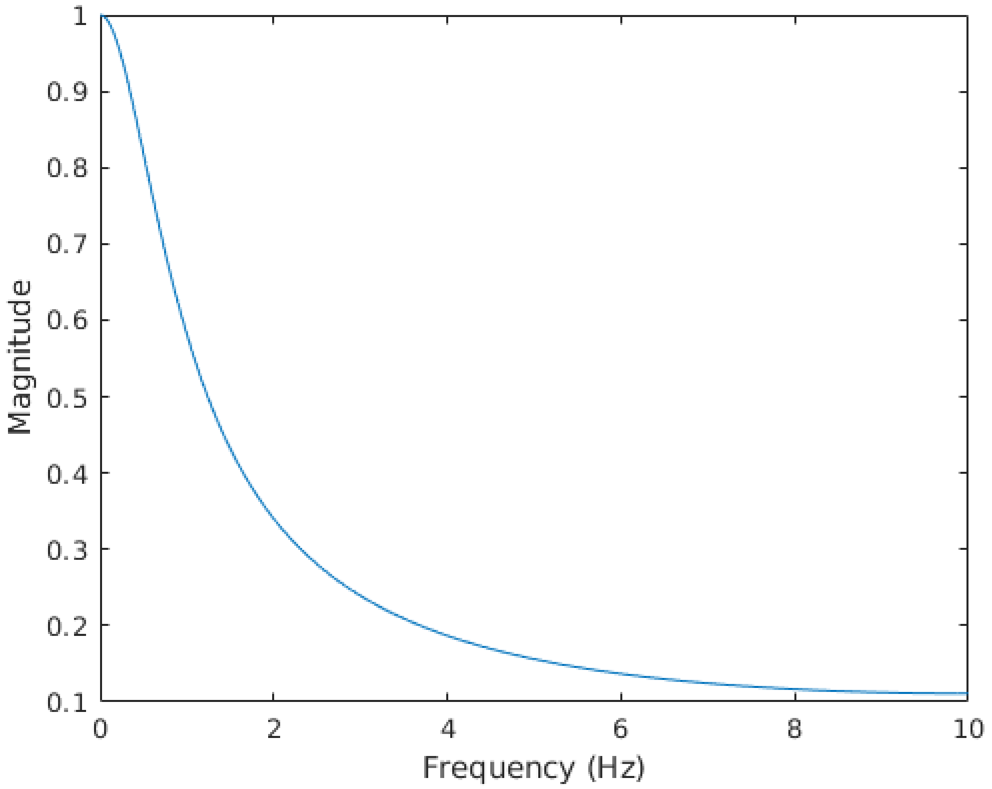
The original difference equation: y[n+1] = 0.8 y[n] + 0.2 x[n], a noncausal system, prompts to prove its BIBO stability, compute steady state and full system response, and plot the magnitude response as a function of frequency up to 10 hz given a 2hz sinusoidal input. The MATLAB source code at the end of the file is able to compute the necessary steps to solve the given problems, although the output may be ugly at times due to lack of proper formula formatting. To properly compute all values, an initial parameter must be set for the noncausal term, in this case y(n+1) represented as y(0) in the transform, which can be changed to modify the behavior of the system.

Image: Magnitude response as a function of frequency for initial parameter y(0) = 0 up to 10 hz



The system represents a low-pass filter by its frequency response for sinusoidal input.

MATLAB Source Code to compute result for all parts below, symbolically:

```
%symbolic solution

syms x(n) y(n) z

assume(n >= 0 & in(n, 'integer'));

%original equation: y[n+1] = 0.8y[n] + 0.2x[n]

diff_eq = y(n+1) - 0.8*y(n) - 0.2*x(n);

big_f = ztrans(diff_eq, n, z);
```

```
%by assignment spec: input is 2hz sinusoidal with sampling
period 0.05 seconds

big_x = ztrans(sin(2*pi*2*0.05*n));

big_f = subs(big_f, ztrans(x(n),n,z), big_x);

syms big_y

big_f = subs(big_f, ztrans(y(n),n,z), big_y);%since system is
noncausal, an initial condition

big_y = solve(big_f, big_y)%will be present symbolically as y(0)

big_h = big_y / big_x

[num_big_h, den_big_h] = numden(big_h);

%part A: is the system BIBO stable?

%if all roots of denominator are z < 1 (within unit circle),
then it's stable

abs(solve(den_big_h == 0))

%verify with matlab's stability function

isstable(big_h)


%part B: full system response (steady state will be further
down)


y_sol = iztrans(big_y); %exact solution

y_sol = vpa(y_sol) %numerical approximation

children(y_sol) %decomposition by terms


%verify with matlab's systems toolkit

verif_sys = filt([0.2], [1, -0.8], 0.05) %transfer (big_h)
function



%part C: magnitude plot
```

```
syms f

big_h_freq = subs(big_h, z, exp(j*2*pi*f*0.05))

big_h_freq = subs(big_h_freq, y(0), 0) %initial value y(0) == 0

%big_h_omga = simplify(rewrite(big_h_omga, 'sincos'));
%optional, looks

%nicer if printing out

freq_range = linspace(0,10,1024);

mag_resp = abs(vpa(subs(big_h_freq, f, freq_range)));

plot(freq_range, mag_resp);

xlabel("Frequency (Hz)");

ylabel("Magnitude");

%verify using matlab toolkit

figure;

bode(verif_sys, {1,20*pi}) %log of frequency

freqresp(verif_sys, 10)
```