BME I5100: Biomedical Signal Processing

Linear systems, Fourier and z-Transform

Lucas C. Parra
Biomedical Engineering Department
City College of New York

CCNY
Schedule

**Week 1: Introduction**
Linear, stationary, normal - the stuff biology is **not** made of.

**Week 1-4: Linear systems** (mostly discrete time)
Impulse response
Moving Average and Auto Regressive filters
Convolution
**Discrete Fourier transform and z-transform**

**Week 5-7: Random variables and stochastic processes**
Random variables
Multivariate distributions
Statistical independence

**Week 8: Electrophysiology**
Origin and interpretation of Biopotentials

**Week 9-14: Examples of biomedical signal processing**
Probabilistic estimation
Linear discriminants  - detection of motor activity from MEG
Harmonic analysis  - estimation of hart rate in Speech
Auto-regressive model  - estimation of the spectrum of 'thoughts' in EEG
Independent components analysis  - analysis of MEG signals
z-Transform

The $z$-transform is a map of a discrete signal $x[n]$ into complex valued function $X(z)$ with complex valued $z$

$$X(z) = \sum_{n=-\infty}^{\infty} x[n] z^{-n}$$

For a **finite** sequence the $z$-transform is defined for all $z$.

For **infinite** sequence it is only defined for values of $z$ for which the sum converges.

$z$-transform is important because of the **Convolution Theorem**:

$$h[n] \ast x[n] \Leftrightarrow H(z) X(z)$$
**z-Transform - Convolution theorem**

\[ h[n] * x[n] \Leftrightarrow H(z) \cdot X(z) \]

Because the \( z \)-transform of the convolution ...

\[
\sum_{n=-\infty}^{\infty} h[n] \ast x[n] z^{-n} =
\]

\[
= \sum_{n=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} h[k] x[n-k] z^{-n}
\]

\[
= \sum_{n'=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} h[k] x[n'] z^{-n'-k}
\]

\[
= \sum_{k=-\infty}^{\infty} h[k] z^{-k} \sum_{n'=-\infty}^{\infty} x[n'] z^{-n'}
\]

\[= H(z) \cdot X(z) \]
**Z-Transform - Difference model**

We approximated the response of an IIR system by the difference model

\[ y[n] = -\sum_{k=1}^{P} a[k] y[n-k] + \sum_{k=0}^{Q} b[k] x[n-k] \]

The \( z \)-transform relates the impulse response \( h[k] \) to \( a[k] \) and \( b[k] \). Define \( a[0]=1 \), then we can rewrite:

\[ \sum_{k=0}^{P} a[k] y[n-k] = \sum_{k=0}^{Q} b[k] x[n-k] \]

After \( z \)-transform on both sides we have

\[ A(z) Y(z) = B(z) X(z) \]
**z-Transform - rational impulse response**

With the z-transform of \( y[n] = h[n] * x[n] \):

\[
Y(z) = H(z) X(z)
\]

We obtain

\[
B(z) X(z) = A(z) Y(z) = A(z) H(z) X(z)
\]

\[
H(z) = \frac{B(z)}{A(z)}
\]

Impulse response is called **rational** if it can be expressed exactly as:

\[
H(z) = \frac{\sum_{k=0}^{Q} b[k] z^{-k}}{1 + \sum_{k=1}^{P} a[k] z^{-k}}
\]

With a sufficient number of terms any \( H(z) \) can be approximated by a rational **polynomial**.
The z-transform of an ARMA filter and its inverse are

$$Y(z) = \frac{B(z)}{A(z)} X(z) \quad X(z) = \frac{A(z)}{B(z)} Y(z)$$

The AR part becomes the MA and the MA becomes AR:

Since in general $b[0] \neq 1$ we have to write:

$$b[0] x[t] = - \sum_{k=1}^{P} b[k] x[n-k] + \sum_{k=0}^{P} a[k] y[n-k]$$
z-Transform -rational impulse response

Note that any polynomial of order -Q can be decomposed into Q factors of the form, \((1-z_k z^{-1})\), with roots \(z_k\).

\[
\sum_{k=0}^{Q} b[k] z^{-k} = b[0] \prod_{k=1}^{Q} (1 - z_k z^{-1})
\]

Example:
Factor representation of square polynomial with roots \(z_1\) and \(z_2\)

\[
(z - z_1)(z - z_2) = z^2 - z(z_1 + z_2) + z_1 z_2
\]

\[
(1 - z_1 z^{-1})(1 - z_2 z^{-1}) = 1 - (z_1 + z_2) z^{-1} + z_1 z_2 z^{-2}
\]

Coefficients of the second order polynomial are then: \(b[0] = 1, b[1] = -z_1 - z_2, b[2] = z_1 z_2\)

For \(b[k] \rightarrow z_k\) use

\[
\text{>> } z = \text{roots}(b);
\]

For \(z_k \rightarrow b[k]\) use

\[
\text{>> } b = \text{poly}(z);
\]
z-Transform - Pole-zero representation of IR

Since $a[k]$ and $b[k]$ define polynomials in $z^{-1}$ this lead to the pole-zero representation of the impulse response:

$$H(z) = \frac{B(z)}{A(z)} = b[0] \frac{\prod_{k=1}^{Q} (1-z_k z^{-1})}{\prod_{k=1}^{P} (1-p_k z^{-1})}$$

For $P=0$ the system is called **all zeros**, for $Q=0$ it is called **all poles**.

See Bruce, page 296 for dependence of $h[n]$ on location of poles.

$$\text{>> plot(filter(b(1)*poly(z),poly(p),[1 zeros(1,100)]))}$$

A rational $H(z)$ is **stable** if the poles $|p_k| < 1$. Hence stability test:

$$\text{>> abs(roots(a))<1}$$

See Bruce for proof on page 450.
z-Transform

Homework: Poles location and Impulse response

Plot the impulse response for filters with the following **configuration of poles** (see below figure for definition of quadrants)
1. TWO conjugate poles, one in quadrant I and the conjugate in quadrant IV.
2. TWO conjugate poles, one in quadrant II and the conjugate in quadrant III.
3. ONE pole in quadrant I.
4. TWO conjugate poles outside of the unit circle.

The final output of your homework should be 4 figures, each figure should be the impulse response of each pole configuration tried.

For the **zeros**, you can try any combination you want.
Discrete Time Fourier Transform (DTFT)

The **Discrete Time Fourier Transform** (DTFT) is the z-transform on the unit circle $z = e^{j\omega} = \cos(\omega) + j\sin(\omega)$

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n] e^{-j n \omega}$$

The DTFT is an invertible transformation

$$x[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} d\omega X(e^{j\omega}) e^{j n \omega}$$

This is simply because

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} d\omega e^{-j \omega n} = 2\pi \delta_n$$

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} d\omega X(e^{j\omega}) e^{j n \omega} = \frac{1}{2\pi} \sum_{k=-\infty}^{\infty} x[k] \int_{-\pi}^{\pi} d\omega e^{-j \omega(k-n)} = x[n]$$
# Fourier Transform Summary

<table>
<thead>
<tr>
<th>Continuous Time Fourier Transform (cont. time, cont. freq.)</th>
<th>Discrete Time Fourier Transform (disc. time, cont. freq.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ F(\nu) = \int_{-\infty}^{\infty} dt , f(t) , e^{-j2\pi \nu t} ]</td>
<td>[ X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n] , e^{-jn\omega} ]</td>
</tr>
<tr>
<td>[ f(t) = \int_{-\infty}^{\infty} d\nu , F(\nu) , e^{j2\pi \nu t} ]</td>
<td>[ x[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} d\omega , X(e^{j\omega}) , e^{jn\omega} ]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Continuous Time Fourier Series (cont. time, disc. freq.)</th>
<th>Discrete Fourier Transform (disc. time, disc. freq.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ a_k = \frac{1}{2T} \int_{-T}^{T} dt , x(t) , e^{-jk\omega t} ]</td>
<td>[ X[k] = \sum_{n=0}^{N-1} x[n] , e^{-j2\pi kn/N} ]</td>
</tr>
<tr>
<td>[ x(t) = \sum_{k=-\infty}^{\infty} a_k , e^{jk\omega t} ]</td>
<td>[ x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] , e^{j2\pi kn/N} ]</td>
</tr>
</tbody>
</table>
The following properties derive directly from the z-transform

\[ x[n] \quad \text{DTFT} \quad X(e^{j\omega}) \]

**Conjugation**
\[ x^*[n] \quad X^*(e^{-j\omega}) \]

**Delay**
\[ x[n-n_0] \quad e^{-j\omega n_0}X(e^{j\omega}) \]

**Time reversal**
\[ x[-n] \quad X(e^{-j\omega}) \]

**Correlation**
\[ \sum_{n=-\infty}^{\infty} x[n+k]y^*[n] \quad X(e^{j\omega})Y^*(e^{j\omega}) \]

Conjugate symmetry for \( x[n] \in \mathbb{R} \)
\[ X(e^{j\omega}) = X^*(e^{-j\omega}) \]
DTFT - System frequency response

Consider stationary oscillatory input to a LSI system \( h[k] \):

\[
x[n] = e^{jn\omega}
\]

\[
y[n] = \sum_{k=-\infty}^{\infty} h[k] x[n-k] = \sum_{k=-\infty}^{\infty} h[k] e^{j(n-k)\omega} = H(e^{j\omega}) e^{jn\omega}
\]

The output is the input times the DTFT of the impulse response

\[
e^{jn\omega} \quad \boxed{H(e^{j\omega})} \quad H(e^{j\omega}) e^{jn\omega}
\]

The oscillation with frequency \( \omega \) has been modified in phase \( \Phi \) and amplitude \( A \)

\[
A(\omega) = |H(e^{j\omega})| \quad \Phi(\omega) = \text{arg}(H(e^{j\omega}))
\]

\[
H(e^{j\omega}) = Ae^{j\Phi}
\]
DTFT - System frequency response

DTFT inversion formula tells us that arbitrary input $x[n]$ can be decomposed into sum of oscillations

$$x[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} d\omega \ X(e^{j\omega}) e^{jn\omega}$$

The system response to that is given by the convolution theorem

$$Y(e^{j\omega}) = H(e^{j\omega}) X(e^{j\omega})$$
DTFT - System frequency response

Example: $a[1] = -0.9, \ b[0] = 0.1$:

$$H(z) = \frac{0.1}{1 - 0.9z^{-1}}$$

Time domain response

```python
>> plot(filter(b,a,u))
```

Frequency domain response

```python
>> freqz(b,a)
```
DTFT - Filter design example

Goal: Highpass filter the DC drift in EEG with minimal latency and memory.

>> sptool

Solution: Select IIR filter, 2\textsuperscript{nd} order Butterworth with cutoff at 1Hz.

Downside to DC removal is long phase delay for eye blinks.
DTFT – Group delay

Phase delay $\varphi(\omega)$ corresponds to delay in time, $\Delta t$, that is frequency dependent:

Group delay in samples

$$n(\omega) = \frac{\varphi(\omega)}{\omega}$$

Group delay in time

$$\Delta t(\omega) = \frac{n(\omega)}{f_s}$$

$$f = \frac{\omega}{2\pi f_s}$$

To avoid phase distortions it is desirable to have a constant group delay, i.e. all frequency components are delayed by the same amount in time. This can be achieved with a linear phase:

$$n(\omega) = \frac{n_o \omega}{\omega} = n_o$$
DTFT - Zero phase and linear phase

A filter is said to have zero phase if it introduces no phase delay

$$H(e^{j\omega}) = |H(e^{j\omega})|e^{j0} = |H(e^{j\omega})|$$

True for all symmetric FIR filters: $$h[-n] = h^*[n]$$

A filter is said to have linear phase if

$$H(e^{j\omega}) = |H(e^{j\omega})|e^{j\omega n_0}$$

Linear phase corresponds to shift in time. Because delay in time corresponds to a multiplication with a linear phase term:

$$x_{n_0} = x[n + n_0] \Rightarrow X_{n_0}(e^{j\omega}) = e^{j\omega n_0} X(e^{j\omega})$$

$$Y(e^{j\omega}) = |H(e^{j\omega})|e^{j\omega n_0} X(e^{j\omega}) = |H(e^{j\omega})| X_{n_0}(e^{j\omega})$$
DTFT - Linear Phase

• A filter with linear phase delays all frequencies by the same amount.
• If we add a constant delay to a zero phase filter we obtain a linear phase filter.
• The shift in time can be removed if the filter can be non-causal. In which case we get a zero phase filter.

Example: delay here 11 samples
DTFT - Pragmatic filter design example

If the filter is allowed to be non causal a **quick fix** is to apply the filter to the time inverted signal resulting in a **zero phase** filter:

\[ Y(e^{j\omega}) = |H(e^{j\omega})|^2 X(e^{j\omega}) \]

```
>> filtfilt(b,a,x)
```
Assignment 4:
Use `sptool()` to design 10Hz bandpass filter (5Hz bandwidth) to measure the power of alpha activity in EEG.
1. IIR filter of low order AND
2. FIR with linear phase
Show corresponding impulse response, magnitude and phase response.
Apply this to the EEG signal posted on the web site (EEG.txt). Show signal before and after filtering.
After finding a good filter take note of the parameters, and use the corresponding filter design function to compute the ARMA filter coefficients. To find the corresponding filter function use for instance:
`>> lookfor chebyshev`
3. Apply filter to all EEG channels and display power in the 10Hz band.

Read Kalouptsidis: 8.8.1 Butterworth filter, 8.8.2. Chebyshev filter, FIR 8.8.4,
Read Bruce: 8.17 FIR filter, 8.19 Biomedical applications
Gabor Quadrature Pair Filter (Morlet wavelets)

A pair of filters (real and complex values) that has compact support in time and frequency domain and $90^\circ$ phase delay – also know as Morlet wavelets:

$$h(t) = \exp(j\omega_0 t) \exp(-\frac{t^2}{2\Delta t^2})$$

$$H(e^{j\omega}) \propto \exp\left(-\frac{(\omega - \omega_0)^2}{2\Delta \omega^2}\right)$$

Time-Frequency Product: $\Delta t \Delta \omega = \frac{1}{2}$

(optimal value for this filter)
Gabor Quadrature Pair Filter (Morlet wavelets)

\[ Q = \frac{\omega}{\Delta \omega} \] (determines number of oscillations)

\[ f = 30\text{Hz} \]
\[ f_s = 250\text{Hz} \]
\[ Q=1 \]
Gabor Quadrature Pair Filter (Morlet wavelets)

\[
\begin{align*}
\text{fs} &= 250; \quad \text{% sampling rate in Hz} \\
\text{fc} &= 10; \quad \text{% center frequency in Hz} \\
\text{Q} &= 1; \quad \text{% Q-factor = f/df;} \\
\text{df} &= \frac{\text{fc}}{\text{Q}}; \quad \text{% bandwidth in Hz,} \\
\text{dt} &= \frac{1}{\text{df}}; \\
\text{t} &= \left(-3\times\text{dt}\times\text{fs}:3\times\text{dt}\times\text{fs}\right)/\text{fs}; \\
\text{b} &= \frac{1}{\sqrt{\pi/2}}\times\frac{\text{fs}}{\text{dt}}\times\exp\left(-\frac{t^2}{2\times\text{dt}^2}\right)\times\exp\left(\sqrt{-1}\times2\pi\times\text{fc}\times t\right); \\
\end{align*}
\]

\[
\begin{align*}
\text{subplot}(2,1,1); \\
\text{plot}(\text{t},[\text{real(\text{b}) imag(\text{b})}]); \text{ axis tight} \\
\text{title('Temporal filter'); xlabel('time (s)');} \\
\text{fbin} &= \left(0:\text{length(\text{b})-1}/\text{length(\text{b})}\times\text{fs}\right); \\
\text{subplot}(2,2,3); \\
\text{plot}(\text{fbin},\log(\text{abs}((\text{fft(\text{[real(\text{b}) imag(\text{b})])]})))); \text{ axis tight} \\
\text{xlim}([0 \text{ fs/2}]); \text{ ax=axis; ylim([-80 ax(4)]); grid on} \\
\text{title('Magnitude response'); xlabel('freq (Hz)'); ylabel('Gain (dB)');} \\
\text{subplot}(2,2,4); \\
\text{plot}(\text{fbin},180/\pi\times\text{angle}(\text{fft(\text{ifftshift([real(\text{b}) imag(\text{b})])]})))); \text{ axis tight} \\
\text{xlim}([0 \text{ fs/2}]); \text{ ax=axis; ylim([-180 180]); grid on} \\
\text{title('Phase response (excluding delay)'); xlabel('freq (Hz)'); ylabel('Phase (deg)');}
\end{align*}
\]
Assignment

Create a bandpass filter for alpha activity (~10 Hz). Filter the data and estimate at what times the person has their eyes open and closed (alpha activity is stronger when the eyes are closed). The output of the homework should be a figure with the following panels.

1) PSD before and after filtering in one figure.
2) Magnitude response of the filter
3) Phase response of the filter
4) Show the instantaneous amplitude and instantaneous frequency.
5) Signal before and after filtering with vertical lines indicating the moments where the eyes are open or closed.

The data is online stored as alpha.mat
Analytic signal

For a band-passed signal $y(t)$ one can estimate an instantaneous power, phase and frequency from the “analytic signal”:

$$y_a(t) = y(t) + i \frac{1}{\pi t} \ast y(t)$$

The convolution in the second (complex) term is known at the Hilbert transform:

$$H[y(t)] = \frac{1}{\pi t} \ast y(t) = \frac{1}{\pi} \int_{-\infty}^{\infty} dt' \frac{1}{t} y(t - t')$$
Instantaneous power, phase, frequency
Instantaneous power, phase, frequency

Instantaneous power: \[ P(t) = |y_a(t)|^2 \]

Instantaneous phase: \[ \Phi(t) = \arg(y_a(t)) \]

Instantaneous frequency: \[ \omega(t) = \frac{\partial \Phi(t)}{\partial t} \]
Discrete Fourier Transform (DFT)

The **Discrete Fourier Transform** (DFT) is the **sampled** DTFT

\[
X \left( e^{j\omega} \right) = \frac{2\pi k}{N} = X[k]
\]

The **N-point DFT** is defined for a signal of length \( N \): (analysis)

\[
X[k] = \sum_{n=0}^{N-1} x[n] e^{-j 2\pi kn/N}
\]

with **inverse N-point DFT**: (synthesis)

\[
x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j 2\pi kn/N}
\]

Note that specifying \( X[0]...X[N-1] \) implies a *synthesized* periodic signal outside \( n = 0...N-1 \):

\[
x[n] = x[n \mod N]
\]
Windowing

Implicitly, the DFT assumes are periodic signal and thus the analysis and thus the analysis equation will capture a potential discontinuity and the period end.

To avoid such artifactual frequency content one should “window”, i.e. multiply the time signal with a windowing function that makes a smooth transition.

Examples are Hanning and Hamming windows.
Discrete Fourier Transform - FFT

In signal processing we always work with the DFT since we can compute Fourier transform only for discrete frequencies.

Important result on computational cost: While computing DFT values $X[k]$, $k=1...N$, would seem to take $N^2$ operations there is an efficient method called Fast Fourier Transform (FFT) of order:

$$N \log_2 N$$

```matlab
>> x = fft(x);
>> x = ifft(X);
```

With this one can implement convolution in $\log_2(P)$ operations per sample rather than $P$!
DFT - circular convolution

Because $X[k]$ corresponds to a periodic $x[n]$ with period $N$ the convolution of two signals is equivalent to a circular convolution:

$$h[n] \circ x[n] = \sum_{k=0}^{N-1} h[k] x[(n-k) \mod N]$$

That is, the circular convolution "wraps around"

It can be implemented with a circular Toeplitz matrix:

$$
\begin{bmatrix}
y[0] \\
y[1] \\
y[2] \\
\vdots \\
y[N-1]
\end{bmatrix}
= 
\begin{bmatrix}
\vdots & \vdots & \vdots & \ddots & \vdots \\
h[N-1] & h[N-2] & h[N-3] & \cdots & h[0]
\end{bmatrix}
\begin{bmatrix}
x[0] \\
x[1] \\
x[2] \\
\vdots \\
x[N-1]
\end{bmatrix}
$$
The convolution theorem for the DFT corresponds now to a circular convolution:

\[ y[n] = h[n] \circ x[n] \quad \Leftrightarrow \quad Y[k] = H[k] X[k] \]

We can use this for a fast implement the linear convolution

\[ y[n] = h[n] * x[n] \]
The $N$-point circular convolution with filter $h[n]$ of length $P$ corresponds to the linear convolution for the last $N-P$ samples. The first $P$ samples do not correspond to the linear convolution as the filter $h[k]$ multiplies with values in the next period. Hence, $y[n]$ computed with the product, $H[k] X[k]$, of the DFTs corresponds to the linear convolution for $n=N-P+1...N$. 

$\text{h[0]...h[P-1]}$

$\text{x[0] ... x[N-1]}$

$\text{x[0] ... x[N-1]}$
DFT - linear convolution with "overlap-save"

The **overlap and save** method for computing the linear convolution using the FFT for a filter of length $N$ consists in:

1. Compute the 2N point FFT of segment $x[n-N]...x[n+N-1]$
2. Multiply by the 2N point FFT of $h[0]...h[N-1]$
3. Invert the result with IFFT
4. Save the last $N$ samples into $y[n]...y[n+N-1]$
5. Increment $n$ by $N$ and return to 1.

**Assignment 5:**

- Implement linear convolutions with "overlap and save" method, and apply to random signal $x$ with some arbitrary filter $h$. Compare this result ($y_{os}$) with a linear convolution implemented with Toeplitz matrix multiplication ($y_{toep}$), and `filter()` function ($y_{filt}$). Plot results $y_{os}, y_{toep}, y_{filt}$. Should be the same.
- Graph speed of all three methods (using `tic; toc;`) for $N=2^n$, $n=1..10$, where $N$ is the size of the filter.
Sampling - Sampling Theorem*

What is the relation between the continuous time Fourier transform (CTFT) of a continuous time signal $x(t)$ given by,

$$X_c(\Omega) = \int_{-\infty}^{\infty} dt \, x(t) e^{-j\Omega t}$$

and the DTFT, $X(e^{j\omega})$ of the signal sampled at frequency $f_s=1/T$, $x[n] = x(nT)$?

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n] e^{-j n \omega}$$

* See Oppenheim and Schaefer for derivation (the classic SP book)
Sampling - Sampling Theorem

According to the Sampling Theorem the relation is:

\[ X \left( e^{j\omega} \right) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X_c \left( \frac{\omega}{T} + \frac{2\pi k}{T} \right) \]

The DTFT repeats the CTFT with a period \(2\pi\).

Contributions above \(\pi\) will overlap with different period!
Sampling - Sampling Theorem

Under which conditions can we determine continuous time \(x(t)\) from discrete time \(x[n]\), \(t=nT\)?

If the signal is \textit{bandlimited}: \(X_c\left(\frac{\omega}{T}\right) = 0\), \(|\omega| \geq \pi\)

Then we can determine \(X_c(\Omega)\) from \(X(e^{j\omega})\) according to the Sampling Theorem:

\[
X(e^{j\omega}) = \frac{1}{T} X_c\left(\frac{\omega}{T}\right)
\]

In that case we can determine \(x[t] \to X(e^{j\omega}) \to X_c(\Omega) \to x(t)\).

After some algebra:

\[
x(t) = \sum_{n=-\infty}^{\infty} x[n] \text{sinc}\left(\frac{t-nT}{T}\right)
\]

\(x(t)\) is \(x[n]\) convolved with \(\text{sinc}(t) = \frac{\sin(\pi t)}{\pi t}\)
Assignment

Use the sampling theorem to reconstruct a signal with high temporal resolution from its low-frequency sampled version:

1. At a high sampling rate, say 1kHz, generate the following 5 signals:
   • Sinusoids at frequencies f=30, 45, and 60 Hz.
   • Gaussian random noise low-pass filtered with stop-band frequency of 40Hz and stop-band frequency of 100Hz.

2. Subsample these 5 signals at a low sampling rate, say 100Hz.

3. Reconstruct the 1kHz signals from the sub-sampled signals.

4. Show the original 1kHz, the sub-sampled and the reconstructed signals.

5. Give the error of you reconstruction as % of variance in the signal for each example (in total you should have 5 numbers giving the % error).

5. For which of the 5 signals above to you expect a vanishing error and for which may there be a significant reconstruction error? Be sure your results agree with that.
Because of the sampling theorem always!

Make sure you lowpass filter the signal to half the sampling frequency (Nyquist) before you sample.

If you can not filter prior to sampling make sure that you choose the sampling frequency to be twice the highest frequency that contains significant signal power.

Do not down sample by simply taking every other sample. First lowpass filter then subsample. Better yet, use either

```matlab
>> x = resample(x,P,Q);
>> x = decimate(x,Q/P);
```