EE 141 DIGITAL SIGNAL PROCESSING

Lab 4: Sampling and Aliasing

In this lab, we will look into fundamentals of sampling from continuous-time signals. We will compare signals with inadequate sampling rates with or without anti-aliasing filters. We will also emulate practical reconstruction filters, while staying completely in discrete-time domain.

Before starting the lab, download the .MAT file provided on iLearn and load it typing

```
load ertem_voice
```

This will load a vector named `audio_sample` and a variable `fs` to the workspace. This is a mono voice signal with sampling rate $f_s = 8,000$Hz. As in the previous lab, to play this one needs to type

```
p = audioplayer(audio_sample,fs);
```

followed by `play(p)`

Question 1:

a) Using the `fft` command, plot the DTFT of the audio signal. Remember, you need to use

```
plot(fftshift(abs(fft(...))))
```

for this.

b) Decrease the sampling rate to $f_s = 4,000$Hz using the process of *subsampling*. That is, create a new vector by keeping only every other sample, using `audio_sample(1:2:end)`. Now plot the DTFT and compare with part **a**. Do you observe new content at high frequencies that did not appear in the original DTFT? Those are the aliasing effects. Play the new audio. Do you hear any aliasing effects?

c) Repeat part **b** with sampling rates $f_s = 2,000$Hz and $f_s = 1,000$Hz. Do you observe more and more activity at high frequencies that did not appear before? Does the aliasing effect become more and more heard?

d) Now, repeat the process by lowpass filtering the original signal *before* subsampling for each of the three sampling rates in **b** and **c**. This should be done by discrete-time lowpass filters with cutoff frequencies $\frac{\pi}{2}$, $\frac{\pi}{4}$, and $\frac{\pi}{8}$ for sampling rates $f_s = 4,000$Hz, $f_s = 2,000$Hz, and $f_s = 1,000$Hz, respectively. You can use simple *averaging* filters of the form

```
h = [1 1]/2
h = [1 1 1 1]/4
h = [1 1 1 1 1 1 1 1]/8
```

for the three cases, respectively. While these filters are not ideal, they are expected to yield better results than no filtering as in parts **b** and **c**. Discuss why that is. Do you still see the replicas as prominently? Do you hear the same aliasing effects, or do you just hear a lowpass filtered version of the signal?

Question 2: You will now emulate *reconstruction* by increasing the sampling rate to $32,000$Hz, thereby making the signal "less discrete" for the lack of a better word.

a) Using the command `upsample`, insert 3 zeros between each sample. This represents the intermediate signal $x_s(t)$ discussed in class. Recall that this signal needs to be lowpass filtered to eliminate all the unwanted replicas. Plot the DTFT of this signal and observe the unwanted replicas.

b) Convolve this intermediate signal with the filter `h=[1 1 1 1]`. This emulates a *zero-order hold* interpolation filter during reconstruction. Plot the DTFT and observe how efficient was the interpolator in eliminating the unwanted replicas. Listen to the audio. Do you hear a mechanical sound? That is probably because the high frequency content of the unwanted replicas still linger.

c) Repeat **b** with the triangular filter `h=[.25 .5 .75 1 .75 .5 .25]`. This emulates a *first-order hold* interpolation filter during reconstruction. Are unwanted replicas suppressed better? Did the mechanical sound get better?

d) Repeat **b** with the filter `h=sinc(-4:.25:4)`. This emulates a *truncated sinc* filter during reconstruction. Are unwanted replicas suppressed better? Did the mechanical sound get better?