Problem 2. [20 pts. total] *Exploring the Closed Loop Transfer Function.*

In class we talked about the closed-loop transfer function of a second-order loop. It was given by the formulas

$$F(s) = \frac{\hat{\Theta}(s)}{\Theta(s)} = \frac{K_t(s+a)}{s^2 + K_t s + aK_t} = \frac{(4\pi\zeta f_n)s + 4\pi^2 f_n^2}{s^2 + (4\pi\zeta f_n)s + 4\pi^2 f_n^2}$$

in terms of the damping coefficient ζ and the undamped natural frequency f_n. Substituting $s = j2\pi f$ into the above gives the CTFT of the closed-loop transfer function.

(a) Plot the closed loop transfer function on a log-log scale for various values of ζ and f_n. If you wish, you can use a frequency variable normalized by f_n as done on the HW solution to simplify the plots. Given this plot comment on the sampling frequency that would seem to be needed for a discrete-time simulation.

(b) Once the continuous-time integrators in the block diagram are replaced by the trapezoidal approximation, the simulation code that remains implements a discrete-time system. Write down the block diagram for the discrete-time system and give the equations that describe it.

(c) Linearizing the discrete-time system by replacing the sine nonlinearity in the phase detector with the identity, find the equivalent LTI discrete-time system that represents the linearized PLL. Plot the DTFT of the resulting closed-loop transfer function and compare with the continuous-time result.

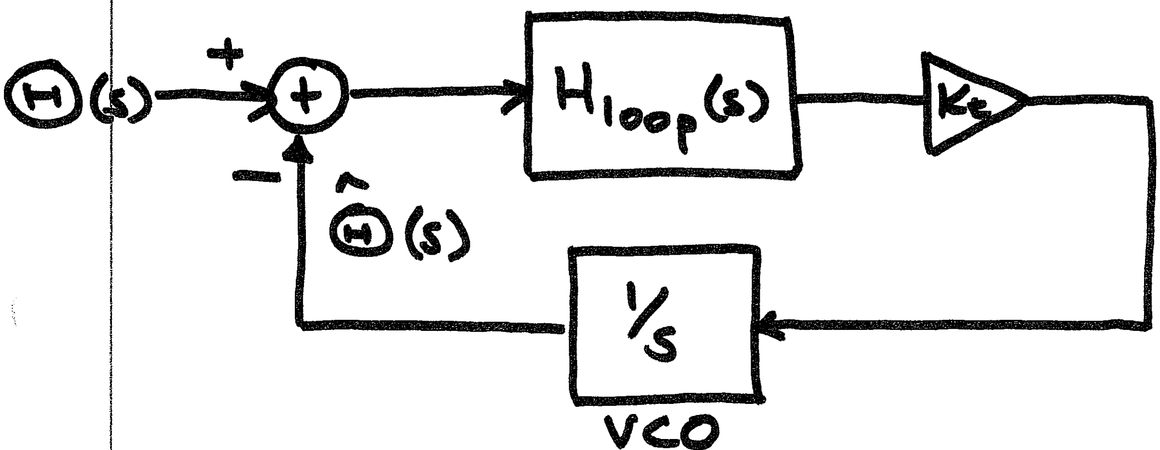
(d) Use Matlab to make plots and compare.

Problem 2

Exploring the Closed Loop Transfer Function.

$$F(s) = \frac{\hat{\Theta}(s)}{\Theta(s)} = \frac{K_t(s+a)}{s^2 + K_t s + aK_t}$$

This came from the Laplace Transform of the linearized baseband model...



The closed loop transfer function above results from inserting

$$H_{loop}(s) = \frac{s+a}{s}$$

into the block diagram, solving for F(s), and simplifying.

F(s) is a generic 2nd order transfer function ... such typically have a denominator polynomial of the form

$$s^2 + 2\xi\omega_n s + \omega_n^2$$

ξ is the damping coefficient and ω_n is the undamped natural frequency in radians per sec. Often write it in Hz ...

$$2\pi f_n = \omega_n = \sqrt{aK_t}$$

$$\boxed{f_n = \frac{1}{2\pi}\sqrt{aK_t}} \Longrightarrow$$

$$2\xi \cdot 2\pi f_n = K_t \Longrightarrow \boxed{\xi = \frac{1}{2}\sqrt{\frac{K_t}{a}}}$$

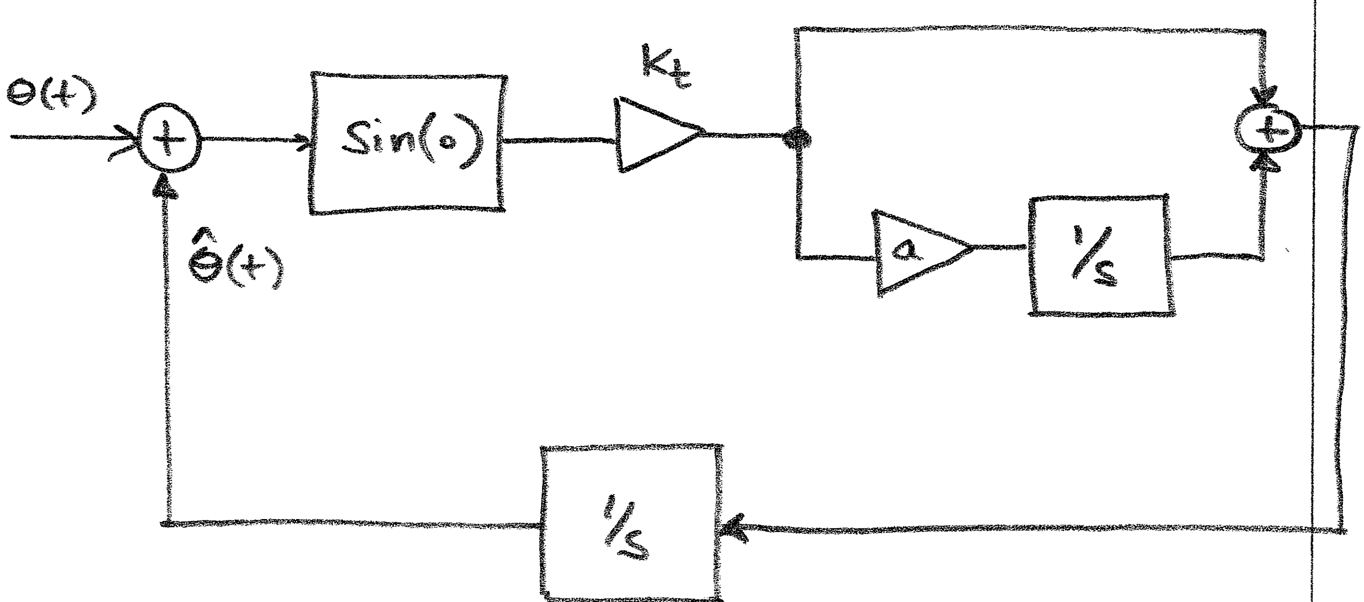
With this the closed loop transfer function is written

$$F(s) = \frac{(4\pi\xi f_n)s + 4\pi^2 f_n^2}{s^2 + (4\pi\xi f_n)s + 4\pi^2 f_n^2}$$

with $s = j2\pi f$ inserted above and simplifying ...

$$F(j2\pi f) = \frac{1 + j2\xi(f/f_n)}{1 - (f/f_n)^2 + j2\xi(f/f_n)}$$

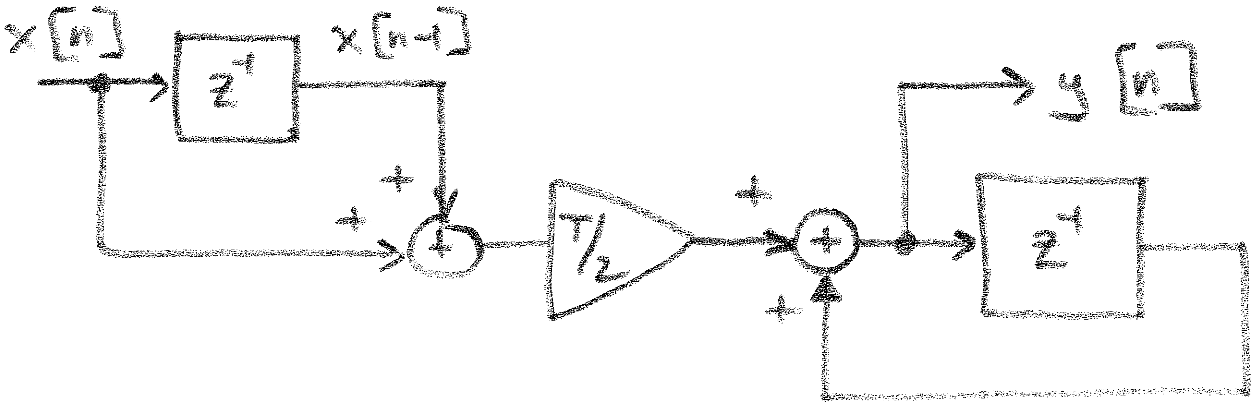
After inserting $H_{loop}(s) = \frac{s+a}{s}$ into the block diagram an implementation is ...



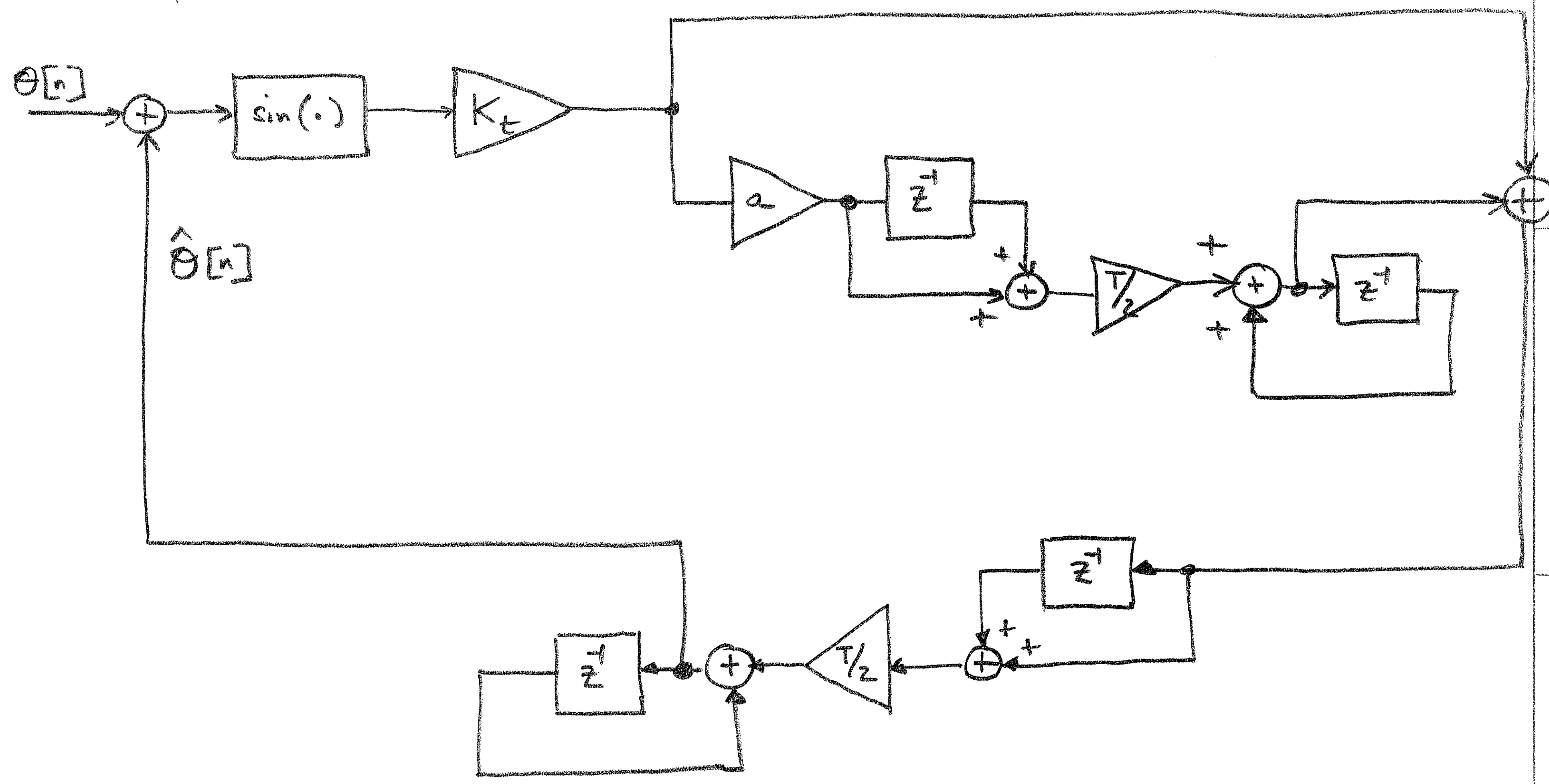
We've been instructed to implement the integrators we a discrete-time recursion ...

$$y[n] = y[n-1] + \frac{T}{2}\left\{ x[n] + x[n-1] \right\}$$

where $x[n]$ is the input and $y[n]$ is the output of the integrator. One possible implementation would be

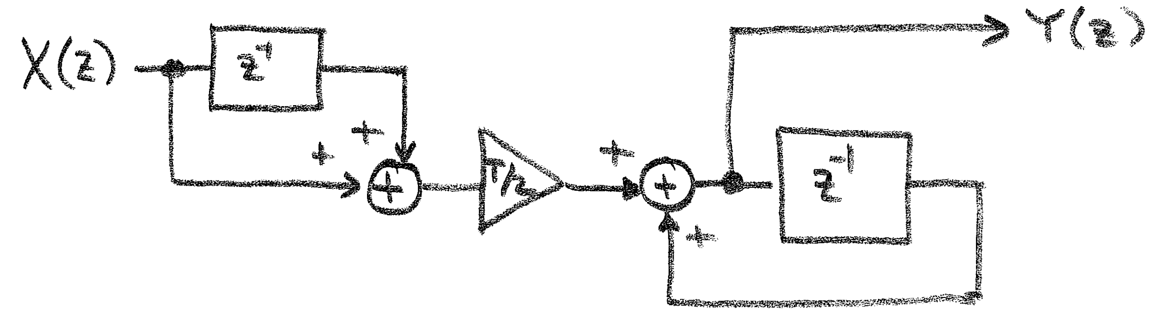


Discretizing $\Theta(nT) = \Theta[n]$, $\hat{\Theta}(nT) = \hat{\Theta}[n]$ and inserting the trapezoidal integration blocks in place of $1/s$ ...



To linearize the previous block diagram we simply replace the sine non-linearity with the identity.

Define T(z) to be the transfer function of a trapezoidal integrator. To find T(z) we take the z-transform of the basic block diagram:



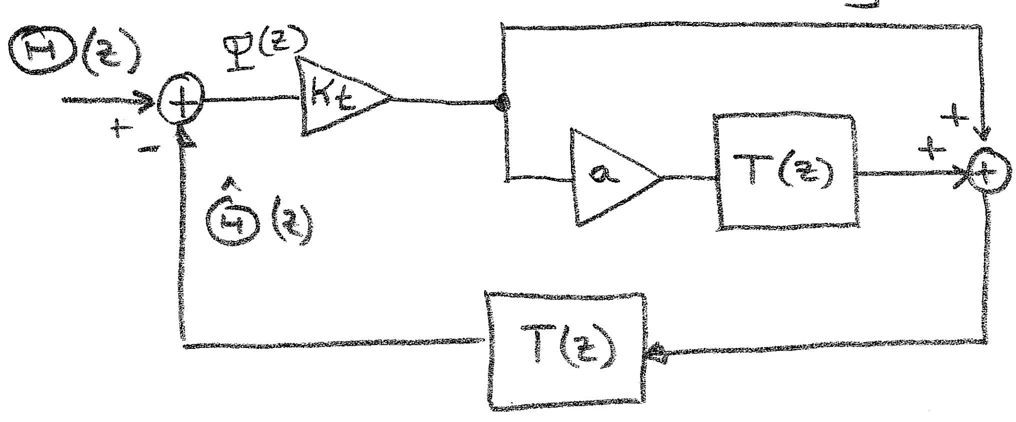
$$\Rightarrow \quad Y(z) = z^{-1}Y(z) + \frac{T}{2}\left\{X(z) + z^{-1}X(z)\right\}$$

$$= z^{-1}Y(z) + \frac{T}{2}\left[1+z^{-1}\right]X(z)$$

$$Y(z)\left[1-z^{-1}\right] = \frac{T}{2}\left[1+z^{-1}\right]X(z)$$

$$\Rightarrow \quad T(z) = \frac{Y(z)}{X(z)} = \frac{\frac{T}{2}\left(1+z^{-1}\right)}{1-z^{-1}}$$

Inserted into the linearized block diagram



$$\hat{\Theta}(z) = T(z)\left[K_t \Psi(z) + aK_t T(z)\Psi(z)\right]$$

$$= K_t T(z)\left[1 + aT(z)\right] \underset{\hat{\Theta}(z) - \hat{\hat{\Theta}}(z)}{\underset{\wedge}{\Psi(z)}}$$

$$\Longrightarrow \hat{\Theta}(z)\left[1 + K_t T(z)\left[1 + aT(z)\right]\right] = K_t T(z)\left[1 + aT(z)\right]\Theta(z)$$

$\Longrightarrow$ Transfer function is

$$\frac{\hat{\Theta}(z)}{\Theta(z)} = \frac{K_t T(z)\left[1 + aT(z)\right]}{1 + K_t T(z)\left[1 + aT(z)\right]}$$

Lets plot this $z = e^{j\omega} = e^{j2\pi\lambda}$ λ is the discrete time frequency variable. Recall that the relationship between discrete-time and continuous-time frequencies is $\lambda = fT$

$T \hookrightarrow$ sampling interval

In the DT transfer function above it might help to normalize wrt f_n as we did when plotting the closed loop transfer function in continuous time.

$$K_t T(z) = K_t \frac{T}{2}\,\frac{1+z^{-1}}{1-z^{-1}} = 4\pi\zeta f_n \frac{T}{2}\,\frac{1+z^{-1}}{1-z^{-1}}$$

$$= 2\pi\zeta (f_n T)\,\frac{1+z^{-1}}{1-z^{-1}}$$

$$aK_t T^2(z) = 4\pi^2 f_n^2 \left(\frac{T}{2}\right)^2 \left(\frac{1+z^{-1}}{1-z^{-1}}\right)^2$$

$$= \pi^2 (f_n T)^2 \left(\frac{1+z^{-1}}{1-z^{-1}}\right)^2$$

Then define $\tilde{T}(z) = \dfrac{1+z^{-1}}{1-z^{-1}}$ and $\lambda_n = f_n T$

$$\frac{\hat{\Theta}(z)}{\Theta(z)} = \frac{2\pi\zeta\lambda_n \tilde{T}(z) + \pi^2 \lambda_n^2 \tilde{T}^2(z)}{1 + 2\pi\zeta\lambda_n \tilde{T}(z) + \pi^2 \lambda_n^2 \tilde{T}^2(z)}$$

$$= \frac{\left[2\zeta + \pi\lambda_n \tilde{T}(z)\right]\pi\lambda_n \tilde{T}(z)}{1 + \left[2\zeta + \pi\lambda_n \tilde{T}(z)\right]\pi\lambda_n \tilde{T}(z)}.$$

$$F(j2\pi f) = \frac{1 + j2\zeta\left(\frac{fT}{f_nT}\right)}{1 - \left(\frac{fT}{f_nT}\right)^2 + j2\zeta\left(\frac{fT}{f_nT}\right)}$$

$f_nT = 0.1$

$\frac{f_n}{0.1} = \frac{1}{T}$

$10 f_n$

$fT = 0.5$

$f_nT = 0.5$

1

$fT = 0.5$

$f = 0.5\frac{1}{T} = 0.5 \cdot 10 f_n$

$= 2f_n.$

$\lambda_n = f_nT = 0.5 \longrightarrow \frac{1}{T} = 2f_n$

$\lambda = fT \qquad -0.5 < \lambda < 0.5$

$L = \frac{\lambda}{\lambda_n} \qquad -1 < L < +1$

$= \frac{fT}{f_nT} = \frac{f}{f_n}$

$\lambda = 0.5$

$fT = 0.5$

$f = \frac{0.5}{T} = f_n$

Probably Correct.

Only question: Z version should be an aliased version of CT... yet it has lower value at foldover. Therefore the copies must be cancelling, not adding.

Phase angle shows why. They all subtract.

Exploring the Closed Loop Transfer Function. Problem 2.

ECE 440. Take Home Exam. Spring 2022.

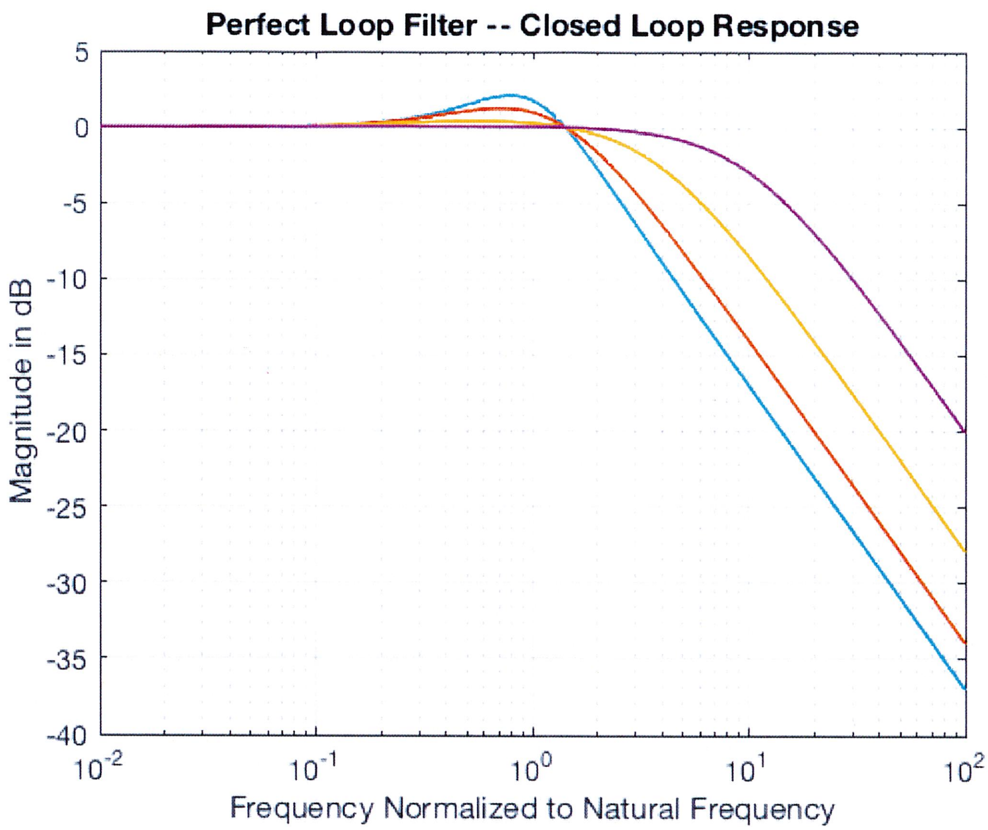
File = THE_P2.m

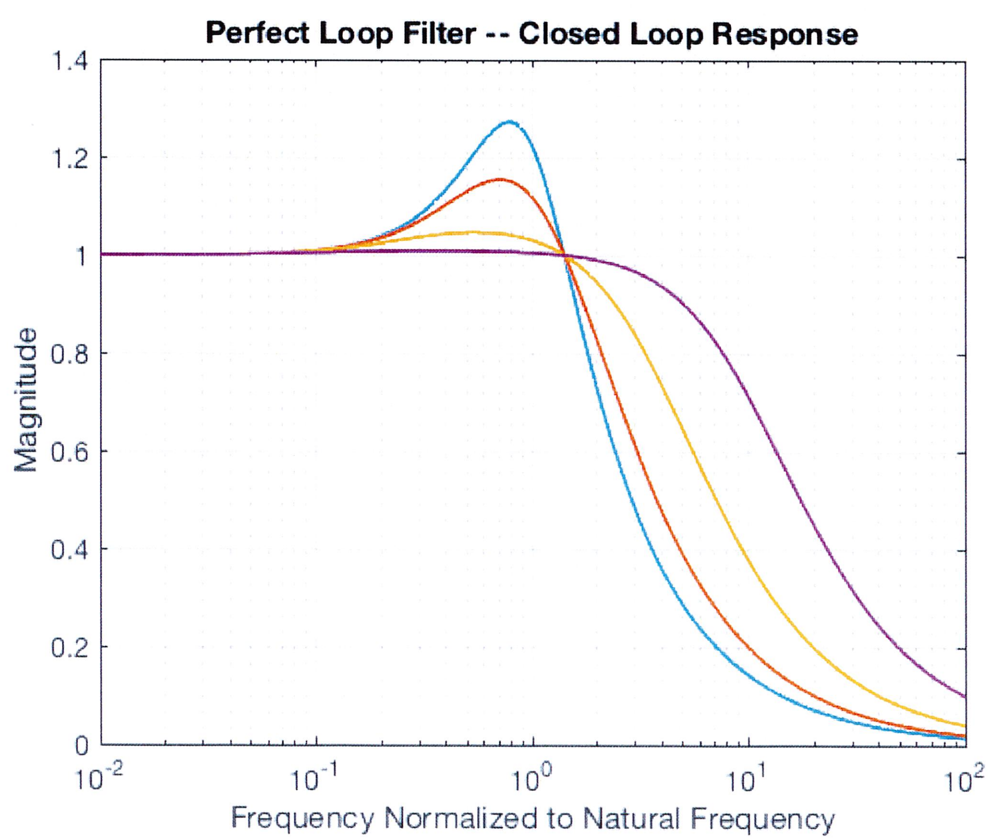
Contents

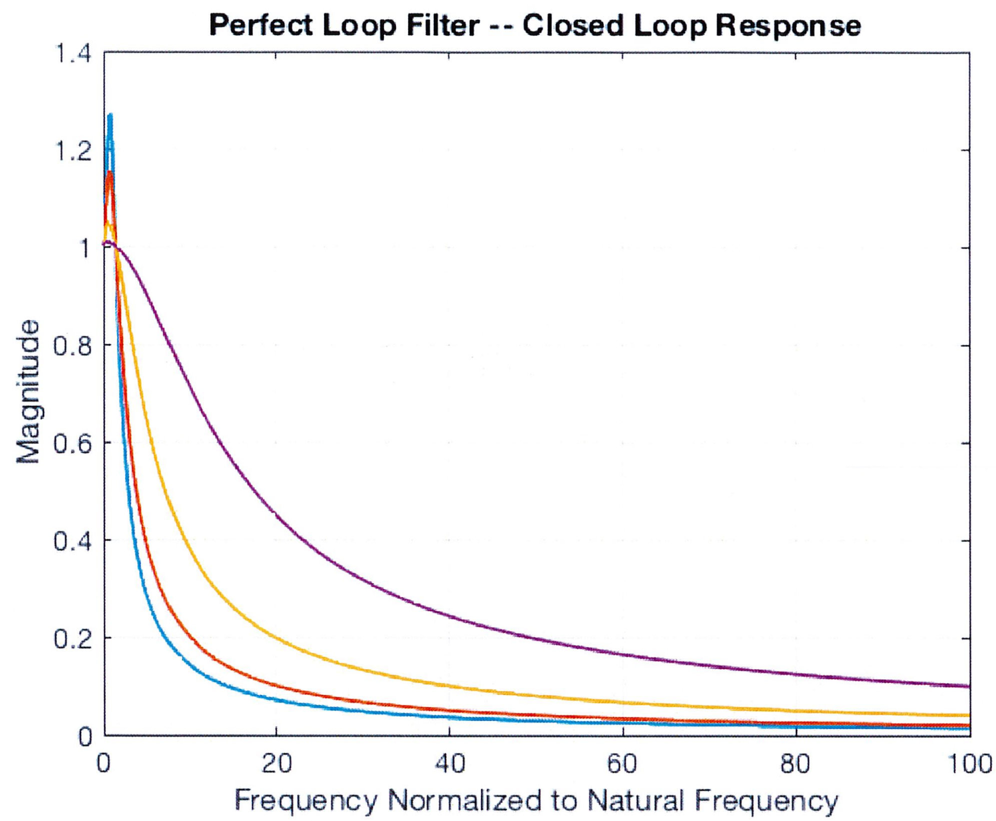
Part (a).

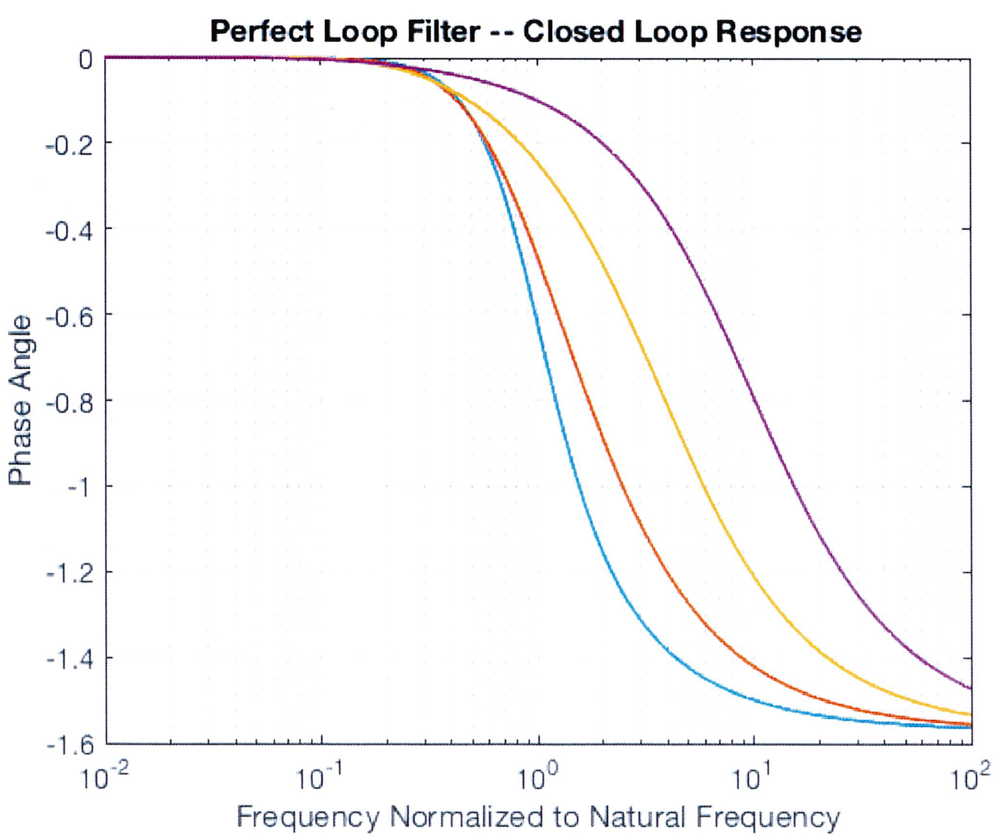
```
clear all;                          %Be safe
close all;

zeta = [0.707, 1.0, 2.0, 5.0];     %Values of damping factor to test
N = length(zeta);

f = logspace(-2,2,200);             %Values for the normalized frequency
                                    %variable

F = zeros(N,length(f));
FdB = zeros(N,length(f));
A = zeros(N,length(f));

for i = 1:N
%    den = (1 - f .^ 2) + j*2*zeta(i)*f;
    num = 1 + j*2*zeta(i)*f;
    den = num - (f .^ 2);
    FdB(i,:) = 20*log10(abs(num ./ den));
    F(i,:) = abs(num ./ den);
    A(i,:) = angle(num ./ den);
end

figure(1)

semilogx(f,FdB,'LineWidth',2)
title('Perfect Loop Filter -- Closed Loop Response')
xlabel('Frequency Normalized to Natural Frequency')
ylabel('Magnitude in dB')
%axis([1e-2 1e2 -90 10]);
grid
set(gca,'Fontsize',14)

figure(2)

semilogx(f,F,'LineWidth',2)
title('Perfect Loop Filter -- Closed Loop Response')
xlabel('Frequency Normalized to Natural Frequency')
ylabel('Magnitude')
%axis([1e-2 1e2 -90 10]);
grid
set(gca,'Fontsize',14)

figure(3)

plot(f,F,'LineWidth',2)
```

```
title('Perfect Loop Filter -- Closed Loop Response')
xlabel('Frequency Normalized to Natural Frequency')
ylabel('Magnitude')
%axis([1e-2 1e2 -90 10]);
grid
set(gca,'Fontsize',14)

figure(4)
semilogx(f,A,'LineWidth',2)
title('Perfect Loop Filter -- Closed Loop Response')
xlabel('Frequency Normalized to Natural Frequency')
ylabel('Phase Angle')
%axis([1e-2 1e2 -90 10]);
grid
set(gca,'Fontsize',14)
```


Perfect Loop Filter -- Closed Loop Response
Magnitude in dB
5
0
-5
-10
-15
-20
-25
-30
-35
-40
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
Frequency Normalized to Natural Frequency


Perfect Loop Filter -- Closed Loop Response
Magnitude
1.4
1.2
1
0.8
0.6
0.4
0.2
0
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
Frequency Normalized to Natural Frequency



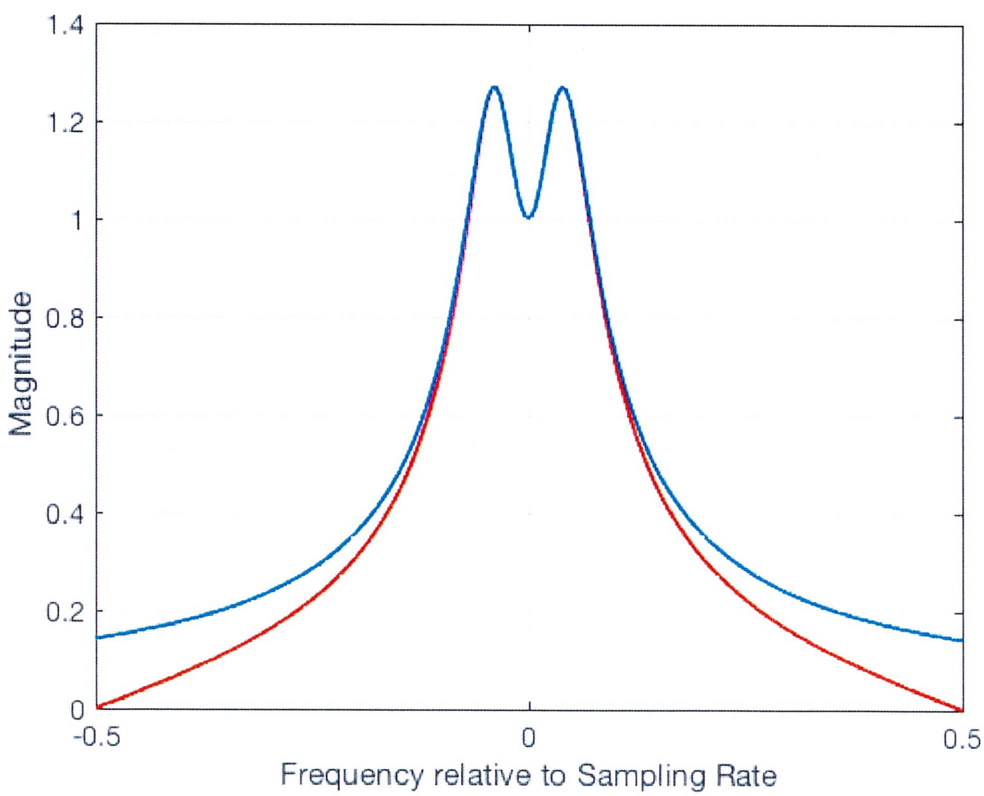


Discrete Time Transfer Function ...

```
lambda = linspace(-0.5,0.5,200);
z = exp(j*2*pi*lambda);
```

```
lambdan = 0.05;
Tztilde = (1 + z .^ (-1)) ./ (1 - z .^ (-1));

zeta = .707;

Num = (2*zeta + pi*lambdan*Tztilde) .* (pi*lambdan*Tztilde);
Den = 1 + Num;

Ftilde = Num ./ Den;

L = lambda/lambdan;

snum = 1 + j*2*zeta*L;
sden = snum - (L .^ 2);

F = snum ./ sden;

figure(5)

plot(lambda,abs(Ftilde),'LineWidth',2,'Color','r')
grid
xlabel('Frequency relative to Sampling Rate')
ylabel('Magnitude')
set(gca,'Fontsize',14)
hold
plot(lambda,abs(F),'LineWidth',2,'Color','b')
```

```
Current plot held
```



Published with MATLAB® R2021a