(a) Original frequency spectrum of 20 Hz sine

(b) Frequency spectrum of sine sampled at 100 Hz

(c) Frequency spectrum of reconstructed sine with ZOH

|\sin \omega t| A

The 'steps' of the ZOH correlate to the additional frequencies introduced into the signal's frequency spectrum.
% Problem 2, Part (a)
% D1 difference equation:
% \( y_1(k) = 1.3*y_1(k-1)-0.8*y_1(k-2)+0.205*y_1(k-3)+u(k)-u(k-1) \)

% D2 difference equation:
% \( y_2(k) = u(k-2)+0.3*u(k-3)-0.42*u(k-4)-0.58*u(k-5)-0.3575*u(k-6)-0.0788*u(k-7)+0.0687*u(k-8) + 0.791*u(k-9)+0.0312*u(k-10)-0.0094*u(k-11) \)

% Matlab does not allow indices less than one, so let j=k+4
y1(1)=0; y1(2)=0; y1(3)=0;
u=ones(29,1);
u(1)=0; u(2)=0; u(3)=0;

for k=0:25
  j=k+4;
  y1(j) = 1.3*y1(j-1)-0.8*y1(j-2)+0.205*y1(j-3)+u(j-2)-u(j-3);
end
y1=y1(4:29);

% Here, need to let j=k+12;
u(1:11)=0; u(12:37)=1;
for k=0:25
  j=k+12;
  y2(j) = u(j-2)+0.3*u(j-3)-0.42*u(j-4)-0.58*u(j-5)-0.3575*u(j-6)-0.0788*u(j-7)+0.0687*u(j-8) + 0.791*u(j-9)+0.0312*u(j-10)-0.0094*u(j-11);
end
y2=y2(12:37);

plot([0:25],y1,'rd'), hold on, plot([0:25],y2,'bo'), hold off
xlabel('time, samples') ylabel('output, y1 and y2')
% Part (b)

% Vectorized computation of Transfer functions, including ZOH
w=logspace(-1,2,100000)*2*pi; w=w';
s=sqrt(-1)*w;
T=1/50;
z=exp(T*s);
ZOH=(1-exp(-s*T))./s;
G1z=(z-1)./(z.^3-1.3*z.^2+0.81*z-0.205);
G2z=(z.^9+0.3*z.^8-0.42*z.^7-0.58*z.^6-0.3575*z.^5-0.0788*z.^4+0.0687*z.^3+0.791*z.^2+0.0312*z-0.0094)./z.^11;
G1=G1z.*ZOH;
G2=G2z.*ZOH;

% Plotting
figure
subplot(211)
semilogx(w/2/pi,20*log10([G1 G2]))
ylabel('Bode Magnitude, dB'), axis([10^(-1) 10^2 -60 -20])
subplot(212)
semilogx(w/2/pi,180/pi*angle([G1 G2]))
xlabel('Frequency, Hz')
ylabel('Bode Phase, deg'), axis([10^(-1) 10^2 -180 180])
Solution with $z^2$ coefficient = 0.0791 instead of 0.791 - - the FIR works a lot better now! (still will not track at low frequency, you can just see it starting to deviate at the low end of the frequency scale in the transfer function).
3. From PS#3:
\[ G(s) = \frac{10}{s(s+0.02)} \times \frac{8}{s^2+0.01s+0.002} \]
\[ G_c(s) = \frac{0.00074(19.77s+1)}{(1.977s+1)} \]

For compensated system:
\[ \omega_c = 0.16 \text{ rad/s} \quad \phi_m = 50^\circ \]

a) choose sample rate:
Nyquist sample rate: \( \omega_s = \frac{\omega_c}{2} \)
\[ \omega_s = \frac{2\pi}{T} \]

As a rule of thumb, want system pole location
to be \( \frac{1}{6} \) of \( \omega_s \)
\( \frac{1}{1.977} < \omega_s / 6 \Rightarrow \omega_s = 3.03 \)
\[ T = \frac{2\pi}{\omega_s} = \frac{2.07s}{s} \]

b) discrete time transfer function
\[ D(z) = G_c(s) \bigg|_{s = \frac{2}{T} \frac{z-1}{z+1}} \]
\[ = \frac{0.00074(19.77 \frac{z-1}{z+1} + 1)}{1.977 \frac{z-1}{z+1} + 1} \]
\[ = \frac{0.029(z-1) + 0.0015(z+1)}{3.95(z-1) + 2.07(z+1)} \]
\[ = \frac{0.0305z - 0.0275}{6.02z - 1.88} \]
3 c) Bode implementation:

\[ G_c = ZOH \cdot \left( D(z) \bigg|_{z = e^{sT}} \right) \]

\[
\frac{1 - e^{-Ts}}{s} \left( \frac{0.0305 e^{13} - 0.0275}{6.02 e^{73} - 1.88} \right)
\]

implementing \( G_c G_c \) in Matlab:

\[
take: \quad G_c G_c \bigg|_{s = j\omega} \]

(see attached plots of continuous and discrete time implementations)

d) The problem with the discrete time implementation is that system uncertainties are large at high frequencies.
PS5 Q3: Continuous-time Controller Implementation

PS5 Q3: Discrete-time Controller Implementation
From the lab handout, the linearized equations of motion around hover are:

\[
\begin{align*}
I_{yy}\delta \dot{\phi} &= \left[-\tau_{\text{coll,}l_{\text{boom}}}\sin(\phi_0)\right]\delta \phi + (K_T\delta \omega_{\text{coll}} - K_c\delta \psi)l_{\text{boom}}\cos(\phi_0) - M_a\delta \dot{\theta} - [Mg\cos(\theta_{\text{rest}})]\delta \theta \\
I_{xx}\delta \dot{\phi} &= (K_T\delta \omega_{\text{cyc}} - K_c\delta \psi)l_h - [mgl_\phi \cos(\phi_0)]\delta \phi - L_p\delta \dot{\phi} \\
I_{zz}\delta \dot{\psi} &= \left[\tau_{\text{coll,}l_{\text{boom}}}\cos(\phi_0)\right]\delta \phi + (K_T\delta \omega_{\text{coll}} - K_c\delta \psi)l_{\text{boom}}\sin(\phi_0) - M_\phi\delta \dot{\theta} - [K_Dl_{\text{boom}}\cos(\gamma_0)]\delta \psi
\end{align*}
\]

In addition, the motor dynamics can be modeled by:

\[
\begin{align*}
\delta \omega_{\text{cyc}} + 6\delta \omega_{\text{cyc}} &= 780\delta V_{\text{cyc}} \\
\delta \omega_{\text{coll}} + 6\delta \omega_{\text{coll}} &= 540\delta V_{\text{coll}}
\end{align*}
\]

The dynamics of the Quanser if the inputs are \(\delta \omega_{\text{cyc}}\) and \(\delta \omega_{\text{coll}}\) in state space are:

\[
\begin{bmatrix}
\delta \dot{\theta} \\
\delta \dot{\phi} \\
\delta \dot{\psi} \\
\delta \omega_{\text{cyc}} \\
\delta \omega_{\text{coll}}
\end{bmatrix} =
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
-Mgs \cos(\theta_{\text{rest}}) & -M_a & -\tau_{\text{coll,}l_{\text{boom}}} \sin(\phi_0) & 0 & 0 \\
I_{yy} & -1 & 0 & 0 & 0 \\
0 & 0 & -mg_l \cos(\phi_0) & -L_p & 0 \\
0 & 0 & 0 & -K_Tl_h & 0 \\
0 & 0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
\delta \theta \\
\delta \phi \\
\delta \psi \\
\delta \omega_{\text{cyc}} \\
\delta \omega_{\text{coll}}
\end{bmatrix} +
\begin{bmatrix}
0 \\
0 \\
0 \\
0 \\
0
\end{bmatrix}
\]

The dynamics of the Quanser if the inputs are \(\delta V_{\text{cyc}}\) and \(\delta V_{\text{coll}}\) in state space are:

\[
\begin{bmatrix}
\delta \dot{\theta} \\
\delta \dot{\phi} \\
\delta \dot{\psi} \\
\delta \omega_{\text{cyc}} \\
\delta \omega_{\text{coll}}
\end{bmatrix} =
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
-Mgs \cos(\theta_{\text{rest}}) & -M_a & -\tau_{\text{coll,}l_{\text{boom}}} \sin(\phi_0) & 0 & 0 \\
I_{yy} & -1 & 0 & 0 & 0 \\
0 & 0 & -mg_l \cos(\phi_0) & -L_p & 0 \\
0 & 0 & 0 & -K_Tl_h & 0 \\
0 & 0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
\delta \theta \\
\delta \phi \\
\delta \psi \\
\delta \omega_{\text{cyc}} \\
\delta \omega_{\text{coll}}
\end{bmatrix} +
\begin{bmatrix}
0 \\
0 \\
0 \\
0 \\
0
\end{bmatrix}
\]
5. a) \( \dot{X} = \begin{bmatrix} -11 & -3 \\ 9 & 1 \end{bmatrix} X \)

Let \( A = \begin{bmatrix} -11 & -3 \\ 9 & 1 \end{bmatrix} \)

Find eigenvalues: \( (sI - A) = 0 \)
\[
\begin{vmatrix} s + 11 & 3 \\ -9 & s - 1 \end{vmatrix} = (s + 11)(s - 1) + 27 = s^2 + 10s + 16 = 0
\]
\( (s + 8)(s + 2) = 0 \) \( \Rightarrow \) \( s_1 = -8 \), \( s_2 = -2 \)

Find eigenvectors:
\( s = -8 \)
\[
\begin{bmatrix} 3 & 3 \\ -9 & -9 \end{bmatrix} v_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Rightarrow v_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}
\]
\( s = -2 \)
\[
\begin{bmatrix} 9 & 3 \\ -9 & -3 \end{bmatrix} v_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Rightarrow v_2 = \begin{bmatrix} 1 \\ -3 \end{bmatrix}
\]

Diagonalization of \( A = S \Lambda S^{-1} \)
\[
A = \begin{bmatrix} \frac{1}{2} & 1 \\ -1 & -3 \end{bmatrix} \begin{bmatrix} -8 & 0 \\ 0 & -2 \end{bmatrix} \begin{bmatrix} -3 & -1 \\ 1 & 1 \end{bmatrix}
\]
5. b. \( x_0 = \begin{bmatrix} 1 \\ -3 \end{bmatrix} \)

\( x_0 \) is aligned with an eigenvector.

\[ x_0 = \begin{bmatrix} 3 \\ -3 \end{bmatrix} \]

\( x_0 \) is aligned with an eigenvector.

c) The initial conditions lie along the eigenvectors.
So, the trajectory over time will lie along the initial condition direction.
Part (a)

\[
\begin{bmatrix}
23 & 49 \\
-19 & -30
\end{bmatrix}
\]
\[
[V, D] = \text{eig}(A)
\]
\[
V =
\begin{bmatrix}
0.8489 & 0.8489 \\
-0.4591 + 0.2620i & -0.4591 - 0.2620i
\end{bmatrix}
\]
\[
D =
\begin{bmatrix}
-3.5000 + 15.1245i & 0 \\
0 & -3.5000 - 15.1245i
\end{bmatrix}
\]
\[
T =
\begin{bmatrix}
0.8489 & 0 \\
-0.4591 & 0.2620
\end{bmatrix}
\]
\[
A_z = \text{inv}(T) \cdot A \cdot T
\]
\[
A_z =
\begin{bmatrix}
-3.5000 & 15.1245 \\
-15.1245 & -3.5000
\end{bmatrix}
\]

Part (b)

\[
x_0 = [1; -3]
\]
\[
\text{alp} = \text{inv}(V) \cdot x_0
\]
\[
\text{alp} =
\begin{bmatrix}
0.5890 + 4.6928i \\
0.5890 - 4.6928i
\end{bmatrix}
\]
\[
t = [0:0.01:20];
\]
\[
x = \text{alp}(1) \cdot V(:,1) \cdot \exp(D(1,1) \cdot t) +
\]
\[
\text{alp}(2) \cdot V(:,2) \cdot \exp(D(2,2) \cdot t);
\]
\[
\text{plot}(\text{real}(x(1,:)), \text{real}(x(2,:)));
\]

Part (c)

\[
z = \text{inv}(T) \cdot x;
\]
\[
\text{plot}(z(1,:), z(2,:), 'r');
\]
\[
\text{axis('equal')}
\]
Repeat for both values of \(x_0\):

Part (d)

The transformation matrix \(T\) provides both scaling and rotation of the states. This scaling and rotation does not change the fundamental properties of the system (the eigenvalues), but the state space picture is changed from a skewed, tilted picture to a ‘canonical’ picture of a harmonic oscillator. Thus the ‘direction’ of the eigenvector indicates how the states \(x_1\) and \(x_2\) (which might be physical, rather than purely mathematical, quantities) are inter-related in the oscillation – in the direction of high ‘stiffness’, the perturbations are smaller, and in the orthogonal direction they are larger.

Note that if we were to compute the outputs of the system \(\dot{x} = Ax + Bu, y = Cx + Du\), as well as the outputs of the transformed system,

\[
\dot{x} = T^{-1} A T x + T^{-1} B u
\]
\[
y = C T x + D u,
\]
Those outputs would be identical for the original and transformed system – in other words the transformation yields a system that is ‘similar’ in an input-output sense; hence the name ‘similarity transformation’.