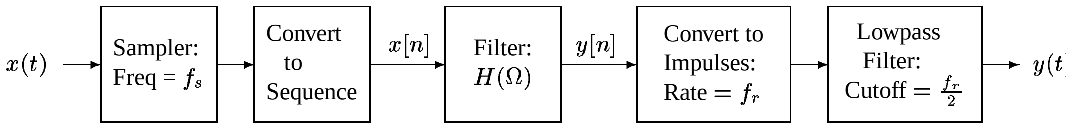
ECE 313 - Signals and Systems, Fall 2012

Practice Problems - Final Exam (More to come!)
Remember also to look at the Practice Problems from Exams #1 and #2!

1. Consider the following system to process continuous-time signals with discrete-time processing.



(a) Suppose that $x(t) = \cos(2\pi 2500t)$, $f_s = f_r = 10000$ Hz, and the discrete-time filter frequency response, which you recall is periodic with period 2π, is given in $[-\pi, \pi]$ as:

$$H(\Omega) = \frac{\Omega^2}{\pi^2}, \qquad |\Omega| < \pi$$

Find the output $y(t)$.

(b) Suppose that $x(t)$ has Fourier Transform $X(f) = p(f/10000)$, $f_s = f_r = 10000$ Hz, and the discrete-time filter frequency response, which you recall is periodic with period 2π, is given in $[-\pi, \pi]$ as:

$$H(\Omega) = \begin{cases} 1, & \frac{\pi}{2} < |\Omega| < \pi \\ 0, & \text{otherwise} \end{cases}$$

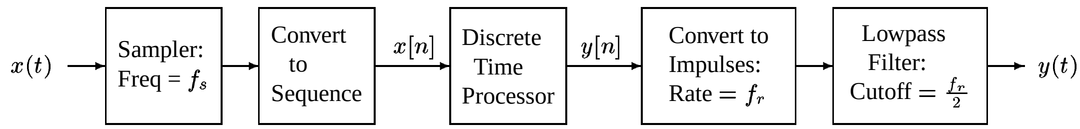
Find the Fourier transform $Y(f)$ of the output $y(t)$, and then use it to find $y(t)$.

(c) Suppose that $x(t) = \cos(2\pi 2500t)$, $f_s = f_r = 10000$ Hz, and the discrete-time filter is defined by the difference equation:

$$y[n] = 0.5y[n-1] + x[n]$$

Find the output $y(t)$ [*Note: You may get some weird trigonometric thing (e.g.* $\cos^{-1}(\frac{3}{7})$*) that you cannot evaluate without your calculator. Just leave it in its trigonometric form (e.g.* $\cos^{-1}(\frac{3}{7})$*) in your answer.*]

2. **This problem was mostly done in class.** Consider the following system to process continuous-time signals with discrete-time processing.



Your discrete-time processing toolkit on your computer has the following four blocks for possible use (when a frequency response is given, it is valid for $\Omega \in [-\pi, \pi]$, and it repeats outside of that range, of course):

Block 1: $H_1(\Omega) = e^{-j\Omega n_0}$, where you can choose any n_0 that you wish

Block 2: $H_2(\Omega) = j\Omega$

Block 3: $$H_3(\Omega) = \begin{cases} 1, & |\Omega| \leq \frac{\pi}{2} \\ 0, & \text{else} \end{cases}$$

Block 4: The fourth block provides output $y[n] = (-1)^n x[n]$ for input $x[n]$.

Using any combinations of these blocks (including multiple versions of any, if needed) and specifying the sampling frequencies f_s and f_r, provide a separate system to implement each of the following desired continuous-time operations with the block diagram above.

(a) The lowpass filter $H(f) = p(f/2000)$, to be applied on input signals $x(t)$ of bandwidth up to 1.5 KHz to yield the signal $y(t)$.

(b) A system that delays the input $x(t)$, of bandwidth less than 10 KHz, by 2 $\times 10^{-3}$ seconds to yield the signal $y(t)$.

(c) A system that yields $y(t) = \frac{d}{dt}x(t)$ for inputs $x(t)$ of bandwidth less than 10 KHz.

(d) A system that implements the bandpass filter:

$$H(f) = \begin{cases} 1, & 5000 < |f| < 10000 \\ 0, & \text{else} \end{cases}$$

on inputs signals $x(t)$ of bandwidth less than 10 KHz to yield the signal $y(t)$.

3. **This problem was done in class.** In this problem, you are going to consider how to do multiplication of two continuous-time signals using discrete-time processing.

(a) Find the Fourier transform $X(f)$ of:

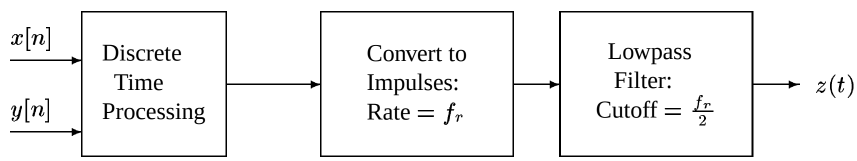
$$x(t) = 5000\text{sinc}(10000t)$$

and the Fourier transform $Y(f)$ of

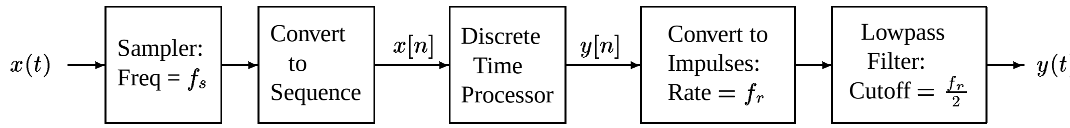
$$y(t) = 5000\text{sinc}(40000t)$$

(b) Find the bandwidth (you do not have to find the full Fourier transform unless you would like) of the signal $z(t) = x(t)y(t)$.

(c) Your buddy samples the signal $x(t)$ at 15 KHz to obtain the sequence $x[n]$ and the signal $y(t)$ at 45 KHz to obtain the signal $y[n]$. Give the discrete-time processing of $x[n]$ and $y[n]$, and the sampling frequency f_r, for the D/C conversion such that the circuit below outputs $z(t) = x(t)y(t)$. [*Notes: (1) Be sure that your circuit would work for any $x(t)$ and $y(t)$ of the same bandwidth as the signals given. (2) You have all discrete-time blocks for your use, including discrete-time filters of any type, upsamplers, downsamplers, mathematical operations of any type, etc.*]



4. Consider the following system to process continuous-time signals with discrete-time processing.



Note that there is no anti-aliasing filter on the front-end of the sampler. As in class, let the output of the five blocks be $x_s(t)$, $x[n]$, $y[n]$, $y_s(t)$, and $y(t)$, respectively, with corresponding Fourier transforms given by $X_s(f)$, $X_{DT}(\Omega)$, $Y_{DT}(\Omega)$, $Y_s(f)$, and $Y(f)$.

Suppose the input to the system is given by:

$$x(t) = 20000 \text{ sinc}(20000t)$$

Below you will be asked to find $Y(f)$, the Fourier transform of the output $y(t)$, for a bunch of different systems. Notes:

- **A sketch of $Y(f)$ is sufficient - no need to give an equation.**
- Be sure to justify your answers. It is sufficient (and I would encourage you) to sketch $X_s(f)$, $X_{DT}(\Omega)$, $Y_{DT}(\Omega)$, $Y_s(f)$, and $Y(f)$, although you can omit $X_s(f)$ and $Y_s(f)$ if you know what you are doing.
- Do not worry about amplitudes!

(a) Suppose $f_s = f_r = 25$ kHz, and the discrete-time processing is the filter

$$H_{DT}(\Omega) = 1 - \frac{|\Omega|}{\pi}, \qquad -\pi < \Omega \leq \pi$$

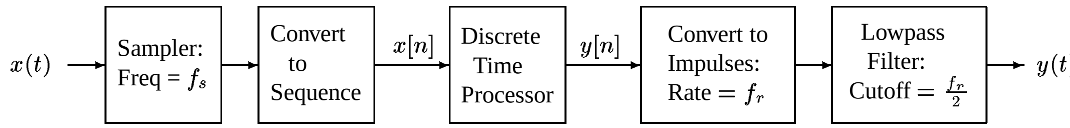
(and the filter is periodic with period 2π, of course). Find $Y(f)$, the Fourier transform of the output $y(t)$.

(b) Suppose $f_s = f_r = 15$ kHz, and there is no discrete-time processing (i.e. $y[n] = x[n]$). Find $Y(f)$, the Fourier transform of the output $y(t)$.

(c) Suppose you want $y(t) = x^2(t)$ (output is the square of the input) and you proceed as follows: You let $f_s = 25$ kHz. Then, realizing you need to move to a higher sampling rate if you want to square, you use the discrete-time processing: (i) interpolate by 8, (ii) decimate by 5, and (iii) $y[n] = x^2[n]$. Because of your new sampling rate, you let $f_r = 40$ kHz. Find $Y(f)$, the Fourier transform of the output $y(t)$. **Also, comment on how well your squarer worked.**

(d) Suppose you want $y(t) = x^2(t)$ (output is the square of the input) and you proceed as follows: You let $f_s = 25$ kHz. Then, realizing you need to move to a higher sampling rate if you want to square, you use the discrete-time processing: (i) decimate by 5, (ii) interpolate by 8, and (iii) $y[n] = x^2[n]$. Because of your new sampling rate, you let $f_r = 40$ kHz. Find $Y(f)$, the Fourier transform of the output $y(t)$.**Also, comment on how well your squarer worked.**

5. Consider the following system to process continuous-time signals with discrete-time processing.



Suppose that the impulse response of our softward module “FILTER” is given by:

$$h[n] = 0.5\delta[n+1] + \delta[n] + 0.5\delta[n-1]$$

where $\delta[n]$ is the standard discrete impulse function.

Note that there is no anti-aliasing filter on the front-end of the sampler.

(a) To test our code, we first give input $x[n] = \delta[n] + \delta[n-3]$ to the module “FILTER” and obtain the output $y[n]$. Find the output $y[n]$.

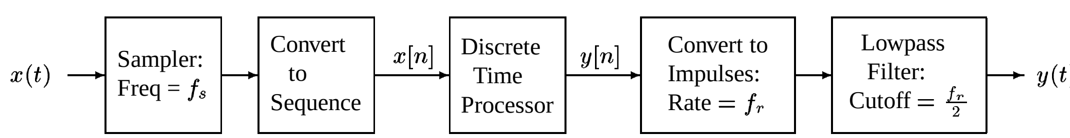
(b) Find the frequency response $H_{DT}(\Omega)$ of “FILTER” and plot its magnitude $|H_{DT}(\Omega)|$

(c) Suppose that the continuous-time signal $x(t) = \cos(2\pi 10000 t)$ is input to our system, with $f_s = f_r = 30$ kHZ, and the discrete-time processing is the application of the filter $h[n]$ to $x[n]$. Find the output $y(t)$.

(d) Suppose you could not figure out part (b), so you turn to MatLab and the discrete Fourier transform (DFT) to help you understand what $h[n]$ will do to various input frequencies. You are planning on employing $f_s = f_r = 30$ kHz, and you want to be able to have an analog frequency resolution of at most 40 Hz. What N should you employ for your DFT?

(e) Your operation from (d) will result in a DFT $H[k], k = 0, 1, \ldots, N-1$. At what k (approximately) will you find the gain that impacts the sinusoid in part (c)?

6. Consider the following system to process continuous-time signals with discrete-time processing.



Note that there is no anti-aliasing filter on the front-end of the sampler.

Suppose that the input is $\cos(2\pi 2000 t)$, and I want to double its frequency, so that the output is $A_1 \cos(2\pi 4000 t)$, where A_1 is any positive constant. Show how this can be done by the proper selection of f_s and f_r without any discrete-time processing.

discrete-time processing.

7. When you sample $x(t)$ to get the sampled version $x_s(t)$, you nearly always get a Fourier transform $X_s(f)$ that is non-zero at places where the $X(f)$ was zero. This implies that the sampler is not a linear time-invariant (LTI) system. Is it non-linear? Or is it time-variant? (or both). (**Be sure to start with the definitions of linearity and time-invariance to get both get this problem correct and get full credit.**)