

AER372: Control Systems

Assignment 3

QiLin Xue

Spring 2023

3.1 We want to turn the characteristic equation into the form $a(s) + L(s)b(s)$ where $a(s), b(s)$ are monic polynomials. We have:

(a) We have the characteristic polynomial

$$(s + c)^3 + A(Ts + 1) = 0. \quad (1)$$

(i) Versus parameter A: We can rewrite

$$(s + c)^3 + AT(s + 1/T) = 0. \quad (2)$$

Therefore,

$$\boxed{a(s) = (s + c)^3, \quad b(s) = s + 1/T, \quad L(s) = \frac{s + 1/T}{(s + c)^3}} \quad (3)$$

(ii) Versus parameter T: We can rewrite

$$(s + c)^3 + A + ATs. \quad (4)$$

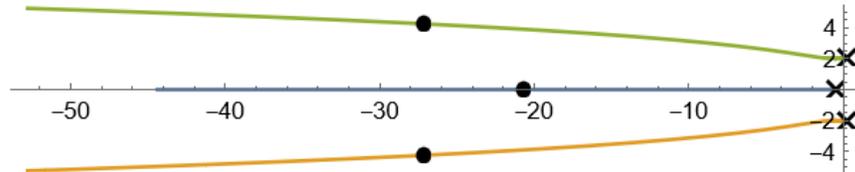
Therefore,

$$\boxed{a(s) = (s + c)^3 + A, \quad b(s) = s, \quad L(s) = \frac{s}{(s + c)^3 + A}} \quad (5)$$

(iii) Versus parameter c : This is impossible since the characteristic equation contains nonlinear terms in c . However, it is still possible to find the roots of the characteristic equation. We can use the cubic equation,

$$c = \omega^k \sqrt[3]{-A(ts + 1) - s} \quad (6)$$

for $k = 0, 1, 2$ and ω is the principal 3rd root of unity. Alternatively (and more realistically), we can use a more superior computer algebra system such as Mathematica to find the roots.



(b) We have

$$1 + \left[k_P + \frac{k_I}{s} + \frac{k_D s}{\tau s + 1} \right] A \frac{c(s)}{d(s)} = 0. \quad (7)$$

(i) Versus k_P : We rewrite

$$\left(1 + \left[\frac{k_I}{s} + \frac{k_D s}{\tau s + 1} \right] A \frac{c(s)}{d(s)} \right) + k_P A \frac{c(s)}{d(s)} = 0. \quad (8)$$

To make everything a polynomial, we multiply by $d(s)s(\tau s + 1)$ to get

$$(d(s)s(\tau s + 1) + A(k_D s^2 + k_I(s\tau + 1))c(s)) + k_P A c(s)s(\tau s + 1) = 0. \quad (9)$$

The highest degree is in the term $s^2\tau d(s)$ since d has a higher degree than $c(s)$. Then dividing by τ , we have

$$(d(s)s(s+1/\tau) + A(k_D s^2/\tau + k_I(s+1/\tau))c(s)) + k_P A c(s)s(s+1/\tau) = 0 \quad (10)$$

$$\implies \left(s^2 d(s) + \frac{sd(s)}{\tau} + \frac{A k_D s^2 c(s)}{\tau} + A k_I s c(s) + \frac{A k_I c(s)}{\tau} \right) + k_P A c(s)(s^2 + s/\tau) = 0. \quad (11)$$

Therefore,

$$a(s) = s^2 d(s) + \frac{sd(s)}{\tau} + \frac{A k_D s^2 c(s)}{\tau} + A k_I s c(s) + \frac{A k_I c(s)}{\tau} \quad (12)$$

$$b(s) = c(s)(s^2 + s/\tau) \quad (13)$$

$$L(s) = \frac{s(s\tau + 1)c(s)}{sd(s) + \tau(A k_I s c(s) + d(s)) + A k_D s^2 c(s) + A k_I c(s)} \quad (14)$$

(ii) Versus k_I : Same thing but make the substitution $k_P \leftarrow \frac{k_I}{s}$ and $k_I \leftarrow k_P s$. This gives

$$\left(s^2 d(s) + \frac{sd(s)}{\tau} + \frac{A k_D s^2 c(s)}{\tau} + A k_P s^2 c(s) + \frac{A k_P s c(s)}{\tau} \right) + k_I A c(s)(s+1/\tau) = 0. \quad (15)$$

Which gives

$$a(s) = s^2 d(s) + \frac{sd(s)}{\tau} + \frac{A k_D s^2 c(s)}{\tau} + A k_P s^2 c(s) + \frac{A k_P s c(s)}{\tau} \quad (16)$$

$$b(s) = c(s)(s+1/\tau) \quad (17)$$

$$L(s) = \frac{(s\tau + 1)c(s)}{s^2 \tau d(s) + sd(s) + A k_D s^2 c(s) + A k_P s^2 \tau c(s) + A k_P s c(s)} \quad (18)$$

(iii) Versus k_D : We can rewrite

$$\left(s^2 d(s) + \frac{sd(s)}{\tau} + \frac{A k_D s^2 c(s)}{\tau} + A k_I s c(s) + \frac{A k_I c(s)}{\tau} \right) + k_P A c(s)(s^2 + s/\tau) = 0 \quad (19)$$

$$\implies \left(s^2 d(s) + \frac{sd(s)}{\tau} + A k_I s c(s) + \frac{A k_I c(s)}{\tau} + k_P A c(s)(s^2 + s/\tau) \right) + k_D \frac{A s^2 c(s)}{\tau} = 0. \quad (20)$$

This gives

$$a(s) = s^2 d(s) + \frac{sd(s)}{\tau} + A k_I s c(s) + \frac{A k_I c(s)}{\tau} + k_P A c(s)(s^2 + s/\tau) \quad (21)$$

$$b(s) = \frac{A s^2 c(s)}{\tau} \quad (22)$$

$$L(s) = \frac{A s^2 c(s)}{s^2 \tau d(s) + sd(s) + A k_I s \tau c(s) + A k_I c(s) + A k_P s^2 \tau c(s) + A k_P s c(s)} \quad (23)$$

(iv) Versus τ : We can rewrite, multiply by τ , and then collect terms

$$\left(s^2 d(s) + \frac{sd(s)}{\tau} + \frac{A k_D s^2 c(s)}{\tau} + A k_I s c(s) + \frac{A k_I c(s)}{\tau} \right) + k_P A c(s)(s^2 + s/\tau) = 0 \quad (24)$$

$$\implies A k_D s^2 c(s) + A k_P s c(s) + A k_I c(s) + sd(s) + A k_I s \tau c(s) + A k_P s^2 \tau c(s) + s^2 \tau d(s) = 0 \quad (25)$$

$$\implies (A k_D s^2 c(s) + A k_P s c(s) + A k_I c(s) + sd(s)) + \tau (A k_I s c(s) + A k_P s^2 c(s) + s^2 d(s)) = 0. \quad (26)$$

Dividing by $A k_D$ gives

$$\left(s^2 c(s) + \frac{k_I c(s)}{k_D} + \frac{k_P s c(s)}{k_D} + \frac{sd(s)}{A k_D} \right) + \frac{\tau}{A k_D} (A k_I s c(s) + A k_P s^2 c(s) + s^2 d(s)) = 0. \quad (27)$$

This gives

$$a(s) = s^2 c(s) + \frac{k_I c(s)}{k_D} + \frac{k_P s c(s)}{k_D} + \frac{sd(s)}{A k_D} \quad (28)$$

$$b(s) = A k_I s c(s) + A k_P s^2 c(s) + s^2 d(s) \quad (29)$$

$$L(s) = A k_D \frac{s^2 d(s) + A k_I s c(s) + A k_P s^2 c(s)}{A k_D s^2 c(s) + A k_I c(s) + A k_P s c(s) + sd(s)}. \quad (30)$$

3.2 (a) Note that $G(s)$ has $m = 0$ roots and has $n = 3$ poles, particularly

$$s = 0, s = -2 \pm i. \quad (31)$$

Asymptotes: The number of asymptotes is $n - m = 3$ with angles

$$\phi_\ell = \frac{180^\circ + 360^\circ(\ell - 1)}{3} = \boxed{60^\circ, 180^\circ, -60^\circ} \quad (32)$$

for $\ell = 1, 2, 3$, and they radiate from the point on the real line,

$$\alpha = \frac{0 + (-2 + i) + (-2 - i)}{3} = \boxed{-\frac{4}{3}}. \quad (33)$$

Breakaway and Break-in Points: They occur when $1 + G(s)$ hits a local minimum/maximum (i.e. a saddle-node bifurcation like behaviour), which leads to when $\frac{dL(s)}{ds} = 0$. We can compute,

$$\frac{d}{ds}(s^3 + 4s^2 + 5s) = 3s^2 + 8s + 5 = 0 \implies \boxed{s = -\frac{5}{3}, s = -1} \quad (34)$$

Angles of departure and arrival: The angle of departure of the pole $-2 + i$ is given by

$$\phi_{\text{dep}} = - \sum_{i \neq \text{dep}} \phi_i - 180^\circ \quad (35)$$

$$= -((180^\circ - \tan^{-1}(1/2)) + 90^\circ) - 180^\circ \quad (36)$$

$$= \boxed{-63.43^\circ}. \quad (37)$$

The angle of departure of the pole $-2 - i$ is given by

$$\phi_{\text{dep}} = - \sum_{i \neq \text{dep}} \phi_i - 180^\circ \quad (38)$$

$$= -((180^\circ + \tan^{-1}(1/2)) - 90^\circ) - 180^\circ \quad (39)$$

$$= \boxed{63.43^\circ}. \quad (40)$$

The angle of departure of the pole 0 is given by

$$\phi_{\text{dep}} = - \sum_{i \neq \text{dep}} \phi_i - 180^\circ \quad (41)$$

$$= -(180^\circ - \tan^{-1}(1/2) + 180^\circ + \tan^{-1}(1/2)) - 180^\circ \quad (42)$$

$$= \boxed{180^\circ}. \quad (43)$$

There are no angles of arrival since the system has no zeros.

Crossings with imaginary axis: For what values of K does $1 + G(s)$ have roots that are purely imaginary? We have,

$$1 + G(s) = 0 \implies s^3 + 4s^2 + 5s + K = 0. \quad (44)$$

Substituting $s = bi$, we have

$$-b^3i - 4b^2 + 5bi + K = 0. \quad (45)$$

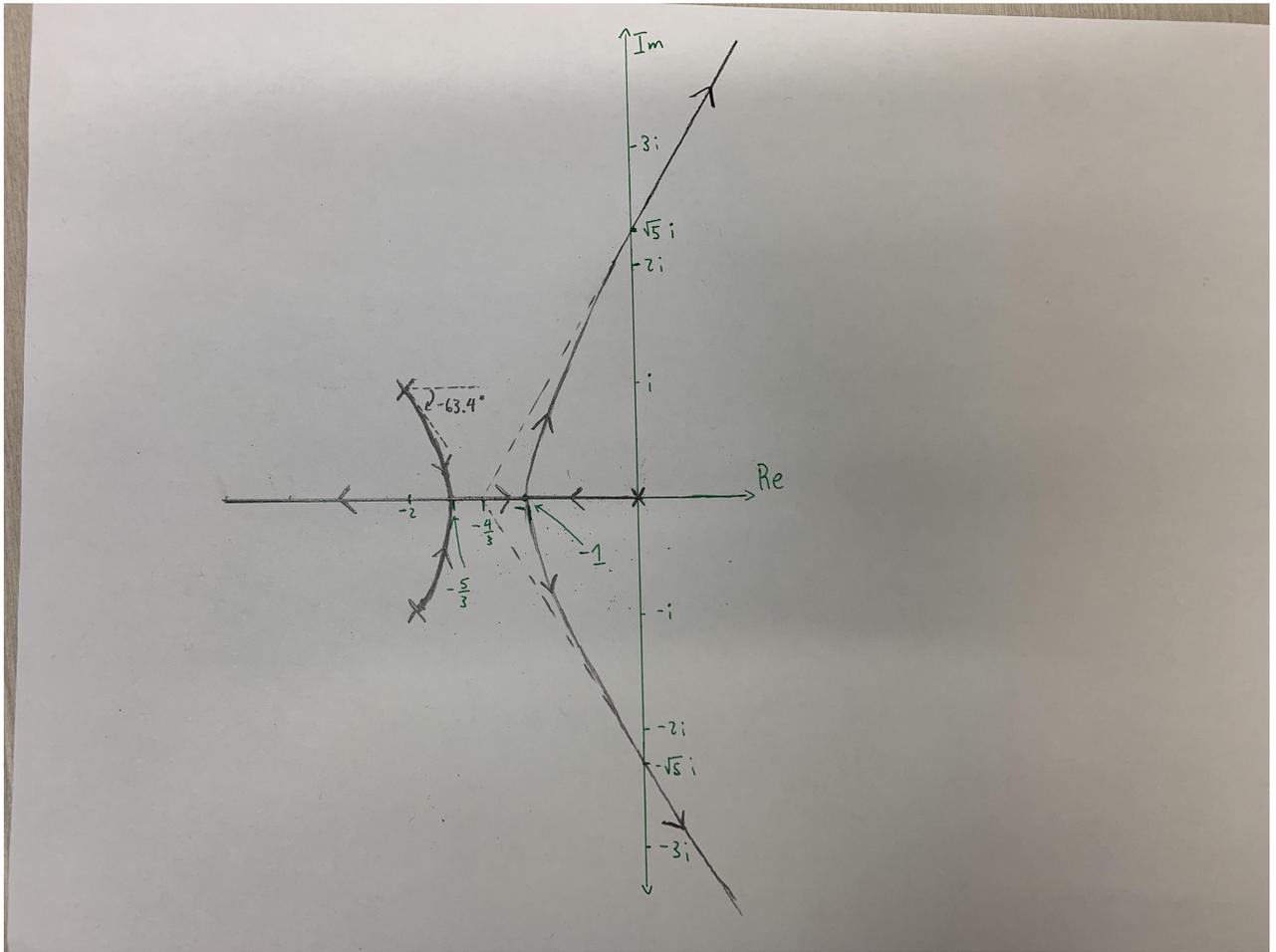
We want the real and the imaginary part to be zero, i.e. this gives the system

$$K = 4b^2 \quad (46)$$

$$5b = b^3 \quad (47)$$

We get $b = 0, \pm\sqrt{5}$, so we get $K = 0, 20$. At $K = 0$ the roots cross the imaginary axis at $s = 0$ and at $K = 20$ the roots cross the imaginary axis at $\boxed{s = \pm\sqrt{5}i}$.

I draw the plot below,



where the arrows indicate the direction the roots are moving with increasing K . I used Matlab to verify these results.

- (b) The system is underdamped when the roots have negative real parts (so it's stable) and a nonzero imaginary part (so it's oscillatory). Similarly, the system is overdamped when the roots have negative real parts with a zero imaginary part. When they transition, we have critical damping, which occurs at $s = -1, -5/3$. We just need to find the corresponding K value.

Recall that $1 + G(s) = 0 \implies s^3 + 4s^2 + 5s + K = 0$. Then Plugging in $s = -1$ gives

$$(-1)^3 + 4(-1)^2 + 5(-1) + K = K - 2 = 0 \implies \boxed{K = 2}, \quad (48)$$

and

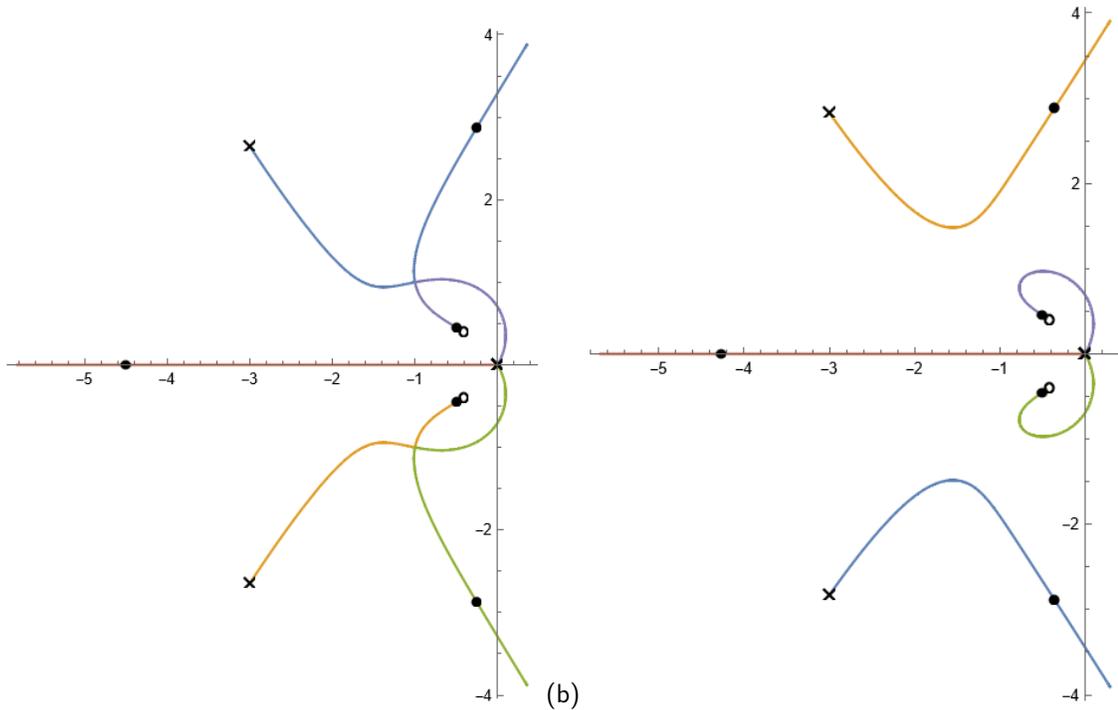
$$(-5/3)^3 + 4(-5/3)^2 + 5(-5/3) + K = K - \frac{50}{27} = 0 \implies \boxed{K \approx 1.85}. \quad (49)$$

Before $K = 1.85$, we have two roots (which originated from $-2 \pm i$) have a nonzero imaginary part and after $K = 2$, we have one root (which originated from $s = 0$ with a nonzero imaginary part). Therefore:

- $K < 1.85$ is underdamped
- $1.85 < K < 2$ is overdamped
- $2 < K < 20$ is underdamped.

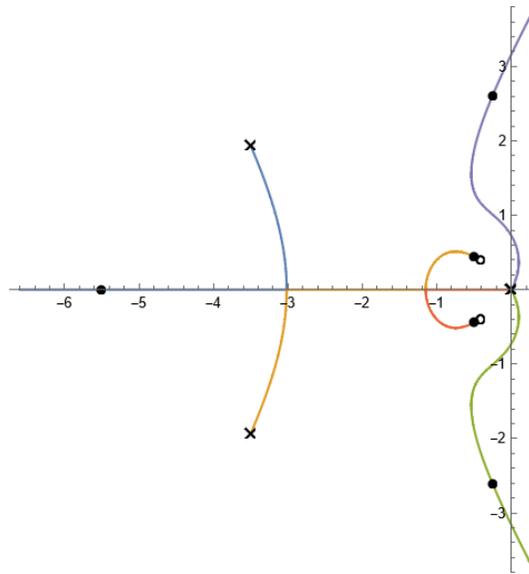
Beyond $K = 20$ it passes the imaginary axis, so it becomes unstable.

3.3 For the three cases, we have the following root-locus plots:



(a)

(b)



(c)

To analyze these plots, we perform some basic computations. First, the zeros occur at

$$s^2 + (5/6)s + (1/3) = 0 \implies s = -\frac{5}{12} \pm \frac{\sqrt{23}i}{12} \quad (50)$$

which corresponds to the two open circles in the plots, i.e. these are where two of the roots will end up as $K \rightarrow \infty$. For all three plots, there are five poles, one at $s = 0$ with multiplicity 3 (so there are three branches coming out) and two poles s, \bar{s} with a negative real part and a non-zero imaginary component. These are represented by an X. We compare the plots pairwise:

- (a) and (b): The breakaway points for part (a) are

$$s = -1 \pm i, -1.56 \pm 0.497i \quad (51)$$

The second pair of breakaway points do not lie on the locus, but $-1 \pm i$ does, so we have an intersection point between two branches. However, for (b), the breakaway points are

$$s = -1.14 \pm 0.976i, -1.42 \pm 0.716i. \quad (52)$$

Here, the real component of the breakaway point has changed slightly, due to modifying the constant term in the denominator. This slight change has made the breakaway point not lie on the locus, so we do not have an intersection point between two branches. Even if we set $\beta = 0.001$ we still have this behaviour.

- (a)/(b) and (c) The major difference here is that there are now two break-away points. We can compute the breakaway points to be

$$s = -3.02, -1.15, -0.807 \pm 0.943i. \quad (53)$$

Here, the conjugate roots are not on the locus, so we do not have an intersection somewhere off the real-axis. However, there are still two intersection points between branches on the real axis, specifically at -3.02 and -1.15 . This means that the other poles and zeros (excluding $s = 0$) need to have a branch that crosses this point and connects it to the real-axis.

Note that all computations here were performed with CAS via Mathematica, along with the plots.

3.4 Solution 1 (Hand computation) Our goal is to find the poles of the closed-loop transfer function. Recall that

$$\zeta = \cos \theta \implies \theta = 60^\circ \quad (54)$$

and

$$\frac{\text{Im } s}{\text{Re } s} = \pm \tan \theta = \pm \sqrt{3}. \quad (55)$$

In other words, we want $s = -\beta + \sqrt{3}\beta i$ to be a root. Our open-loop transfer function is

$$G(s)D_c(s) = \frac{10(s+a)}{s(s+1)(s+8)} \quad (56)$$

and the closed-loop transfer function is

$$\frac{G(s)D_c(s)}{1 + G(s)D_c(s)} = \frac{\frac{10(s+a)}{s(s+1)(s+8)}}{1 + \frac{10(s+a)}{s(s+1)(s+8)}} \quad (57)$$

$$= \frac{10(a+s)}{10a + s(s+1)(s+8) + 10s}. \quad (58)$$

Substituting in $s = \beta + \sqrt{3}\beta i$ to the characteristic equation gives

$$10a + (-\beta + \sqrt{3}\beta i)(-\beta + \sqrt{3}\beta i + 1)(-\beta + \sqrt{3}\beta i + 8) + 10(-\beta + \sqrt{3}\beta i) = 0 \quad (59)$$

$$\implies 10a + 8\beta^3 - 18\beta^2 - 18\sqrt{3}i\beta^2 - 18\beta + 18\sqrt{3}i\beta = 0. \quad (60)$$

Separating into real and imaginary gives

$$0 = 10a + 8\beta^3 - 18\beta^2 - 18\beta \quad (61)$$

$$0 = -18\sqrt{3}\beta^2 + 18\sqrt{3}\beta. \quad (62)$$

The second equation gives $\beta = 0, 1$. Plugging this into the first equation gives

$$a = \frac{-8(1)^3 + 18(1)^2 + 18(1)}{10} = 2.8 \quad (63)$$

$$a = \frac{-8(0)^3 + 18(0)^2 + 18(0)}{10} = 0. \quad (64)$$

Note that $a = 0$ is not a valid choice, as it gives $s = 0$ which does not have $\zeta = 0.5$. However, $a = 2.8$ is valid, and is the only choice that gives a pole at $s = -\beta \pm \sqrt{3}\beta i$.

Solution 2 (Root Locus): The characteristic equation is

$$1 + G(s)D_c(s) = 0 \quad (65)$$

$$\implies 1 + \frac{10}{s(s+1)} \cdot \frac{s+a}{s+8} = 0 \quad (66)$$

$$\implies 10a + s^3 + 9s^2 + 18s = 0 \quad (67)$$

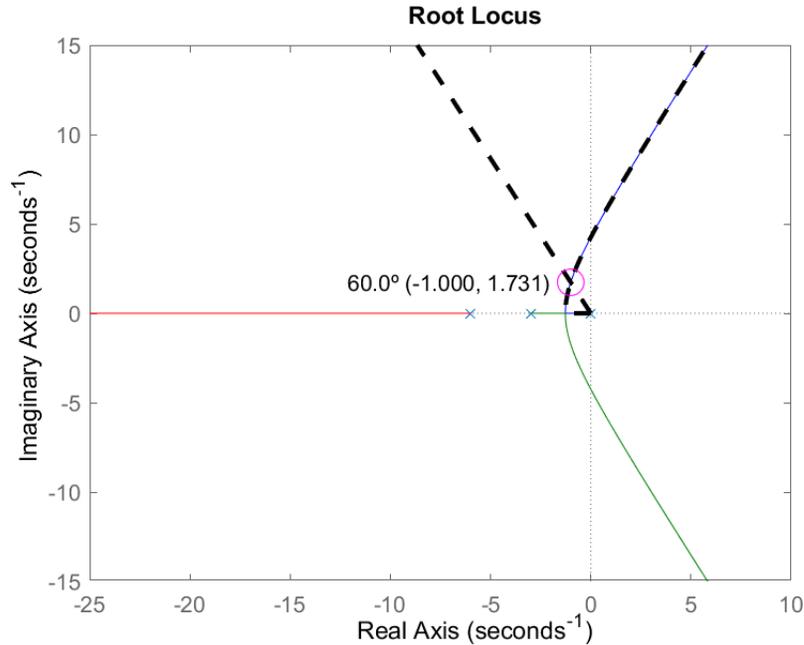
which can be put in the form of

$$(s^3 + 9s^2 + 18s) + a(10), \tag{68}$$

so we can use the root-locus method on

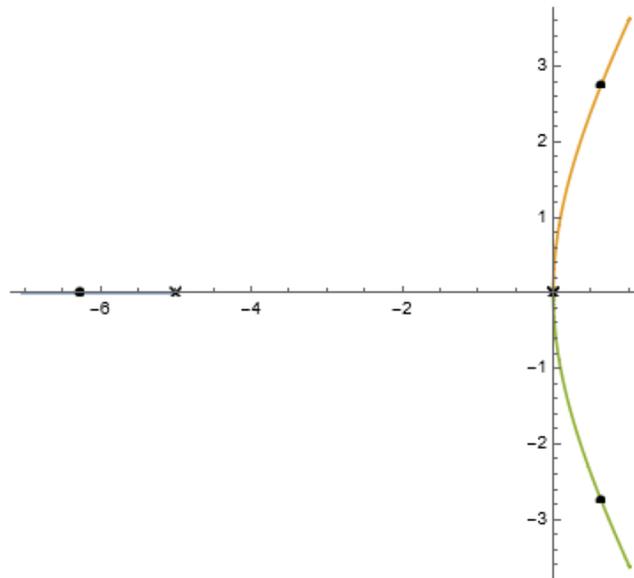
$$\frac{10}{s^3 + 9s^2 + 18s}, \tag{69}$$

which gives us the following plot on Matlab:



where using the intersect tool, gives $a = 2.8$ as well. Note that I borrowed code from [Adam Danz](#) from MathWorks to get the intersection working.

3.5 (a) This is impossible. Refer to the root locus plot below for $KG(s)$.



For any $K > 0$ there are two roots with a positive real part, which is unstable.

(b) The open loop gain is

$$K \cdot \frac{1/(s^2(s+5))}{1 + K_t s \cdot 1/(s^2(s+5))} = \frac{K}{s(K_t + s(s+5))}. \tag{70}$$

This has no zeros, and has poles at $s = 0$ and when $s^2 + 5s + K_t = 0$, which occurs at

$$s = -\frac{5}{2} \pm \frac{\sqrt{25 - 4K_t}}{2}. \quad (71)$$

For $0 < K_t < 25/4$, we have the two roots $s_+, s_- < 0$. The angle of departure from the $s = 0$ root is

$$\phi_{\text{dep}} = -(180^\circ + 180^\circ) - 180^\circ = 180^\circ. \quad (72)$$

Similarly, for $K_t > 25/4$, we have two roots s, \bar{s} with a negative real part and a non-zero imaginary component. The angle of departure from the $s = 0$ root is still 180° (similar computation as Q2). Finally, we should expect the same behavior at $K_t = 25/4$ because we should expect the locus to change continuously as a function of K_t .

Because the angle of departure is 180° , for small values of K we can guarantee that all roots have a negative real component. However, for large enough K we could still have roots that go off to have a positive real component, so yes, we should expect a relation between K and K_t to guarantee a stable response.

(c) We have already computed some of these ranges:

- $0 < K_t < 25/4$: all roots lie on the real axis with negative real part
- $K_t = 25/4$: there are only two distinct roots (where one has a multiplicity of 2), both on the real axis with negative real part
- $25/3 > K_t > 25/4$: there are two roots with a negative real part and a non-zero imaginary component. We will see where the $25/3$ comes from later.

Note that $s = 0$ is a root for all of them. Also, we can compute where the break-away points are:

$$\frac{d}{ds} (sK_t + s^3 + 5s^2) = K_t + 3s^2 + 10s = 0 \implies s = -\frac{5}{3} \pm \frac{\sqrt{25 - 3K_t}}{3}. \quad (73)$$

For $K_t < \frac{25}{3}$, there are two distinct breakaway points. But at $K_t = 25/3$ there is only one, and as $K_t > 25/3$ there are none, signifying a change in behaviour. This gives two more regions:

- $K_t = \frac{25}{3}$: there is one breakaway point
- $K_t > \frac{25}{3}$: there are no breakaway points.

(d) No overshoot occurs when the system is critically or over-damped, i.e. the poles are on the negative real axis (so not oscillations). The rise time is determined by how far away the poles are from the imaginary axis. Because of this, we will look for when this root has a multiplicity of 2 (i.e. if it didn't, then one root would always be to the right (and to the left) of this double root, so a local min/max can only exist at a double root).

Note that these are the poles of the closed-loop system, which are the roots of

$$s^3 + 5s^2 + K_t s + K = 0. \quad (74)$$

The double root can only occur when the derivative is zero, i.e.

$$3s^2 + 10s + K_t \implies s = -\frac{5}{3} \pm \frac{\sqrt{25 - 3K_t}}{3} \quad (75)$$

but since $K_t \leq 5$ is the upper bound, we will choose $K_t = 5$ which gives $s = -5/3 \pm \sqrt{103}$, i.e. $s = -0.613, -2.73$. We can plug these values back into the original equation to get the corresponding K values:

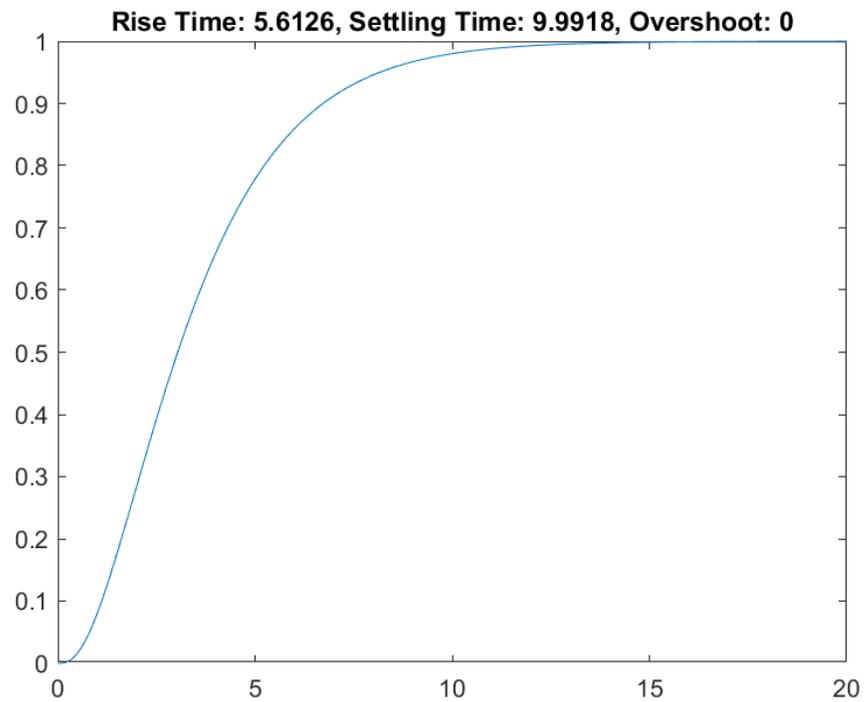
$$(-0.613)^3 + 5(-0.613)^2 + 5(-0.613) + K = 0 \implies K = 1.416501397 \quad (76)$$

$$(-2.73)^3 + 5(-2.73)^2 + 5(-2.73) + K = 0 \implies K = -3.268083 \quad (77)$$

so only the first value is valid since $K > 0$. Thus, we pick

$$\boxed{(K_t, K) = (5, 1.4)}. \quad (78)$$

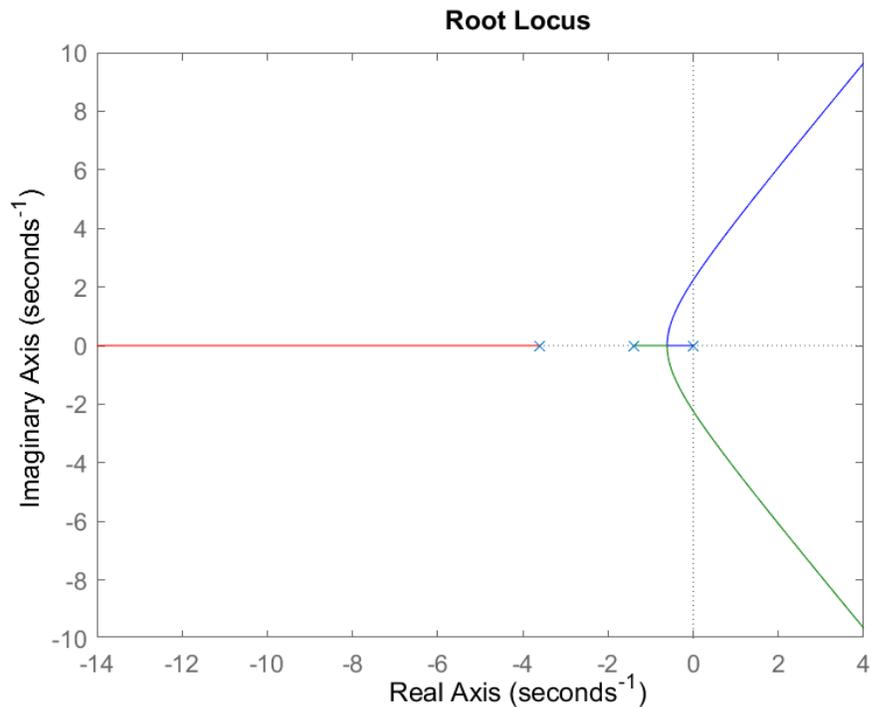
We can plot this in MATLAB,



(e) Using MATLAB, I was able to make the root-locus plot for the open-loop transfer function

$$\frac{K}{s^3 + 5s^2 + 5s} \quad (79)$$

where $K_t = 5$ was used. This value was used because of the earlier discussion of how increasing K_t will cause the rise time to increase. The plot is shown below,



Using `rlocfind()`, we can determine the location of the breakaway point (i.e. when the system changes from overdamped to underdamped) and get around

$$\boxed{K = 1.42}, \quad (80)$$

which agrees with our earlier analysis. To verify this, we can try a few different values for K_t , and we will see that this is maximized at $K_t = 5$.

3.6 Let's translate the design specifications into more useful quantities.

- DS1: System needs to be type 1, i.e. have a pole at $s = 0$. This means we should have an integrator.
- DS2: We want $1/K_v < 0.28$ or equivalently

$$\lim_{s \rightarrow 0} sGD_{cl}(s) > 3.57. \quad (81)$$

- DS3: We use the equation for percent overshoot (note that here it is approximate, but we will use it as a starting point)

$$e^{-\pi\zeta/\sqrt{1-\zeta^2}} < 0.05 \implies \zeta > 0.752 \quad (82)$$

This corresponds to an angle $\theta = \cos^{-1} 0.752 = 41.218^\circ$, and

$$\left| \frac{\text{Im } s}{\text{Re } s} \right| = \tan \theta < 0.876 \quad (83)$$

- DS4: The real part should be smaller than $-4/1.5 \approx -2.7$.

Consider a controller in the form of

$$k_P(1 + \alpha/s) \quad (84)$$

where $\alpha = k_I/k_P$. We guess the poles (nice numbers that satisfy DS3,DS4)

$$s = -3 \pm 2i \quad (85)$$

in order to satisfy DS3 and DS4. Using root-locus and a bit of trial and error, we were able to obtain the poles $-2.71 \pm 1.77i$ with $\alpha = 0.7$ and $k_P = 8.5938$. To ensure it satisfies DS2, we can compute

$$\lim_{s \rightarrow 0} s \cdot 8.5938(1 + 0.7/s) \cdot \frac{1}{(s+1)(s+5)} = \lim_{s \rightarrow 0} \frac{8.5938 \cdot 0.7 \cdot (2s+1)}{(s+1)(s+5)} = 1.2, \quad (86)$$

which is almost three times smaller than 3.57 Repeated experimentation leads us to believe that the location of viable roots cannot change significantly by changing k_P, α , so we look into a three parameter controller, such as a PID controller. We have the ansatz

$$k_P(1 + \alpha/s + \beta s) \quad (87)$$

where $\beta = k_D/k_P$. The condition for DS2 becomes

$$\lim_{s \rightarrow 0} s k_P(1 + \alpha/s + \beta s) \cdot \frac{1}{(s+1)(s+5)} = \lim_{s \rightarrow 0} \frac{k_P(\alpha + s(\beta s + 1))}{(s+1)(s+5)} = \frac{\alpha k_P}{5} \quad (88)$$

which is the same condition as before. If we set $\alpha = 0.5$ to a more "safe" value, then we need $k_P > 35.7$.

Guessing $\beta = 0.17$ with root-locus allows us to pick the roots $-6.58 \pm 0.2356i$ with $k_P = 45.05$, which meets DS2. Therefore, our proposed values are

$$\boxed{k_P = 45.05, k_I = 22.53, k_D = 7.66}. \quad (89)$$

As a quick sanity check, we need to make sure the system is stable. The plant is given by

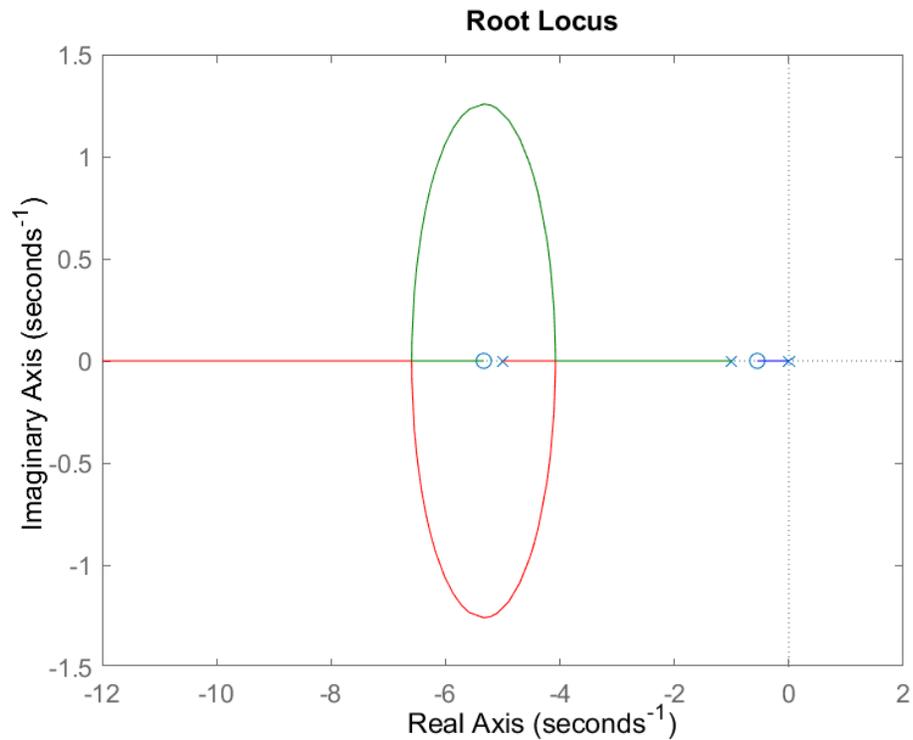
$$G(s) = \frac{1}{s^2 + 6s + 5} \quad (90)$$

so $\omega_n = \sqrt{5}, \zeta = 3/\sqrt{5} = 1.34$, and $K = 1/5$. Routh's Stability Criterion gives

$$\frac{(2\zeta + k_D K \omega_n)(1 + k_P K) \omega_n}{K} = \frac{(2 * 1.34 + 7.55 * 1/5 * \sqrt{5})(1 + 45.05 * 1/5) \sqrt{5}}{1/5} = 677.8 > k_I, \quad (91)$$

so the system is stable.

The corresponding root-locus plot (for documentation purposes) is



Note that at the root around $-6.6 \pm 0.2i$, there is a third root on the real-axis near $s = 0$. We are able to ignore this pole, because it cancels out one of the zeros from the controller. This is allowed behavior since we are not cancelling out any zeros from the plant, only the controller.

Listing 1: Code to Compute Design Specifications

```

1 %% Define the transfer function
2 %  $kp(1+a/s) = kp s + a*kp / s$ 
3 kp=45.049;
4 alpha=0.5;
5 beta=0.17;
6
7 C = tf([kp*beta kp kp*alpha],[1 0]);
8 G = tf(1, [1 6 5]);
9 sys = G*C/(1+G*C);
10
11 %% Get Step Input Result
12 step_sys = feedback(sys, 1);
13 step_info = stepinfo(step_sys);
14
15 % Simulate the response to a step input
16 t = 0:0.01:20;
17 y = step(sys, t);
18
19 % Calculate the steady-state error
20 steady_state_value_step = y(end);
21 input_mag_step = 1;
22
23 %% Get Ramp Input Result
24 % Simulate the response to the ramp input
25 t = 0:0.1:10;
26 ramp_input = t;
27 [y, t] = lsim(sys, ramp_input, t);
28

```

```

29 % Calculate the steady-state error
30 steady_state_value = y(end);
31 input_mag = max(ramp_input);
32
33 disp(DS1: Step SS Error=+abs(steady_state_value_step - input_mag_step)/input_mag_step)
34 disp(DS2: Ramp SS Error=+abs(steady_state_value - input_mag)/input_mag)
35 disp(DS3: Overshoot=+step_info.Overshoot)
36 disp(DS4: Settling time=+step_info.SettlingTime)

```

which gives us

```

DS1: Step SS Error=1.7317e-06
DS2: Ramp SS Error=0.22137
DS3: Overshoot=0
DS4: Settling time=0.40168

```

so everything is satisfied. Note that this is not the only solution. I can run a grid search algorithm to find the best solution within a certain region:

Listing 2: Code to Perform Grid Search on Design Specifications

```

1 % Define the criteria
2 DS1_target = 0.01;
3 DS2_target = 0.28;
4 DS3_target = 5;
5 DS4_target = 1.5;
6
7 % Define the parameter ranges
8 kp_range = 10:1:50;
9 alpha_range = 0.3:0.05:0.7;
10 beta_range = 0.1:0.01:0.2;
11
12 % Initialize the minimum kp and the minimum DS4
13 min_kp = Inf;
14 min_alpha = Inf;
15 min_Beta = Inf;
16 % Loop over all possible combinations of kp, alpha, and beta
17 for kp = kp_range
18     for alpha = alpha_range
19         for beta = beta_range
20             % Calculate the DS1, DS2, DS3, and DS4
21             [DS1, DS2, DS3, DS4] = calculate_DS(kp, alpha, beta);
22
23             % Check if the criteria are satisfied and update the minimum
24             if DS1 < DS1_target && DS2 < DS2_target && DS3 < DS3_target && DS4 < DS4_target
25                 if kp < min_kp
26                     min_kp = kp;
27                     min_alpha = alpha;
28                     min_beta = beta;
29                 end
30             end
31         end
32     end
33 end
34
35 % Print the minimum kp and minimum DS4
36 if min_kp == Inf
37     disp('No combination of parameters satisfies the criteria.')
38 else
39     disp('Minimum kp!: + min_kp)
40     disp('Minimum alpha: + min_alpha)

```

```

41     disp(Minimum beta:  + min_beta)
42 end
43
44 [DS1, DS2, DS3, DS4] = calculate_DS(min_kp, min_alpha, min_beta)

```

which gives us

```

Minimum kp: 19
Minimum alpha: 0.95
Minimum beta: 0.1
DS1 = 1.1769e-11
DS2 = 0.2770
DS3 = 4.7056
DS4 = 0.9436

```

We can compute

$$\frac{k_p \alpha}{5} = 3.61 > 3.57 \quad (92)$$

as desired. Therefore, the optimal solution in this range is

$$(k_p, k_I, k_D) = (19, 18.05, 1.9) . \quad (93)$$

where we are optimizing for smaller gains, which although give the same behaviour, is desired in real applications because there is less noise and it is more robust when actually implementing it (i.e. smaller voltages). Similarly, the system is stable since

$$\frac{(2\zeta + k_D K \omega_n)(1 + k_P K) \omega_n}{K} = \frac{(2 * 1.34 + 1.9 * 1/5 * \sqrt{5})(1 + 19 * 1/5) \sqrt{5}}{1/5} = 189 > k_I, \quad (94)$$

so this is stable as well.

The function `calculate_DS(kp, alpha, beta)` is defined via the following code:

Listing 3: Function to Compute Design Specifications

```

1 function [DS1, DS2, DS3, DS4] = calculate_DS(kp, alpha, beta)
2     % Define the transfer function
3     C = tf([kp*beta kp kp*alpha],[1 0]);
4     G = tf(1, [1 6 5]);
5     sys = G*C/(1+G*C);
6
7     % Get Step Input Result
8     step_sys = feedback(sys, 1);
9     step_info = stepinfo(step_sys);
10    t = 0:0.01:20;
11    y = step(sys, t);
12
13    % Calculate the steady-state error for step input
14    steady_state_value_step = y(end);
15    input_mag_step = 1;
16
17    % Get Ramp Input Result
18    t = 0:0.1:10;
19    ramp_input = t;
20    [y, t] = lsim(sys, ramp_input, t);
21
22    % Calculate the steady-state error for ramp input
23    steady_state_value = y(end);
24    input_mag = max(ramp_input);
25
26    % Return DS1, DS2, DS3, and DS4 as outputs
27    DS1 = abs(steady_state_value_step - input_mag_step)/1;

```

```
28     DS2 = abs(steady_state_value - input_mag)/1;  
29     DS3 = step_info.Overshoot;  
30     DS4 = step_info.SettlingTime;  
31 end
```

For transparency, GPT4 was used to assist in generating the grid search optimization.