Guidelines for PS # 3.0

Frequency Response

Consider a causal and stable LTI system with input $x[n]$, output $y[n]$ and a difference equation representation of the form:

$$\sum_{k=0}^{N} a[k]y[n-k] \;=\; \sum_{k=0}^{N} b[k]x[n-k].$$

The frequency response of the system $H(e^{j\omega})$ is given by:

$$H(e^{j\omega}) \;=\; \frac{\sum_{k=0}^{M} b[k]\exp\left(-jk\omega\right)}{\sum_{k=0}^{N} a[k]\exp\left(-jk\omega\right)} \;=\; \frac{B(e^{j\omega})}{A(e^{j\omega})}, \quad \omega \in [-\pi, \pi].$$

This frequency response however, is still a continuous a function in the frequency variable ω and is not computable. Instead we compute it at a set of discrete frequencies:

$$\omega_k = \left(\frac{2\pi}{L}\right) k, \quad 0 \leq k \leq (L-1).$$

In MATLAB to compute the frequency response at L points we make use of the function `freqz.m` with synopsis:

```
% L: # of points
% b: coeffecient vector for $B(z)$.
% a: coefficient vector for $A(z)$.
>> [H,W] = freqz(b,a,L);
```

Direct Form Implementation

The difference equation can be implemented in the direct form realization using the MATLAB function `filter.m` using:

```
% b : numerator coefficient vector
% a : denominator coefficient vector
% x : input
% y : output
% state : initial state vector
>> y = filter(b,a,x,state)
```

FIR Filter Design

A lowpass filter with system function $H(z)$ and cut-off frequency, $\omega_c = \frac{\pi}{M}$ where M is the oversampling factor is required in the upsampling and noise shaping assignment. Towards this end use the MATLAB function `fir1.m` with synopsis:

```
% L  : order of the FIR filter
% wn : cut--off frequency for filter
>> b = fir1(L,wn)
```

Autocorrelation

For a random WSS sequence $x[n]$ of length L, the autocorrelation sequence is defined via $R_{xx}[k] = E\left(x[n]x[n-k]\right)$. In the absence of information regarding the underlying probability distributions this is typically approximated using:

$$R_{xx}[k] \;\approx\; \frac{1}{L-|k|}\sum_{n=0}^{L-1} x[n]x[n+k].$$

The normalization factor is so that we obtain a unbiased estimate of the actual autocorrelation sequence. In MATLAB this computation is done via the function `xcorr.m`:

```
%  normalized correlation function
>> r_xx = xcorr(x,'coeff')
```

Power Spectral Density

The power spectrum of the random signal $x[n]$ is defined as the DTFT of the autocorrelation sequence $R_{xx}[k]$ via:

$$P_{xx}(e^{j\omega}) = \sum_{k=-\infty}^{\infty} R_{xx}[k]\exp(-j\omega k).$$

In MATLAB this is computed at a discrete set of frequencies via the command `psd.m` with synopsis:

```
% NFFT : # of discrete points
% Fs   : Sampling frequency
% 'win': Type of window used: default 'hamming'
>> Pxx = psd(X,NFFT,Fs,'win')
```

Average Power

The average power of a random signal $x[n]$ is defined via:

$$P_{\text{ave}}^{x} = E\left(|x[n]|^2\right) = \frac{1}{2\pi}\int_{-\pi}^{\pi} P_{xx}(e^{j\omega})d\omega.$$

In the absence of knowledge of the underlying distributions this is estimated using the sample variance of the sequence $x[n]$ via:

$$P_{\text{ave}}^{x} \approx \frac{1}{L-1}\sum_{k=0}^{L-1}(x[k]-\mu_x)^2,$$

where μ_x is the sample mean. In MATLAB this is implemented via the `std.m` command.

Histogram

To obtain a probabilistic picture of the quantization error sequence $e_q[n]$ we use the `hist.m` command in MATLAB that computes the histogram of the sequence. To obtain an approximation to the PDF underlying the random process we normalize the histogram:

```
% N : frequency of observations
% X : class marks of classes
>> [N,X] = hist(x);
```

Uniform Quantization

Use the program `fxquant.m` on the course webpage to implement the uniform (B+1) bit quantizer:

```
[x_q,alphab,SNR] = fxquant(x,Xmax,X_max,B,opt);
  B : Bit resolution of quantizer (including sign bit);
  X_max : Clip-off magnitude
  opt : 1 --> round--off, 2--> saturation
  x_q : quantized output
  alphab : output alphabet
  SNR    : SNR of quantized signal
```