LAB 5: IIR Filters and Arbitrary Waveforms

AUTHOR: Himanshu Sharma **Roll Number**: 1610110149 **Email**: hs583@snu.edu.in **Instructor**: Prof. Vijay K. Chakka

```
% define the main namespace
function a = main()
    recursive_function(500);
    question1(0.9, 0.01, '$$h(n)=0.9^{n} u(n)$$ at initial value equal
 to ');
    question1(0.9, 0.001, '$$h(n)=0.9^{n} u(n)$$ at initial value
 equal to ');
    question2('audio.mp3', 2);
    question1(-0.9, 0.01, '$$h(n)=(-0.9)^{n} u(n)$$ at initial value
 equal to ');
    question1(-0.9, 0.001, '$$h(n)=(-0.9)^{n} u(n)$$ at initial value
 equal to ');
    two_causal_seq(-0.9, 0.9, 0.02, '$$h(n)=(-0.9)^{n} u(n) + 0.9^{n}
 u(n)$$ at initial value equal to ');
    two_causal_seq(-0.9, 0.9, 0.002, '$$h(n)=(-0.9)^{n} u(n) + 0.9^{n}
 u(n)$$ at initial value equal to ');
    two_causal_seq(0.5, 0.9, 0.02, '$$h(n)=0.5^{n} u(n) + 0.9^{n}
 u(n)$$ at initial value equal to ');
    two_causal_seq(0.5, 0.9, 0.002, '$$h(n)=0.5^{n} u(n) + 0.9^{n}
 u(n)$$ at initial value equal to ');
    fifth_a();
    fifth_b();
    fifth_c();

    % to use the recursive formula we create a general function for
 that.
    function shift = recursor(x)
        % take the first sample as 0.
        shift = 0;
        for i = 0:length(x)-1
           shift = shift + x(i+1)*dirac_delta(i);
        end
    end

    % define delta function
    function y = dirac_delta(n)
        if n == 0
            y = 1;
        else
            y = 0;
        end
    end

    % define the recursive equation as a function with base condition
    % as for a causal system.
    function Y = recursive_function(iterations)
        % define an arbitrary random sequence R.
```

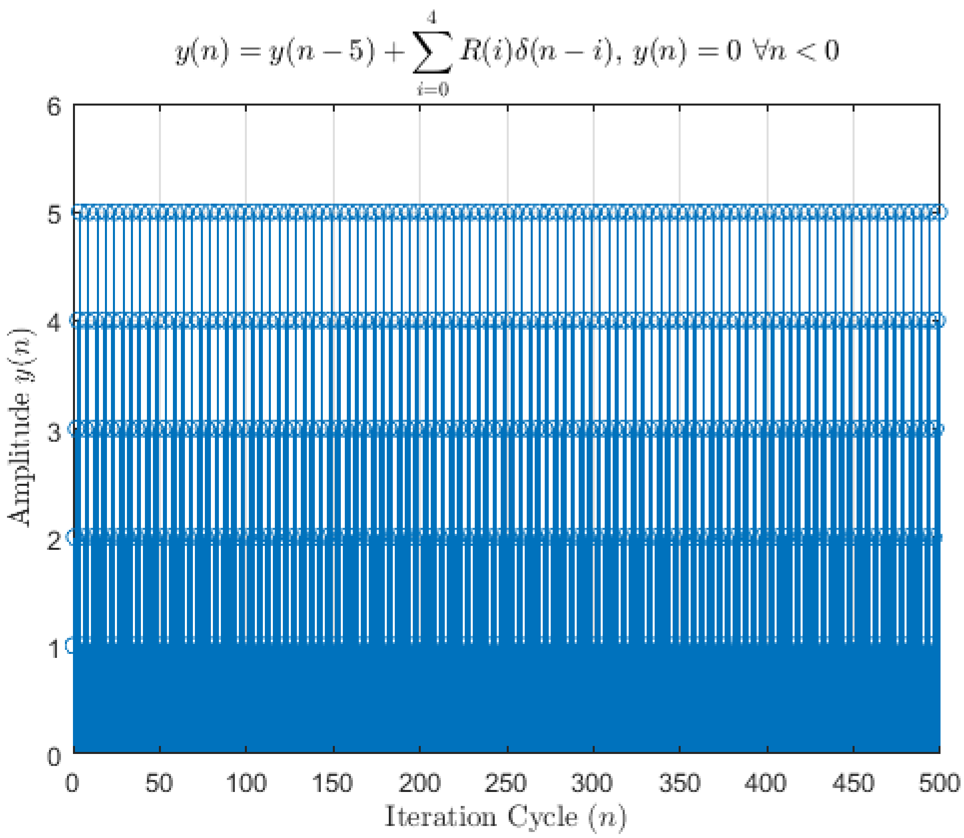
```
        R = [1,2,3,4,5];

        % define empty Y which will store result after each cycle.
        Y = [];
        % define x axis.
        x = [];

        % iterate over iterations.
        for iter = 0:iterations-1
            % if the index of 'y' is less than 0.
            shift = R(1)*dirac_delta(iter) + R(2)*dirac_delta(iter-1)
 + R(3)*dirac_delta(iter-2) + R(4)*dirac_delta(iter-3) +
 R(5)*dirac_delta(iter-4);
            if iter - 5 < 0
                y = shift;
            else
                % iter - 5 + 1 because in matlab indexing starts from
 1.
                y = Y((iter-5)+(1)) + shift;
            end
            % store the output in Y.
            Y = [Y y];
            x = [x iter];
        end

        % plot Y.
        figure;
        stem(x, Y);
        ylim([0, 6]);
        %xlim([-5, 105]);
        grid on;
        xlabel('Iteration Cycle ($$n$$)', 'interpreter', 'latex');
        ylabel('Amplitude $$y(n)$$', 'interpreter', 'latex');
        title('$$y(n) = y(n-5) + \sum_{i=0}^{4}R(i)\delta(n-i)$$, $
$y(n) = 0$$ $$\forall n<0$$', 'interpreter', 'latex');
    end

    % implement the unit step function
    function y = unit_step(n)
       if n >= 0
           y = 1;
       else
           y = 0;
       end
    end

    function question1(x, i_val, Title)
       % x = a for h(n) = a^n u(n)
       % calculate the initial value
       % let it be i_val. Hence, we equate, i_val = 0.9^n.
       % this gives n = 44.7 = n. Take round off value of n.
       uplimit_n = round(log(i_val)/log(abs(x)));
       n = 0:uplimit_n-1;
```

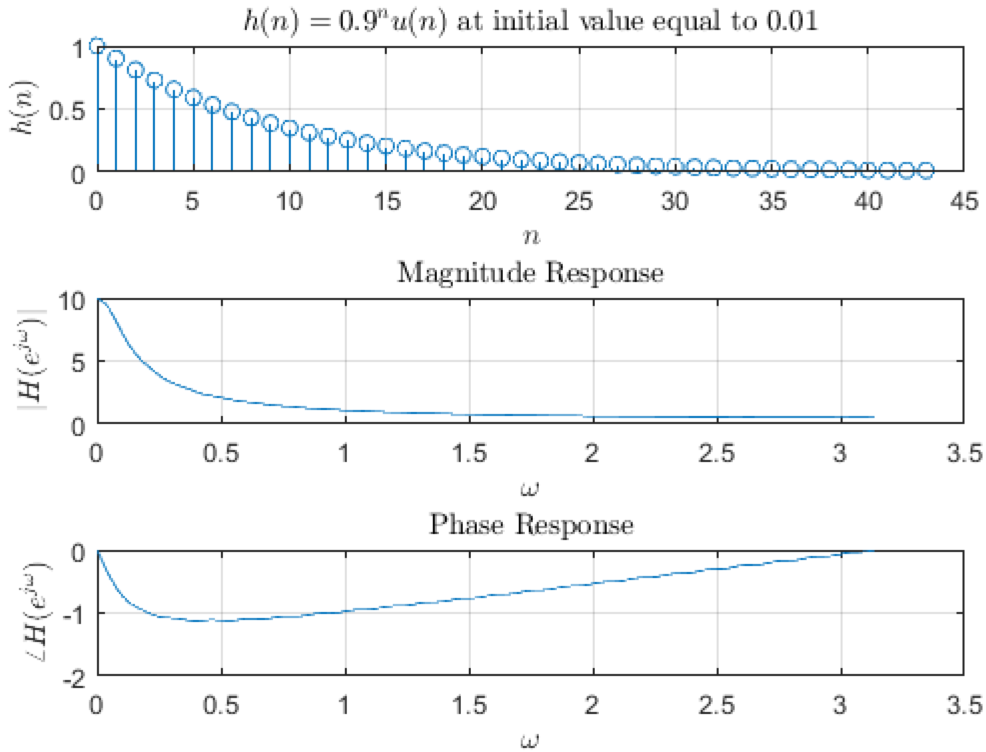
```
    % now implement h(n).
    h = (x.^n);

    % take the magnitude response here.
    [H, omega] = freqz(h);
    angle_array = atan2(imag(H), real(H));

    % now stem h.
    figure;
    subplot(3,1,1);
    stem(n, h);
    xlabel('$$n$$', 'interpreter', 'latex');
    ylabel('$$h(n)$$', 'interpreter', 'latex');
    title([Title, num2str(i_val)], 'interpreter', 'latex');
    grid on;
    subplot(3,1,2);
    plot(omega, abs(H));
    xlabel('$$\omega$$', 'interpreter', 'latex');
    ylabel('$$|H(e^{j \omega})|$$', 'interpreter', 'latex');
    title('Magnitude Response', 'interpreter', 'latex');
    grid on;
    subplot(3,1,3);
    plot(omega, angle_array);
    xlabel('$$\omega$$', 'interpreter', 'latex');
    ylabel('$$\angle H(e^{j \omega})$$', 'interpreter', 'latex');
    title('Phase Response', 'interpreter', 'latex');
    grid on;
end

% define a function for two terms as in b and c parts.
function two_causal_seq(x1, x2, i_val, Title)
    % x1 has the dominant ROC.
    uplimit_n = round(log(i_val)/log(abs(x1)));
    n = 0:uplimit_n-1;

    % implement h(n).
    h = x1.^n + x2.^n;

    % calculate the frequency response.
    [H, omega] = freqz(h);
    angle_array = atan2(imag(H), real(H));

    % now stem h.
    figure;
    subplot(3,1,1);
    stem(n, h);
    xlabel('$$n$$', 'interpreter', 'latex');
    ylabel('$$h(n)$$', 'interpreter', 'latex');
    title([Title, num2str(i_val)], 'interpreter', 'latex');
    grid on;
    subplot(3,1,2);
    plot(omega, abs(H));
    xlabel('$$\omega$$', 'interpreter', 'latex');
    ylabel('$$|H(e^{j \omega})|$$', 'interpreter', 'latex');
```

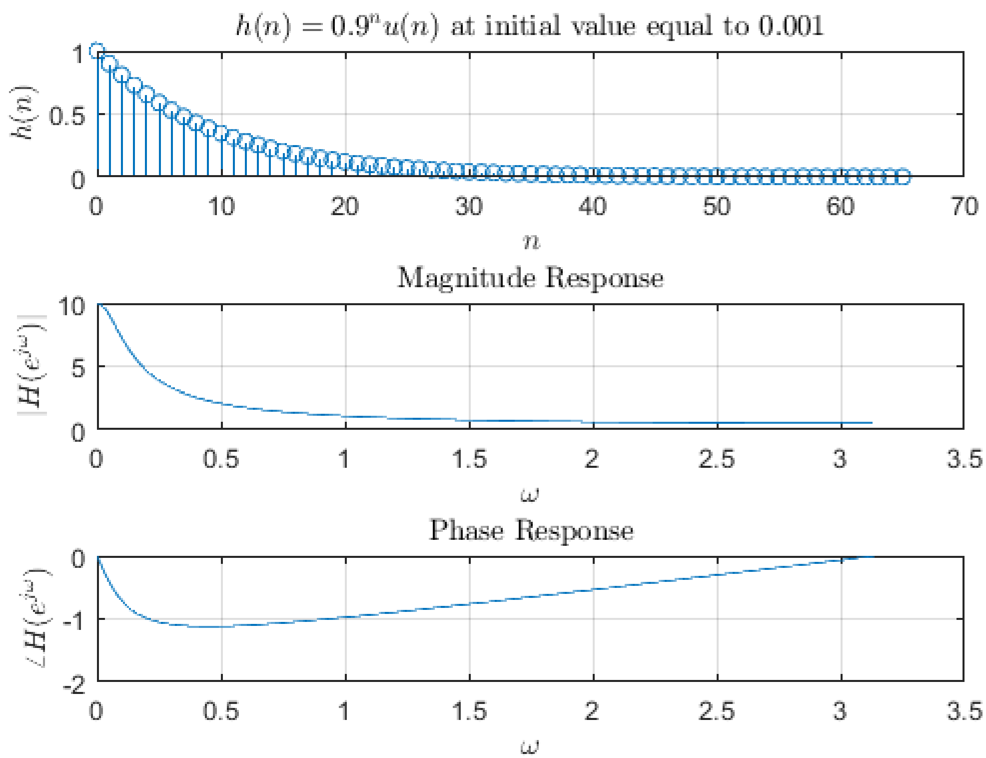
```
        title('Magnitude Response', 'interpreter', 'latex');
        grid on;
        subplot(3,1,3);
        plot(omega, angle_array);
        xlabel('$$\omega$$', 'interpreter', 'latex');
        ylabel('$$\angle H(e^{j \omega})$$', 'interpreter', 'latex');
        title('Phase Response', 'interpreter', 'latex');
        grid on;
    end

    % define a function for question 2.
    function Y = question2(audio, time)
         Y = [];
         % read the audio
        [y, fs] = audioread(audio);

        % extract 'time' seconds of audio from the y.
        v = y(1 : time*fs + 1);

        % define time
        t = 0:1/fs:(1/fs)*(length(v)-1);
        %disp(length(t));

        for i = 0:length(v)-1
            if i - 1 < 0
                % implement equation when n < 0.
                % i + 1 because indexing starts from 1 in MATLAB.
                block = v(i+1);
            else
                % implementation of recursive equation.
                % i - 1 + 1 because of MATLAB indexing.
                block = 0.9*Y(i-1+1) + v(i+1);
            end
            % store the result in Y.
            Y = [Y block];
        end

        figure;
        subplot(2,1,1);
        plot(t, v);
        title('Audio sample $$x(n)$$ of 2
 seconds', 'interpreter', 'latex');
        xlabel('time in seconds');
        ylabel('$$x(n)$$', 'interpreter', 'latex');
        grid on;
        subplot(2,1,2);
        plot(t, Y);
        title('$$y(n)=0.9y(n-1)+x(n)$$ and $$H(z)=\frac{1}
{1-0.9z^{-1}}$$ with $$|z|>0.9$$', 'interpreter', 'latex');
        xlabel('time in seconds');
        ylabel('$$y(n)$$', 'interpreter', 'latex');
        grid on;
    end
   % function to plot 5 a.
```

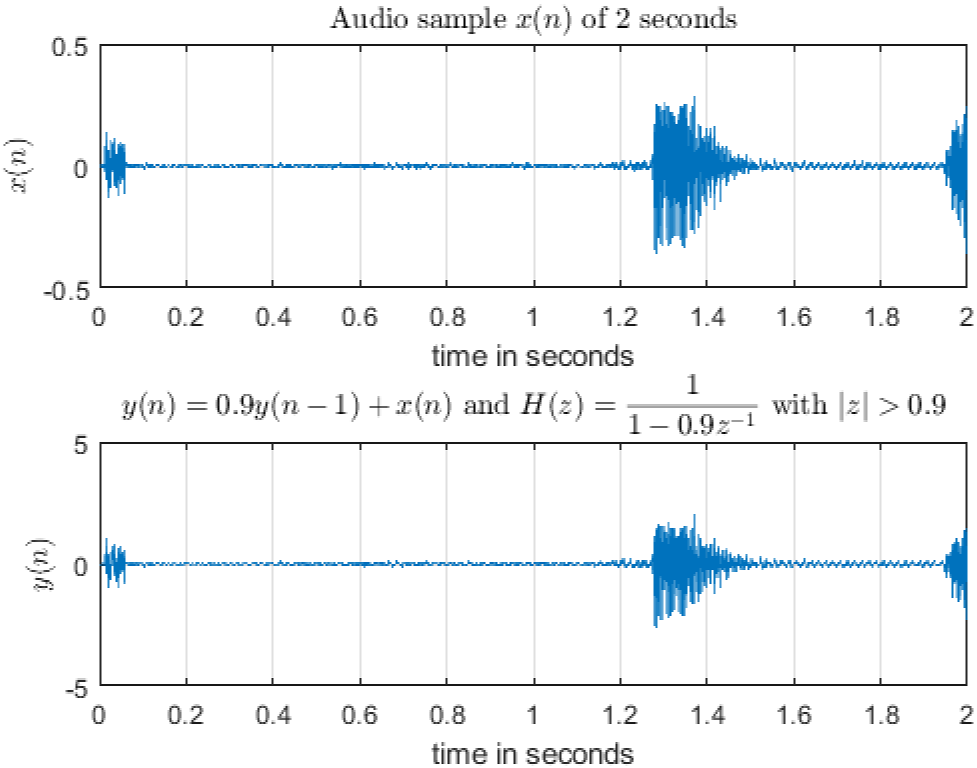
```
function fifth_a()
    % define the omega axis.
    omega = 0:0.001:pi;

    % define the complex Fourier transform
    H = 1./(1-(0.9.*exp(-j*omega)));
    ang = atan2(imag(H), real(H));

    % plot the response now.
    figure;
    subplot(1,2,1)
    plot(omega, abs(H));
    title('$$|H(e^{j \omega})| = |\frac{1}{1-0.9e^{-j\omega}}|$
$', 'interpreter', 'latex');
    xlabel('$$\omega$$', 'interpreter', 'latex');
    grid on;
    subplot(1,2,2)
    plot(omega, ang);
    title('$$\angle H(e^{j \omega})$$', 'interpreter', 'latex');
    xlabel('$$\omega$$', 'interpreter', 'latex');
    grid on;
end

% function to plot 5 b.
 function fifth_b()
    % define omega.
    omega = 0:0.001:pi;

    % define H.
    H = (0.1.*exp(-j*omega))./(1-0.1.*exp(-j*omega)).^2;
    ang = atan2(imag(H), real(H));

    % plot the response now.
    figure;
    subplot(1,2,1)
    plot(omega, abs(H));
    title('$$|H(e^{j \omega})| = |\frac{0.1e^{-j\omega}}{(1-0.1e^{-
j\omega})^{2}}|$$', 'interpreter', 'latex');
    xlabel('$$\omega$$', 'interpreter', 'latex');
    grid on;
    subplot(1,2,2)
    plot(omega, ang);
    title('$$\angle H(e^{j \omega})$$', 'interpreter', 'latex');
    xlabel('$$\omega$$', 'interpreter', 'latex');
    grid on;
 end

 % function for 5 c.
 function fifth_c()
    % define omega.
    omega = 0:0.001:pi;

    % define H.
    H = (1./(1-0.5.*exp(-j*omega))) + (1./(1-2.*exp(-j*omega)));
```

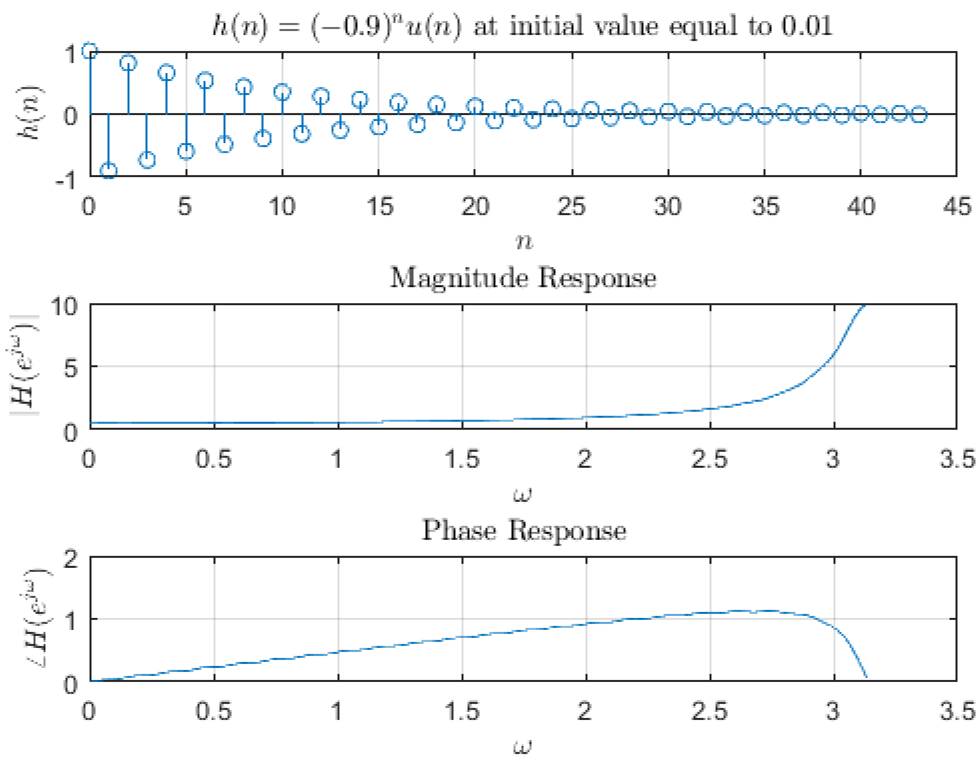
```
        ang = atan2(imag(H), real(H));

        % plot the response now.
        figure;
        subplot(1,2,1)
        plot(omega, abs(H));
        title('$$|H(e^{j \omega})| = |\frac{1}{(1-5e^{-j\omega})} +
  \frac{1}{(1-2e^{-j\omega})}|$$', 'interpreter', 'latex');
        xlabel('$$\omega$$', 'interpreter', 'latex');
        grid on;
        subplot(1,2,2)
        plot(omega, ang);
        title('$$\angle H(e^{j \omega})$$', 'interpreter', 'latex');
        xlabel('$$\omega$$', 'interpreter', 'latex');
        grid on;
    end
end
```

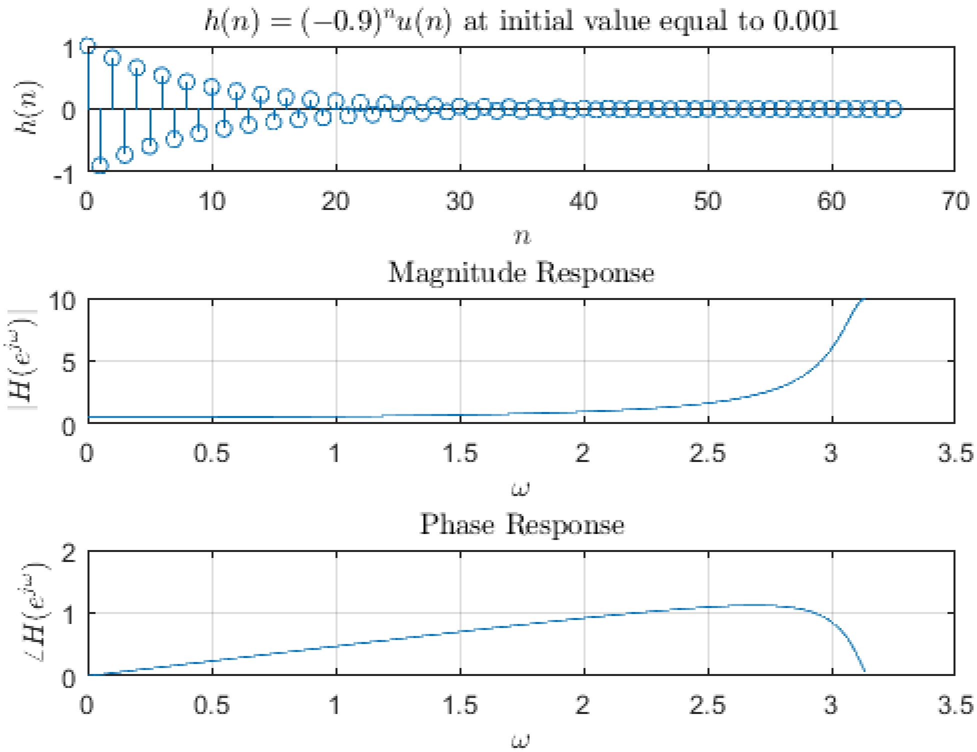


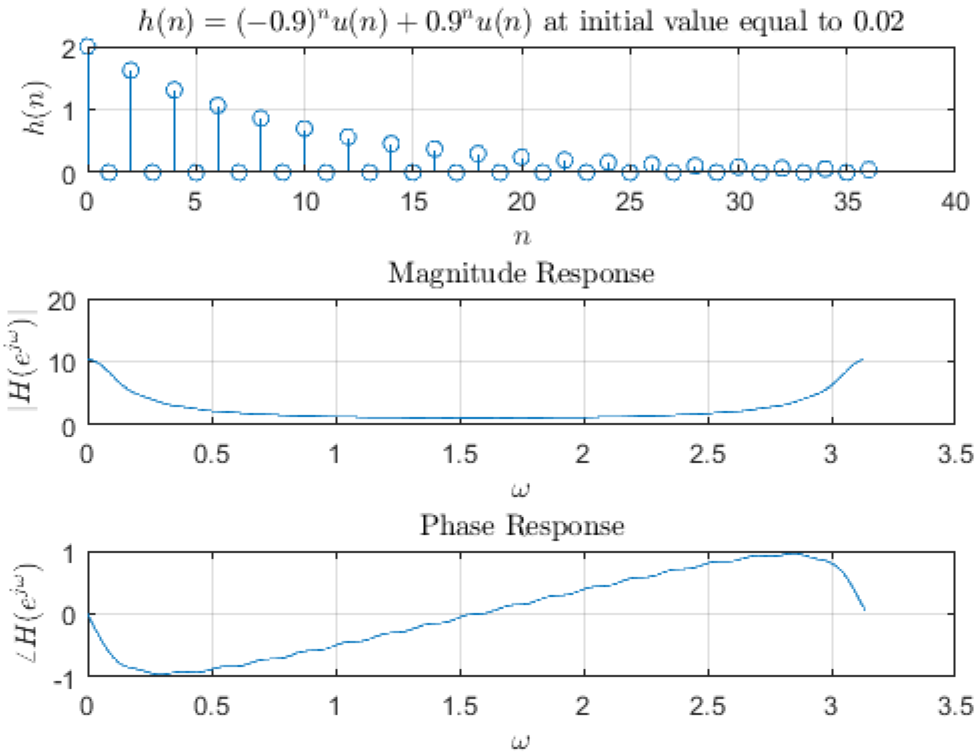

$h(n) = 0.9^n u(n)$ at initial value equal to 0.01
$h(n)$
n
Magnitude Response
$|H(e^{j\omega})|$
ω
Phase Response
$\angle H(e^{j\omega})$
ω

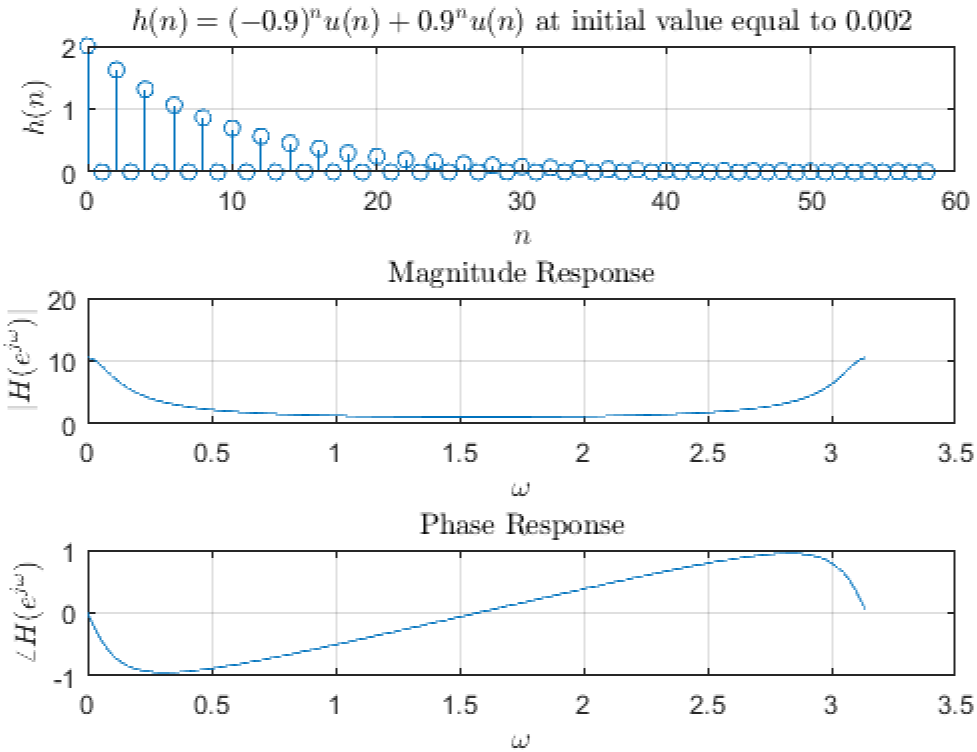

$h(n) = 0.9^n u(n)$ at initial value equal to 0.001
$h(n)$
n
Magnitude Response
$|H(e^{j\omega})|$
ω
Phase Response
$\angle H(e^{j\omega})$
ω

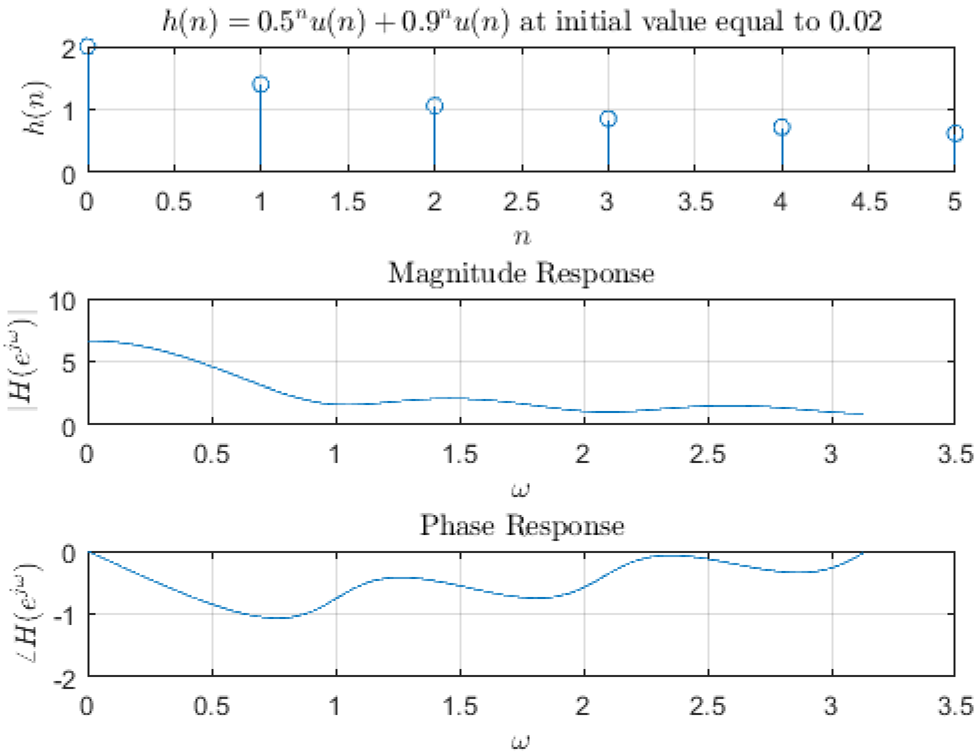

Audio sample $x(n)$ of 2 seconds
$x(n)$
0.5
0
-0.5
0 0.2 0.4 0.6 0.8 1 1.2 1.4 1.6 1.8 2
time in seconds
$y(n) = 0.9y(n-1) + x(n)$ and $H(z) = \frac{1}{1-0.9z^{-1}}$ with $|z| > 0.9$
$y(n)$
5
0
-5
0 0.2 0.4 0.6 0.8 1 1.2 1.4 1.6 1.8 2
time in seconds

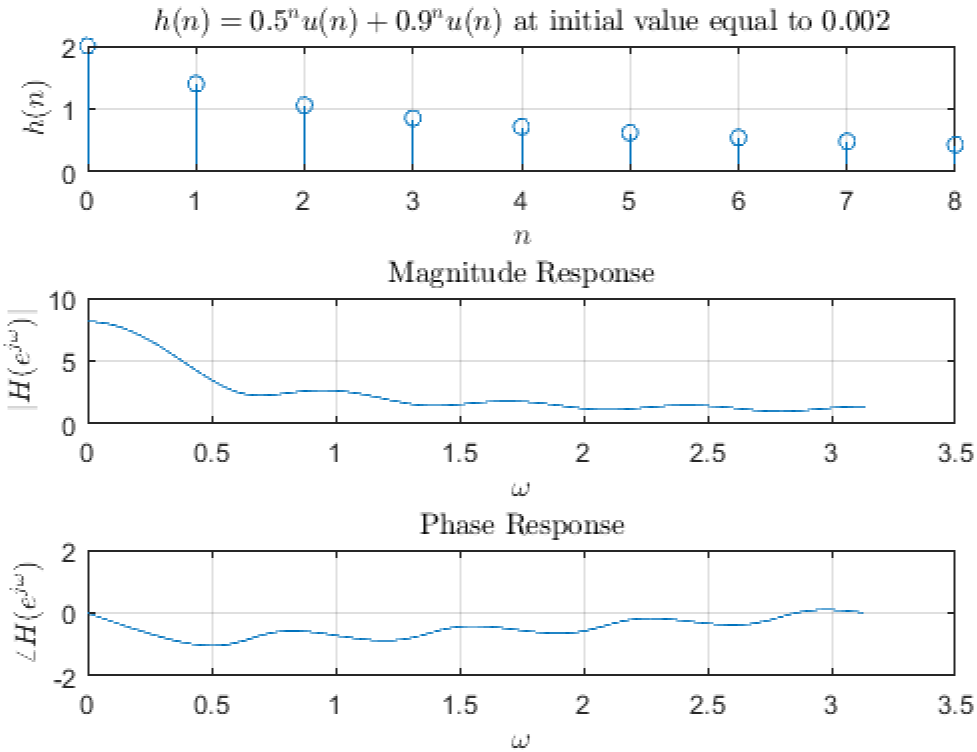

$h(n) = (-0.9)^n u(n)$ at initial value equal to 0.01
$h(n)$
1
0
-1
0 5 10 15 20 25 30 35 40 45
n
Magnitude Response
$|H(e^{j\omega})|$
10
5
0
0 0.5 1 1.5 2 2.5 3 3.5
ω
Phase Response
$\angle H(e^{j\omega})$
2
1
0
0 0.5 1 1.5 2 2.5 3 3.5
ω

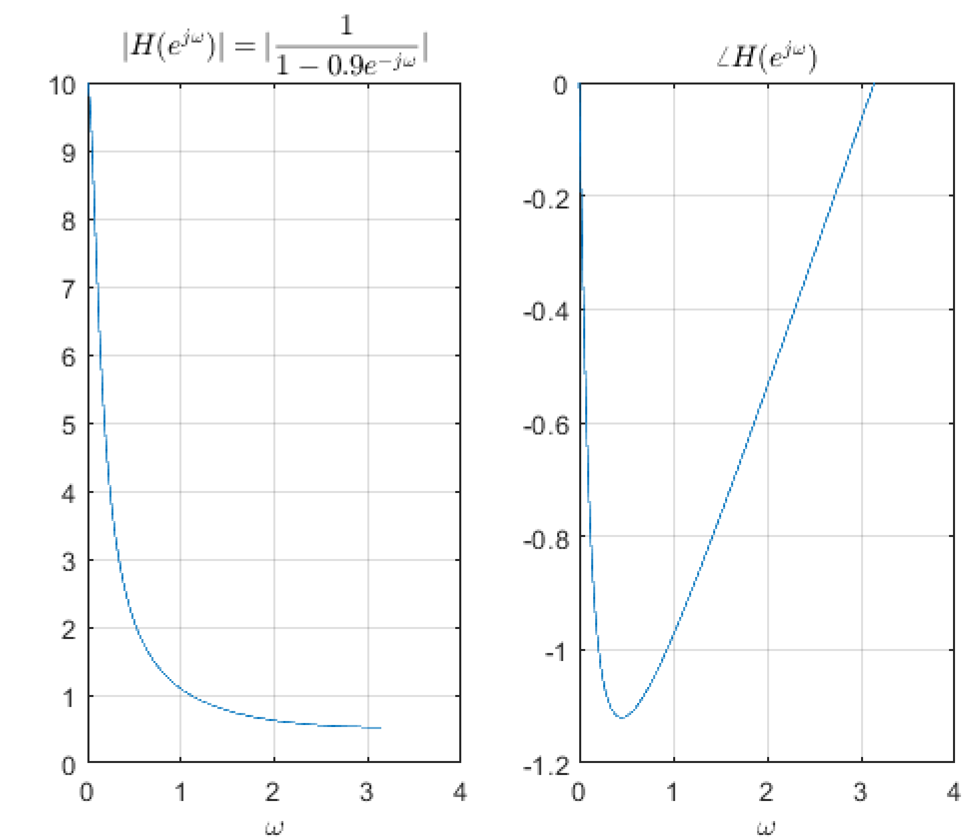

$h(n) = (-0.9)^n u(n)$ at initial value equal to 0.001
$h(n)$
n
Magnitude Response
$|H(e^{j\omega})|$
ω
Phase Response
$\angle H(e^{j\omega})$
ω

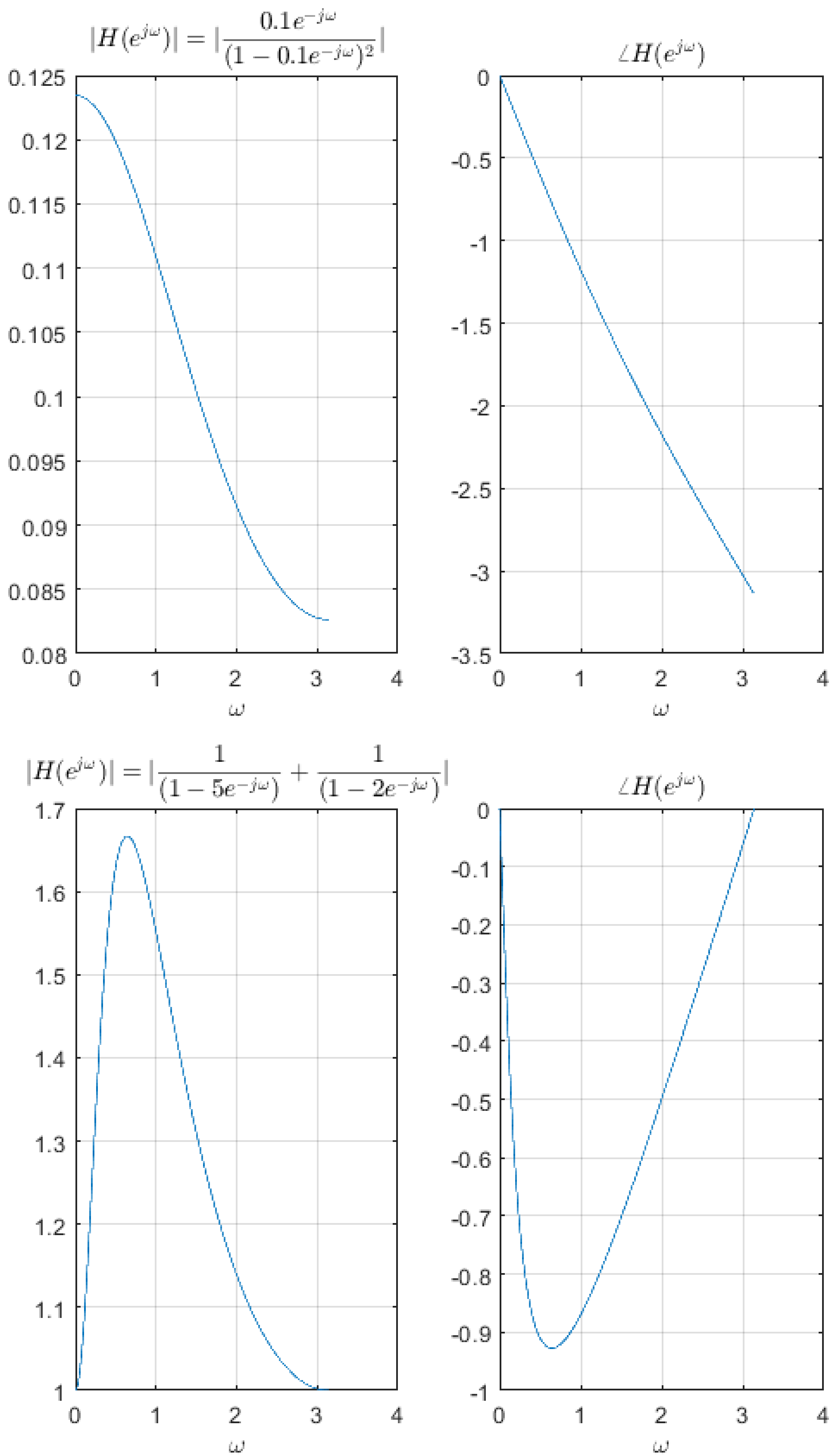

$h(n) = (-0.9)^n u(n) + 0.9^n u(n)$ at initial value equal to 0.02
$h(n)$
n
Magnitude Response
$|H(e^{j\omega})|$
ω
Phase Response
$\angle H(e^{j\omega})$
ω


$h(n) = (-0.9)^n u(n) + 0.9^n u(n)$ at initial value equal to 0.002
$h(n)$
n
Magnitude Response
$|H(e^{j\omega})|$
ω
Phase Response
$\angle H(e^{j\omega})$
ω


$h(n) = 0.5^n u(n) + 0.9^n u(n)$ at initial value equal to 0.02
$h(n)$
n
Magnitude Response
$|H(e^{j\omega})|$
ω
Phase Response
$\angle H(e^{j\omega})$
ω


$h(n) = 0.5^n u(n) + 0.9^n u(n)$ at initial value equal to 0.002
$h(n)$
n
Magnitude Response
$|H(e^{j\omega})|$
ω
Phase Response
$\angle H(e^{j\omega})$
ω


$|H(e^{j\omega})| = |\frac{1}{1-0.9e^{-j\omega}}|$
$\angle H(e^{j\omega})$
ω
ω



$|H(e^{j\omega})| = |\frac{0.1e^{-j\omega}}{(1-0.1e^{-j\omega})^2}|$
$\angle H(e^{j\omega})$
$|H(e^{j\omega})| = |\frac{1}{(1-5e^{-j\omega})} + \frac{1}{(1-2e^{-j\omega})}|$
$\angle H(e^{j\omega})$
ω

CONCLUSION ABOUT AUDIO

```
%
% It is clear from the frequency response of 0.9u(n) that it acts like
 a high
% pass filter. So when a recursive equation is made out of it and an
 audio of
% length 2 sec is applied to this recursive equation, we get those
 parts of
% the audio which have high pitch. This is because, a high pass filter
 allows
% to pass high pitch (frequency) signal through it and supresses the
 low
% pitch parts of it. So when we play this sound, we get some 'tick'
 like
% sounds which corresponds to high pitch parts of the same 2 sec.
 audio.
% The same is clear from the plots as well.
```

Published with MATLAB® R2017b