ELG4177 - DIGITAL SIGNAL PROCESSING
Lab8

By:Hitham Jleed

http://www.site.uottawa.ca/~hjlee103/

uOttawa

Assignment #8

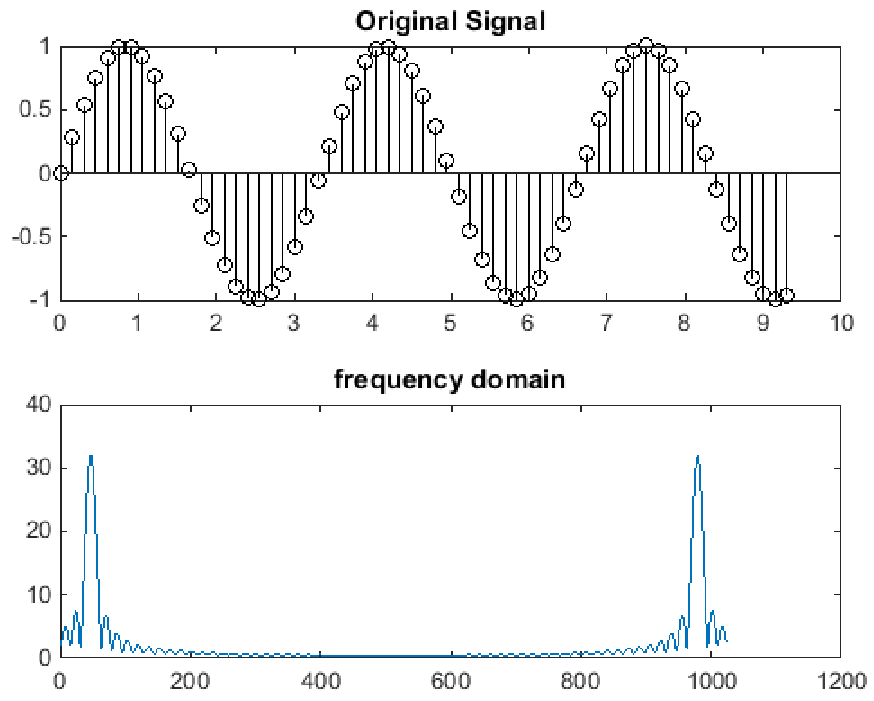
DECIMATION/INTERPOLATION/MULTIRATE SIGNAL PROCESSING

uOttawa

Change of sampling rate

Note: we don't use polyphase filters. Do not use the interp and decimate functions

$$x[n] = \sin(2\pi 0.3n)$$



Up-sampling

$$x[n] \longrightarrow \boxed{\uparrow L} \longrightarrow x_u[n]$$

$$|H(e^{j\omega})| = \begin{cases} L, & |\omega| \leq \omega_c/L, \\ 0, & \pi/L \leq |\omega| \leq \pi. \end{cases}$$

Matlab Operation

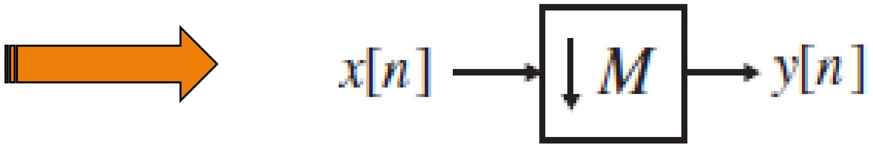
```
x_new = zeros(1, L*length(x_old));
x_new([1: L: end]) = x_old;
```

Matlab Operation

```
b = fir1(50, wc);
X_up = L*filter(b , 1, x_new);
```

Down-sampling

$$|H(e^{j\omega})| = \begin{cases} 1, & |\omega| \le \omega_c/M, \\ 0, & \pi/M \le |\omega| \le \pi. \end{cases}$$



Matlab Operation

```
b = fir1(50, wc);
X_new = filter(b , 1, x);
```

Matlab Operation

```
X_down = x_new([1: M: end]);
```

The two operations together

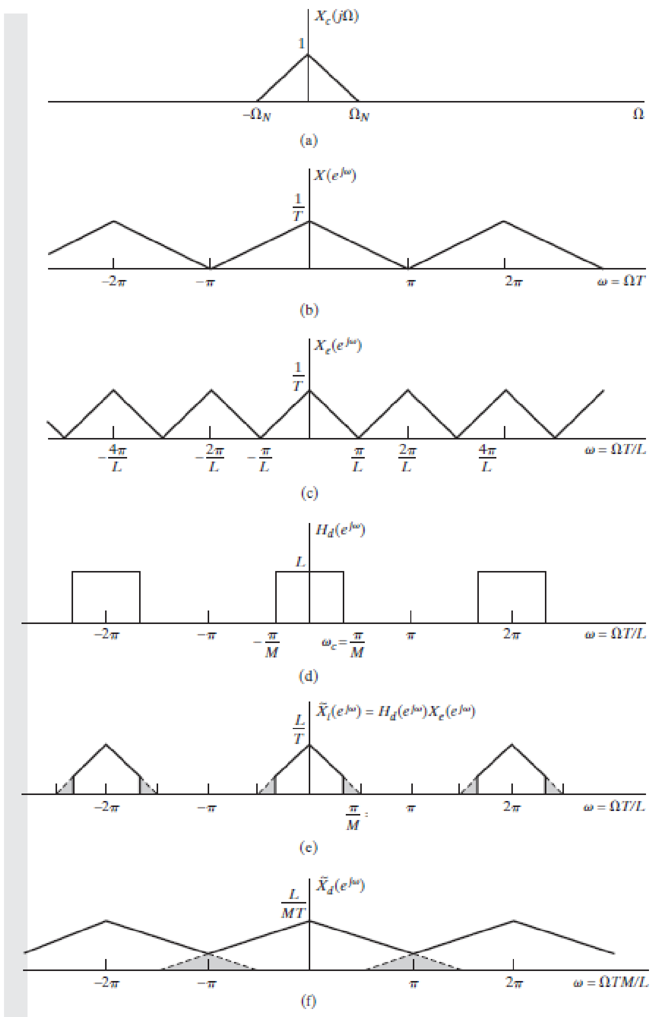


Figure 4.30 Illustration of changing the sampling rate by a noninteger factor.

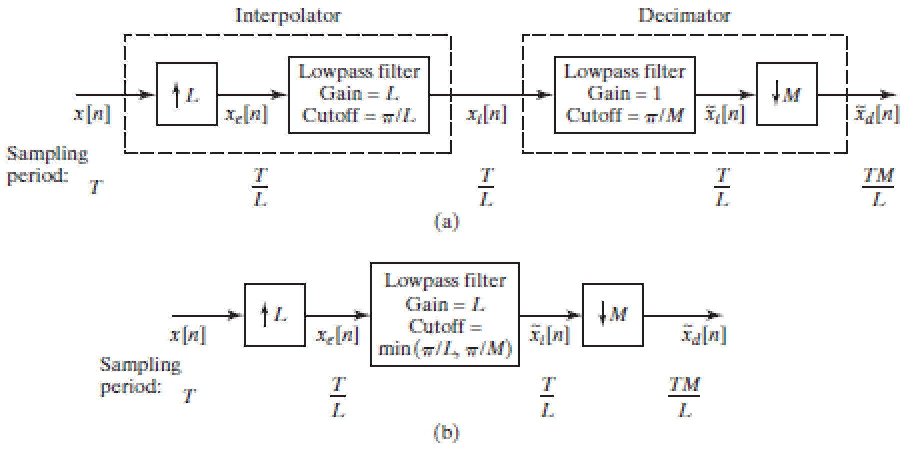


Figure 4.29 (a) System for changing the sampling rate by a noninteger factor. (b) Simplified system in which the decimation and interpolation filters are combined.

$$\omega_s = \min\left(\frac{\pi}{L}, \frac{\pi}{M}\right).$$

Matlab Operation

```
wc = min(1/L, 1/M);
 b = fir1(50, wc);
```

Change of sampling rate

a) It is desired to increase the sampling rate of the signal $x[n] = \sin(2\pi 0.3n)$, by a factor of 3/2, using interpolation and decimation techniques. Which operation should be performed first : interpolation by 3 or decimation by 2 ? (Note: we don't use polyphase filters in this assignment). Does it matter ? Perform the required interpolation and decimation (not necessarily in that order). Do NOT use the *interp* and *decimate* functions because they implicitly perform low-pass filtering. Instead, design your own low-pass filter and do your own filtering in the time domain. Plot the time domain signals and their Fourier transform at the various steps of the interpolation/decimation process.

```
L = 3;   % Upsample rate
M = 2;   % Downsample rate
```

Hint:

```
T = 0.15;     n = 0:T:3*pi;
x = sin(2*pi*0.3*n);
```

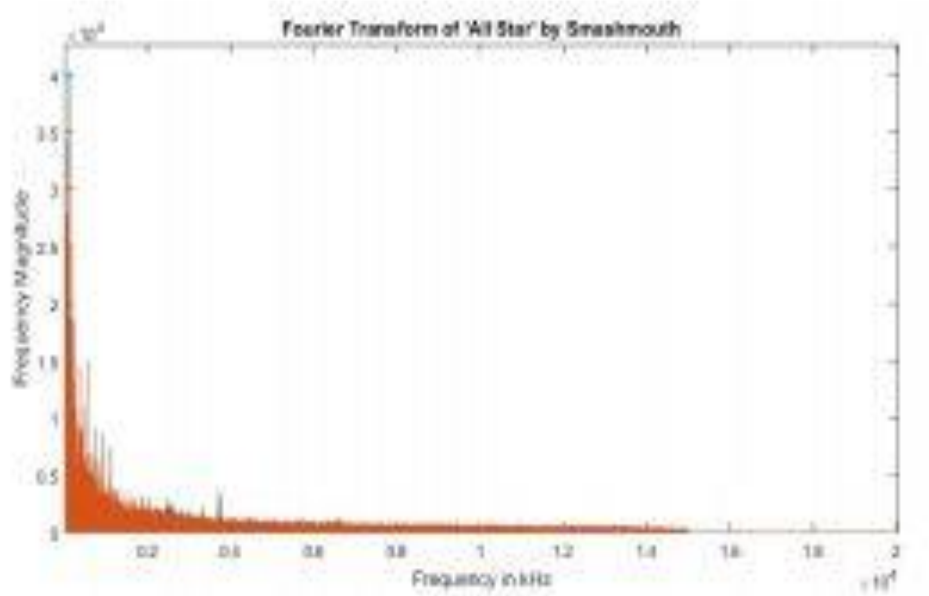
b) For a signal $x[n] = \sin(2\pi 0.2n)$, if the same increase of the sampling rate by a factor of 3/2 is desired, which operation should be performed first : interpolation by 3 or decimation by 2 ? Does it matter ?

```
x = sin(2*pi*0.2*n);
```

c) It is desired to decrease the sampling rate of the signal $x[n] = \sin(2\pi 0.3n)$, by a factor of 2/3, using interpolation and decimation techniques. Which operation should be performed first : interpolation by 2 or decimation by 3 ? Does it matter ? Perform the required interpolation and decimation (not necessarily in that order). Plot the time domain signals and their Fourier transform at the various steps of the interpolation/decimation process.

```
L = 2;   % Upsample rate
M = 3;   % Downsample rate
```

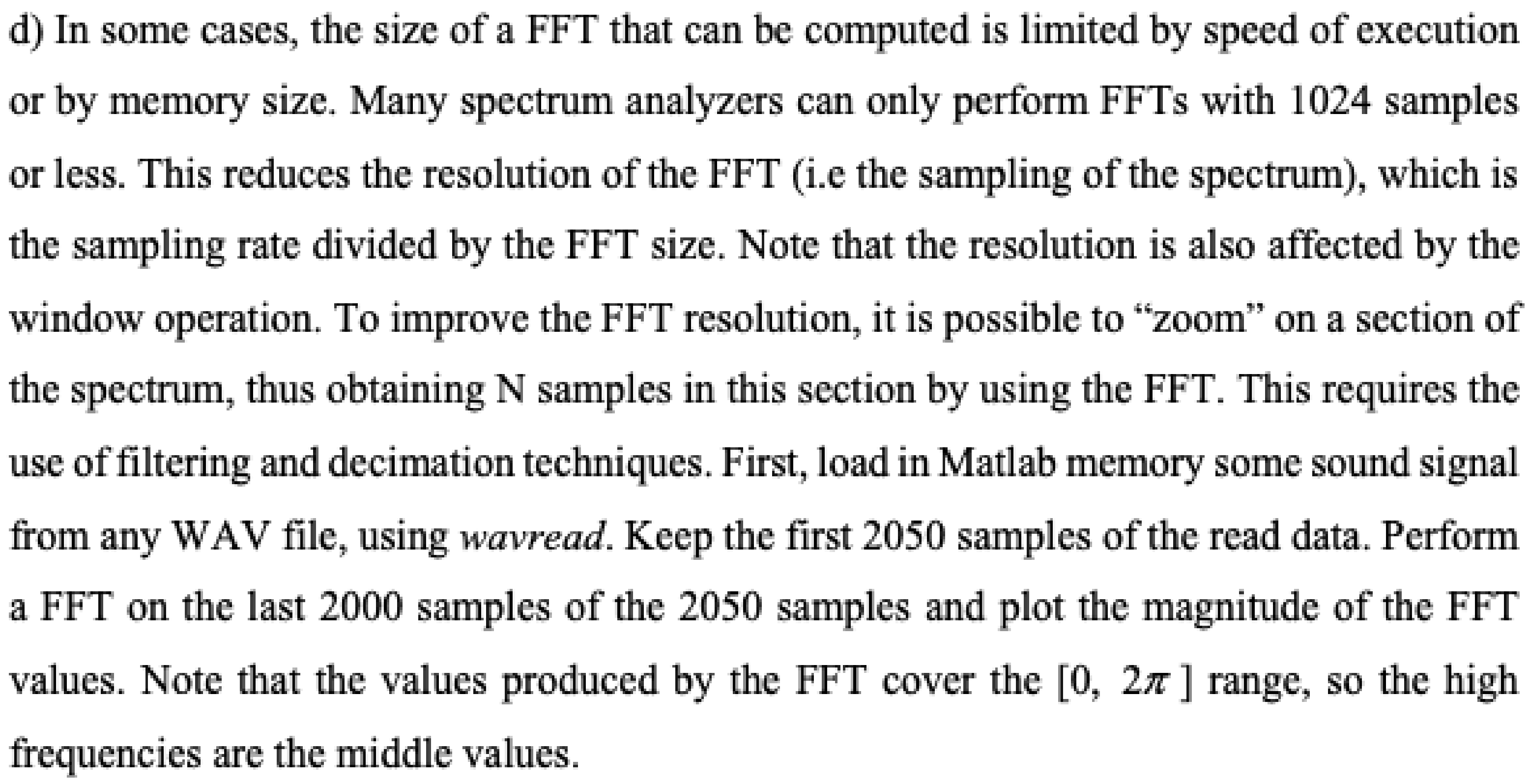
Audio in Matlab



If you need help, see these links:

1. https://www.mathworks.com/help/matlab/import_export/read-and-get-information-about-audio-files.html
2. https://www.mathworks.com/help/matlab/math/basic-spectral-analysis.html
3. https://blogs.umass.edu/Techbytes/2019/02/20/handling-media-files-in-matlab/

Frequency Zoom


d) In some cases, the size of a FFT that can be computed is limited by speed of execution or by memory size. Many spectrum analyzers can only perform FFTs with 1024 samples or less. This reduces the resolution of the FFT (i.e the sampling of the spectrum), which is the sampling rate divided by the FFT size. Note that the resolution is also affected by the window operation. To improve the FFT resolution, it is possible to "zoom" on a section of the spectrum, thus obtaining N samples in this section by using the FFT. This requires the use of filtering and decimation techniques. First, load in Matlab memory some sound signal from any WAV file, using *wavread*. Keep the first 2050 samples of the read data. Perform a FFT on the last 2000 samples of the 2050 samples and plot the magnitude of the FFT values. Note that the values produced by the FFT cover the [0, 2π] range, so the high frequencies are the middle values.

Hint:

```
[y,Fs] = audioread('Lab8.wav', [1,2050]);
Y = fft(y(51:2050));
```

e) Now suppose that the maximum size of FFT that can be used is N=1000, but the same resolution is required as in d), where N=2000 was used. This can be done for different parts of the spectrum, by filtering and decimation. Let's consider the case of the high frequencies ($[\frac{\pi}{2}, \frac{3\pi}{2}]$ interval). Design a half-band highpass FIR filter with a length of 50. Filter the 2050 samples by the FIR filter, and keep the 2000 last values of the output samples.

Multiply the resulting signal by $e^{-j\frac{\pi}{2}n}$. Decimate the resulting signal by a factor of 2, thus keeping only 1000 values. Perform the FFT of those values, and plot the magnitude of the FFT values. Compare with the middle (high frequencies) FFT values found in d). You should get similar values and similar resolutions. Explain analytically (with graphs) the process that was performed. The same process could also be applied to zoom low frequency values. In fact, if the spectrum is divided in D parts, with D integer, a zoom can be performed on any part using filtering, modulation and decimation.

Hint:

```
b = fir1(50,0.5,'high');
y = filter(b,1,x);
```

```
mod = exp(-j*(pi/2)*n');       % Modulation
y = y.*mod;
```

Finish the lab and submit your report

The END