Table of Contents

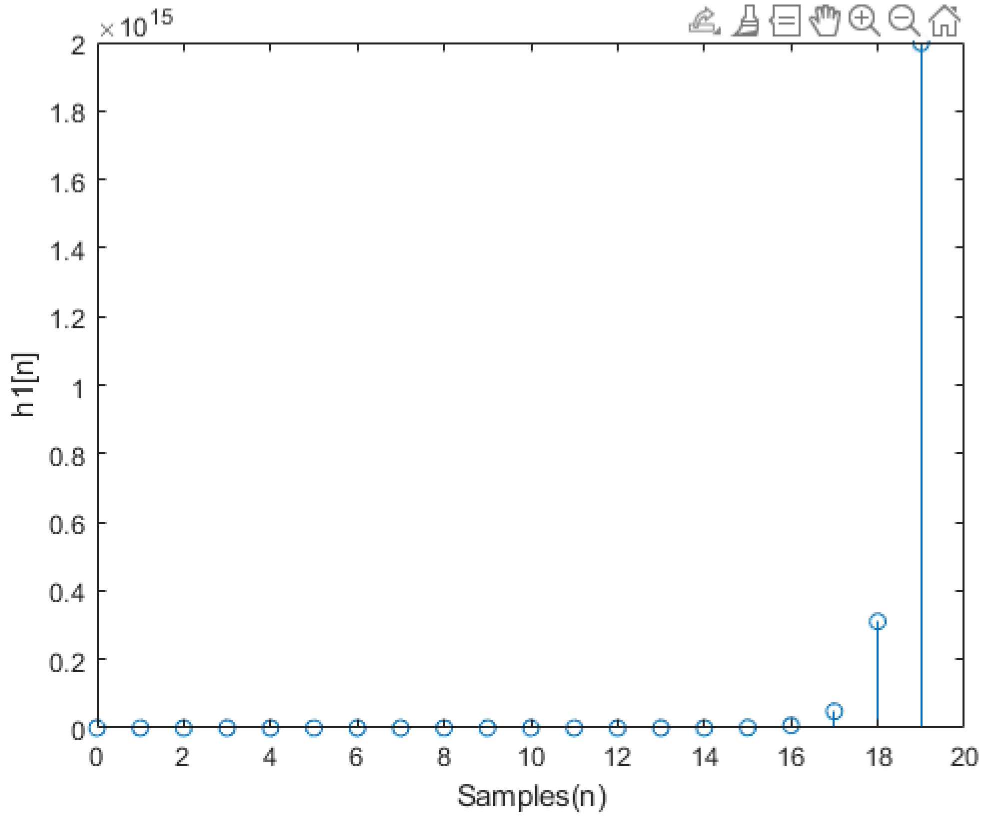

```
% Section: 28944
% Group members: Connor Dupuis, Rachel Romaine, Corinne Meyers, Sara
 Kinzbruner
% Isabella Perlmutter

clear
close all
clc
```

Part A

```
Beta = 4;
```

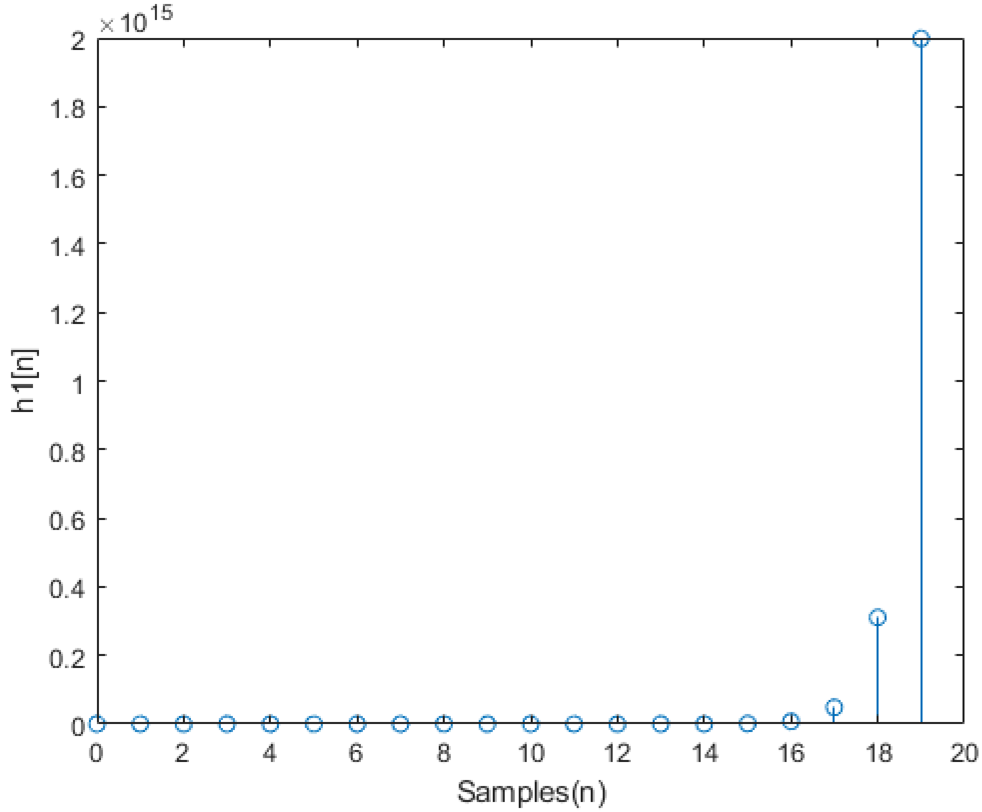
Part B

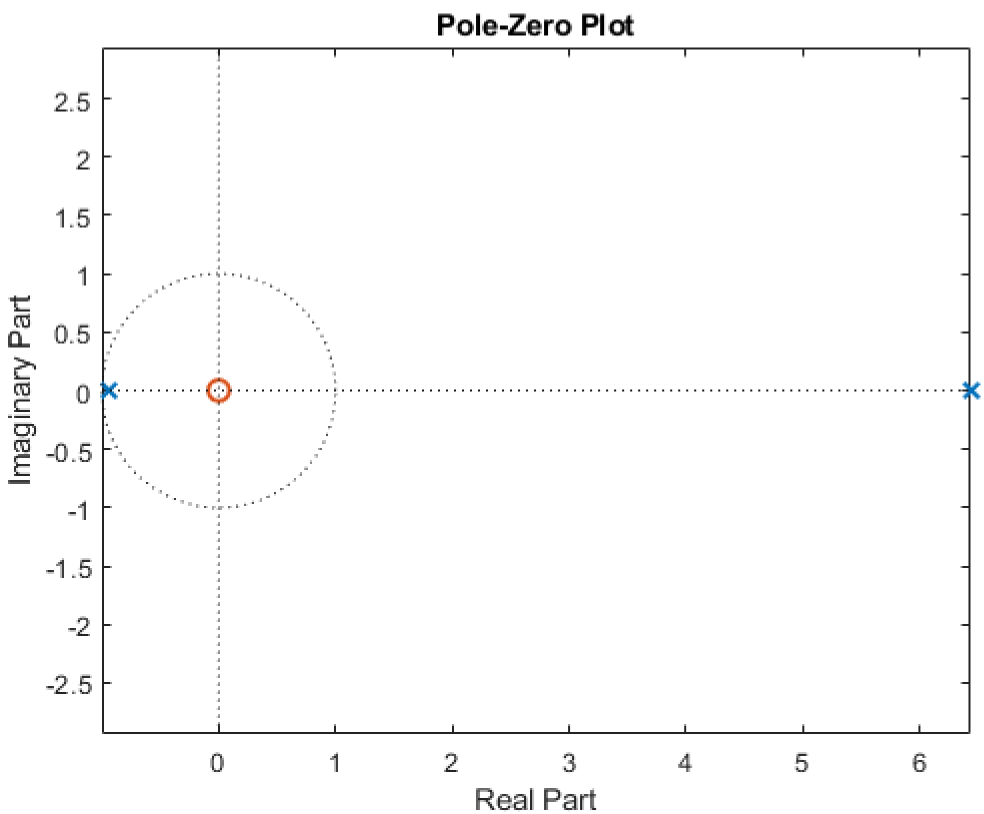
```
N = 20;
n = 0:(N-1);

% we are creating a delta function

x = zeros(N,1);
x(1) = 1;

h1 = group_system(x, Beta);

figure
stem(n, h1);
ylabel('h1[n]');
xlabel('Samples(n)');

% the system is unstable because as n approaches infinity, the impulse
% response approaches infinity
```

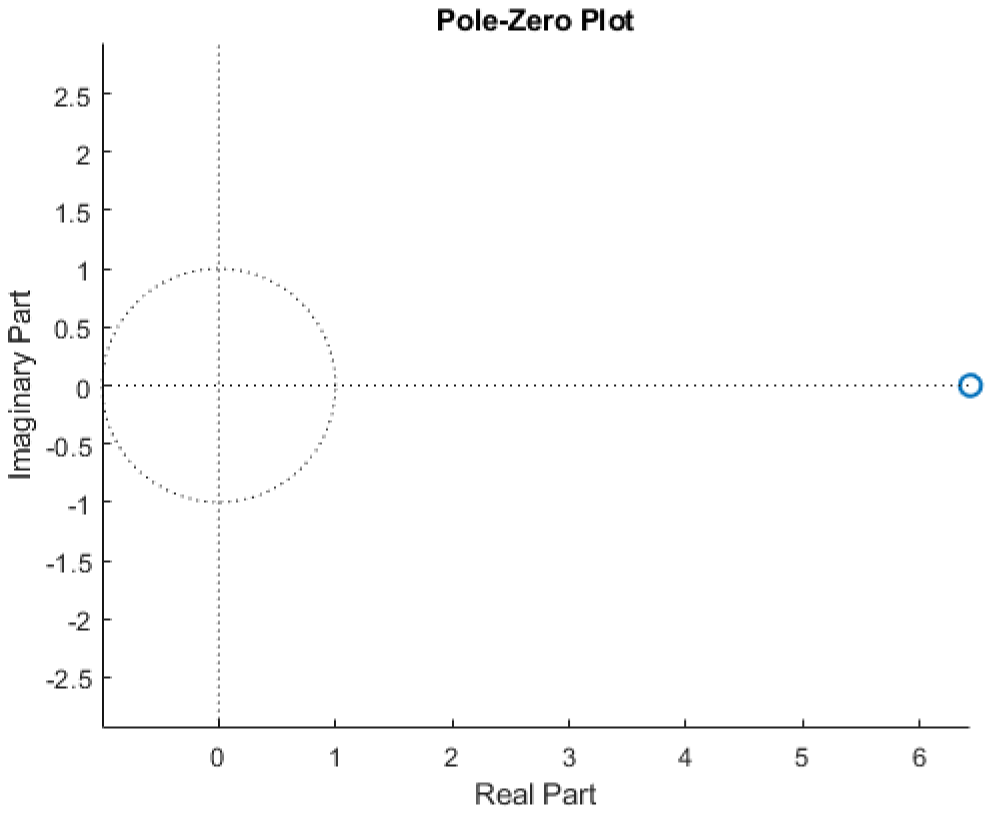


Part C

```
%Y(z) = (1.5+Beta)*Y(z)*z^-1+1.5*Beta*Y(z)*z^-2+X(z)
%Y(z) - ((1.5+Beta)*Y(z)*z^-1+1.5*Beta*Y(z)*z^-2) = X(z)
%Y(z) (1 - ((1.5+Beta)*z^-1 + 1.5*Beta*z^-2) = X(z)
%Y(z)/X(z) = H(z) = 1/(1 - ((1.5+Beta)*z^-1 + 1.5*Beta*z^-2))

a1 = [1, -(1.5+Beta), -1.5*Beta];
b1 = 1;

figure
pzplot(b1,a1);

a2 = [0 1];
b2 = [1, -6.433];

figure
pzplot(b2,a2);
```


×10^15
h1[n]
Samples(n)
0
2
4
6
8
10
12
14
16
18
20
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2


Pole-Zero Plot
Imaginary Part
Real Part
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
0
1
2
3
4
5
6



Part D

```
h2 = filter(b2,a2,h1);

figure
stem(1:length(h2), h2);
ylabel('h1[n]');
xlabel('Samples(n)');

% Unable to apply filter because of 0 in a(1) position.
% The transfer function of our system is:
% H(z) = Z - 6.433 (Non-causal)
% H(z) = (1-6.433z^-1)/z^-1 (Delayed to make causal)

% This system has zeros in the numerator that cancels out the unstable
% poles of the original system.

Error using filter
First denominator filter coefficient must be non-zero.

Error in eellab7finale (line 52)
h2 = filter(b2,a2,h1);
```

Part E

This approach requires to us to know the transfer function, therefore not applicable to arbitrary system in the real world where it is difficult to find the transfer function.

```
function pzplot(b,a)
% PZPLOT(B,A)  plots the pole-zero plot for the filter described by
% vectors A and B.  The filter is a "Direct Form II Transposed"
% implementation of the standard difference equation:
%
%    a(1)*y(n) = b(1)*x(n) + b(2)*x(n-1) + ... + b(nb+1)*x(n-nb)
%                          - a(2)*y(n-1) - ... - a(na+1)*y(n-na)
%

    % MODIFY THE POLYNOMIALS TO FIND THE ROOTS
    b1 = zeros(max(length(a),length(b)),1); % Need to add zeros to get
 the right roots
    a1 = zeros(max(length(a),length(b)),1); % Need to add zeros to get
 the right roots
    b1(1:length(b)) = b;    % New a with all values
    a1(1:length(a)) = a;    % New a with all values

    % FIND THE ROOTS OF EACH POLYNOMIAL AND PLOT THE LOCATIONS OF THE
 ROOTS
    h1 = plot(real(roots(a1)), imag(roots(a1)));
    hold on;
    h2 = plot(real(roots(b1)), imag(roots(b1)));
    hold off;

    % DRAW THE UNIT CIRCLE
    circle(0,0,1)

    % MAKE THE POLES AND ZEROS X's AND O's

 set(h1, 'LineStyle', 'none', 'Marker', 'x', 'MarkerFaceColor','none', 'linewidth'
 1.5, 'markersize', 8);

 set(h2, 'LineStyle', 'none', 'Marker', 'o', 'MarkerFaceColor','none', 'linewidth'
 1.5, 'markersize', 8);
    axis equal;

    % DRAW VERTICAL AND HORIZONTAL LINES
    xminmax = xlim();
    yminmax = ylim();
    line([xminmax(1) xminmax(2)],[0 0], 'linestyle', ':', 'linewidth',
 0.5, 'color', [1 1 1]*.1)
    line([0 0],[yminmax(1) yminmax(2)], 'linestyle', ':', 'linewidth',
 0.5, 'color', [1 1 1]*.1)

    % ADD LABELS AND TITLE
    xlabel('Real Part')
    ylabel('Imaginary Part')
    title('Pole-Zero Plot')
```

```
end

function circle(x,y,r)
% CIRCLE(X,Y,R)  draws a circle with horizontal center X, vertical
 center
% Y, and radius R.
%

    % ANGLES TO DRAW
    ang=0:0.01:2*pi;

    % DEFINE LOCATIONS OF CIRCLE
    xp=r*cos(ang);
    yp=r*sin(ang);

    % PLOT CIRCLE
    hold on;
    plot(x+xp,y+yp, ':', 'linewidth', 0.5, 'color', [1 1 1]*.1);
    hold off;

end
```

Published with MATLAB® R2020a