332:347 – Linear Systems Lab – Fall 2016

Lab 3 – S. J. Orfanidis

In this lab, besides the theoretical and numerical questions, you will need to produce three graphs for problem 1, and 18 graphs for problem 2.

1. In this part, you will study the sinusoidal steady-state and transient response of a filter. Consider the following sinusoidal input signal and filter:

$$x(t) = \sin(\omega_0 t)\,u(t)\,, \quad H(s) = \frac{s+3}{s^2 + s + 1.25}$$

where $\omega_0 = 4$ rad/sec and t is in seconds.

(a) Determine the poles of the filter and calculate its 40-dB time constant in sec. Determine an analytical expression for the impulse response $h(t)$ of this filter (you may use the symbolic toolbox here).

Define a vector of 1001 equally-spaced time points t spanning the interval $0 \le t \le 10$, and evaluate and plot $h(t)$ over that interval.

(b) Calculate the value of the frequency response at $\omega = \omega_0$, that is, $H(j\omega_0)$, as well as its magnitude $|H(j\omega_0)|$, and phase $\phi_0 = \arg H(j\omega_0)$.

For the same time vector t of part (a), evaluate the corresponding steady-state output signal due to the sinusoidal input $x(t)$, that is,

$$y_{\text{st}}(t) = |H(j\omega_0)|\,\sin(\omega_0 t + \phi_0) \tag{1}$$

(c) The exact response to a complex sinusoidal input $e^{i\omega_0 t}u(t)$ was worked out in class for any filter (with single poles):

$$y_c(t) = H(j\omega_0)\,e^{j\omega_0 t}u(t) + \sum_{i=1}^{M} R_i e^{p_i t}u(t) \tag{2}$$

where p_i, $i = 1, 2, \dots, M$, are the filter poles, M is the order of the denominator polynomial of $H(s)$, assuming that the order of the numerator polynomial was at most M. The exact response due to a real-valued sinusoidal input $\sin(\omega_0 t)u(t)$ is obtained by calculating the imaginary part of Eq. (2).

Using the function **residue**, compute the quantities R_i, p_i that appear in Eq. (2), and then evaluate $y_c(t)$ at the same vector of t's defined above, and extract its imaginary part $y(t) = \text{Im}\big[y_c(t)\big]$.

On the same graph, plot $y(t)$, $y_{\text{st}}(t)$, and the input $x(t)$, and observe how $y(t)$ tends to $y_{\text{st}}(t)$ as t increases.

(d) Calculate the theoretical phase delay, $t_{\text{ph}} = -\phi_0/\omega_0$, in seconds. Then, estimate it from the computed graphs as follows: Consider the later time interval, $8 < t < 9$, and using the built-in function **max**, determine the time instants, say, t_1, t_2, at which $x(t)$ and $y(t)$ reach their maximum values, then, compute the estimated phase shift as the difference, $t_{\text{est}} = t_2 - t_1$, and compare it with the theoretical value t_{ph}. Add the pairs of points $t_1, x(t_1)$ and $t_2, y(t_2)$ on the graph of part (c) using markers (see example graph at end).

(e) On a separate graph, plot only the transient part $y_{\text{tr}}(t) = y(t) - y_{\text{st}}(t)$. Is its rate of decrease consistent with the 40-dB time constant that you calculated above?

2. A signal $x(t)$ consists of a sinusoid plus random noise:

$$x(t) = \sin(\omega_0 t) + v(t) \tag{3}$$

It is desired to process $x(t)$ through a bandpass filter $H(s)$ that lets the sinusoid pass through unchanged, while it substantially attenuates the noise component, so that the output signal would have the form:

$$y(t) = \sin(\omega_0 t) + y_v(t) \tag{4}$$

where $y_v(t)$ denotes the filtered noise, which must be much weaker than the input noise, i.e., the RMS value of $y_v(t)$ must be much less than the RMS value of $v(t)$, or in terms of their variances, $\sigma_{y_v}^2 \ll \sigma_v^2$. Such a bandpass filter can be designed to have transfer function and I/O differential equation:

$$H(s) = \frac{\alpha s}{s^2 + \alpha s + \omega_o^2} \quad \Leftrightarrow \quad \ddot{y}(t) + \alpha \dot{y}(t) + \omega_0^2 y(t) = \alpha \dot{x}(t) \tag{5}$$

This is complementary to the notch filter discussed in lab-2. Such bandpass filter could represent, for example, a simple radio receiver tuned to the carrier frequency ω_0 of a radio station, allowing through only a small band of frequencies around the carrier, and rejecting all other frequencies. Its magnitude frequency response, obtained by setting, $s = j\omega$, is given by:

$$|H(j\omega)|^2 = \frac{\alpha^2\omega^2}{\left(\omega^2 - \omega_0^2\right)^2 + \alpha^2\omega^2} \tag{6}$$

It has a narrow peak centered at ω_0 and unity gain there, i.e., $H(j\omega_0) = 1$. Its 3-dB width is $\Delta\omega = \alpha$ (see graphs at end). As discussed in class, its impulse response is given by:

$$h(t) = \alpha e^{-\alpha t/2}\left[\cos(\omega_r t) - \frac{\alpha}{2\omega_r}\sin(\omega_r t)\right]u(t), \quad \omega_r = \sqrt{\omega_0^2 - \frac{\alpha^2}{4}} \tag{7}$$

Assuming the noise component to be white noise with broadband flat spectrum, the narrow peak of the filter will only let through a small part of the noise (whatever lies within the effective width of the peak), so that the output noise power will be proportional to the bandwidth parameter α — it can be shown that $\sigma_{y_v}^2/\sigma_v^2 \approx T\alpha/2$, where T is the sampling time step.

Thus, the smaller the α, the more the noise reduction. On the other hand, as can be seen from Eq. (7), the smaller the α, the longer the effective time constant $t_{\text{eff}} = 2/\alpha$ of the filter,† resulting in longer transients. In this lab, you will study this tradeoff between noise reduction and speed of response.

(a) Start with the values $\omega_0 = 5$ and $\alpha = 0.2$. Plot the magnitude response squared $|H(j\omega)|^2$ versus ω in the interval $0 \le \omega \le 10$. Then, plot the phase response, $\arg\left[H(j\omega)\right]$, versus the same values of ω.

(b) Generate 2001 equally-spaced noisy sinusoidal samples of $x(t)$ in the interval $0 \le t \le 40$, e.g., using the code:

```
Tmax = 40;  T = Tmax/2000;      % T = 0.02 sec
t = 0:T:Tmax;
seed = 2016; rng(seed);         % initialize random number generator
v = randn(size(t));             % zero-mean, unit-variance, gaussian
x = sin(w0*t) + v;              % noisy input
```

Compute the filter output samples $y(t)$ using the function **lsim**,

```
y = lsim(tf(num,den),x,t, [0;0], 'zoh');
```

† the 60-dB time constant is actually, $t_{60} = \ln(1000)\cdot 2/\alpha = 13.8/\alpha$.

where this syntax, as opposed to, `y=lsim(tf(num,den),x,t)`, forces the use of the zero-order-hold method of integration.

On two separate graphs, plot $x(t)$ and $y(t)$ versus t. Observe the initial transients and the steady-state output (it's not quite equal to the sinusoid because a small amount of noise survives the filtering process.)

(c) To observe what happens to the noise itself, filter the noise signal $v(t)$ through this filter to obtain the filtered noise $y_v(t)$. On two separate graphs, but using the same vertical scales, plot the signals $v(t)$ and $y_v(t)$ versus t.

(d) Repeat parts (a–c) for the values $\alpha = 0.5$ and $\alpha = 1$, discussing the tradeoffs between noise reduction, speed of response, and quality of the resulting desired signal.

(e) The zero-order-hold method implemented by the function **lsim** is equivalent to replacing the continuous-time transfer function $H(s)$ of Eq. (5) by the following discrete-time transfer function and corresponding input/output difference equation:

$$\begin{aligned} H_d(z) &= \frac{Gz^{-1}(1-z^{-1})}{1+a_1z^{-1}+a_2z^{-2}} \\ y_n + a_1y_{n-1} + a_2y_{n-2} &= G(x_{n-1}-x_{n-2}) \end{aligned} \tag{8}$$

with coefficients:

$$\begin{aligned} G &= \frac{\alpha}{\omega_r}\, e^{-\alpha T/2}\sin(\omega_r T) \\ a_1 &= -2e^{-\alpha T/2}\cos(\omega_r T) \\ a_2 &= e^{-\alpha T} \end{aligned} \tag{9}$$

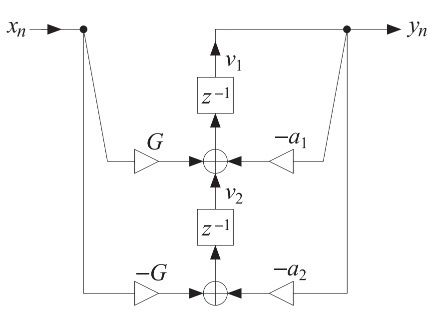
We will derive this result in class later. Also to be derived is the corresponding discrete-time impulse response, for $n \geq 0$,

$$h_d(n) = g(n) - g(n-1)\,, \quad g(n) = \frac{\alpha}{\omega_r}\, e^{-\alpha nT/2}\sin(\omega_r nT)\, u(n) \tag{10}$$

Notice that $h_d(0) = 0$ as expected from the presence of the factor z^{-1} in the numerator of $H_d(z)$. Note also that $h_d(n)$ can be computed as the following integral of the continuous-time impulse response $h(t)$,

$$h_d(n) = \int_{nT-T}^{nT} h(t)\,dt\,, \quad n \geq 0 \tag{11}$$

Eqs. (10) and (11) can also be derived with MATLAB's symbolic toolbox. The output samples $y_n = y(t_n)$ corresponding to the input samples $x_n = x(t_n)$ can be computed by a repetitive loop that solves the difference equation (8), for example, choosing the *transposed* realization depicted below, we have,



initialize $v_1 = v_2 = 0$, then,
for each $n = 0, 1, 2, \dots$ do:
  $y_n = v_1$
  $v_1 = v_2 + Gx_n - a_1y_n$
  $v_2 = -Gx_n - a_2y_n$

(12)

The built-in function **filter** uses exactly the same transposed realization to compute the output. Its syntax is as follows, where `num,den` are the numerator and denominator coefficients of the discrete transfer function $H_d(z)$,

```
y = filter(num,den,x);
```

For the values $\omega_0 = 5$, $\alpha = 0.2$, $T = 0.02$ (same T as above), use the iteration of Eq. (12) to compute the discrete-time output signal $y_n = y(t_n)$ for the same sampled input of part (b). Then, compute it again using the function **filter**. Do not plot the signals $y(t_n)$ since they are virtually indistinguishable from those calculated with **lsim**.

However, using the built-in function **norm**, do compare the resulting output vectors from the three methods, by computing the Euclidean norms of the corresponding error differences, that is, the quantities,

$$E_{\text{lsim}} = \|\mathbf{y}_{\text{lsim}} - \mathbf{y}_{\text{iter}}\| = \text{norm}(\mathbf{y}_{\text{lsim}} - \mathbf{y}_{\text{iter}})$$
$$E_{\text{iter}} = \|\mathbf{y}_{\text{iter}} - \mathbf{y}_{\text{filter}}\| = \text{norm}(\mathbf{y}_{\text{iter}} - \mathbf{y}_{\text{filter}})$$

where we expect E_{lsim} to be tiny, and E_{iter} to be zero.

(f) You may have learned in your Probability & Random Processes course—and we will be deriving this at a later date—that if you filter a zero-mean white-noise signal sequence $v(t_n)$ of variance σ_v^2, through a discrete-time filter, then the variance $\sigma_{y_v}^2$ of the corresponding output noise signal $y_v(t_n)$ is related to σ_v^2 by,

$$NRR = \frac{\sigma_{y_v}^2}{\sigma_v^2} = \sum_{n=0}^{\infty} |h_d(n)|^2 \tag{13}$$

This ratio is a measure of the noise reduction capability of the filter, and is known the *noise reduction ratio* (NRR). For the filter of Eq. (10), the exact value of NRR can be determined in closed-form,

$$NRR_{\text{exact}} = 2\,\text{Re}\left[\frac{A^2 p^2}{1-p^2}\right] + 2\,\frac{|A|^2|p|^2}{1-|p|^2} \tag{14}$$

where p, A are given by,

$$p = e^{-\alpha T/2} e^{j\omega_r T}, \quad A = \frac{G e^{\alpha T/2}(1 - e^{\alpha T/2} e^{-j\omega_r T})}{2j \sin(\omega_r T)}$$

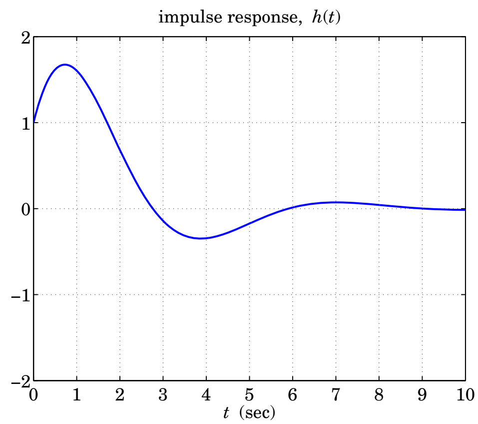
The NRR can also be estimated by the following three approximations,

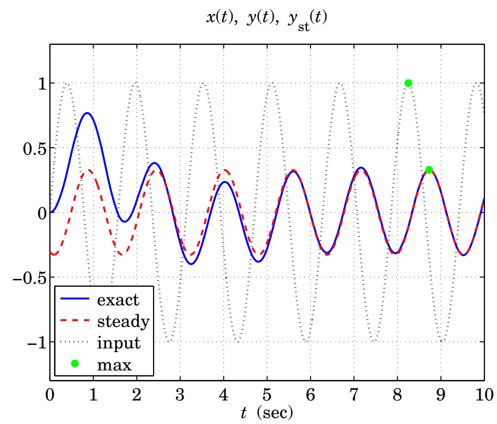
$$\begin{aligned}
NRR_1 &= \sum_{n=0}^{N-1} |h_d(n)|^2 = \text{truncated sum with } N \text{ terms} \\
NRR_2 &= \frac{\hat{\sigma}_{y_v}^2}{\hat{\sigma}_v^2} = \text{using sample variances} \\
NRR_3 &= T\,\frac{\alpha}{2} = \text{based on the continuous-time } h(t)
\end{aligned} \tag{15}$$

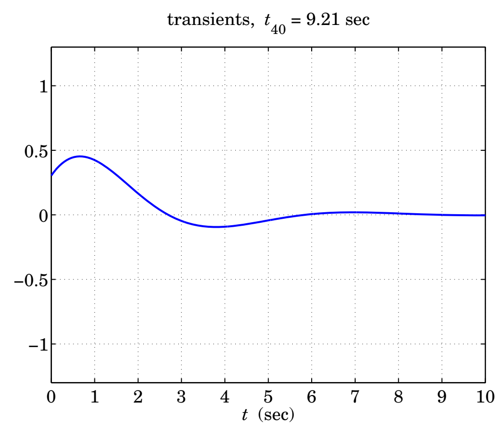
where N can be chosen to correspond to two 60-dB time constants, i.e., $N = 2t_{60}/T$, and the sample variances can be calculated using the built-in function **std** for the standard deviation of a random vector, and the third expression was mentioned in the introduction.

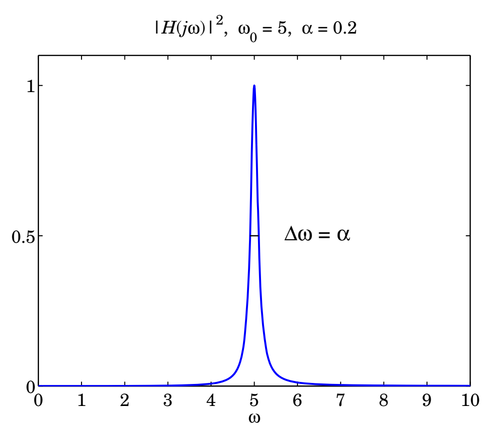
Calculate and compare the NRR values from Eqs. (14) and (15) for the values, $\omega_0 = 5$, $\alpha = 0.2$, $T = 0.02$. For NRR_2, the necessary output noise vector $y_v(t_n)$ can be calculated using **filter** with input $v(t_n)$.

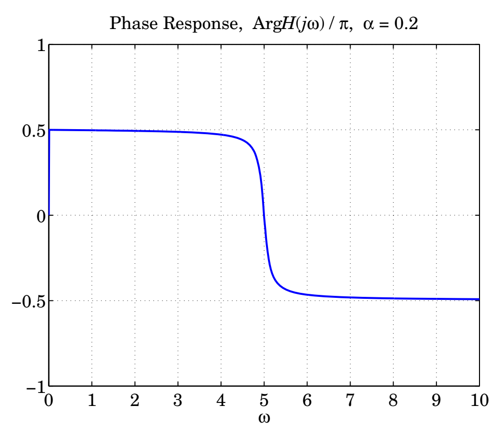
Typical Outputs

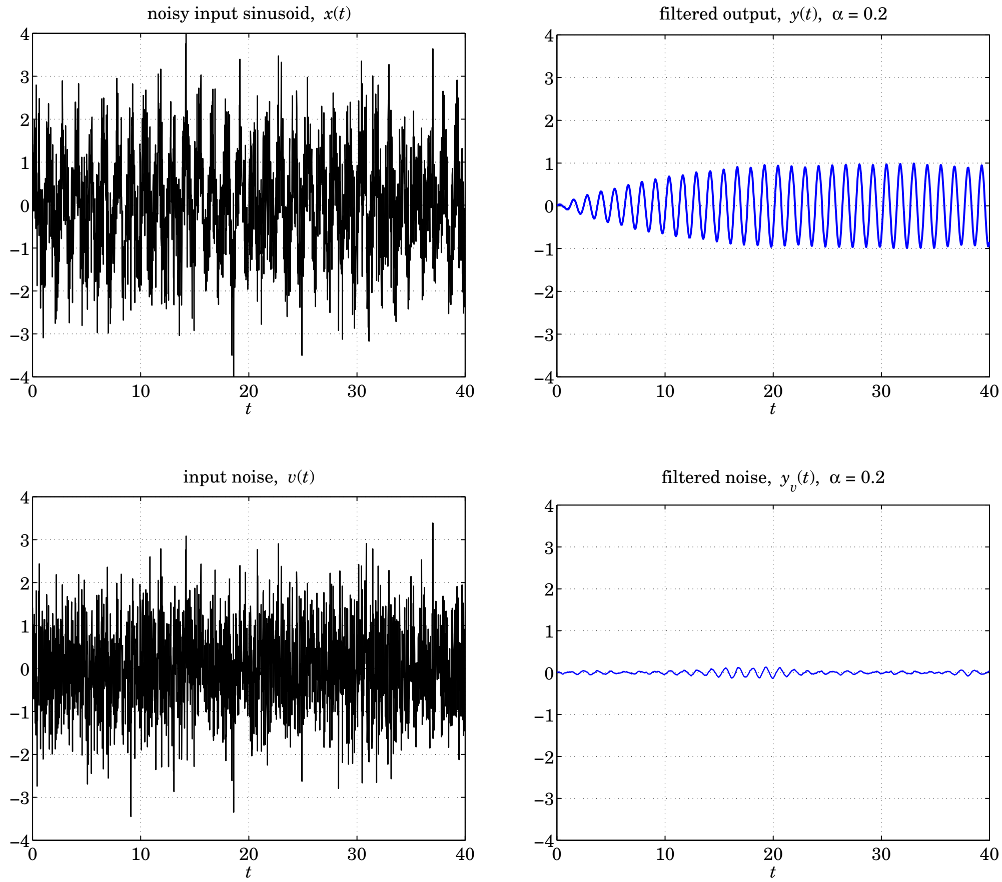











noisy input sinusoid, x(t)
filtered output, y(t), α = 0.2
input noise, v(t)
filtered noise, y_v(t), α = 0.2
t