.

EC431H1 Digital Signal Processing
LAB 3.
Binary Digital Transmission in noisy and dispersive channels.
Introduction

The purpose of this experiment is to become familiar with some aspects of modeling and performance evaluation of binary digital communication channels. It is recommended that you use MATLAB to program and execute the required steps of the experiment. Your lab report (one per group of two students, 5 page approximately) should include: a)printouts of your programs, b) printouts/plots of the obtained results, and c) answers to questions and brief but critical discussion of the obtained results.

Experiments

PART A. Binary transmission in additive white Gaussian Noise channel.

In a binary communication system , binary data consisting of a sequence of 1's and -1's are transmitted by means of two waveforms each of duration T sec. After modulation, transmission through a noisy communication channel, demodulation and synchronous sampling every T sec, the equivalent discrete baseband model that describes the process is

$$r[n] = x[n] + w[n], \qquad n = 1, 2, ..., N$$

where $x[n]$ is the transmitted binary sequence , w[n] is noise from the communication channel and $r[n]$ is the received signal. Usually a hard decision (threshold decoding) is applied on $r[n]$ in order to detect the true value of $x[n]$, that is

$$\hat{x}[n] = 1 \qquad \text{if} \qquad r[n] \geq 0$$

or

$$\hat{x}[n] = -1 \qquad \text{if} \qquad r[n] < 0$$

Due to the presence of the distortion the decision will be characterized by a probability of error defined as

$$P(e) = \frac{\text{number of wrong decisions}}{N}$$

In the first part of the experiment we will calculate the Probability of error, $P(e)$ when the channel distortion is white (that is completely random) Gaussian noise and we will draw the $P(e)$ as a function of the Signal to Noise Power SNR defined as:

$$SNR = 10log_{10}\frac{\frac{1}{N}\sum_{n=1}^{N}|x[n]|^2}{\frac{1}{N}\sum_{n=1}^{N}|w[n]|^2} \qquad \text{dB.}$$

For $N = 1,000$ and $SNR = 0, 1, ..., 5$ dB:

1. Generate the signal $x[n]$, $n = 1, \ldots, N$ as a random sequence taking the values $\{\pm 1\}$ with equal probability. One way to achieve this is to use MATLAB to generate N samples of a uniform random variable between [0,1]. Then assign the value -1 to any sample in the interval [0,0.5] and the value 1 to any sample in the interval [0.5,1].

2. Generate a zero mean, unit variance Gaussian random sequence $w[n]$, $n = 1, ..., N$ by using the corresponding MATLAB random generator.

3. Generate $r[n] = x[n] + G \cdot w[n], \qquad , n = 1, \ldots, N$ where, G controls the corresponding SNR. Since the power of $x[n]$ is 1, then,

$$G = 10^{-\frac{SNR}{20}}$$

.

4. Obtain $\hat{x}[n]$, $n = 1, ..., N$ from $r[n]$ by threshold decoding.

5. By comparing $x[n]$ and $\hat{x}[n]$ calculate and draw the $P(e)$ vs SNR.

PART B. Binary transmission in presence of Intersymbol Interference (ISI) and additive white Gaussian Noise (AWGN).

Most wired and wirelles communication channels introduce intersymbol interference in addition to Additive White Gaussian noise. In a baseband discrete model ISI manifests itself as the result of LTI filtering , that is, the received sequence $r[n]$ can be written as

$$r[n] = h[n] * x[n] + w[n] \qquad n = 1, 2, ..., N$$

where, $h[n]$ is the impulse response of a LTI filter. Thus, at the receiver side in order to recover the information, one has to estimate $h[n]$ and remove its effect by convolving (filtering) $r[n]$ with the 'inverse impulse response' $h^{in}[n]$, defined as $h^{in}[n] * h[n] = \delta[n]$. This operation is called 'equalization'. Equalization will remove the ISI but it may enhance the power of the additive noise which will also be filtered by the $h^{in}[n]$.

For $N = 1,000$ and $SNR = 2, 4, 6, 8, 10$ dB and $h[n] = \delta[n] - 0.6\delta[n-1]$:

1. Generate the binary sequence $x[n]$, and the noise sequence $w[n]$, $n = 1, \ldots, N$ as in part A.

2. Generate $r[n] = h[n] * x[n] + G \cdot w[n]$ where, in this case, G is defined as the ratio of the power of $h[n] * x[n]$ to the power of the noise. Since the power of $h[n] * x[n]$ is $1 \cdot \sum_n |h[n]|^2$, then ,

$$G = \sqrt{\sum_n |h[n]|^2} \cdot 10^{-\frac{SNR}{20}}$$

.

3. Obtain $\hat{x}[n]$, $n = 1, ..., N$ from $r[n]$ by threshold decoding.

4. By comparing $x[n]$ and $\hat{x}[n]$ calculate and draw the $P(e)$ vs SNR.

5. Assuming that the detector knows $h[n]$ obtain $h^{in}[n]$ (how?). Since in our case $h[n]$ is finite length, $h^{in}[n]$ is infinite length. In practice, however, finite length equalizers are obtained by properly truncating $h^{in}[n]$. Choose L (justify your choice), obtain $h^{in}_{tr}[n]$, $n = 1, ..., L$ and equalize the sequence $r[n]$ to obtain $r_{eq}[n] = h^{in}_{tr}[n] * r[n]$.

6. Obtain $\hat{x}_{eq}[n]$, $n = 1, ..., N$ from $r_{eq}[n]$ by threshold decoding.

7. By properly comparing $x[n]$ and $\hat{x}_{eq}[n]$ calculate and draw the $P(e)$ vs SNR (for comparison draw the results of item 3 and item 6 in the same graph.)

8. Draw $|H(\omega)|$, $|H^{in}(\omega)|$ and$|H^{in}_{tr}(\omega)|$ using FFT and comment on how ISI and noise are affected.

PART C. Adaptive equalization of binary data in presence of ISI and AWGN .

Often in practice the communication channel $h[n]$ is not known but needs to be estimated. Also, it is desirable to take into account the presence of noise in the equalization process. One way to achieve this is to use a training sequence with an adaptive equalization procedure to estimate the best $h_{tr}^{in}[n]$, $n = 1, ..., L$ that minimizes ISI and AWGN in a combined manner, then use it to equalize the actual information sequence. A popular process is the Least Mean Square (LMS) algorithm that tries to minimize the mean square error between the original and the output of the equalizer filter.

For $N = 1,500$ and $SNR = 2, 4, 6, 8, 10$ dB and $h[n] = \delta[n] - 0.6\delta[n-1]$:

1. Generate the binary sequence $x[n]$, and the noise sequence $w[n]$, $n = 1, \ldots, N$ and $r[n] = h[n] * x[n] + G \cdot w[n]$ as before.

2. We will assume that $x[n]$, $n = 1, ..., 500$ will be used for training the equalizer filter and the remaining 1,000 samples are the actual information symbols. Choose L and initialize the equalizer filter $h^{(0)}[n] = 0$, $n = 1, ..., L$.
   At iteration $i = 1, \ldots, 500$,

   (a) Calculate the output of the equalizer filter $y[i] = \sum_{n=1}^{L} h^{(i-1)}[n] \cdot r[i - n + 1]$.

   (b) Calculate the error between the output of the equalizer filter and the corresponding true value of the training sequence $e[i] = x[i] - y[i]$ and the so called adaptation step size $\mu[i] = \frac{1}{\sum_{n=1}^{L} r^2[i-n+1]}$.

   (c) Update the coefficients of the equalizer filter

   $$h^{(i)}[n] = h^{(i-1)}[n] + \mu[i] \cdot e[i] \cdot r[i - n + 1], \qquad n = 1, \ldots, L$$

3. Draw $|e[i]|$, $n = 1, \ldots, 500$ and comment on its convergence. Fix $h_{tr}^{in}[n] = h^{(500)}[n]$, $n = 1, ..., L$ and obtain the corresponding $|H_{tr}^{in}(\omega)|$. Compare with the results of part B. If the overal process is not satisfactory , you may wish to try different values of L or length of training data.

4. With $h_{tr}^{in}[n] = h^{(500)}[n]$, $n = 1, \ldots, L$ obtain $r_{eq}[n] = h_{tr}^{in}[n] * r[n]$, $n = 501, \ldots, N$.

5. Obtain $\hat{x}_{eq}[n]$, $n = 501, ..., N$ from $r_{eq}[n]$ by threshold decoding.

6. By properly comparing $x[n]$ and $\hat{x}_{eq}[n]$ calculate and draw the $P(e)$ vs SNR (for comparison draw the corresponding results from PART B and this item in the same graph.) Comment accordingly.