ECS602U Digital Signal Processing Lab 2

Marco Datola

November 13, 2016

Contents

1 Preface

This is the report after the second lab of ECS602U. In the first exercise two sinusoidal signals of frequencies $f_1 = 128Hz$ and $f_2 = 220Hz$ are sampled [1] at a sampling rate of $F_s = 2048Hz$ evaluated over 32 samples. Their the magnitude spectrum is evaluated [2] and further observations for the sampling precision of the second signal are also made [3].
In the second exercise the sonogram of a female voice is plotted [4], with observations on the energy distribution and a discussion on the highest decimation factor with which the signal can be undersampled[5].

[1] relevant code 2.1 , **??**
[2] relevant code 2.2 , **??**
[3] relevant code 2.7 , **??**
[4] relevant code 3.1
[5] relevant code 3.5

2 Are you down with DFT? How about FFT?

2.1 Sampling a real sinusoidal ($f = 128HZ$)

32 samples of a sinusoidal signal with frequency $f = 128Hz$ are sampled at a sampling frequency $F_s = 2048Hz$. The sampled signal is plotted below.

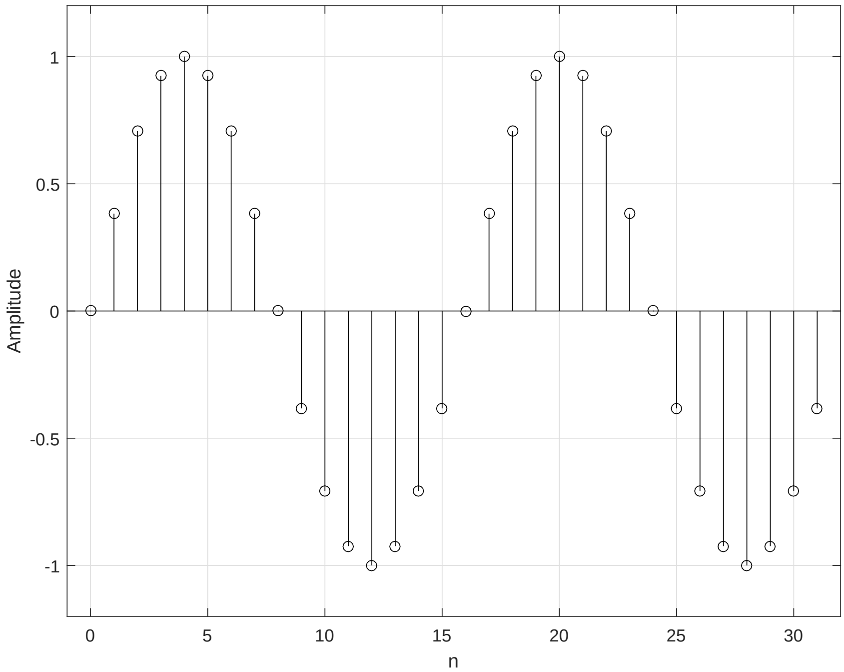

Figure 1: A real 128 Hz sinusoid sampled at 2048Hz over $0 \leq n \leq 31$.

It is achieved using the following Matlab code.

```
a = 1;
f = 128;
Fs = 2048;
w = 2*pi*f;

N = 32;
n = [0:N-1];

x = a*sin(w*n/Fs);
figure;
stem(n,x,'k');
grid on;
axis([-1 N -(a+0.2) (a+0.2)]);
ylabel('Amplitude');
xlabel('n');
print(gcf,'-depsc2','ex11.eps');
```

2.2 Magnitude spectrum: FFT of sinusoidal $f = 128HZ$ signal

To find the DFT of the signal we utilise the FFT and plot the absolute value of its magnitude. To plot against the number of frequency values evaluated by the FFT set $n_f = \frac{F_s}{N} n_s$. The spectogram of the sinusoidal signal is shown below.

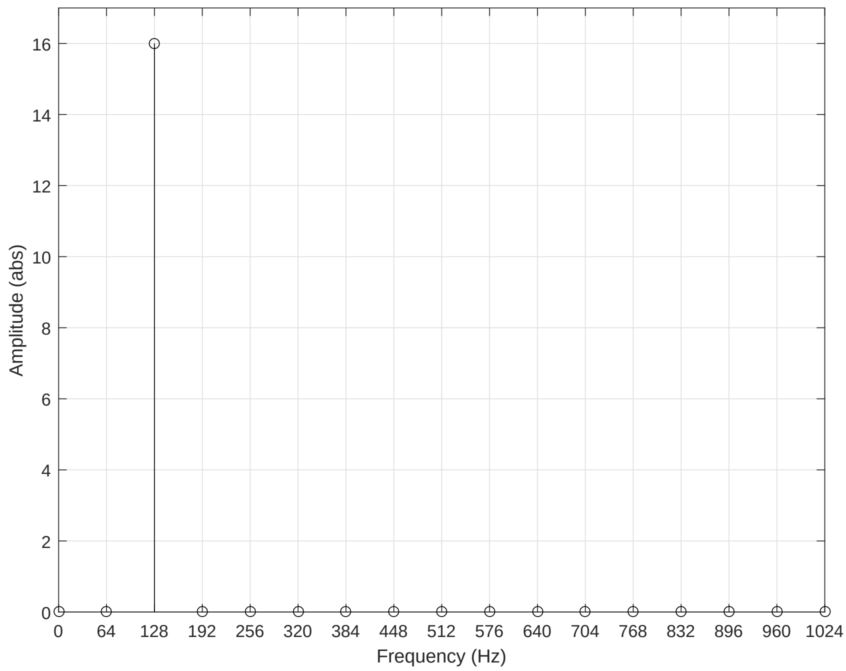

Figure 2: Magnitude spectrum of a real 128 Hz sinusoid sampled at 2048Hz over $0 \leq n \leq 31$.

It is achieved using the following Matlab code.

```

a = 1;
f = 128;
Fs = 2048;
w = 2*pi*f;

N = 32;
n = [0:N-1];

x = a*sin(w*n/Fs);
mfft = fft(x);
k = Fs/N*n;
figure;
stem(k,abs(mfft),'k');
grid on;
axis([0 Fs/2 0 max(abs(mfft))+1]);
set(gca, 'Xtick', [0:Fs/N:Fs/2]);
ylabel('Amplitude (abs)');
xlabel('Frequency (Hz)');
print(gcf,'-depsc2','ex12.eps');
```

2.3 Phase shift

Given we are analysing a real signal, the angle (phase shift/delay) there will be between the cosinusoidal (real) component of the FFT at $f = 128Hz$ and the sinusoidal signal will be $\frac{-\pi}{2}$. This is visible in the below figure.

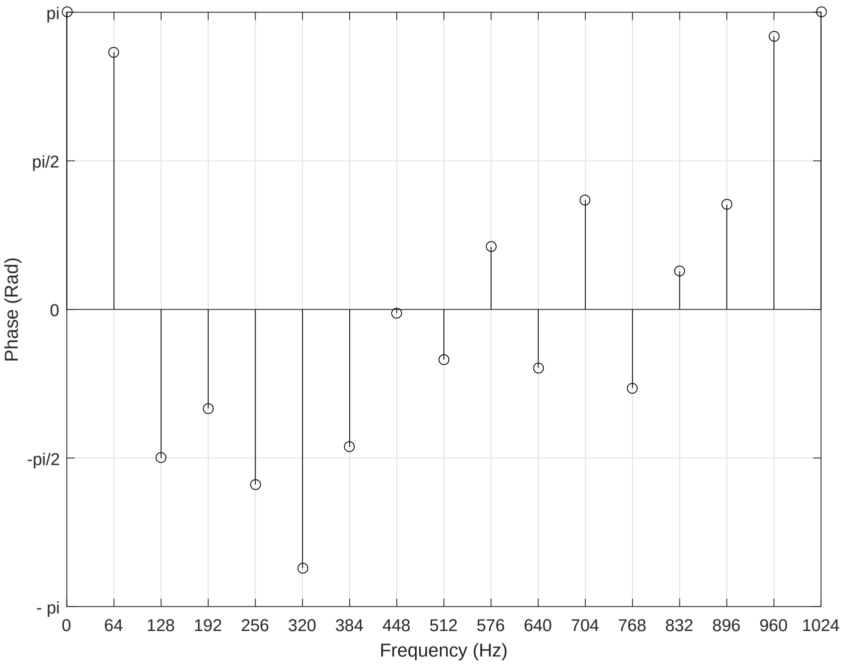

Figure 3: Phase shift of a 128 Hz sinusoid sampled at 2048Hz over $0 \leq n \leq 31$.

It is achieved using the following Matlab code.

```matlab
a = 1;
f = 128;
Fs = 2048;
w = 2*pi*f;

N = 32;
n = [0:N-1];

x = a*sin(w*n/Fs);
mfft = fft(x);
k = Fs/N*n;
ang = angle(mfft);

figure;
stem(k,ang,'k');
grid on;
axis([0 Fs/2 -pi pi]);
```

```
set(gca, 'Xtick', [0:Fs/N:Fs/2], ...
        'Ytick',[-pi:pi/2:pi],'Yticklabel',{'- pi','-pi/2', '0', 'pi/2', 'pi'});
ylabel('Phase (Rad)');
xlabel('Frequency (Hz)');
print(gcf,'-depsc2','ex13.eps');
```

2.4 Sampling a real sinusoidal ($f = 220HZ$)

32 samples of a sinusoidal signal with frequency $f = 220Hz$ are sampled at a sampling frequency $F_s = 2048Hz$. The sampled signal is plotted below.

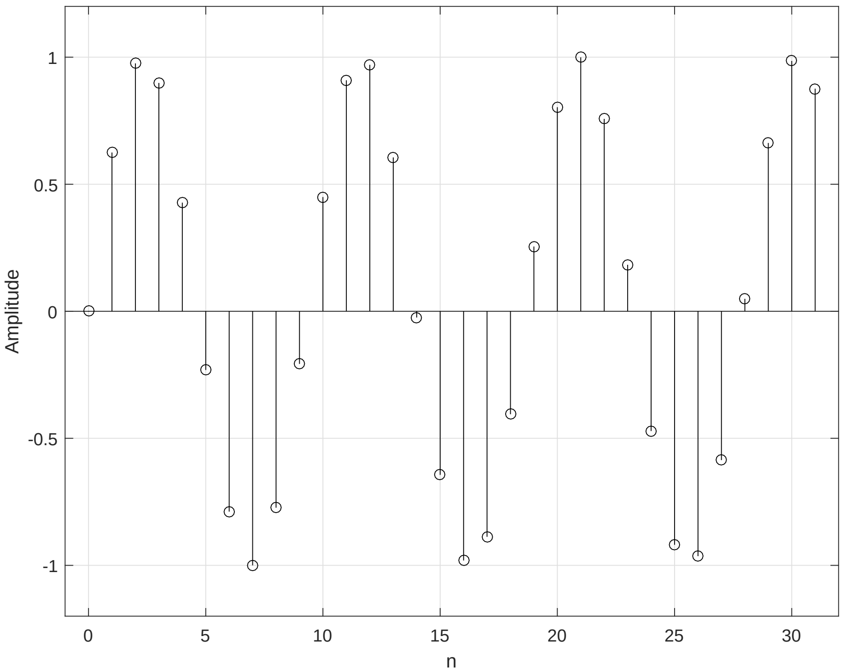

Figure 4: A real 220 Hz sinusoid sampled at 2048Hz over $0 \leq n \leq 31$.

2.5 Magnitude spectrum hypothesis

Given this sinusoidal signal has a sole real sinusoidal component, the magnitude spectrum is expected to be zero for all frequencies except than at $f = 220Hz$. This is show in the below figure.

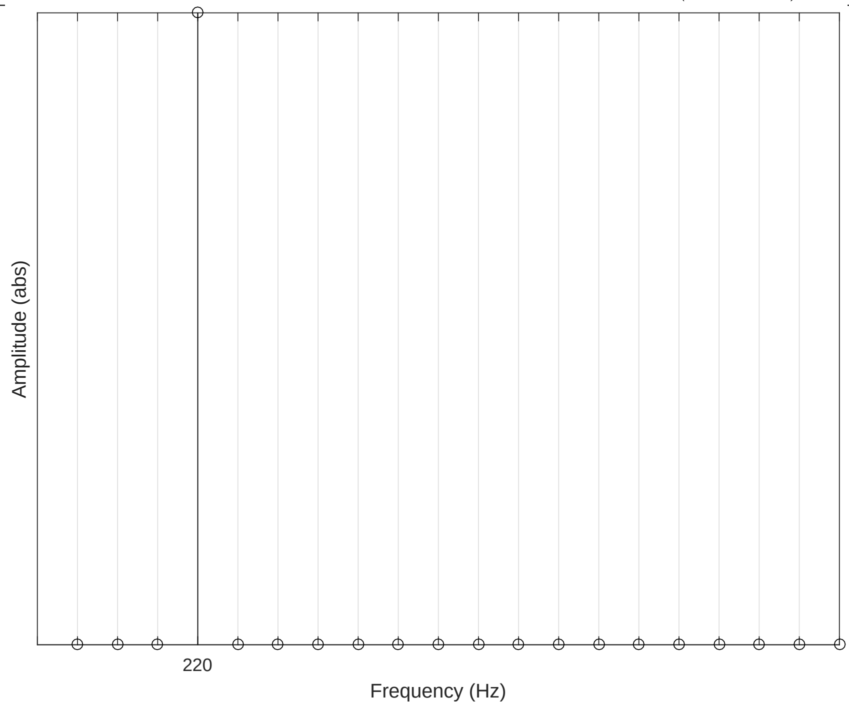

Figure 5: Hypothesis of the magnitude spectrum of a real 220 Hz sinusoid sampled at 2048Hz over $0 \leq n \leq 31$.

2.6 Magnitude spectrum: FFT of a sinusoidal $f = 220HZ$ signal $(N_s = 32)$

To find the DFT of the signal we utilise the FFT and plot the absolute value of its magnitude. To plot against the number of frequency values evaluated by the FFT set $n_f = \frac{F_s}{N} n_s$. The spectogram of the sinusoidal signal is shown below.

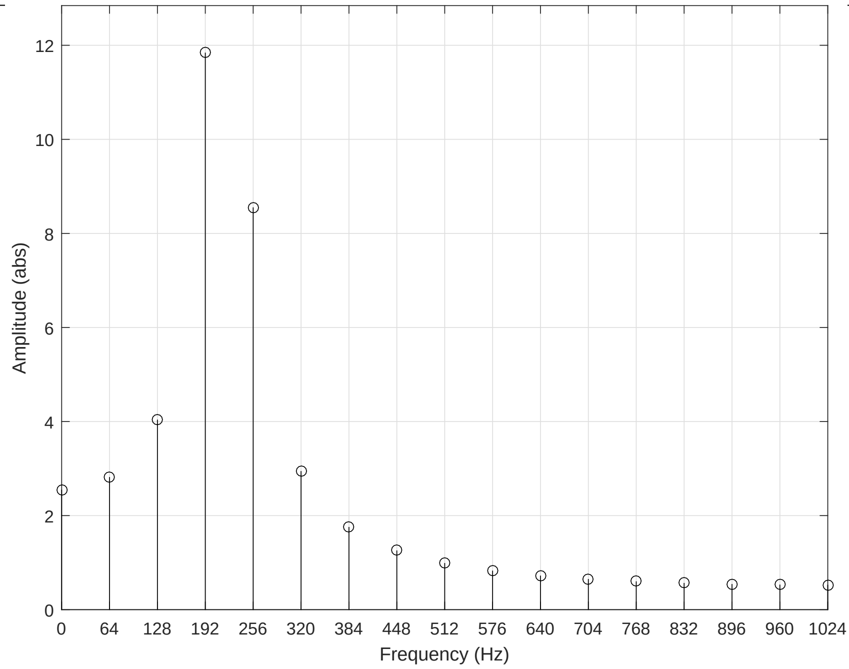

Figure 6: Magnitude spectrum of a real 220 Hz sinusoid sampled at 2048Hz over $0 \leq n \leq 31$.

The spectrogram appears to have a different trend compared to 3, hence the hypothesis 5 is wrong. Instead of showing only the sinusoidal component of the real signal ($220Hz$), the spectrogram in figure shows all the components of the FFT are contributing towards its discrete representation.

2.7 Signal reconstruction: IFFT

The IFFT function is utilised to reconstruct the real signal from the complex phasors the FFT returns. The result is show below.

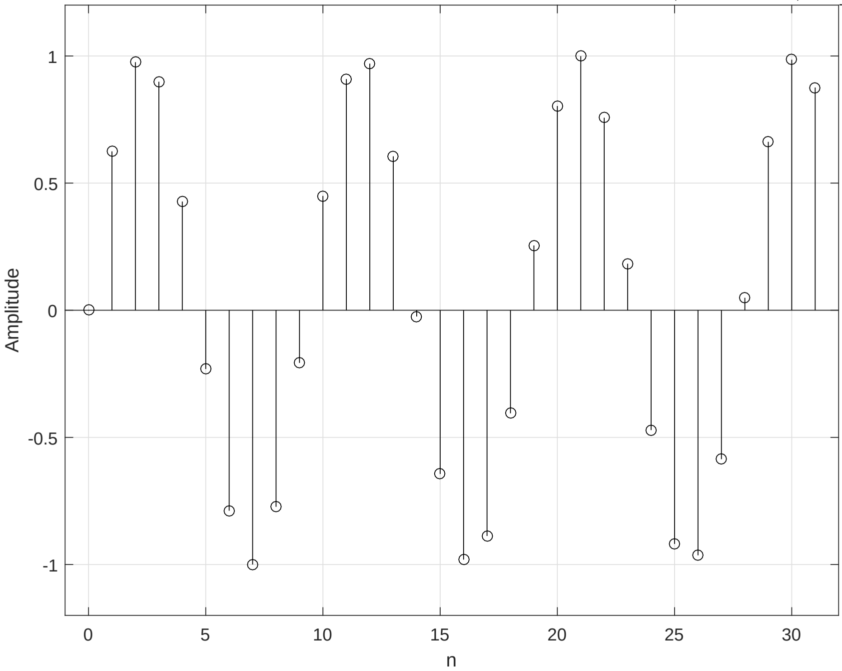

Figure 7: Signal reconstruiction: IFFT of the complex phasors representing a real $f = 220HZ$ sinusoid sampled at 2048Hz over $0 \leq n \leq 31$.

As expected it corresponds to figure 4.
It is achieved using the following Matlab code.

```
a = 1;
f = 220;
Fs = 2048;
w = 2*pi*f;

N = 32;
n = [0:N-1];

x = a*sin(w*n/Fs);
mfft = fft(x);
mifft = ifft(mfft);
k = Fs/N*n;
figure;
stem(n,mifft,'k');
grid on;
axis([-1 N -(a+0.2) (a+0.2)]);
ylabel('Amplitude');
xlabel('n');
print(gcf,'-depsc2','ex17.eps');
```

2.8 Magnitude spectrum: FFT of a sinusoidal $f = 220HZ$ signal ($N_s = 512$)

The spectogram of the sinusoidal signal is shown below.

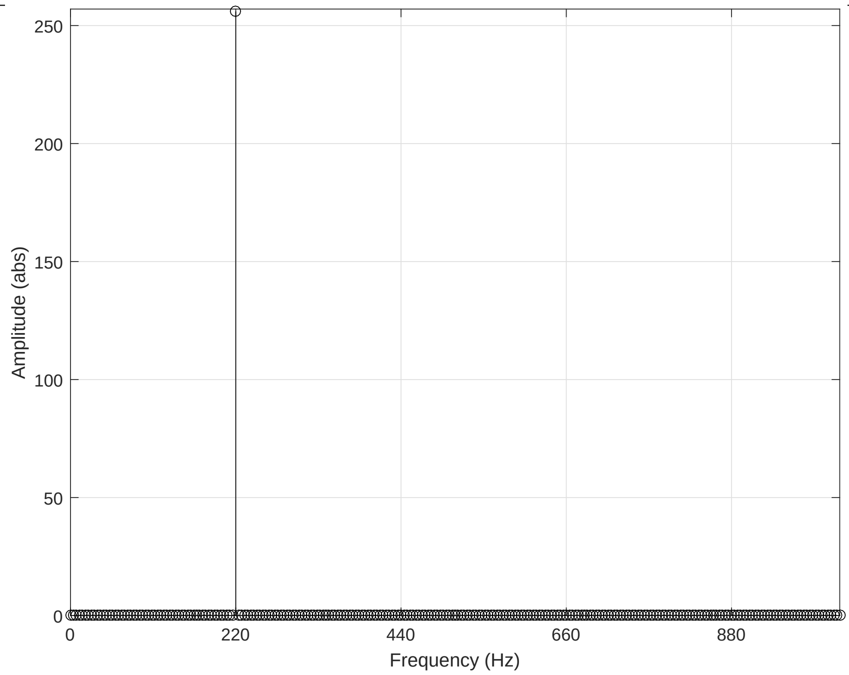

Figure 8: Magnitude spectrum of a real 220 Hz sinusoid sampled at 2048Hz over $0 \leq n \leq 511$.

3 Under-sampling and Aliasing

3.1 Sonogram of a woman speaking

Comments of the provided code are presented.

```
clear all;
sndfile = 'speech_female.wav';
% stores sampled audio file in x
% and sampling frequency in Fs
[x,Fs] = audioread(sndfile);
% number of samples per window
N = 512;
% stores the spectrogram achieved via STFT
% a vector F of the frequencies of the STFT's phasors
% a vector T of the time at which the STFT is performed
% %
% the 1st argument specifies only on 1.4s
% of the sampled data the STFT will be performed
% %
% the 2nd argument sets the widow size to be
% equal to the number of samples defined above
% %
% the 3rd argument sets the overlap between
% the windows equal to 3/4 of their size
% this means every window we progress of N/4 samples
% %
```

```
% the 4th argument sets the frequencies for which
% the DFT is performed to N&4
% %
% the 5th argument specifies the sampling frequency
[S,F,T] = spectrogram(x(1:Fs*1.4),N,3*N/4,N*4,Fs);
% styling and positioning of figure
f = figure('Position',[500 300 700 500],'MenuBar','none', ...
'Units','Normalized');
set(f,'PaperPosition',[0.25 1.5 8 5]);
axes('FontSize',14);
colormap('jet');
% displays the power of each frequency component
% at the different time instants
% %
% the 1st argument provides the time vector
% %
% the 2nd argument provides the frequencies in KHz
% %
% the 3rd argument evaluates the spectral power
% of the signal, at each time instant
imagesc(T,F./1000,20*log10(abs(S)));
% stylinmg of figure
axis xy;
set(gca,'YTick',[0:2000:Fs/2]./1000,'YTickLabel',[0:2000:Fs/2]./1000);
ylabel('Frequency (kHz)');
xlabel('Time (s)');
print(gcf,'-depsc2','ex21.eps');
```

The sonogram produced by this code is visible in the below figure.

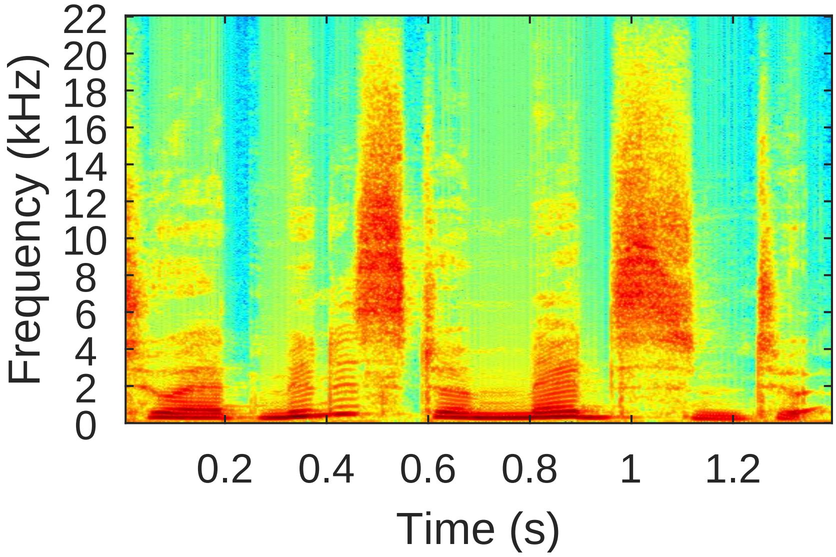

Figure 9: A sonogram of a woman speaking.

3.2 Considerations: Distribution of energy and minimum sampling frequency

The sonogram shows us most of the energy is located between 0 and 6 kHz. Hence, applying the Nyquist-Shannon ($F_s = 2f_{max}$) theorem we would be able to reconstruct the signal, not without errors, sampling it at a frequency of approx $f_{min} = 12kHz$.

3.3 Considerations: Sounds with high frequency content

Given the abundance of high frequency components (and lack of low frequency components), the sounds between 4 and 18 kHz coincide with 'st' in 'administer' and 'ci' in 'medicine'.

3.4 Evaluation of highest decimation factor

Having determined the minimum sampling frequency from the Nyquist theorem $f_{min} = 12kHz$, the decimation factor that should be adopted is equivalent to $dec = \frac{F_s}{f_{min}} = \frac{44100}{12000} = 3.675 \rightarrow dec = 4$.

3.5 Evaluation of highest decimation factor

Below the code to downsample the waveform and hear the results. I observed the speech was clear down to a decimation factor of 25, this means sampling at $f_{min} = 4410025 = 1764Hz$. I'm not sure what to think about this result?

```
dec = 25;
sndfile = 'speech_female.wav';
[x,Fs] = audioread(sndfile);
sound(x(1:dec:end),Fs/dec);
```

3.6 Optimizing animal care

Given the speech was understandabel at such with such a low sampling frequency, it's probable animals could be cured mixing less medicine..... with more butter!