

**Problem Set 2: Sampling & Filters****Total marks:** 100**Due Date:** Sunday, May 12, 2019 at 23:59 AEST [*end of week 10*]

**Note:** This assignment is worth **20%** of the final course mark. Please submit answers via [Platypus](#). Solutions, including equations, should be typed please and **submitted directly in Platypus** (preferred) or as **PDF**. Note that Microsoft **Word** documents and scanned images of **handwritten pages** are specifically **disallowed**. The grade is determined by the teaching staff directly (which may be formed after peer reviews are entered). Please double-check that your name is not in the solution directly or via the associated metadata.

Also, the tutors will **not** assist you further unless there is real evidence you have attempted the questions.



Finally, a note of remembrance for our colleagues and compatriots in Paris. Where there is noise, there is also a new, indelible, signal.

*Thank you. :-)*

**Questions**

Explain your solutions as if you are trying to **teach a peer**. Demonstrate your insight and understanding. Answering an entire question with bare equations, lone numbers or without any verbal explanation is not acceptable. Marks may be reduced if an answer is of poor quality, demonstrates little effort or significant misunderstanding.

**Q1. A Quick Sample<sup>1</sup>****[10 points]**

Consider a continuous time signal  $s(t)$  sampled at  $f_s$  and is bandlimited to a frequency less than  $f_n = f_s/2$ . It has a Nyquist rate of  $\omega_n = 2\pi f_s = 4\pi f_n$ . **Determine and briefly justify** the required **Nyquist rate** ( $\omega_n$ ) of the following variations of this signal.

(a)  $a(t) = s(t) + s(t - 3004)$

(b)  $b(t) = \frac{ds(t)}{dt}$

(c)  $c(t) = s^2(t)$

(d)  $d(t) = s(t) \cdot (\cos(\omega_n t))$

<sup>1</sup> Correction:  $\omega_n$  was defined incorrectly with respect to  $f_n$  (instead of the correct  $f_s$ ). See also, Lathi, p.321. However, the answers are relative to the given  $\omega_n$  so they would have all scaled across. (Also, an apology, it seems that Google Doc does not allow equations in Footnotes).

## Q2. Filtering Smoothly & Orderly

[15 points]

[Caution: PS 2 (2017) Question 3 asks a similar looking question, but for a **low pass** filter]

Filters run the gamut. Let us consider two cases.

(a) Butterworth: Smooth & Creamy?

Design a high pass, Butterworth filter with a stopband stopping at  $0.3\pi$  rads/sample, a passband starting at  $0.5\pi$  rads/sample, and a minimum rise of 60dB between them.

What is the lowest order filter you can have which satisfies these criterion?

Please plot the poles and zeros of this filter in the  $z$ -plane.

[See Also: Matlab functions: buttord, butter, bode, tf, zplane, and freqz]

(b) Chebyshev: Sharp, but Rough?

Please now design a Chebyshev filter to the same specifications. What's the lowest order it can take, while satisfying the criterion? Please justify your choice of Chebyshev filter.

[See Also: The Matlab functions: cheby1, cheb1ord, cheb1ap, and cheby2]

(c) What would happen to the minimum order of the Chebyshev filter if the minimum rise between the stopband and passband was changed to 30 dB? Or 90 dB? **Please generalize and explain** how the minimum rise relates to filter order for this case.

## Q3. An Exponentially Fun Filter

[25 points]

Consider an exponential impulse response given by:  $y[n] = a^n x[n]$

(for  $0 < a < 1$ , thus  $|a| < 1$ ).

Initially consider  $x[n] = u[n]$ , where  $u[n]$  is the discrete-time unit step function.<sup>2</sup>

(a) What is the **z-Transform** of this function? (i.e., What is  $Y(z)$ ?)

(b) What is the corresponding Magnitude response for this filter?

[Hint: what is  $Y(e^{j\omega})$  for this case?]

(c) What is the corresponding Phase response for this filter?

[Hint: Euler's Formula could be helpful depending on how one approaches this]

(d) Please plot the Magnitude and Phase response of this filter,  $Y(z)$ , for a generalized  $a$  against the frequency  $\omega$  for a range of  $-2\pi \leq \omega \leq 2\pi$ .

(e) Is this function (and its plot) periodic?

If so, is there a shorter frequency interval that this could then be displayed for (i.e., for an interval given by  $p \leq \omega \leq q$ , what might be values for  $p$  and  $q$ ?)

---

<sup>2</sup> (April 30) Clarification on the initial value of  $x[n]$  as  $x[n]=u[n]$ .

#### Q4. Getting Past the Echo Chamber

[25 points]

The Zonama *cho* –  $e^{\pi i}$  is a smart device<sup>3</sup> that needs to receive voice signals in small room with an echo (i.e., reverberation). The goal is to see if a digital filter can provide needed clarity.

Given that receiving signals in the presence of an echo is a commonplace and practical task, this question looks at a means of understanding and potentially overcoming the echo with a series of filters.

Consider a microphone (receiver) that records a signal  $m(t)$  based on the original spoken sound  $s(t)$  and its echo,  $a \cdot s(t - T_0)$ , an attenuated ( $|a| < 1$ ) and delayed copy of  $s(t)$ . Thus,  $m(t) = s(t) + a \cdot s(t - T_0)$ .

The general idea is to sample the received signal  $m(t)$  to a discrete sample train  $m[n]$  to then filter this via a digital filter (difference equation),  $h[n]$ , to give an output  $o[n]$ . This output is then filtered via an ideal low pass filter of gain  $K$  as part of<sup>4</sup> being reconstructed as an analog output  $y(t)$ .

Assume that the source sound,  $s(t)$ , is band limited ( $|S(j\omega)| = 0$  for  $|\omega| > \omega^*$ )

- (a) Please draw up a system diagram for the process described above. **Succinctly explain** both (1) why there is an ideal low pass filter at the reconstruction end?, and (2) Is an ideal low pass filter needed before sampling the signal  $m(t)$ ?
- (b) If  $T_0 < \pi/\omega^*$  and the system is sampled at a fixed sampling period ( $T_s$ ) that is no more than  $T_0$  (i.e.,  $\alpha T_s = T_0$ ,  $\alpha \geq 1$ ), then determine a difference equation for the filter,  $h[n]$ , such that the final output  $y(t)$  is proportional to  $s(t)$ .  
[Hint: start with  $\alpha=1$ , then consider the case for  $\alpha \gg 1$ ]
- (c) Specify the gain  $K$  of the low-pass filter such that the final output  $y(t) = s(t)$   
[Hint: start with  $\alpha=1$ , then consider the case for an arbitrary  $\alpha$ ]
- (d) *Okay do cho* –  $e^{\pi i}$ , is there aliasing? Suppose the echo gets worse and the delay gets larger. That is, (for this part (d)) assume  $\pi/\omega^* < T_0 < 2\pi/\omega^*$ . Are we out of luck? Is there a sampling period  $T_s$ , filter gain  $K$ , and filter  $h[n]$  such that  $y(t) = s(t)$ ?

---

<sup>3</sup> Some claim it's the number  $-\exp(\pi i)$  smart device for ELEC3004 problem sets. It's certainly the most complex named unit out there ☺.

<sup>4</sup> (May 5): clarified that the filter is part of the reconstruction, which happens before the final end output.

**Q5. Deep FIR: No Fake News (on) Filters****[25 points]**

Filters are a type of system. So are Deep Convolutional Neural Networks (CNNs). We investigate cascading FIR filters and their similarity to a popular machine learning tool.

**Recall:** that the term “windowing” refers to multiplying a signal  $x[n]$  or an ideal impulse response  $h[n]$  by a finite duration sequence  $w[n]$  of length  $N$  (i.e.,  $w[n] = 0$  for  $n < 0$ ,  $n \geq N$ ) to produce a finite-duration result  $x_w[n] = x[n]w[n]$  or  $h_w[n] = h[n]w[n]$ . The latter is particularly useful for filtering. Of the number of Window functions that have been designed, one particularly useful filter is the Hamming Window:  $w_{HAM}[n] = (0.54 - 0.46 \cdot \cos(\frac{2\pi n}{N-1}))(u[n] - u[n-N])$ . The filter’s output is then given by:

$$y[n] = h_w(1) \cdot x[n] + h_w(2) \cdot x[n-1] + \dots + h_w(nh_w + 1) \cdot x[n - nh_w]$$

(a) *Does it have to be this way?*

**Briefly explain**, is it possible to have an online FIR filter without a window function?

(b) *Low-Pass Filter<sup>5</sup>*

Consider the ideal impulse response of a low pass filter (of cut-off frequency  $\omega_c$ ) given by  $h_{LP}[n] = \omega_c \text{sinc} \omega_c(n - \frac{N-1}{2})$  for a filter size  $N$ .

Thus,  $h_{W_{LP}}[n] = h_{LP}[n]w_{HAM}[n]$

Now consider the case where  $N=100$  and  $\omega_c = 0.3004$ . Determine  $h_{W_{LP}}[n]$  using the formula above. Then generate a FIR filter in Matlab (or equivalent) using the command `fir1`.

**Briefly explain**, are these filters the same?

Using the plotting tool `freqz` to see the difference of these filters.

(c) *Filters, on the Double*

If one filter is useful, are then, perhaps, two filters, doubly useful? Consider applying the same the filter twice, that is  $h_{double}[n] = h_{LP}[n]h_{LP}[n]w_{HAM}[n]$ .

Building on the result from part (b), compute and plot  $h_{double}[n]$ . Please describe what happens to the filter’s performance and attenuation relative to  $h_{W_{LP}}[n]$ .

(d) *Make it Sing!*

Load a song into Matlab of your choice (such as `chip [load chirp]`). Apply the two filters ( $h_{W_{LP}}[n]$  and  $h_{double}[n]$ ) using the `filtfilt` command. Clearly and scientifically plot the output of the song using the two filters. Briefly compare and contrast the filtered outputs.

(e) *Go Deep!*

**[Challenging! — Bonus Marks]**

Consider the case of a cascade of filters, say four, such that

$h_{forfun}[n] = h_A[n]h_B[n]h_C[n]h_D[n]w_{HAM}[n]$ . Also imagine that we do not want to start with  $h[n]$ , but rather “learn” filters the  $h_A[n]$ , ...,  $h_D[n]$  from examples of noisy and clean songs. If  $N=100$ , how many degrees of freedom are there in total? How many song pairs are needed to make solve this? Is there something we can do make the approach faster?

<sup>5</sup>(May 2)  $\omega_c$  and 0.3004 updated (previous version had  $\omega_c/\pi$  and 0.3004 $\pi$ )