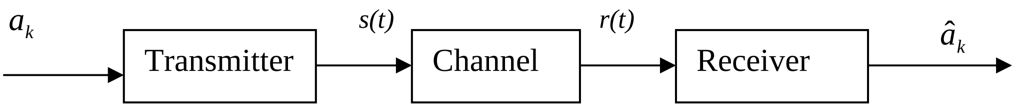
SIMULATION OF DIGITAL COMMUNICATION SYSTEMS

Prof. Dr. Yalçın Tanık, METU
October, 2008

This report highlights some basic points in discrete-time simulation of typical digital communication systems.

Model of the Actual System:



The sequence a_k is commonly assumed to be binary with equally likely and independent bits. We assume that the rate of the sequence is R_b and $T_b = 1/R_b$.

The transmitter output is assumed to be a bandpass signal that can be expressed by

$$s(t) = \Re e\left\{u(t)e^{j2\pi f_c t}\right\}$$

where *u(t)* is the lowpass equivalent of *s(t),* and f_c is the reference frequency which is commonly taken as the carrier frequency of the modulated wave. We denote the bandwidth of *s(t)* by *B* where it is defined such that the PSD of *s(t)* is zero outside the interval

$$(f_c + B/2, f_c - B/2).$$

Usually, the channel is assumed to be linear and either time invariant or slowly varying. Thus, the bandwidth of the signal at the channel output is essentially the same as that its input. We consider that its output is corrupted by additive noise. Therefore

$$r(t) = s(t) * h(t) + n(t)$$

where *h(t)* is the impulse response of the channel (here we assume the channel is stationary).

The receiver processes its input, which can be expressed by

$$r(t) = \Re e\left\{\left[v(t) + z(t)\right]e^{j2\pi f_c t}\right\}$$

where *v(t)* is the lowpass equivalent of the signal component and *z(t)* is the lowpass equivalent of *n(t).* Notice here that we assume the noise process to be of the same bandwidth as the signal. Let P_r denote the average received power and N_0 be the one-sided height of PSD[1] of noise which is assumed to be flat. Then, the bit-SNR of the received signal is

$$\frac{E_b}{N_0} = \frac{P_r T_b}{N_0} = \frac{P_r}{N_0 R_b}$$

Note that this SNR has no relation to the signal bandwidth. On the other hand, if we consider the power SNR which is the average received power to average noise power ratio within the signal bandwidth:

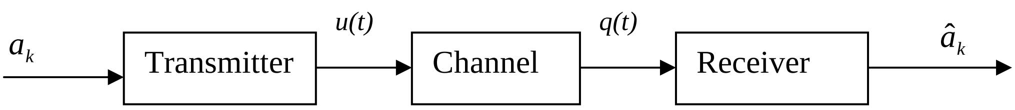
$$SNR_P = \frac{P_r}{N_0 B},$$

which is dependent on the signal bandwidth. However, bit-SNR is commonly preferred as the input SNR condition instead of power SNR. Note that

$$SNR_p = \frac{E_b}{N_0}\frac{R_b}{B}$$

which shows the relationship between the two SNR's. The ratio R_b / B is just the spectral efficiency (bps/Hz).

Lowpass Equivalent System:



The first step we take for simulation (and analysis too) is to obtain a low-pass equivalent of the original system. For this purpose, we use *u(t)* instead of *s(t)* for representing the transmitter output. Similarly,

$$q(t) = v(t) + z(t)$$

represents the receiver input as its lowpass equivalent. Note that two-sided bandwidth of these signals are *B* (one sided bandwidth is *B/2*).

Observe that since lowpass equivalent signals have twice as much power as their originals, the power of *v(t)* is $2P_r$, and the two-sided PSD of *z(t)* becomes $2N_0$ (The average power of *z(t)* must be $2N_0 B$). However, of course, neither the bit-SNR nor the power SNR values are affected.

[1] N_0 can be computed from $N_0 = FkT$, where F is the noise figure of the receiver, k=Boltzman constant (1.38×10^{-23} $Joules/Kelvin$) and T is the ambient tempearature in Kelvin).

Notice that *z(t)* is a complex noise process. For example, if we take *n(t)* as a flat, Gaussian process then

$$z(t) = n_I(t) + jn_Q(t)$$

where $n_I(t)$ and $n_Q(t)$ are zero-mean, independent processes each of which has a rectangular PSD of height N_0 and width *B*.

Finally, we model the channel by its lowpass equivalent:

$$h(t) = 2\Re e\{h_l(t)e^{j2\pi f_c t}\}$$

where $h_l(t)$ is the lowpass equivalent of *h(t)*. Note that this definition is a little bit different than those we use for signals due to the factor 2 that appears in the expression (It is there for mathematical convenience). Thus, we have

$$v(t) = h_l(t) * u(t) .$$

That is, just a conventional filtering for the low-pass equivalents. We can find the lowpass equivalent channel as

$$H_l(f - f_c) = \begin{cases} H(f) & f > 0 \\ 0 & f \le 0 \end{cases} .$$

If we have a flat channel with no distortion but just delay and attenuation, then, the received signal can be approximately found from

$$v(t) = \alpha e^{j\theta} u(t)$$

where the effect of delay manifests itself as a phase change of the carrier and the delay in the envelope is omitted (or, compensated by synchronization).

If we have a channel that introduces a small amount of Doppler-shift, then the following approximation is usually good:

$$v(t) = \alpha e^{j(2\pi f_d t + \theta)} u(t)$$

where f_d is the Doppler shift on the modulated signal.

<u>Discrete-Time Low-Pass Equivalent System Model:</u>

Since most computer tools, like MATLAB, are based on discrete-time signal and system models, we have to obtain a discrete-time model of the system described in previous section. The approach is straightforward: The signals of interest are low-pass, and they can be represented by their samples taken at a rate greater than the Nyquist rate:

$$u_k = u(kT_s)$$

where $T_s = 1/f_s$ and f_s is the sampling rate we use for simulation. By the sampling theorem we must have

$$f_s \geq 2(B/2) = B$$

Thus, it seems that the choice of sampling rate is quite arbitrary. However, to facilitate simulations we should choose it as an integer multiple of the symbol rate. For example, for QAM class with a raised-cosine spectrum pulse shape with roll-off<100% it can easily be seen that $f_s = 2/T$ where T is the symbol period is a valid (and good) choice.

Channel Model

When we consider a channel which is of the same bandwidth as the signal, we simply sample the channel impulse response at rate f_s and obtain the channel model:

$$h_k = h(kT_s)$$

where $h(.)$ denotes the lowpass equivalent of the actual bandpass channel. Then, we obtain its output as

$$v_k = h_k \otimes u_k$$

When a radio channel is to be simulated, the tapped-delay line model can be used:

$$v_k = \sum_{n=0}^{L-1} h_n u_{k-n}$$

where $L = T_m / T_s = f_s T_m$ and T_m is the delay spread of the channel. h_n are complex, Gaussian, independent r.v.'s with variances following the average power delay profile of the channel. In case a slowly time varying channel is simulated, h_n are actually slowly varying random processes PSD of which are the Doppler spectrum of the channel at delay nT_s which can be obtained from the scattering function of the channel.

Received Signal

The signal part of the received signal is obtained as explained in the previous section as the channel output. The noise part can be obtained as the samples of a lowpass complex noise process of bandwidth *B/2*. However, doing so, since the sampling rate is a whole multiple of the symbol rate, the noise samples become correlated, in general. In order to avoid this undesired result, we can assume that noise samples are obtained by filtering the complex white noise with an ideal rectangular filter of bandwidth $f_s / 2$ and sampling at the Nyquist rate. Of course, the spectral density of white noise is to be chosen as $2N_0$ in order to maintain equivalance. The use of a wider filter instead of the actual one is justified in that there will always be subsequent filtering in the receiver blocks, and by keeping the PSD height the same no change is observed as far as noise effects are concerned.

In order to generate the noise samples we note that

$$z_k = n_{I,k} + jn_{Q,k}$$

where the variance of z_k is $2N_0 f_s$. Since the quadrature parts have equal variances, then the real and imaginary parts of z_k are each $N_0 f_s$. For a certain received signal of bit-SNR E_b / N_0, the required sample-SNR can be obtained as follows

$$E\left\{|s_k|^2\right\} = 2P_r ,$$

$$E\left\{|z_k|^2\right\} = 2N_0 f_s .$$

The sample-SNR is

$$\rho = \frac{E\left\{|s_k|^2\right\}}{E\left\{|z_k|^2\right\}} = \frac{P_r}{N_0 f_s} = \frac{P_r T_b}{N_0 f_s T_b} = \frac{R_b}{f_s}\frac{E_b}{N_0}$$

Of course, in order to simulate the SNR condition one doesn't have to adjust the powers of the signals to their original values: We can set the average power of s_k to unity, and the power of z_k can be set equal to $1/\rho$, i.e., powers of the real and imaginary components of z_k can each be set to $1/(2\rho)$. Conversely, the power of z_k can be set equal to unity (powers of the real and imaginary components of z_k set to ½ each), and the average power of s_k is set to to ρ.

For the special case $f_s = R_s = R_b / M$ where M is the number of bits per symbol (symbol rate sampling), we have

$$\rho = \frac{E_s}{N_0}$$

where E_s is the average energy per symbol.