Lab 2 Report

Discrete System Analysis

Alex Hirzel

Submitted to Yang Liu for EE4252

Due October 4, 2012

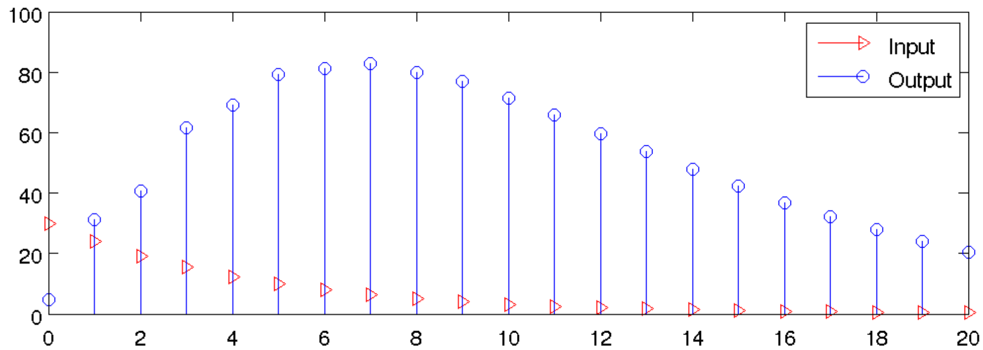
Michigan Technological University

Project 1

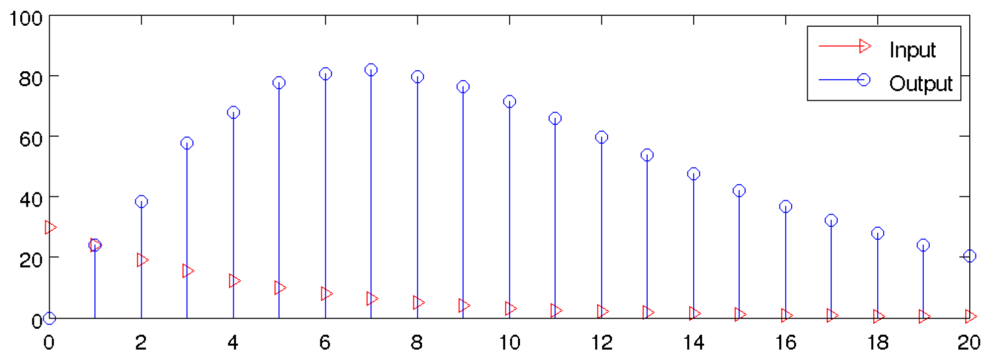
Considering the system

$$y[n] - 0.49y[n-2] = x[n] \tag{1}$$

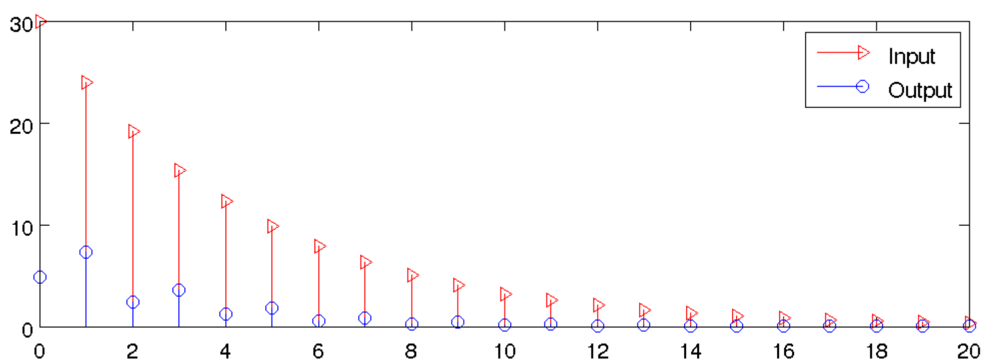
with $x[n] = 30(0.8)^n u[n]$, $y[-1] = 15$ and $y[-2] = 10$,



(a) Total response.



(b) Zero-input response.



(c) Zero-state response.

Figure 1: The zero-input and zero-state responses for system (1).

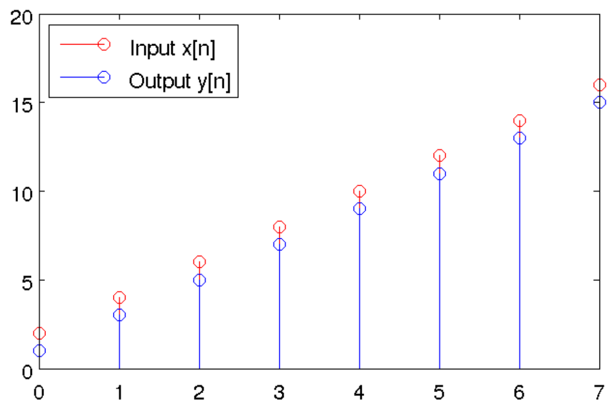


Figure 2: System (2) applied to the data set $\{\overset{\Downarrow}{2}, 4, 6, 8, 10, 12, 14, 16\}$ with output shown on the same plot.

Project 2

In this case, the two-point averaging filter is

$$y[n] = 0.5x[n] + 0.5x[n-1] \tag{2}$$

or $h[n] = \{\overset{\Downarrow}{0.5}, 0.5\}$, which agrees with the requested MATLAB output in Listing 2. As an example, the effect of system (2) is shown in Figure 2. In general, an N-point averaging filter has an impulse response of

$$\boxed{N\text{-point averaging filter:}\ \underbrace{\left\{\overset{\Downarrow}{\frac{1}{N}}, \frac{1}{N}, \cdots, \frac{1}{N}, \frac{1}{N}\right\}}_{N \text{ terms in length}} \quad \text{or} \quad h[n] = \frac{1}{N}\sum_{i=0}^{N-1} \delta[n-i]}.$$

For example, with $N = 4$ the filter will be a 4-point averaging filter and can be expressed as

$$\boxed{4\text{-point averaging filter:}\ \left\{\overset{\Downarrow}{\tfrac{1}{4}}, \tfrac{1}{4}, \tfrac{1}{4}, \tfrac{1}{4}\right\} \quad \text{or} \quad h[n] = \tfrac{1}{4}\delta[n] + \tfrac{1}{4}\delta[n-1] + \tfrac{1}{4}\delta[n-2] + \tfrac{1}{4}\delta[n-3]}.$$

When an averaging filter is used, the output signal is affected in two ways:

1. **Time shift** - the filter adds a $t_s(N-1)/2$ time lag due to the causal nature of the filter. Specifically, this is because the filter cannot make use of samples that occur in the future with respect to a given position. Because of this, any given point in the output is the result of the algorithm using the $N-1$ previous samples and the "current" sample (for N samples in total). This range of samples is centered $(N-1)/2$ samples in the past, making the output appear $(N-1)/2$ samples "too late".

2. **Attenuation** - the output of the filter can be reduced by the averaging process. The attenuation factor at a given time T in the original signal can be approximated (excluding discretation effects) by

$$\mathrm{att}_T(T) \approx \frac{1}{x(T)} \int_{T-t_s(N-1)/2}^{T+t_s(N-1)/2} x(t)\,\mathrm{d}t, \tag{3}$$

   which is the mean signal value over the averaging region of the filter. This can be put in terms of n and the original signal as follows:[1]

$$\begin{aligned} \mathrm{att}_n(n) &= \mathrm{att}_T\left(nt_s - \overbrace{t_s[N-1]/2)}^{\text{time shift}}\right. \\ &= \mathrm{att}_T\left(-t_s[N-2n-1]/2\right) \approx \frac{\int_{-t_s(N-n-1)}^{T+nt_s} x(t)\,\mathrm{d}t}{x(nt_s - t_s[N-1]/2)}. \end{aligned}$$

[1] This is which is a differential equation—still in terms of $x(t)$!—that I don't want to expand any further. That said, one could now write $x(t)$ in terms of both $x[n]$ and $\mathrm{att}_n(n)$, convert to a discrete-time summation instead of an integration and arrive at an enormously practical result. In practice, the original signal is often the desired output from a system with this attenuation. This so-called "reverse averaging" problem is difficult. After all, it boils down to deconvolution! I think this is the right start, though.

Averaging a noisy sinusoid

The requested plots are shown and described in Figure 4 and Figure 3. The attenuation is adjusted by using a factor of 0.75732, which would change the `b` coefficient to $1/200/0.75732 = 0.0066$ instead of 0.005.

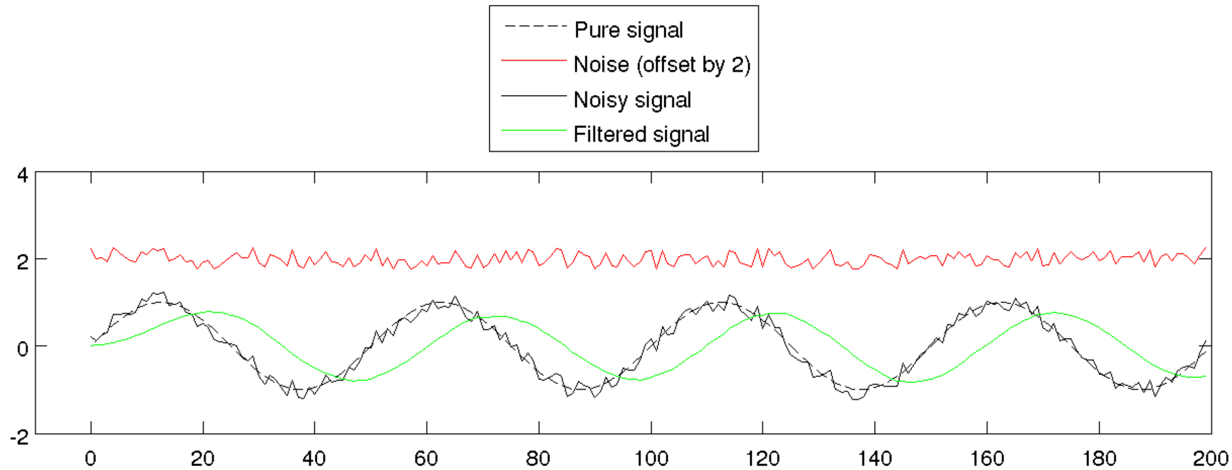


Figure 3: Original sinusoid (in dashed black), the original sinusoid with noise added (in black) and the output of the 20-point averaging filter (in green) with noise shown isolated in red.

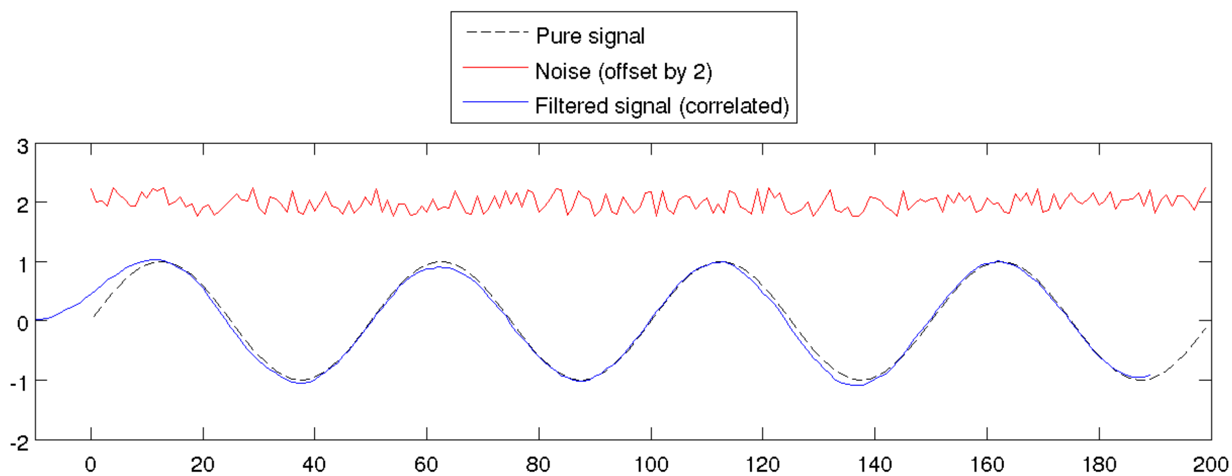


Figure 4: Original signal (in dashed black) shown with colocated filter output (in blue). The filter output is attenuated from the pure original sinusoid by a factor of 0.75732 and offset horizontally by a shift of $10t_s$. The isolated noise from Figure 3 is also shown (in red) for reference.

Appendix: MATLAB Source Code

What follows is a listing of the MATLAB source code (listing 1)—and the output of this code (listing 2)—used to generate the figures and other information presented in this report.

Listing 1: The MATLAB script used for this report, `Lab02_ahirzel.m`.

```
% EE4252 Lab 2
% Alex Hirzel <ahirzel@mtu.edu>
% 2012-09-27

addpath ../../ClassWorkspace;
delete 'generated/diary.txt'; diary 'generated/diary.txt'; diary on


% Project 1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
n=0:20;

[yt, yzs, yzi] = sysresp1('z', ...
    [1 0 0], [1 0 -0.49],      ... % y[n] - 0.49y[n-2] = x[n]
    [30 0.8 1 0 0 0],          ... % 30 (0.8)^n n^1
    [15 10]);                      % y[-1] = 15, y[-2] = 10

% Total response
stem(n, 30*(0.8).^n, 'r>'); hold on
stem(n, eval(yt),    'o')
legend('Input', 'Output')
mysaveas('p1_total', 8, 2.5)

% Zero-input response
stem(n, 30*(0.8).^n, 'r>'); hold on
stem(n, eval(yzs),   'o')
legend('Input', 'Output')
mysaveas('p1_zir',   8, 2.5)

% Zero-state response
stem(n, 30*(0.8).^n, 'r>'); hold on
stem(n, eval(yzi),   'o')
legend('Input', 'Output')
mysaveas('p1_zsr',   8, 2.5)


% Project 2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
MA = @(n) ones(1,n)/n;

% Impulse reponse of 2-point averaging filter
r = sysresp1('z', MA(2), [1 0]);
disp(['Impulse response of y[n]: ' r])

% Example for 2-point averaging filter
x = [2 4 6 8 10 12 14 16];
y = filter(MA(2), [1 0], x);
stem(0:7, x, 'or'); hold on
stem(0:7, y, 'ob')
legend('Input x[n]', 'Output y[n]', 'Location', 'NorthWest')
mysaveas('p2_averaging_example', 5, 3)

% Sinusoid
n = 0:199;
rand('seed', 324324);
pure     = sin(2*pi*n/50);
noise    = 0.5*randist(pure, 'uni', 0);
noisy    = pure + noise;
filtered = filter(MA(20), [1 zeros(1,19)], noisy);
plot(n, pure,     '--k', ...
     n, 2+noise,  '-r', ...
     n, noisy,    '-k', ...
     n, filtered, '-g')
legend('Pure signal', ...
       'Noise (offset by 2)', ...
       'Noisy signal', ...
       'Filtered signal', ...
       'Location', 'NorthOutside')
xlim([-10 200]);
mysaveas('p2_sinusoid_filtered', 10, 3.5)

% Calculate attenuation factor
att = max(filter(MA(20), [1 zeros(1,19)], pure));
disp(['Attenuation factor: ' num2str(att)])
```

```
% Correlated sinusoid
plot(n,                pure,          '--k', ...
     n,                2+noise,       '-r', ...
     n-10*ones(1,200), filtered/att, '-b')
legend('Pure signal', ...
    'Noise (offset by 2)', ...
    'Filtered signal (correlated)', ...
    'Location', 'NorthOutside')
xlim([-10 200])
mysaveas('p2_sinusoid_correlated', 10, 3.5)

diary off
```

Listing 2: The output of listing 1, `diary.txt`.

```
Impulse response of y[n]: 0.5*udelta(n)+0.5*udelta(n-1)
Attenuation factor: 0.75732
```