ITESO

Universidad Jesuita
de Guadalajara

Sistemas de Comunicaciones Digitales

**Práctica 3**

Luis Fernando Rodriguez Gutierrez ie705694

Omar Humberto Longoria Gándara

11/05/2020

OBJETIVO

El objetivo de esta práctica es transmitir bits de una computadora a otra, utilizando todas las técnicas que hemos aprendido hasta ahora. La mayor parte del trabajo ya ha sido realizado en las tareas; lo significativo de la práctica será la integración de todo este trabajo en un solo programa, y el reporte.

La transmisión será como mínimo:

1. Obligatorio: Una imagen de 2Mb (como la imagen de la Lena512x512). Con esto se quiere corroborar que la transmisión debe durar por lo menos un tiempo igual al que toma la constelación en cerrarse, abrirse y volverse a cerrar.
2. Opcional: Un archivo de audio comprimido (OPUS por ejemplo)
3. Puede elegirse TDM o FDM, es decir compartiendo el ancho de banda espectral disponible por el canal utilizado.

MODULOS AGREGADOS.

En esta fase, en la cual es el desarrollo de la practica 3. Lo que se realizo primero fue la optimización de los valores para lograr una recuperación correcta de la LENA512 completa. En donde dicha información lo que se hace en guardarla en un archivo .wav para ser enviada desde un teléfono.

Dicha grabación, tiene un tiempo aproximado de 87 segundos, esta para lograr la transmisión total de la imagen. Tomando los siguientes valores.

```
Fs      =   96e3;               % Samples per second
Ts      =   1/Fs;               %
beta    =   0.22;                % Roll-off factor
B       =   16e3;               % Bandwidth available
Rb      =   2*B/(1+beta);       % Bit rate = Baud rate
mp      =   ceil(Fs/Rb)         % samples per pulse
Rb      =   Fs/mp;              % Recompute bit rate
Tp      =   1/Rb;               % Symbol period
B       =   (Rb*(1+beta)/2)     % Bandwidth consumed
D       =   6;                  % Time duration in terms of Tp
type    =   'srrc';             % Shape pulse: Square Root Rise Cosine
E       =   Tp;                 % Energy
[pbase ~] = rcpulse(beta, D, Tp, Ts, type, E);    % Pulse Generation
```

Una vez enviada la información que se quiere transmitir, en este caso una imagen de la LENA. Esta se recupera por medio de Audacity, para luego ser tomada en Matlab.

El siguiente proceso que se realizo fue el normalizar la señal con un rango de valores de 1 y -1. Esto buscando facilitar la automatización del sistema.

```
%% Rx side
clear all ; clc;
format longg

%Audio read from WAV file.
filename= 'lena_rx.wav';
INFO = audioinfo(filename)
nBits = INFO.BitsPerSample;
Fs_audio = INFO.SampleRate;
%samples = [1,sec*Fs_audio];
[Rx_signal,FsWav] = audioread(filename);
Rx_signal = normalize(Rx_signal, 'range', [-1 1]);
```

De manera consiguiente lo que se busco fue eliminar los silencios como se muestra en el código. Obtenemos los siguientes 500 valores.

```
%%
threshold = 0.1; % Detecting the channel energization
start = find(abs(Rx_signal)> threshold,3,'first'); % Initial
stop = find(abs(Rx_signal)> threshold,1,'last'); % End
Rx_signal = Rx_signal (start:stop);
```

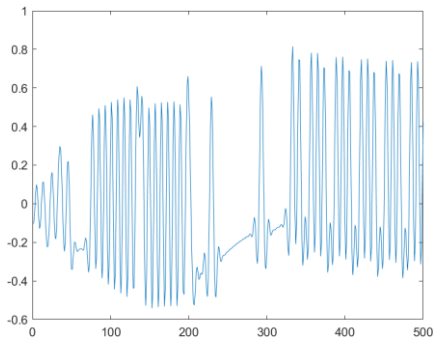


*Ilustración 1 Señal recibida por Audacity*

De manera consiguiente se realiza una convolución de la señal recibida con el pulso base, de manera que ahora que tenemos tanto una señal normalizada y “recuperada”. Obtenemos el diagrama de ojo para poder visualizar que tan abierto quedo este para la recuperación de información.

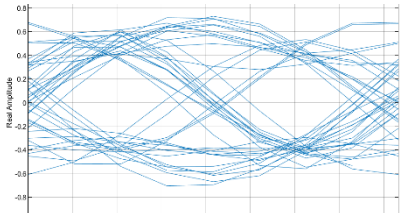

*Ilustración 2. Diagrama de ojo de la señal Rx*

SYNC

Este proceso lo que busca es evitar el buscar por medio de “a ojo de buen cubero” el punto adecuado para empezar a muestrear cada mp muestras. De misma forma que este proceso se busque automatizar este proceso y se logre recuperar información de paquetes mas grandes.

```
%% Sync
% Sampler:
symSync = comm.SymbolSynchronizer('TimingErrorDetector','Early-Late (non-data-aided)','SamplesPerSymbol', mp);
% El sincronizador utiliza la señal del Match Filter
% y también muestrea, dejando solamente los símbolos
rxSym = symSync(norm_signal);
release(symSync); % Liberar el objeto
```

Ahora los valores obtenidos, los decodificamos y convertimos a bits.

```
%% Convert rxSym to bits

bits_rxSym = rxSym(:);

bits_rxSym(bits_rxSym < 0) = 0;
bits_rxSym(bits_rxSym > 0) = 1;
figure; plot(bits_rxSym(1:60));
```

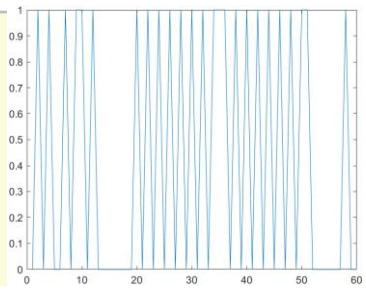


*Ilustración 3. Valores recuperados a bits*

Recuperamos el preámbulo por medio del siguiente código, en este lo que realiza es por medio del preámbulo ya conocido. La función de este pedazo de código es, en un máximo de muestras es que este mismo busque los bits. De manera consiguiente elimina los bits basura, para ya obtener los puros bits.

```
%% Preambulo
preamble_bits = [1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 1 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 1]';

preamble_detect = comm.PreambleDetector(preamble_bits,'Input','Bit');
% Deteccion de preámbulo. El índice indica dónde termina la trama
idx = preamble_detect(bits_rxSym(1:128)) % Ventana de 128 bits
% Una vez que encuentra el índice, se descartan los “bits basura”
% Una forma de hacerlo es la siguiente:
rxData= bits_rxSym(idx+1-numel(preamble_bits):end);
```

En el siguiente segmento de código lo que buscamos es reutilizar el código de la P02. Sin embargo no se puede usar tal como se implemento, se busco optimizar y que lográramos obtener los mismos resultados para recuperar lo que es el header.

```
%%
%preamble num elements
preamble_size = numel(preamble_bits);
header_size = 16;

header_bits = rxData(preamble_size+1:preamble_size+(2*header_size));

header_row = bi2de(header_bits(1:numel(header_bits)/2)','left-msb');
header_col = bi2de(header_bits(numel(header_bits)/2:end)','left-msb');
```

Por ultimo lo único que se hizo fue la conversión de información, y de misma manera recuperar lo que es el payload para recuperar la imagen de la LENA.

```
%% Payload
payload_size = header_row * header_col * 8;
bits_payload = rxData(preamble_size+(header_size*2)+1:end);

%Recover img
bi2mat = vec2mat(bits_payload,8);
lena_bin2dec = bi2de(bi2mat,'left-msb');
new_mat = vec2mat(lena_bin2dec,header_row);
new_mat = new_mat';
figure; imshow(uint8(new_mat));
```

*Ilustración 4. Lena Recuperada, con bajo BER*