ELEC 3004/7312: Signals Systems & Controls
Assignment 3, worked solutions – updated

1. Digitise the following systems, assuming 10 Hz sample rate
   i. As difference equations, using Euler's method

   ii. As z-domain transfer functions between *x* and *u*

   a. $\dot{x} = -3x + u$
   b. $\ddot{x} = -2\dot{x} - 4x + \dot{u} + u$
   c. $x = 10\left(u + 0.1\int u\,dt + 2\frac{du}{dt}\right)$
   d. $\ddot{x} = \dot{y} - 2\dot{x},\ \ddot{y} = -y - \dot{x} + u$
   e. $\dot{x} = -\sin(t) - x - \dot{u}$

i. Difference equations are not expected to be causal (but causal forms will be accepted)

a.
$$\frac{x(k+1) - x(k)}{T} = -3x(k) + u(k)$$
$$x(k+1) - x(k) = -3Tx(k) + Tu(k)$$
$$x(k+1) - (1-3T)x(k) = Tu(k)$$
$$\boxed{x(k+1) - 0.7x(k) = 0.1u(k)}$$

b. Note that Euler's method can be applied to multiple orders of differential. By using a place-holder variable for a second order or higher derivative, it is possible to approximate multiple derivatives. Eg. $\ddot{x} = \dot{q} \cong \frac{q(k+1)-q(k)}{T}$ and then recognizing that $\dot{x}(k)$ can be likewise approximated. This process can be greatly simplified using the substitution $s^n = \left(\frac{z-1}{T}\right)^n$
$$x(k+2) - 2x(k+1) + x(k) =$$
$$-2Tx(k+1) + 2Tx(k) - 4T^2x(k) + Tu(k+1) - Tu(k) + T^2u(k)$$
$$x(k+2) + (2T-2)x(k+1) + (1 - 2T + 4T^2)x(k)$$
$$= Tu(k+1) + (T^2 - T)u(k)$$
$$\boxed{x(k+2) - 1.8x(k+1) + .84x(k) = 0.1u(k+1) - 0.09u(k)}$$

c. Differentiating both sides puts the relationship into a derivative expression
$$\dot{x} = 10(\dot{u} + 0.1u + 2\ddot{u})$$
$$x(k+1) - x(k) = 10\left(\frac{u(k+1)-u(k)}{T} + 0.1u(k) + 2\frac{u(k+2)-2u(k+1)+u(k)}{T^2}\right)$$
$$T^2x(k+1) - T^2x(k) =$$
$$10\big(2u(k+2) + (T-4)u(k+1) + (2 - T + 0.1T^2)u(k)\big)$$
$$\boxed{0.01x(k+1) - 0.01x(k) = 20u(k+2) - 39u(k+1) + 19.01u(k)}$$

Bonus other way: If you solved the typo version where 1 was written instead of *u*, the solution is

$x(k+1) - x(k) = 10\left(1 + 0.1u(k) + 2\frac{u(k+2)-2u(k+1)+u(k)}{T^2}\right)$

$T^2x(k+1) - T^2x(k) = 10\left(T^2 + 2u(k+2) - 4u(k+1) + (2+0.1T^2)u(k)\right)$

$$\boxed{0.01x(k+1) - 0.01x(k) = 0.1 + 20u(k+2) - 40u(k+1) + 20.01u(k)}$$

d. Note that the second equation can be expressed as a transfer function between y and $(-\dot{x}+u)$:

$s^2y = -y - sx + u$

$(s^2+1)y = -sx + u$

$y = \frac{-sx+u}{s^2+1}$

Thus,

$s^2x = s\frac{-sx+u}{s^2+1} - 2sx$

$s(s^2+1)x = -sx + u - 2(s^2+1)x$

$s^3x + 2s^2x + 2sx + 2x = u$

Returning to ODEs:

$\dddot{x} + 2\ddot{x} + 2\dot{x} + 2x = u$

Therefore:

$$\frac{x(k+3) - 3x(k+2) + 3x(k+1) - x(k)}{T^3} + 2\frac{x(k+2) - 2x(k+1) + x(k)}{T^2} + 2\frac{x(k+1) - x(k)}{T} + 2x(k) = u(k)$$

$x(k+3) - 3x(k+2) + 3x(k+1) - x(k) + 2T(x(k+2)$
$-2x(k+1) + x(k)) + 2T^2(x(k+1) - x(k)) + 2T^3x(k) = T^3u(k)$

$x(k+3) + (2T-3)x(k+2) + (2T^2 - 4T + 3)x(k+1)$
$+(2T^3 - 2T^2 + 2T - 1)x(k) = T^3u(k)$

$$\boxed{x(k+3) - 2.8x(k+2) + 2.62x(k+1) - 0.818x(k) = 0.001u(k)}$$

e. Obviously, finding an Euler approximation for sin(*t*) is not directly sensible. To generate a difference equation from this function requires a quick dip into the s domain:

$sx = -\frac{1}{s^2+1} - x - su$

Then,

$s(s^2+1)x = -1 - (s^2+1)x - s(s^2+1)u$

$s^3x + s^2x + sx + x = -1 - s^3u + su$

Returning to ODEs:

$\dddot{x} + \ddot{x} + \dot{x} + x = -1 - \ddot{u} + \dot{u}$

$$\frac{x(k+3)-3x(k+2)+3x(k+1)-x(k)}{T^3}+\frac{x(k+2)-2x(k+1)+x(k)}{T^2}+\frac{x(k+1)-x(k)}{T}+x(k)=$$
$$-\frac{u(k+3)-3u(k+2)+3u(k+1)-u(k)}{T^3}+\frac{u(k+1)-u(k)}{T}-1$$

$$x(k+3)-3x(k+2)+3x(k+1)-x(k)+T(x(k+2)-2x(k+1)+x(k))$$
$$+T^2\big(x(k+1)-x(k)\big)+T^3x(k)=$$
$$-u(k+3)+3u(k+2)-3u(k+1)+u(k)+T^2(u(k+1)-u(k))-\mathrm{T}^3$$

$$x(k+3)+(T-3)x(k+2)+(T^2-2T+3)x(k+1)+$$
$$(T^3-T^2+T-1)x(k)=$$
$$-u(k+3)+3u(k+2)+(T^2-3)u(k+1)+(1-T^2)u(k)-T^3$$

$$\boxed{\begin{aligned}&x(k+3)-2.9x(k+2)+2.81x(k+1)-0.9090x(k)=\\&\quad -u(k+3)+3u(k+2)\ -2.99u(k+1)+0.99u(k)-0.001\end{aligned}}$$

Bonus other way: approximate sin(*x*) as a Taylor series expansion and repeat. No worked solution here, because this would take a long time and isn't especially interesting – but kudos if you tried!

ii.

a. To convert between difference equations and the z-domain representation, replace instances of $q(k+n)$ with $z^n q(z)$. Using the result from part i:

$$zx(z)-0.7x(z)=0.1u(z)$$

Now put into a ratio of $x(z)$ and $u(z)$,

$$\boxed{x(z)=\frac{0.1}{z-0.7}u(z)}$$

b.

$$z^2x(z)+0.4x(z)=zu(z)$$
$$x(k+2)-1.8x(k+1)+.84x(k)=0.1u(k+1)-0.09u(k)$$

$$\boxed{x(z)=\frac{0.1z-0.09}{z^2-1.8z+0.84}u(z)}$$

c.

$$0.01zx(z)-0.01x(z)=20z^2u(z)-39zu(z)+19.01u(k)$$

$$\boxed{x(z)=\frac{20z^2-39z+19.01}{0.01z-0.01}u(k)}$$

d.

$$z^3x(z)-2.8z^2x(z)+2.62zx(z)-0.818x(z)=0.001u(z)$$

$$\boxed{x(z)=\frac{0.001}{z^3-2.8z^2+2.62z-0.818}u(z)}$$

e.

Easy enough to convert to z-domain:

$$z^3x(z) - 2.9z^2x(z) + 2.81zx(z) - 0.9090x(z) =$$
$$-z^3u(z) + 3z^2u(z) - 2.99zu(z) + 0.99u(z) - 0.001$$

How to handle that constant? Simply express *x*(*z*) as a function of *u*(*z*), and the constant:

$$\boxed{x(z) = \frac{-z^3+3z^2 -2.99z+0.99}{z^3-2.9z^2+2.81z-0.9090}u(z) - \frac{0.001}{z^3-2.9z^2+2.81z-0.9090}}$$

2. Find the steady-state value of x at $t = \infty$ (if any) for each system in part 1

To find the steady state value of a function, use the Final Value Theorem for digital systems on the transfer functions of part 1.ii The impulse response is acceptable, but the step response is preferred.

The FVT states that

$$\lim_{t\to\infty} x(t) = \lim_{z\to 1} F(s)\frac{z-1}{z}$$

For a step response, $u(z) = \frac{z}{z-1}$, and so for system $H(z)$, given in part 1, then $F(z) = H(z)\frac{z}{z-1}$ and

$$\lim_{t\to\infty} x(t) = \lim_{z\to 1} H(z)$$

i.

$$\lim_{t\to\infty} x(t) = \frac{0.1}{1-0.7} = \boxed{0.33\bar{3}}$$

b.

$$\lim_{t\to\infty} x(t) = \frac{0.1-.09}{1^2-1.8+0.84} = \boxed{0.25}$$

c.

$$\lim_{t\to\infty} x(t) = \frac{20*1^2-39*1+19.01}{0.01*1-0.01} = \boxed{\infty}$$

d.

$$\lim_{t\to\infty} x(t) = \frac{0.001}{1^3-2.81^2+2.621-0.818} = \boxed{0.5}$$

e.

The root factorization of $\frac{1}{z^3-2.9z^2+2.81z-0.9090}$ is $\frac{1}{(z-0.9)(z-1+0.1i)(z-1-0.1i)}$.

Two of the roots lie outside the unit circle. Consequently, the system is unstable; the final value of the system is unbounded, and oscillates between ∞ and $-\infty$.

3. Sketch the root locus of the system $\ddot{x} = -10x + u$ in feedback with a 5 Hz controller implementing a

i. lead compensator
ii. lag compensator
iii. PID compensator

Which is more robust to arbitrarily large system gains?

The z-transform of this system is easy to compute using the transform tables:

$$s^2 x = -10x + u$$

$$x(s) = \frac{1}{s^2 + 10} u(s)$$

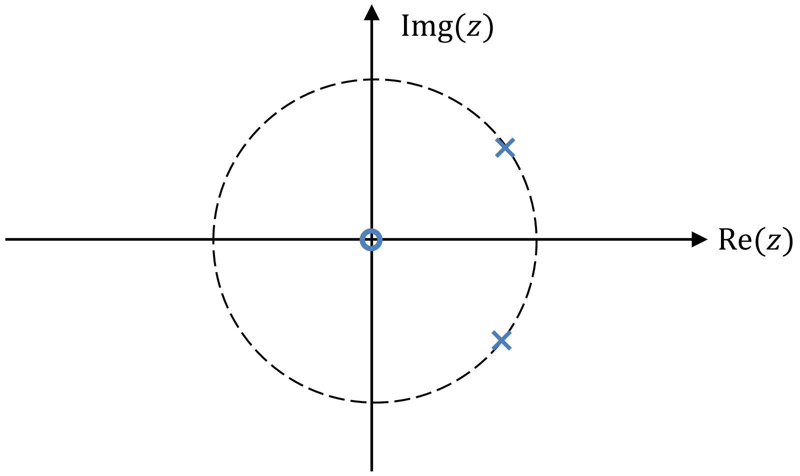
With sample rate 5 Hz, the period is 0.2. This yields the z-domain system:

$$x(z) = \frac{z \sin(0.2\sqrt{10})}{z^2 - 2\cos\left(0.2\sqrt{10}\right) z + 1} u(z)$$

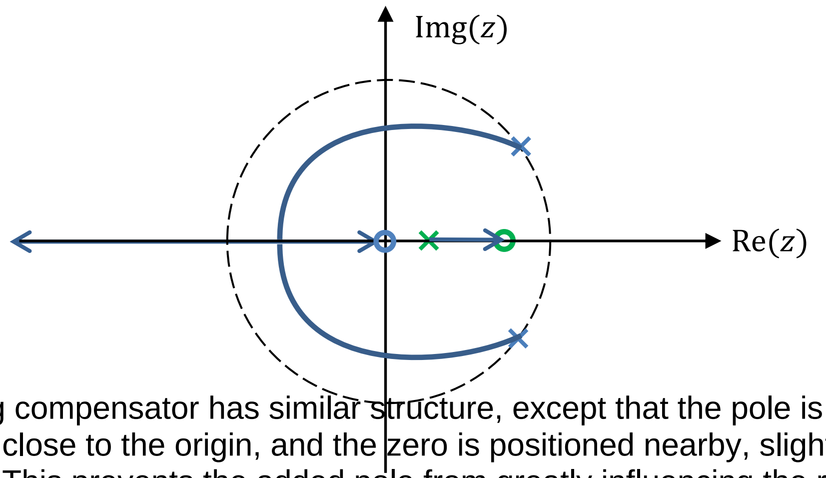
With quick application of the quadratic formula, the poles of the system will be at:

$$z = 0.8066 \pm 0.5911i$$

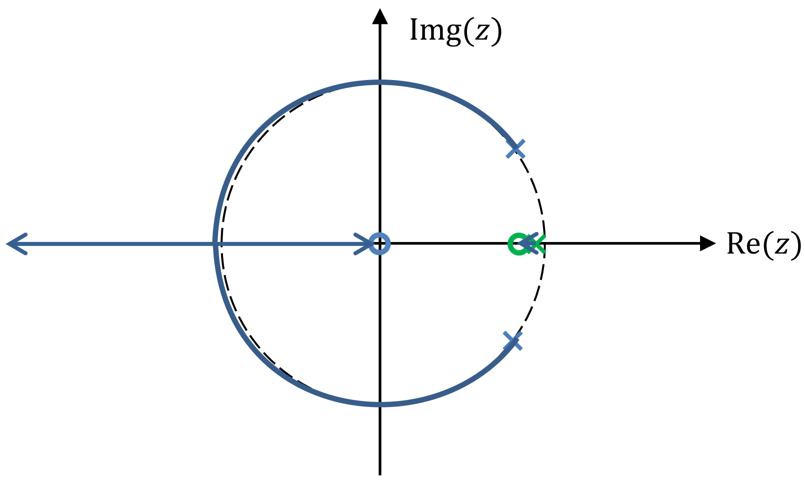
Plotted on the z-plane, this is:



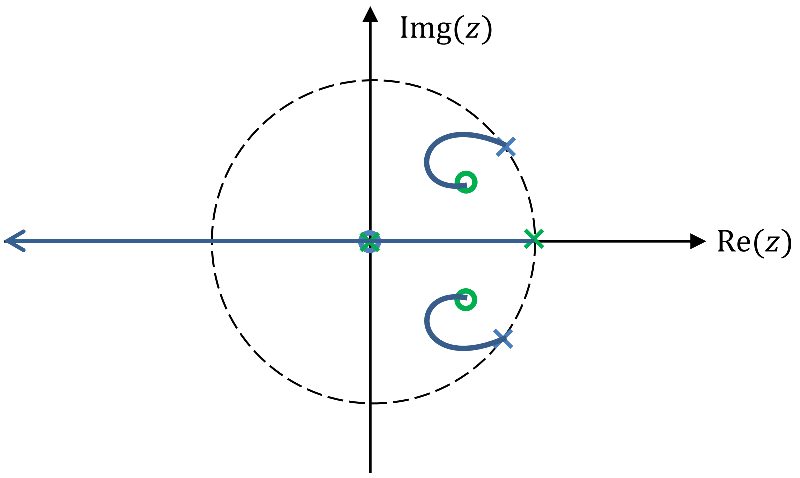
i. A lead compensator has the form $\frac{1-\alpha z^{-1}}{1-\beta z^{-1}}$ where the zero is positioned close to the dynamics of the plant, and the pole is a 'fast' pole closer to the origin. The pole-zero excess is one, therefore one pole will go to $z = -\infty$ on the real axis. Without compensation, the poles would circle the zero before meeting at $z = -1$ and one going out to the left and one being absorbed by the pole at the origin. The compensator pole is absorbed by the compensator zero



ii. A lag compensator has similar structure, except that the pole is located very close to the origin, and the zero is positioned nearby, slightly to the left. This prevents the added pole from greatly influencing the rest of the system dynamics. The resulting root locus will be very similar to the simple proportional closed-loop feedback case, in which the root locus is nearly circular:



iii. Discrete PID compensators add two zeros, a pole at $z = 1$ and a pole at the origin. The added pole at the origin cancels with the zero there; the pole zero excess is the same as the other controllers. The two zeros will each absorb a pole; the path taken from the oscillatory poles to the zero can either be direct, or through a break-in and subsequent break-out from the real axis.



Comparison question: All of these compensator forms will result in the system going unstable for arbitrarily large gains. However, in general a properly designed PID controller will allow more gain to be used.

4. For what range of proportional feedback gains are the following systems stable?

i. $H(z) = \frac{0.1052}{z-1.105}$

ii. $H(s) = \frac{2}{s^2+4s+4}$, 20 Hz (approximate using Tustin's method)

iii. $H(s) = \frac{2}{s^2+4s+4}$, 20 Hz (exact z-transform)

These systems will be stable in proportional feedback if the roots of the Characteristic Polynomial (CP): $0 = H_d + kH_n$ are within the unit circle.

i. CP: $0 = z - 1.105 + 0.1052k$

The pole is at $z_p = 1.105 - 0.1052k$

This will be stable if $|z_p| < 1$, which occurs when:

$$\boxed{0.9981 < k < 20.0095}$$

ii. Tustin's method uses the substitution $s = \frac{2}{T}\frac{z-1}{z+1}$. Substitute this into $H(s)$, and apply a lot of elbow grease. From inspection, it is clearly going to be a second order in both the numerator and denominator. A small dose of cleverness will indicate that you could do the substitution after you've computed the CP:

$$\text{CP: } 0 = s^2 + 4s + 4 + 2k$$

$$0 = \frac{2^2}{T^2}\frac{(z-1)^2}{(z+1)^2} + 4\frac{2}{T}\frac{z-1}{z+1} + 4 + 2k$$

This is a second-order polynomial in z. The pole-zero excess is zero; that means each pole will be attracted to a zero (both located at $z = -1$). Using the root-locus drawing guideline, it is clear that under feedback gain, the roots will remain complex until they converge with the zeros.

Put in simpler polynomial form, the CP becomes:

$$0 = 4(z-1)^2 + 8T(z-1)(z+1) + T^2(4+2k)(z+1)^2$$

This expands easily to:

$$(8T + 4T^2 + 4 + 2T^2k)z^2 + (8T^2 + 4T^2k - 8)z + (2T^2k - 8T + 4T^2 + 4)$$

This is a quadratic in z, so the roots will be given by the quadratic formula:

$$\frac{-b}{2a} \pm \frac{\sqrt{b^2 - 4ac}}{2a}$$

For stability, $|z| < 1$. The magnitude of the roots can be determined by the Pythagorean theorem. Note that :

$$Re\left(\frac{-b}{2a}\right)^2 + Re\left(\frac{\sqrt{b^2-4ac}}{2a}\right)^2 < 1^2$$

Observe that if the roots are complex, $b^2 - 4ac$ will be negative; b^2 is always positive, so $-4ac > b^2$. We can multiply the terms within the square root by -1 and take i outside the bracket (which is promptly annihilated by the $Re(\cdot)$ operator.

This yields:

$$\frac{b^2}{4a^2} + \frac{4ac - b^2}{4a^2} < 1$$

$$4ac < 4a^2$$

Or just simply $0 < a - c$

This simplifies to $16T$, which is always positive and has no dependence on k; thus the system is always stable provided our gain is sensible:

$$\boxed{k > 0}$$

iii. The exact z-transform is conveniently given by the transform table in the class slides:

$$\frac{1}{(s+a)^2} \rightarrow \frac{zTe^{-aT}}{(z - e^{-aT})^2}$$

From this we can go directly to the CP with little difficulty:

$$0 = (z - e^{-2T})^2 + 2kzTe^{-2T}$$

$$0 = z^2 + 2(kT - 1)e^{-2T}z + e^{-4T}$$

Once again we have a polynomial in z, except this time the pole-zero excess is one; a pole will be absorbed at the origin, and the other will escape to the left-half plane. Obviously, when the system goes unstable, the root will be at $z = -1$; therefore, we can use the factorisation to figure out what k must be at that point.

The CP will factorise to be:

$$0 = (z + 1)(z + q)$$

where q is the other root. From the coefficients of the CP, we observe that:

$$(1 + q) = 2(kT - 1)e^{-2T}$$

and

$$q = e^{-4T}$$

Thus:

$$k < \frac{(1 + e^{-4T})}{T2e^{-2T}} + \frac{1}{T}$$

Which in evaluates to $\boxed{k < 40.1001}$

5. Determine analytically the lowest frequency that does not destabilise $H(s) = \frac{1}{s^2+s+1}$, given proportional feedback gain of 48.

A quick look at the poles here indicates that this system is an oscillatory pair. No approximation was specified; we could convert this to a digital system in a number of ways. For fun, this solution uses the exact transform. **Don't panic** – your approximation probably won't break the answer *too* much. The z-transform for this system is:

$$H(s) = \frac{\omega}{(s + a)^2 + \omega^2} \rightarrow H(z) = \frac{ze^{-aT}\sin(\omega T)}{z^2 - 2e^{-aT}\cos(\omega T)\, z + e^{-2aT}}$$

Here, $a = 0.5$ and $\omega = \sqrt{0.75}$.

Using the same technique as before, the CP is:

$$z^2 + (\omega^{-1}48e^{-aT}\sin(\omega T) - 2e^{-aT}\cos(\omega T))z + e^{-2aT}$$

This time it is more difficult to tell whether the system will destabilize by crossing the unit circle with complex components or real components. The insight needed is that oscillatory poles will always orbit a zero at the origin; this zero is a important feature of the exact transform. As a consequence, as the poles follow the isocline path described with changing T, their root locus will always track a circle around the origin until they touch the negative real axis. From the breakpoint, one pole will be absorbed by the zero at the origin and the other will escape to the left-half plane.

The situation is now analogous to that of part 3.iii and the solution proceeds in the same way, with the exception that the trigonometric function in T must also be solved.

The root equivalence yields:

$$(1 + q) = \omega^{-1}48e^{-aT}\sin(\omega T) - 2e^{-aT}\cos(\omega T)$$

and

$$q = e^{-2aT}$$

Thus,

$$1 + e^{-2aT} = \omega^{-1} 48 e^{-aT} \sin(\omega T) - 2e^{-aT} \cos(\omega T)$$

That looks rather scary to solve; throw it into Matlab and it will (eventually) return $\boxed{T\ =\ 0.0833}$ (or something rather close to $1/12$).

The hand solution employs a third-order Taylor series expansion. Note the following approximations:

$$e^x \cong \frac{x^2}{2} + x + 1$$
$$\sin(x) \cong x$$
$$\cos(x) \cong 1 - \frac{x^2}{2}$$

Apply elbow-grease, and rather conveniently, the relation simplifies to:

$$0 = -\,24T^2\ +\ 50T\ -\ 4$$

Solving for T, the old fashioned way, you get $\boxed{T\ =\ 0.0833.}$

6. What design parameter governs the slowest allowable controller sample rate, and why? What design assumptions must be made in order to achieve this slowest rate?

   The parameters that sets the slowest admissible sample time is the bandwidth of the reference signal that the closed-loop system must track; the sample time must be greater than twice the desired bandwidth. This can be substantially slower than the open-loop bandwidth of the uncontrolled system, provided the plant is stable. An example of this is where a plant has unmodelled high-frequency mechanics such as vibrations; the real plant has a very high bandwidth the controller can ignore it and consider only the low desired low-frequency components. A filter may be required to prevent higher frequency dynamics do not alias into the range of the controller; they would appear as noise to the slow control loop.

7. A missile guidance system has a control computer that operates at 50 Hz. The missile's heading is governed by the following dynamic equations:

$$\mathrm{I}\ddot{\theta} = -c\dot{\theta} + k\theta + \tau$$

   Where $\mathrm{I} = 0.1$ is the rotational inertia of the missile, $c = 1.5$ is the aerodynamic damping applied by the stabiliser fins, k =10 is the torque due to nose drag, τ is the control torque generated by the guide fins, as commanded by the controller. The guide fin actuators are driven by a servomechanism with a two sample communication delay.

   Design a controller to regulate the missile's heading, and prove its stability analytically. Write a program that implements your controller in pseudo-code, noting all stored values and constants.

Computing the s-domain transfer function is once again straight-forward:

$$\mathrm{I}s^2\theta = -cs\theta + k\theta + \tau$$

$$\frac{\theta}{\tau} = \frac{10}{s^2 + 15s - 100}$$

$$\frac{\theta}{\tau} = H(s) = \frac{10}{(s-5)(s+20)}$$

For fun, I'm going to use Tustin's method and substitute $s = \frac{2}{T}\frac{z-1}{z+1}$.

$$H(z) = \frac{kT^2(z+1)^2}{(2(z-1) + aT(z+1))(2(z-1) + bT(z+1))}$$

Some algebraic elbow grease here:

$$H(z) = \frac{kT^2(z+1)^2}{(2Ta + 2Tb + T^2ab + 4)z^2 + (2T^2ab - 8)z + (T^2ab - 2Tb - 2Ta + 4)}$$

which evaluates to:

$$H(z) = \frac{0.004(z+1)^2}{4.56z^2 - 8.08z + 3.36}$$

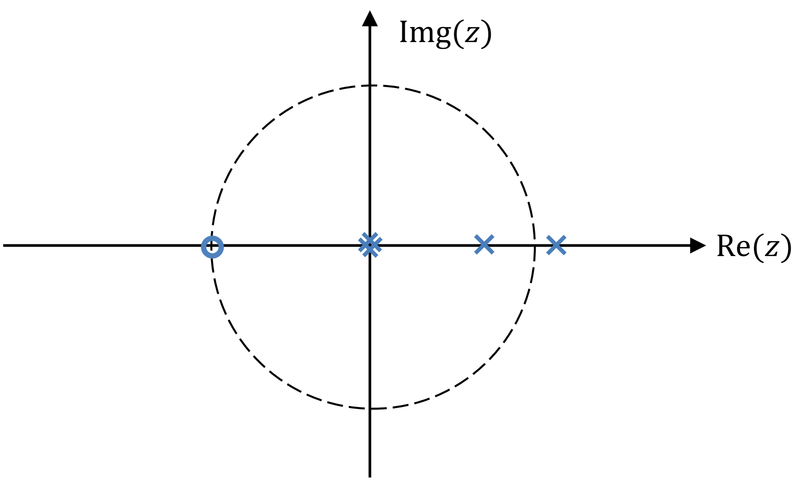
Or in ZPK form:

$$H(z) = \frac{8.77 \times 10^{-4}(z+1)^2}{(z-1.105)(z-0.6667)}$$

The effect of the two-sample delay is to add a pair of poles at $z = 0$:

$$H(z) = \frac{8.77 \times 10^{-4}(z+1)^2}{z^2(z-1.105)(z-0.6667)}$$

What does this look like on the root locus?

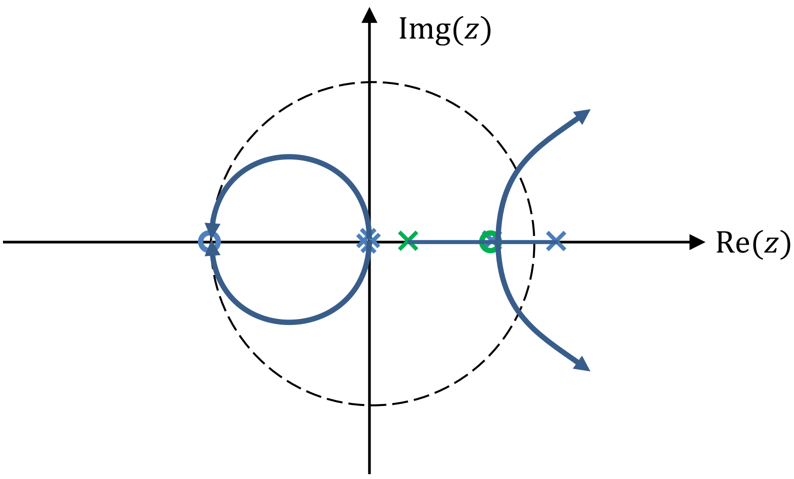


Clearly, the system is unstable. The initial instinct might be to throw a PID controller at it and use the zeros to absorb the dominant poles near $z = 1$. This is quite workable, but would make life a little harder: observe that the pole added by PID at $z = 1$ will take off into the right-half plane due to the influence of the two poles at the origin.

Note that there are many possible valid controllers that would successfully stabilize the plant. The sneaky solution is to simply apply proportional gain – the two poles near the origin share a breakpoint just inside the unit circle at $z = 0.96$. It is moderately straightforward to calculate the required gain to produce a double pole at that point. However, having the system dominant poles so close to the stability boundary is not generally good practice.

A more robust approach is to use a lead compensator, which cause the two dominant poles to move closer to added zero (and absorb one). The added pole will interact with the poles at the origin, but provided control gain is low enough, they should not escape the unit circle.

Put the zero somewhere nice and close to the stable dominant pole; in fact, as this pole is stable, the pole could be cancelled by setting the zero $z = 0.66$. The associated lead pole would go at a faster position, such as $z = 0.25$.



The pole-zero excess is three, and so two of the dynamics/controller poles will head off to the right-half plane, one of the delay poles will be absorbed by the zero at $z = -1$ and one will head off into the left-half plane. By inspection, the distance that the delay poles must travel is great, whereas that of the dynamics poles is small. We can stop worrying about the delay pole and focus on the stability goal.

We can approximate the gain required by calculating the breakpoint as if the system only consisted of the dynamics poles and the controller. This need not be exact, as we only desire a sense of the required gain magnitude.

The approximate system is (using the FVT DC gain for scaling):

$$H(z)_{approx} = \frac{2.66k(z-0.667)}{(z-0.25)} \frac{0.0035}{(z-1.105)(z-0.667)}$$

Cancelling the matched pole and zero, the CP of the approximate feedback system will be:

$$0 = (z-0.25)(z-1.105) + k0.0093$$
$$0 = z\text{^}2 - 1.355\,z + 0.2763 + k0.0093$$

A safe stable configuration can be reached by completing the square such that the two poles are co-located.

That is, when $(0.2763 + k0.0093) = \left(\frac{1.355}{2}\right)^2$; this occurs at $k = 19.6458$

Thus, an example control solution would be:

$$\boxed{C(z) = 52.25\frac{(z - 0.667)}{(z - 0.25)}}$$

This can be further expanded upon by adding lag controller close to the origin – not enough to affect the dominant dynamics, but enough to reduce the DC tracking error.

In practice, this approach underestimates the effect of the gain, due to the influence of the delay poles. It is important to double-check that a system with this gain would in fact be stable. Throwing everything back into the full system CP:

$$0 = z^4 - 1.356\,z^3 + 0.322\,z^2 + 0.09159\,z + 0.04584$$

You can prove the stability of this polynomial using Routh-Hurwitz or direct factorization if so desired.

To convert the controller to a program, you need to convert the system to a difference equation:

$$\frac{\tau}{\theta} = 52.25\frac{(z - 0.667)}{(z - 0.25)}$$

$$\tau(z - 0.25) = 52.25(z - 0.667)\theta$$

$$\tau(k + 1) - 0.25\tau(k) = 52.25\theta(k + 1) - 34.85\theta(k)$$

$$\tau(k + 1) = 52.25\theta(k + 1) - 34.85\theta(k) + 0.25\tau(k)$$

This system must be made causal, otherwise it cannot be implemented. Arguably, we should have taken account of that when designing our controller… but we'll let it slide. ☺

$$\tau(k) = 52.25\theta(k) - 34.85\theta(k - 1) + 0.25\tau(k - 1)$$

Now we can put this into a program. Note that we have to store the past history of both the input and output for one sample each.

```
/*This doesn't have to be super-hardcore realistic code or
 *anything.  Mostly you just need to show that you put the
 *controller into causal form and then implemented it with
 *a time-history of previous values. If you actually used
 *some kind of periodic flow control on executing the
 *algorithm, then that's even better
 */

//Definitions here
A = 52.25
B = 34.85
C = 0.25

while(1)
     //Some sort of periodic flow control here
     if(control_interrupt_flag = 1)
     {
          theta0 = theta;
          tau0 = tau;
          theta = get_new_measurement();
          tau = A*theta - B*theta0 + C*tau0;
          output(tau);
     }
```