Bryan Melanson

How to Not Fail
Control Systems

While never going to class

Computer Engineering 2020 $\boxed{\mathcal{BM}}$

Contents

1 Modeling in the Frequency Domain **2**

2 Modeling in the Time Domain **3**

3 Time Response **4**

4 Reduction of Multiple Systems **5**

5 Stability **6**
5.1 Routh-Hurwitz Criteria 6
5.2 Routh-Hurwitz Special Cases 6

6 Steady State Errors **7**
6.1 Steady State Error for Unity Feedback Systems 8
6.1.1 Steady State Error in Terms of G(s) 9
6.2 Static Error Constants and System Type 9
6.3 Steady-State Error Specifications 10
6.4 Forming Equivalent Unity from Non-Unity Systems 10

7 Root Locus Techniques **11**
7.1 Sketching the Root Locus 11
7.1.1 Number of Branches 11
7.1.2 Symmetry 11
7.1.3 Real Axis Segments 11
7.1.4 Starting and Ending Points 11
7.1.5 Behavior at Infinity 12
7.2 Refining the Sketch 12
7.2.1 Breakaway/Break-in Point 12
7.2.2 $j\omega$-Axis Crossings 13
7.2.3 Angles of Departure and Arrival 14
7.3 Plotting and Calibrating the Root Locus 15
7.4 Generalized Root Locus 16
7.5 Positive Feedback Systems 16

8 Design via Root Locus **18**

9 Frequency Response Techniques **19**

10 Design via Frequency Response **20**

1 Modeling in the Frequency Domain

2 Modeling in the Time Domain

3 Time Response

4 Reduction of Multiple Systems

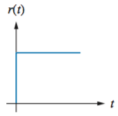
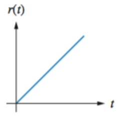
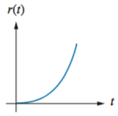
5 Stability

5.1 Routh-Hurwitz Criteria

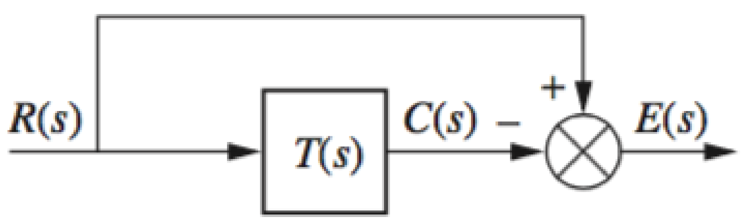
5.2 Routh-Hurwitz Special Cases

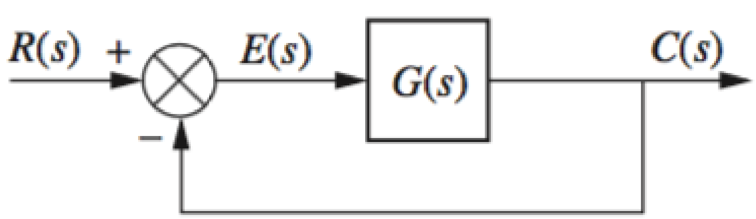
6 Steady State Errors

Steady State Error is defined as the difference between the input and output as t $\rightarrow \infty$. When testing for factors such as constant position, constant velocity and constant acceleration, inputs such as unit steps $u(t)$, ramps $r(t)$ and parabolas are used. This discussion is limited to stable systems.

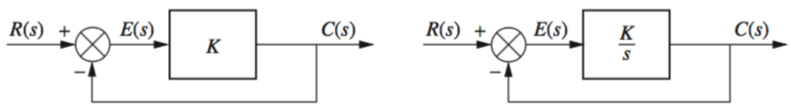
Waveform	Name	Physical interpretation	Time function	Laplace transform
	Step	Constant position	1	$\frac{1}{s}$
	Ramp	Constant velocity	t	$\frac{1}{s^2}$
	Parabola	Constant acceleration	$\frac{1}{2}t^2$	$\frac{1}{s^3}$

Most steady state errors $E(s)$ arise from the input and/or the configuration of the system, as seen in the diagrams below for general closed loop and unity feedback systems.





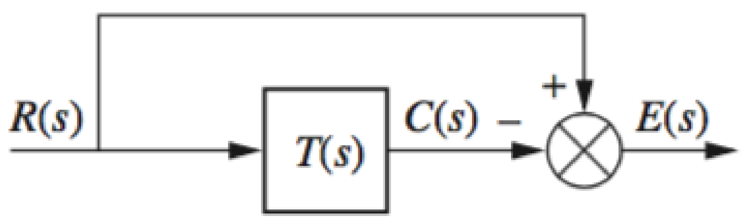
In the first case $E(s) = R(s) - C(s)$ is the error. If the input $R(s)$ is a step input, then $C(s)$ should $= R(s)$ and $E(s) = 0$. However, if gain K is introduced, $C(s) = KR(s)$ and $E(s)$ must be finite and non-zero.



From these systems we see that $C(s) = KE(s)$, or $E(s) = \frac{1}{K}C(s)$

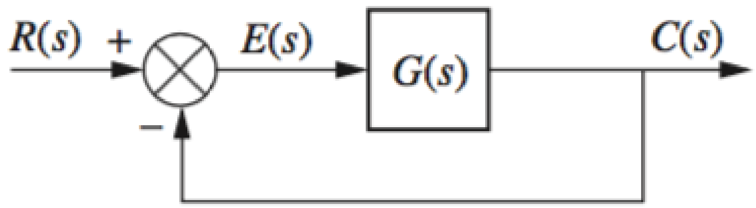
6.1 Steady State Error for Unity Feedback Systems

Steady-state error can be calculated from transfer function $T(s)$ or the open loop transfer function $G(s)$. Once $E(s)$ is found, the steady state error can be found using the *Final Value Theorem*, which states that the value at infinity is equal to the Laplace as $s \to 0$.



$$E(s) = R(s) - C(s)$$
$$C(s) = R(s)T(s)$$
$$E(s) = R(s)[1 - T(s)]$$
$$e(\infty) = \text{lim}_{s\to\infty} e(t) = \text{lim}_{s\to 0} sE(s)$$
$$e(\infty) = \text{lim}_{s\to 0} sR(s)[1 - T(s)]$$

6.1.1 Steady State Error in Terms of G(s)



$$E(s) = R(s) - C(s)$$
$$C(s) = E(s)G(s)$$
$$E(s) = \frac{R(s)}{1+G(s)}$$
$$e(\infty) = \lim_{s\to\infty} \frac{sR(s)}{1+G(s)}$$

6.2 Static Error Constants and System Type

The steady-state error for unit step inputs is

$e(\infty) = \frac{1}{1+\lim_{s\to 0} G(s)}$

The steady-state error for ramp inputs of unit velocity is

$e(\infty) = \frac{1}{\lim_{s\to 0} sG(s)}$

The steady-state error for parabolic inputs of unit acceleration is

$e(\infty) = \frac{1}{\lim_{s\to 0} s^2G(s)}$

The terms in the denominator are known as k_p, k_v, k_a, the **static error constants**, representing position, velocity and acceleration, respectively.

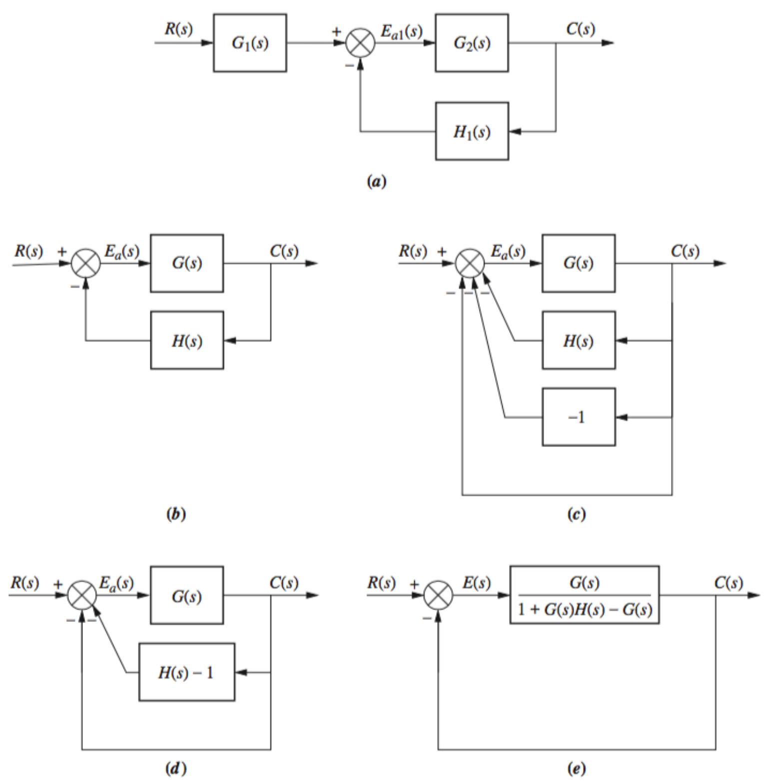
Systems can also be defined by **system type**. This defines the number of pure integrations in the forward path, assuming a unity feedback system. Increasing the system type decreases the steady-state error as long as the system is stable.

		Type 0		Type 1		Type 2	
Input	Steady-state error formula	Static error constant	Error	Static error constant	Error	Static error constant	Error
Step, $u(t)$	$\frac{1}{1+K_p}$	$K_p = \text{Constant}$	$\frac{1}{1+K_p}$	$K_p = \infty$	0	$K_p = \infty$	0
Ramp, $tu(t)$	$\frac{1}{K_v}$	$K_v = 0$	∞	$K_v = \text{Constant}$	$\frac{1}{K_v}$	$K_v = \infty$	0
Parabola, $\frac{1}{2}t^2u(t)$	$\frac{1}{K_a}$	$K_a = 0$	∞	$K_a = 0$	∞	$K_a = \text{Constant}$	$\frac{1}{K_a}$

6.3 Steady-State Error Specifications

The steady-state error is inversely proportional to the static error constant - the larger the constant, the smaller the steady-state erorr. Increasing gain increases the static error constant, thus, increasing the gain decreases the steady-state error if the system is stable.

6.4 Forming Equivalent Unity from Non-Unity Systems



7 Root Locus Techniques

The following sections apply to **Negative Feedback Closed Loop** systems.

7.1 Sketching the Root Locus

7.1.1 Number of Branches

The number of branches in a root locus equal the number of poles.

7.1.2 Symmetry

The root locus is symmetrical about the real axis.

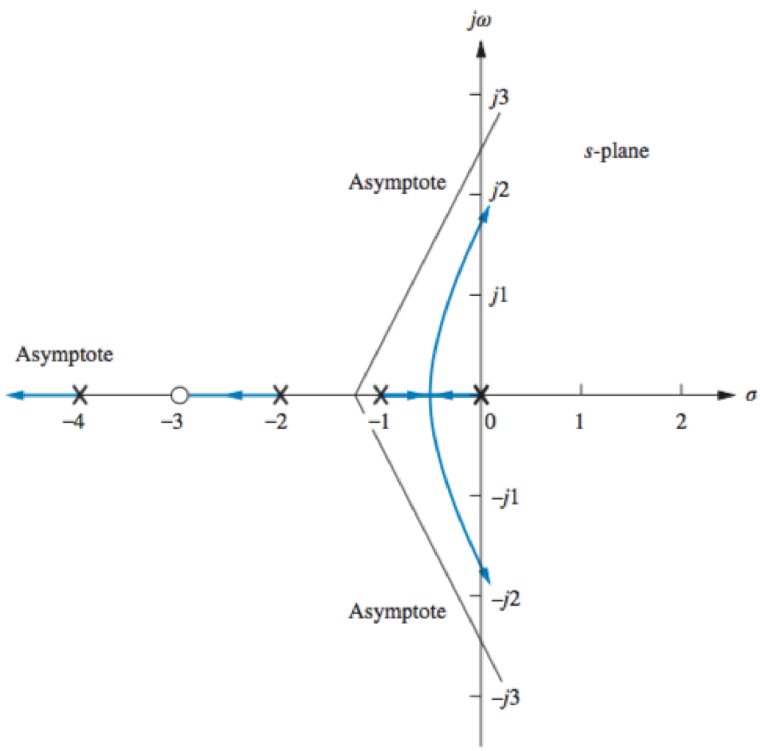
7.1.3 Real Axis Segments

On the real axis, for $K > 0$ the root locus exists to the left of an odd number of real-axis, finite open-loop poles and/or finite open-loop zeros.

7.1.4 Starting and Ending Points

The root locus begins at the finite and infinite poles of $G(s)H(s)$ and ends at the finite and infinite zeros of $G(s)H(s)$.

7.1.5 Behavior at Infinity



7.2 Refining the Sketch

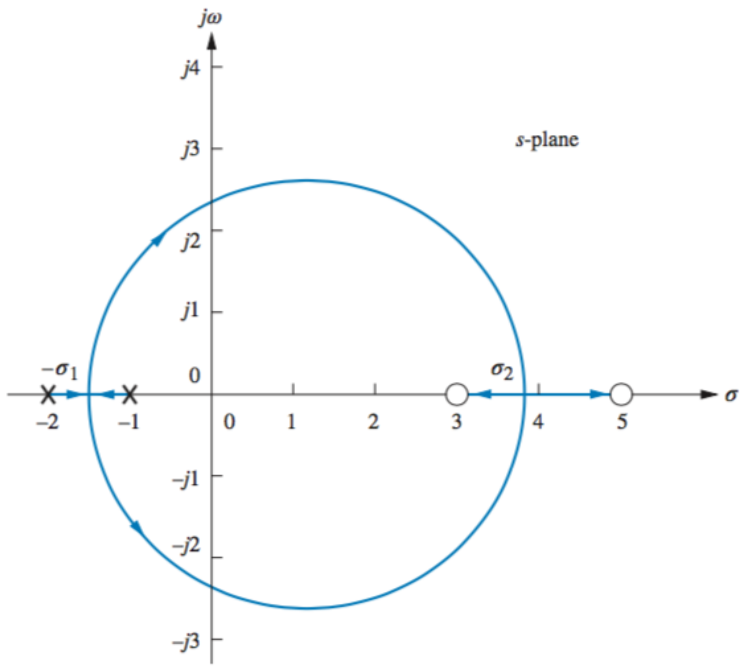
7.2.1 Breakaway/Break-in Point

At the breakaway or break-in point, the branches of the root locus form an angle of $180/n$ with the real axis, where n is the number of closed-loop poles arriving at or departing from the single breakaway or break-in point on the real axis. These are the points when gain is at its minimum and maximum, respectively.

For all points on the root locus,

$K = -\frac{1}{G(s)H(s)}, \frac{1}{G(s)H(s)} = -1$ and by differential calculus,

$K = -\frac{1}{G(\sigma)H(\sigma)}$ where breakpoints occur - setting σ to 0 will produce K.



Or, conversely,

$\sum \frac{1}{\sigma+z_i} = \sum \frac{1}{\sigma+p_i}$

Where z and p are the zero and pole values. By equating the two sides and simplifying to a single equation, factoring can produce σ.

7.2.2 $j\omega$-Axis Crossings

The crossing of the $j\omega$ axis defines when the system becomes unstable. The crossing of the ω axis deines the frequency of oscillation, while the gain at the $j\omega$ axis yields the maximum positive gain for system stability.

The $j\omega$ axis crossings can be found using the Routh-Hurwitz criterion. Forcing a row of zeros yields the gain, then going back a row and solving for the roots yields the frequency at the imaginary axis crossing.

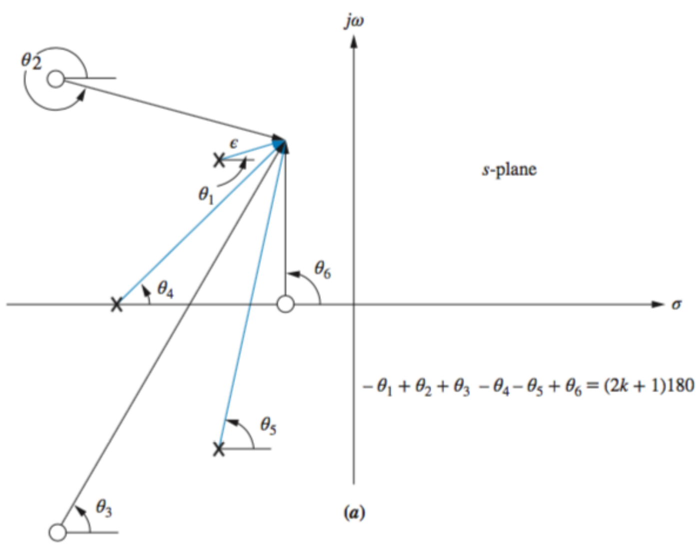
7.2.3 Angles of Departure and Arrival

If we assume a point on the root locus ϵ close to a complex **pole**, the sum of angles drawn from all finite poles and zeros to this point is an odd multiple of 180°. Except for the **pole** that is ϵ close to the point, we assume all angles drawn from all other poles and zeros are drawn directly to the **pole** that is near the point. Thus, the only unknown angle in the sum is the angle drawn from the **pole** that is ϵ close. We can solve for this unknown angle, which is also the angle of departure from this complex **pole**.

$$-\theta_1 + \theta_2 + \theta_3 - \theta_4 - \theta_5 + \theta_6 = (2k+1)180^\circ$$

or

$$\theta_1 = \theta_2 + \theta_3 - \theta_4 - \theta_5 + \theta_6 - (2k+1)180^\circ$$

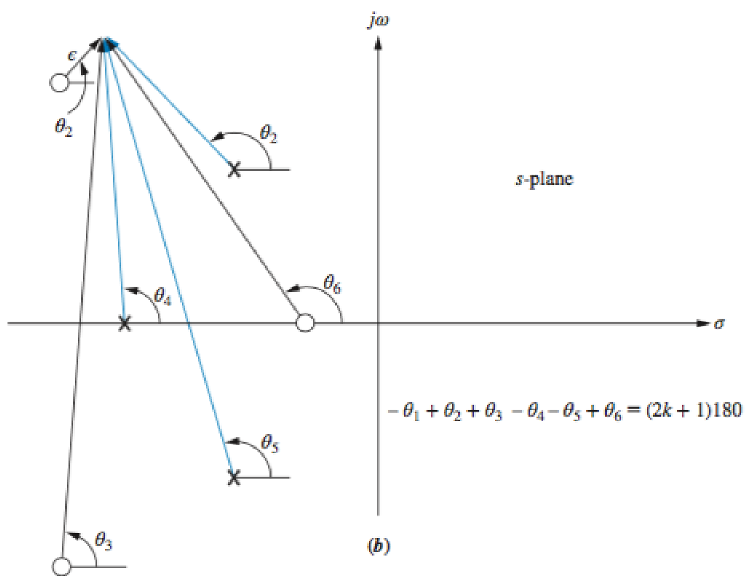


If we assume a point on the root locus ϵ close to a complex **zero**, the sum of angles drawn from all finite poles and zeros to this point is an odd multiple of 180°. Except for the **zero** that is ϵ close to the point, we can assume all angles drawn from all other poles and zeros are drawn directly to the **zero** that is near the point. Thus, the only unknown angle in the sum is the angle drawn from the **zero** that is ϵ close. We can solve for this unknown angle, which is also the angle of arrival to this complex **zero**.

$-\theta_1 + \theta_2 + \theta_3 - \theta_4 - \theta_5 + \theta_6 = (2k+1)180°$

or

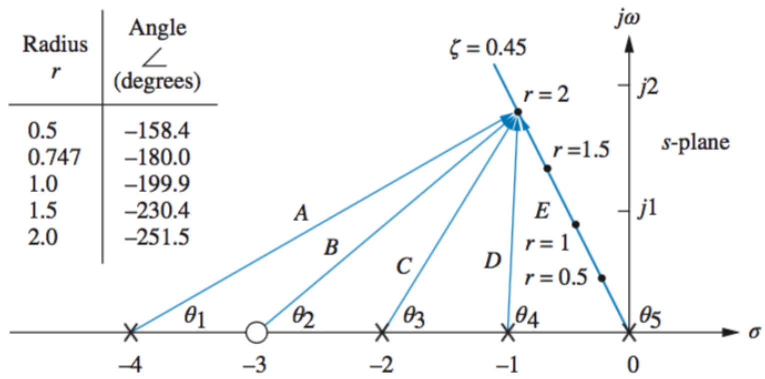
$\theta_2 = \theta_1 - \theta_3 + \theta_4 + \theta_5 - \theta_6 - (2k+1)180°$



7.3 Plotting and Calibrating the Root Locus

When locating points on the root locus and finding their specified gain, for example as it crosses the radial line representing 20% overshoot, such as the below graph where $\zeta = 0.45$

Evaluating the graph at points along the line, and summing the angles from poles and zeros, it can be determined if a point is on the root locus if the angles are a multiple of 180°.

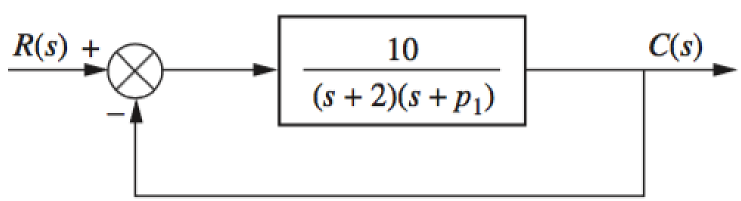


Radius r	Angle $\angle$ (degrees)
0.5	–158.4
0.747	–180.0
1.0	–199.9
1.5	–230.4
2.0	–251.5

7.4 Generalized Root Locus

If finding the root locus of a system concerning a single parameter instead of gain K, an equivalent system can be used where the denominator is represented as $1 + p_1 G(s)H(s)$. From the below system,

$$T(s) = \frac{KG(s)H(s)}{1+KG(s)H(s)} = \frac{10}{s^2+(p_1+2)s+2p_1+10} = \frac{10}{s^2+2s+10+p_1(s+2)}$$



7.5 Positive Feedback Systems

$KG(s)H(s) = 1 = 1\angle k360°$ $\qquad k = 0, 1, 2, 3...$

1. **Number of Branches**
   No change
2. **Symmetry**
   No change

3. **Real Axis Segments**
   On the real axis, the root locus for positive-feedback systems exists to the left of an **even** number of real-axis, finite open-loop poles and/or finite open-loop zeros.

4. **Starting and Ending Points**
   The root locus for positive-feedback systems begins at the finite and infinite poles of $G(s)H(s)$ and ends at the finite and infinite zeros of $G(s)H(s)$.

5. **Behavior at Infinity**
   The root locus approaches straight lines as asymptotes as the locus approaches infinity. Further, the equations of the asymptotes for positive-feedback systems are given by the real-axis intercept, σ_a, and angle, θ_a, as follows:

$$\sigma_a = \frac{\sum FinitePoles - \sum FiniteZeros}{\#FinitePoles - \#FiniteZeros}$$
$$\theta_a = \frac{k2\pi}{\#FinitePoles - \#FiniteZeros}$$

8 Design via Root Locus

9 Frequency Response Techniques

10 Design via Frequency Response