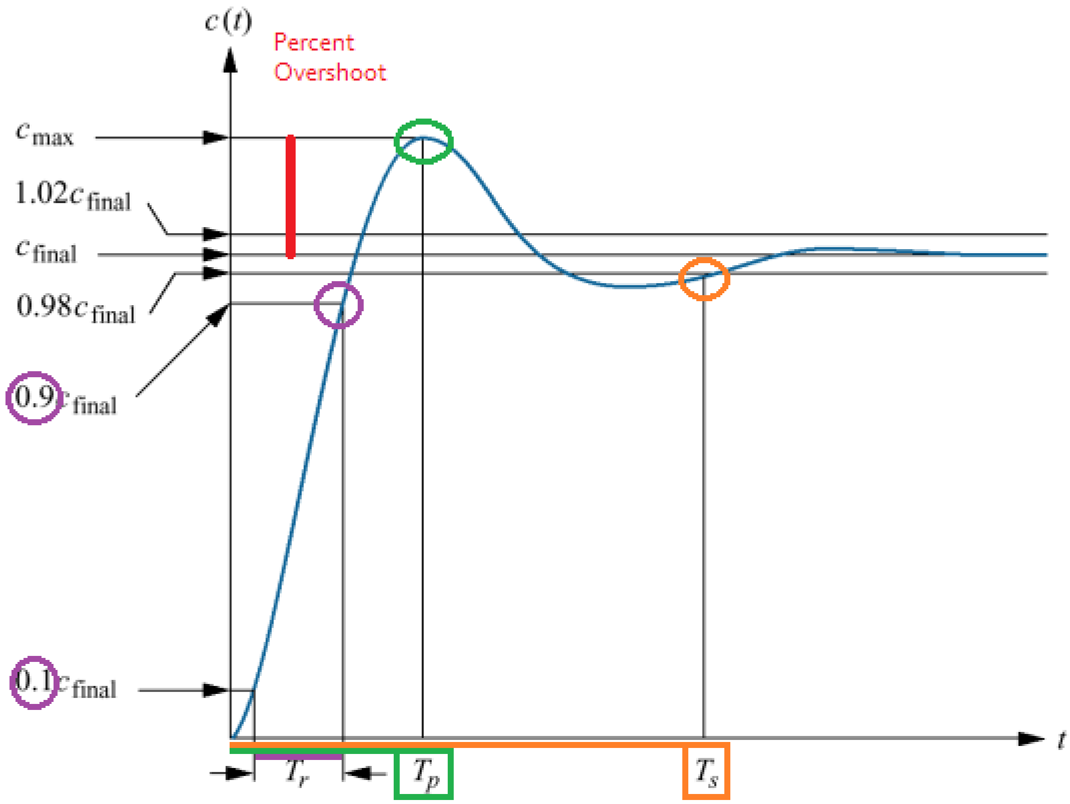
$$T_s = \frac{4}{\zeta\omega_n}$$

Damping Ratio [ζ]: $\zeta = \frac{-\ln(\%\text{OS})}{\sqrt{\pi^2+\ln^2(\%\text{OS})}}$

Percent overshoot [%OS]: how much further is the peak from the final

$$\%\text{OS} = \frac{c_{\max}-c_{\text{final}}}{c_{\text{final}}}\times 100\%$$

Time Constant [τ]: the time it takes the system's step response to reach 1 – 1/e = 63.2% of c_{final}
Second-order:



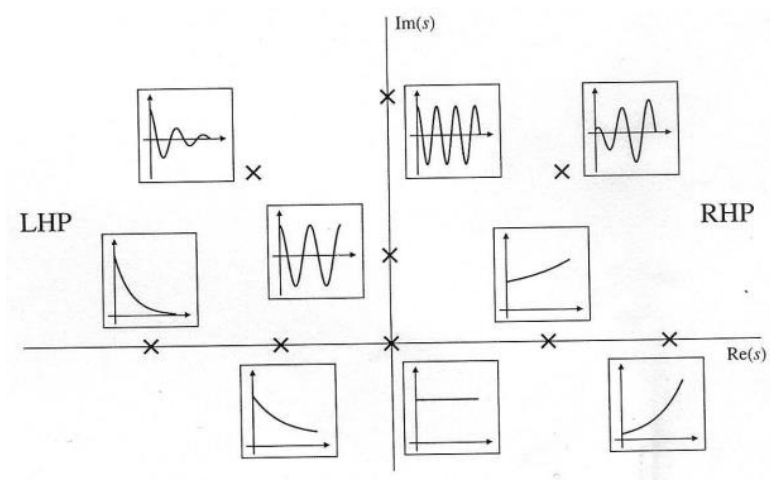


Figure 2.5 from Dorf and Bishop, *Modern Control Systems (10th Edition)*, Prentice-Hall, 2004.

$K = c_{\text{final}} \times a$

For each pole,

$G(s) = \frac{K_1}{s+a_1} + \frac{K_2}{s+a_2} + \text{etc}, a = \frac{1}{\tau}$

Forced response: when a = 0
Natural response: when a > 0

Nonminium-phase system: Initially the system starts in the wrong direction, then stabilizes at the right place

Non-/Linear Systems

5. Op Amps are linear
6. If you don't have enough voltage, your motor magnets won't have enough power to switch poles, so they require a minimum voltage

You can't model non-linear systems, until you linearize it. To do this, we find the slope and approximate the equation of the line, using y=mx+b

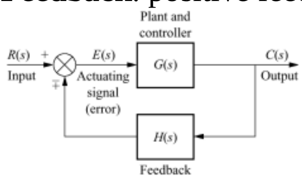
Block Diagrams

A way of representing a system

Summing junction: could be an X or +, but usually an X in this course

Cascade: subsystems in series are multiplied
Parallel: parallel subsystems have a *summing junction* at the end, so you just add everything together

Feedback: positive feedback is bad

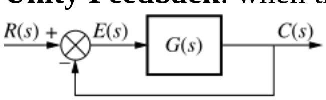


Positive: $\frac{G(s)}{1-G(s)H(s)}$

Negative: $\frac{G(s)}{1+G(s)H(s)}$

Simplification:

Unity Feedback: when the feedback path has multiplicative value of 1



State Space Equations

Yeah, you think you know them from 2MX3, but you don't really know them. Apparently the ABCD variables actually have names.

- **System Matrix [A]**:
- **Input Matrix [B]**:
- **Output Matrix [C]**:
- **Feedforward Matrix [D]**:

$$G(s)=\mathbf{C}(s\mathbf{I}-\mathbf{A})^{-1}\mathbf{B}$$

Transfer Function -> State Space

Phase Variable Approach:
The *n* state variables will consist of:

- *y*
- the derivatives of *y*

$A^{-1} = \frac{1}{\det(A)}\operatorname{adj}(A)$

adjoint(A) = (cofactor matrix(A))^T

$$A=\begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & 9\end{bmatrix}$$

$$\text{cofactor(A)} = \begin{bmatrix}A_{11} & A_{21} & A_{31}\\ A_{12} & A_{22} & A_{32}\\ A_{13} & A_{23} & A_{33}\end{bmatrix}$$

$$A_{11}=\begin{vmatrix}5 & 6\\ 8 & 9\end{vmatrix}=45-48=-3$$

Stability

Root Mean Square (RMS): the effective DC value of an AC current, by finding a special average

$$f(t)_{\text{RMS}}=\sqrt{T\int_0^T \left(f(t)\right)^2 \text{d}t}$$

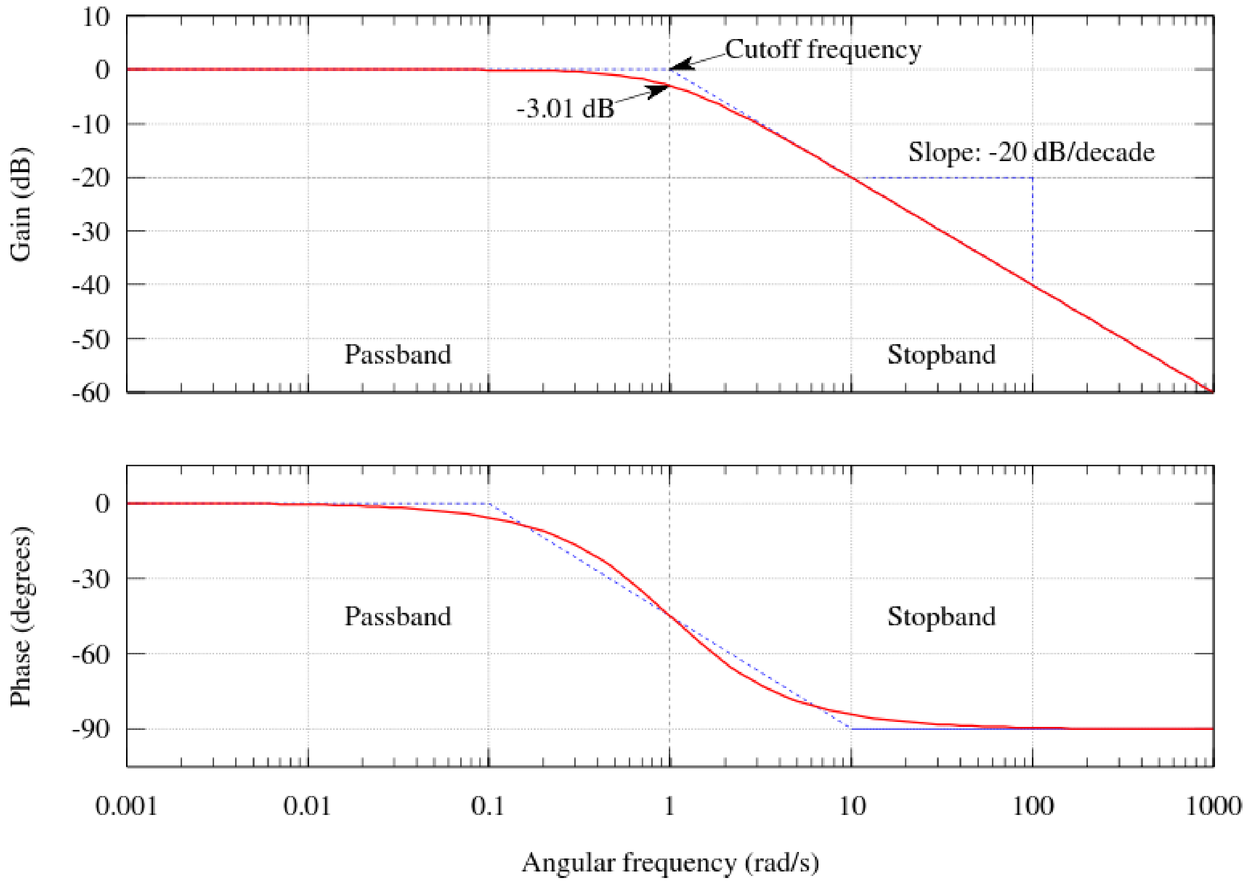
Gain [K]:

Bode plot: graph of frequency response of a system, using a *phase graph* and *gain graph*

1. Find all poles, zeroes, and K value
2. Represent each pole and zero in terms of a fraction added to a 1, i.e. $(s+5)\Rightarrow 5\left(\frac{s}{5}+1\right)$

Cutoff Frequency: (a.k.a. *breakaway point*) low pass filter is said to pass frequencies lower than ω_c and reject those that are higher than ω_c. In other words, the pass(ing) band is $\omega < \omega_c$.
How to find from chart:

- magnitude = −3Db
- phase = −45°
- $\omega_c = \omega((½)^{½} \times \text{amplitude}_{max}) = \omega(0.707 \times A_{max})$



Types:

1. Constant(K): M = 20log(K), Ø = 0
2. Integration(1/s): M = –20log(|jω|)
3. Derivative(s):
4. 1st order lag $\left(\dfrac{1}{\left(\frac{s}{\omega_n}+1\right)}\right)$: focus on poles
5. 1st order lead $\left(\frac{s}{\omega_n}+1\right)$: focus on zeroes
6. 2nd order lag $\left(\dfrac{1}{\frac{s^2}{\omega_n^2}+2\frac{\zeta s}{\omega_n}+1}\right)$: focus on poles
7. 2nd order lead $\left(\frac{s^2}{\omega_n^2}+2\frac{\zeta s}{\omega_n}+1\right)$: focus on zeroes

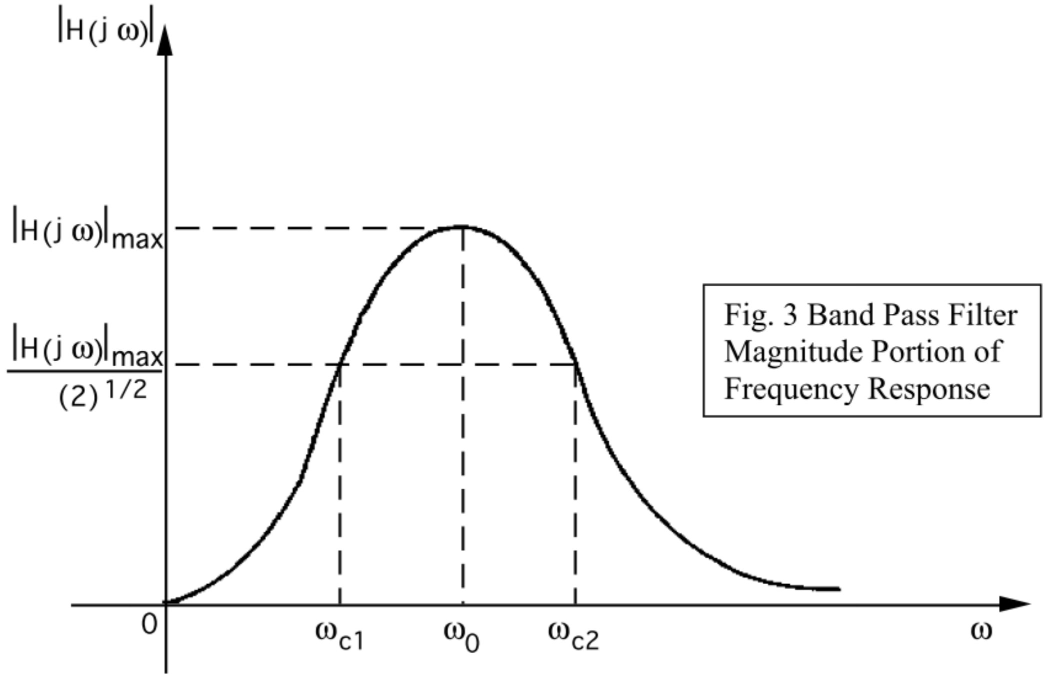
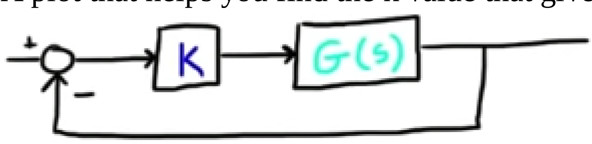


Fig. 3 Band Pass Filter Magnitude Portion of Frequency Response

Root Locus

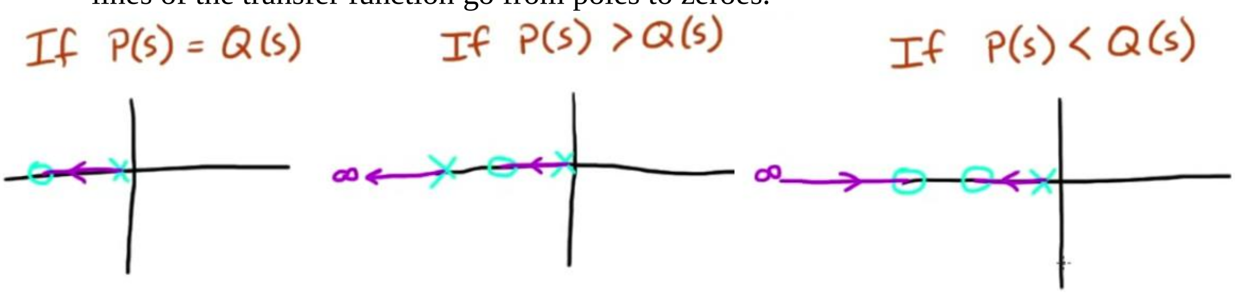
A plot that helps you find the k value that gives your system your desired level of stability.



Branch: the lines on the root locus that represent the stable range of the transfer function, starting at a pole

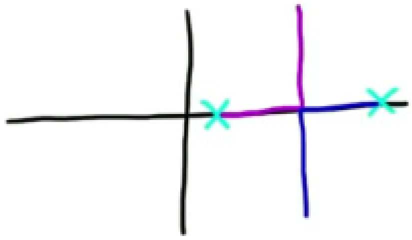
- i.e. open-loop-zero
- branches can be endless, going to infinite

1. Number of poles/zeros (whichever is greater) = number of branches
2. As K moves from 0→∞, roots move from poles of G(s) to zeros of G(s). In other words, lines of the transfer function go from poles to zeroes.

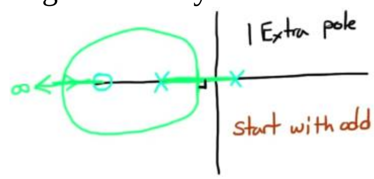


3. Roots that are complex, i.e. not on the real axis, always come in pairs of positive and negative, i.e. above and below at the same r location. In fact, the path is completely mirrored.

4. The path the roots take will never cross itself, unless 2 roots meet, in which case the lines



   break out
5. Right-to left priority.
6. Lines only break out at 90°.
7. Poles with no zeros on the left will go to infinity. Zeros with no poles on the right will



   have lines coming from infinity.
8. To find the position of the asymptote (8.27): $\boxed{\sigma_a = \frac{\sum \text{poles} - \sum \text{zeros}}{\#\text{poles} - \#\text{zeros}}}$, where you're summing the positions of the poles and the angles of the asymptotes are $\boxed{\theta_a = \frac{(2k+1)\pi}{\#\text{poles} - \#\text{zeros}}}$
9. Hint: if there are 2/+ lines going to infinite, Σroots = constant

You can have multiple σ's: ||poles − zeros|| = # of σ's
Break out points:
Solve for σ, where:

$$0 = \frac{\mathrm{d}}{\mathrm{d}\sigma}(\text{denominator, i.e. poles})$$

Break in points:
Solve for σ, where

$$0 = \frac{\mathrm{d}}{\mathrm{d}\sigma}(\text{numerator, i.e. zeros})$$

Sometimes you'll need to put extra poles/zeros to get the proper stability because K isn't enough.

Cancelling zeroes / poles

e.g.)

If you have k in a weird random spot in the denominator, you need a way to access it.

$$\frac{1}{s^3 + 4s^2 + ks + 1}$$

1) Group all k terms $\Rightarrow s^3 + 4s^2 + ks + 1 = 0$
2) Divide by non-k terms

$$\frac{s^3+4s^2+1}{s^3+4s^2+1}+\frac{ks}{s^3+4s^2+1}=0$$

$$1+k\frac{s}{s^3+4s^2+1}=0$$

Second Order Approximation

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

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

$$\Sigma\text{zero angles}-\Sigma\text{pole angles}=(2k+1)\pi$$

$$\sum_{i=1}^{m}\angle(s+z_i)-\sum_{j=1}^{n}\angle(s+p_j)=(2k+1)\pi$$

For dumb people who don't use radians, use 180 degrees instead of π. I hate degrees though, so I'm not even going to write it out.

Compensator Zero [z_c]: eventually cancelled out,
[θ_{zc}]: the angle that is the result of the z_c.

Graph method

noodles

Routh-Hurwitz Table

s^3	$a_3 = 1$	$a_1 = 80$	0
s^2	$a_2 = 18$	$a_0 = k$	0
s^1	$b_1=\frac{-\begin{vmatrix}a_3 & a_1\\ a_2 & a_0\end{vmatrix}}{a_2}=\frac{1440-k}{18}$	$\frac{\begin{vmatrix}1 & 0\\ 18 & 0\end{vmatrix}}{k}$ 0	0
s^0	$k = 6$	0	0

$$b_i=\frac{-\begin{vmatrix}a_n & a_{n-1}\\ a_{n-2i} & a_{n-2i-1}\end{vmatrix}}{a_{n-1}}$$, where the det is the 4 values in the square above it

$$c_i=\frac{-\begin{vmatrix}a_{n-1} & a_{n-2i-1}\\ b_1 & b_{i+1}\end{vmatrix}}{b_1}$$

Think: each determinant has the first column in the left column and the column to its right in the right column

a_n	a_{n-2}	…	a_1
a_{n-1}	a_{n-3}	…	a_0

If the first column has a sign change as you down, your system is unstable.

[M_G]: Gain Magnitude

$$M_G = \frac{\prod_{i=1}^{m} |s + z_i|}{\prod_{i=1}^{n} |s + p_i|}$$

Hint: you can take out any common factor from any row

Exceptions

- Zero in first column of any row
  - Represent the zero by epsilon
  - Test to see the sign when epsilon is + and –
  - If both make the system unstable, it's unstable. Otherwise, it's stable!
- Zero in any column if there is a nonzero value in the column to the right
  - Represent by epsilon
- Row full of zeroes
  - Form a polynomial using the entries in the row above zeros
  - Find the derivative of this polynomial
  - Replace the row with the coefficients of the derivative

Phase Variable

Phase-variable representation: pretty much the same as state space representation

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -8 & -6 & -4 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} r(t)$$

$$\mathbf{y} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \mathbf{x} + 0$$

Phase margin [Φ]: $\Phi = \Omega$

Steady-State Error

Difference between the final value of the step response and the predicted final value

There are 3 static error constants (K_p, K_v, K_a)

- Step:
  - $y = m$
  - $R(t) = u(t)$
  - $\frac{1}{s}$
  - $e_{\text{step}} = \lim_{s \to 0} \frac{s\left(\frac{1}{s}\right)}{1 + G(s)} = \lim_{s \to 0} \frac{1}{1 + G(s)} = \frac{1}{1 + K_p}$
  - K_p (position constant) $= \lim_{s \to 0} G(s)$

- ○
- Ramp:
  - $y = mx$
  - $R(t) = tu(t)$
  - $\frac{1}{s^2}$
  - $e_{\text{ramp}}(\infty) = \lim_{s\to 0} \frac{s\left(\frac{1}{s^2}\right)}{1+G(s)} = \lim_{s\to 0} \frac{\frac{1}{s}}{1+G(s)} = \boxed{\lim_{s\to 0} \frac{1}{sG(s)}} = \frac{1}{K_v}$
  - K_v (velocity constant) $= \lim_{s\to 0} sG(s)$
- Parabolic:
  - $y = mx^2$
  - $R(t) = t^2 u(t)$
  - $\frac{n!}{s^{n+1}} = \frac{2!}{s^3}$
  - $e_{\text{parabolic}}(\infty) = \lim_{s\to 0} \frac{s\left(\frac{2!}{s^3}\right)}{1+G(s)} = \lim_{s\to 0} \frac{\frac{2!}{s^2}}{1+G(s)} = \boxed{\lim_{s\to 0} \frac{1}{s^2 G(s)}} = \frac{1}{K_a}$
  - K_A (acceleration constant) $= \lim_{s\to 0} s^2 G(s)$

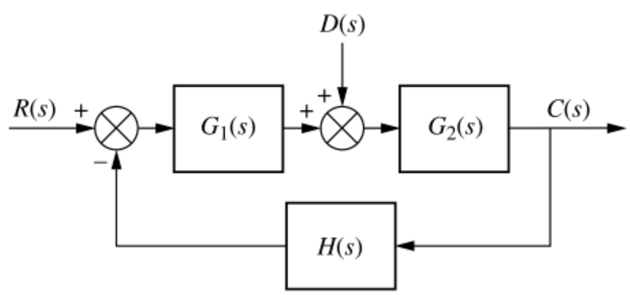


FIGURE 7.17 Nonunity feedback control system with disturbance

$$e_\infty = \lim_{s\to 0} s\mathrm{E}(s)$$
$$= \lim_{s\to 0} s\left\{\left[1 - \frac{G_1(s)G_2(s)}{1+G_1(s)G_2(s)H(s)}\right]R(s) - \left[\frac{G_2(s)}{1+G_1(s)G_2(s)H(s)}D(s)\right]\right\}$$

Disturbance [D]:

Proportional Integral (PI):

PI: K_p & K_i enter system

Proportional Derivative (PD): a.k.a. proportional differential

Proportional-Integral-Derivative (PID): K_p & K_i & K_d enter the system

Natural Frequency [ω_n]: only in 2nd order

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

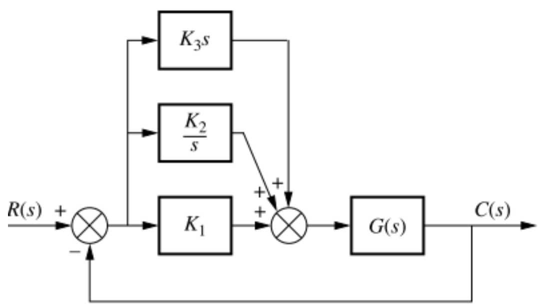
[M_p]: $M_P=\dfrac{1}{2\zeta\sqrt{1-\zeta^2}}$

Note: If your gears are vibrating, your PID is probably too high

Compensator: a way of altering your system to your design constraints, mathematically

Some types of compensators:

Compensator equation: a.k.a. controller equation $K_D s^2+K_p s+K_i=K_C(s+z_1)(s+z_2)$
[K_d]: the derivative of the error of the system (K_3)
[K_p]: a proportion that is multiplied by the error of the system (K_1)
[K_i]: an integral of the error of the system (K_2)



Canonical Form

Phase Variable

Phase variable: $\dot{\vec{x}}=\begin{bmatrix}0 & 1\\ -11 & -7\end{bmatrix}\vec{x}+\begin{bmatrix}0\\1\end{bmatrix}r(t)$

$$\mathbf{y}=\begin{bmatrix}1 & 0\end{bmatrix}\mathbf{x}+0$$

$$\mathbf{x}=\begin{bmatrix}x_1\\x_2\end{bmatrix}$$

rotate matrices=>

Controller Canonical: $\dot{\mathbf{x}}=\begin{bmatrix}-7 & 11\\ 1 & 0\end{bmatrix}\mathbf{x}+\begin{bmatrix}1\\0\end{bmatrix}r(t)$

$$y=\begin{bmatrix}0 & 1\end{bmatrix}\mathbf{x}$$

$$\mathbf{x} = \begin{bmatrix} x_2 \\ x_1 \end{bmatrix}$$

Control

$C_m = \begin{bmatrix} B & AB & A^2B & \dots & A^{n-1}B \end{bmatrix}$, where n is the rank of B

$$\mathbf{x} = A\mathbf{x} + Br(t)$$

$$y = C\mathbf{x} + D$$

Rank: length of the smallest size when it's a triangular matrix

Observer

Lala

Digital Control

Convert frequency domain → z domain and apply the above formulas, you have 80% of the question

Sampling Period [T]:
Position, **V**elocity, **A**cceleration

$$K_P = \lim_{z \to 1} G(z)$$

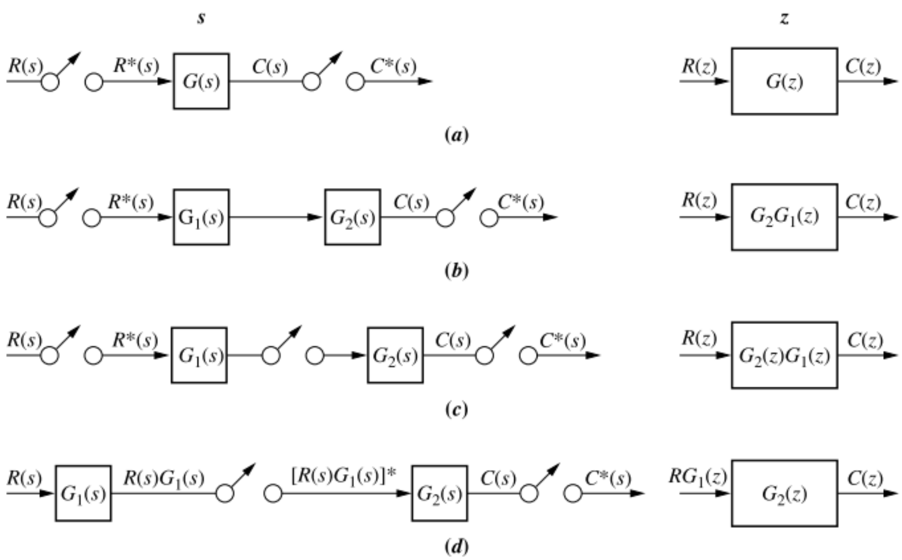
$$K_V = \frac{1}{T} \lim_{z \to 1} (z-1) G(z)$$

$$K_A = \frac{1}{T^2} \lim_{z \to 1} (z-1)^2 G(z)$$

$$e_P^*(\infty) = \frac{1}{1+K_P}$$

$$e_V^*(\infty) = \frac{1}{K_V}$$

$$e_A^*(\infty) = \frac{1}{K_A}$$



Correction: $G_2G_1(z) = G_2(G_1(z))$

Z Transform

1. Convert from frequency domain → time domain, using Laplace^{-1}.
2. Convert from time domain to the complex frequency domain (z), using Z Transform.

TABLE 13.2 z-transform theorems

	Theorem	Name
1.	$z\{af(t)\} = aF(z)$	Linearity theorem
2.	$z\{f_1(t) + f_2(t)\} = F_1(z) + F_2(z)$	Linearity theorem
3.	$z\{e^{-aT}f(t)\} = F(e^{aT}z)$	Complex differentiation
4.	$z\{f(t - nT)\} = z^{-n}F(z)$	Real translation
5.	$z\{tf(t)\} = -Tz\dfrac{dF(z)}{dz}$	Complex differentiation
6.	$f(0) = \lim\limits_{z\to\infty} F(z)$	Initial value theorem
7.	$f(\infty) = \lim\limits_{z\to 1}(1 - z^{-1})F(z)$	Final value theorem

Note: kT may be substituted for t in the table.

Zero-Order Hold (ZOH): Discretization

$$G(z) = \frac{z-1}{z}\mathcal{Z}\left\{\frac{G(s)}{s}\right\}$$

MATLAB Stuff

- angle:
- atan2:
- evalfr:
- `feedback`: for all your transfer function needs
- `pole(compensator)`: input

- `roots(polynomial)`: column vector of polynomial roots
- rlocus: graphs the root locus
- `sisotool(compensator)`:
- step(compensator): step input to compensator
- s=tf('s'): transfer function
- [num,den] = ss2tf(A, B, C, D, 1): State Space to transfer function, outputting
- zpk([zero1 zero2], [pole1 pole2], gain): zero pole gain