COMM.SYS.300 COMMUNICATION THEORY

Matlab Exercise #3

Digital transmission: Baseband PAM (Pulse Amplitude Modulation), Pulse-shaping and Eye-diagrams

1 SYMBOLS AND SINC-PULSE

1.1 GENERATION OF SYMBOLS

In this exercise we consider only real-valued PAM (Pulse Amplitude Modulation) symbols.

- Generate a 4-PAM symbol alphabet, which consists of all possible symbol values

```
alphabet = [-3 -1 1 3]; % Symbol alphabet for 4-PAM
```

- Generate 10 random symbols from the 4-PAM alphabet

```
N_symbols = 10;

% Random vector that includes integers (="symbol indices") between 1 and 4:
symbol_ind = randi(length(alphabet),1,N_symbols);
% I.e., these numbers represent the symbols indices in the alphabet
% (1 means the symbol "-3", 2 means the symbol "-1", and so on...

symbols = alphabet(symbol_ind); % Generate a random symbol sequence
```

- Plot the symbol sequence

```
figure
stem(symbols); % Plots the symbol sequence
```

1.2 ILLUSTRATION OF A BASEBAND SIGNAL BY USING SINC-PULSES

Since discrete symbols cannot be transmitted in practical communications channel, the symbol sequence must be presented in a continuous-wave format. This can be achieved by using overlapping pulses $p(t)$ which satisfy the Nyquist criterion:

$$p(t) = \begin{cases} 1, \text{ when } t = 0 \\ 0, \text{ when } t = \pm kT, \text{ where } k \text{ is an integer and } T \text{ is the symbol duration} \end{cases}$$

Now, by recalling the lecture slides, the PAM signal $x(t)$ is given as

$$x(t) = \sum_k a_k p(t - kT) \ ,$$

where a_k represents the symbol sequence (similar to what we generated earlier in 1.1). One of possible pulses (a classic theoretical example) is the sinc-pulse. Now, by using sinc-pulses our target is to illustrate the PAM signal $x(t)$ in such manner that we can also see the individual (a_k -weighted) pulses $a_k\, p(t\text{-}kT)$.

- Let's consider the following parameters:
    - Oversampling factor r = 10 (this is the number of samples per one pulse/symbol)
    - Symbol time interval T = 1μs
    - Consider the symbols which we earlier generated in task 1.1
- Generate a sinc-pulse (make sure you understand the difference between the symbol time interval and the sample time interval!)

```
r = 10;                 % samples per pulse/symbol (oversampling factor)

T = 1e-6;               % Symbol time interval [s].
Fs = r/T;               % Sampling frequency
Ts = 1/Fs;              % Sample time interval


t = 0:Ts:N_symbols*T;  % Time vector (sampling intervals) for all symbols

% Generation of the pulse:

% Option #1 (traditional)
% p = sinc(t/T); % The pulse given in fixed time vector defined by t

% Option #2 (The pulse given in function defined by the time vector
% parameter t. In Matlab this is called a handle function and it is more
% convenient in our case, so we use it in the following. Here p is not a
% double-variable, but it is a function that can be called any time for
% different time vectors)

p = @(t) sinc(t/T); % Sinc pulse handle function
```

- Plot one sinc-pulse p between time -10μs…10μs

```
t_plot = -10e-6:Ts:10e-6; %plot time interval

figure
plot(t_plot,p(t_plot))
xlabel('Time [s]')
ylabel('Amplitude')
grid on
xlim([-10e-6 10e-6]) % x-axis limits
title('Sinc-pulse')
```

- Generate the PAM signal x(t) for the whole symbol sequence and plot the signal in time domain. Include also plots of different pulses in the same figure by using different colors.

```
figure
hold on % allows us to plot on top of the previous plots
x = 0; %initialize the sum of pulses (i.e. the total signal x(t))
a = symbols; % Renaming the symbol vector according to the equation notation
% For each symbol index k, we generate the weighted pulse and add it to the
% overall sum defining the PAM signal x
for k = 0:N_symbols-1

    % Remember that Matlab indexing starts from 1 (a_0=a(1), a_1=a(2), etc.)

    kth_pulse = a(k+1)*p(t-k*T); %kth pulse weighted by the symbol a_k


    x = x + kth_pulse; % add the kth pulse to the overall sum (PAM signal)
```

```
    plot(t,kth_pulse,'LineWidth',2)
    % Notice that Matlab changes the line color automatically between plots

end

plot(t,x,'k','LineWidth',3) % Plotting the total signal x with black color

% In the end, we also plot the symbol values in their correct time instants
plot(0:T:T*(N_symbols-1),symbols(1:N_symbols),'ko','MarkerSize',10)

hold off
grid on
```

Now, study the generated PAM signal and pay attention to the Nyquist criterion. For example, at which time instants you should take samples from the PAM signal in order to recover the transmitted symbols?

2 BASEBAND SIGNAL BY USING A RAISED-COSINE PULSE

Sinc-pulses are not a practical solution for generating PAM signals. For example, they are infinite in length, for which reason they have to be cut in practice. Cutting the sinc-pulse often leads to undesired spectrum spreading, which causes interference to signals in adjacent frequencies. Raised-cosine pulses are weighted with an appropriate windowing function, and thus, the pulse shape is given as

$$p(t) = \mathrm{sinc}\left(\frac{t}{T}\right)\frac{\cos\left(\frac{\pi\alpha t}{T}\right)}{1-\left(\frac{2\alpha t}{T}\right)^2},$$

where α is a roll-off factor which defines amount the excess bandwidth (the larger is α the wider is spectrum). In the following, we generate the PAM signal by using the raised-cosine pulse as a transmitter filter. Unlike in task 1.2, where the PAM signal was generated by adding separate pulses together (a mathematical approach), here we achive the same PAM signal by filtering the symbol sequence with the transmitter filter (defined as the pulse shape)

- Now, let's consider the following parameters
  - Oversampling factor r = 3 (this is more practical than r=10)
  - Roll-off factor $\alpha = 0.3$
  - Symbol time interval T = 1μs
  - The overall pulse duration is 10 symbols (I.e., the time vector t of one pulse is defined over period of 10 symbols)

```
r = 3;              % Oversampling factor
T = 1e-6;           % Symbol time interval [s].
Fs = r/T;           % Sampling frequency
Ts = 1/Fs;          % Sampling time
t = -5*T:Ts:5*T;    % Time vector (sampling intervals) for one pulse
alfa = 0.30;        % Roll-off factor
```

- Generate 500 4-PAM symbols like did earlier in task 1.1
- Generate the raised cosine pulse for the given time vector t and plot the pulse in time

```
% Raised-Cosine FIR filter:
p = sinc(t/T).*((cos(alfa*pi*t/T))./(1-((2*alfa*t/T).^2)));

figure
plot(t,p)
```

```
hold on
stem(t,p)
xlabel('Time [s]')
ylabel('Amplitude')
hold off
```

- Filter the symbol sequence with the pulse-shape p in order to get the PAM signal $x(t)$
    - Notice that the samples of the symbol sequence and the pulse shape are not given in the same sampling rate (i.e. the sample durations of the symbol sequence and the pulse-shape are T and Ts, respectively)
    - Thus, we must first match the sampling rates by oversampling the symbol sequence

```
% Zero vector initialized for Up-sampled symbol sequence
symbol_upsampled = zeros(size(1:r*N_symbols));

% Up-sampled sequence, i.e. a1, 0, 0, a2, 0, 0, a3, 0, 0, a4, 0, ...
symbol_upsampled(1:r:r*N_symbols) = symbols;

% Transmitter filtering
xn = filter(p,1,symbol_upsampled);

% Remove filter delay. In general it is the sample index of the the maximum
% value of the impulse response. Thus, check "figure,stem(1:length(p),p)",
% where the maximum is at the 16th sample. Thus, the first symbol
% corresponds the 16th sample
filter_delay = (length(p)-1)/2;
xn = xn(filter_delay+1:end);
```

- Plot the first 200 samples from the generated PAM signal

```
figure
plot(0:Ts:199*Ts,xn(1:200)); % Plotting a piece of the generated signal
xlabel('Time [s]')
ylabel('Amplitude')
```

3 EYE-DIAGRAM AND SPECTRAL CONSIDERATIONS

3.1 PLOTTING THE EYE DIGRAM

Recall from the lecture notes the concept of eye-diagram.

- Plot the eye-diagram for the generated 4-PAM signal

```
figure
hold on
for i = 1:2*r:(length(xn)-2*r)
    plot(xn(i:i+2*r),'b');
end
hold off
grid on
```

- Add noise to the signal
    - SNR = 20dB

```
SNR_target = 20; % SNR in dB

zn = randn(size(xn));        % White Gaussian random noise

%Signal power
P_xn = var(xn);              % relies on the ergodicity of the signal model
```

```
% P_xn = mean(abs(xn).^2);  % exact power for this realization

%Noise power
P_zn = var(zn);              % relies on the ergodicity of the signal model
noise_scaling_factor = sqrt(P_xn/P_zn/10^(SNR_target/10));

% Noisy signal
yn = xn + noise_scaling_factor*zn;

% Make sure that the SNR is OK (just in case…):
P_zn_scaled = var(noise_scaling_factor*zn);
SNR_check = 10*log10(P_xn/P_zn_scaled)
```

- Plot the first 200 samples of the noisy signal
- Plot the eye-diagram for the noisy 4-PAM signal

Recall how different non-idealities (noise, timing errors and jitter) affect the eye-opening of the eye-diagram. What do you see? What is the difference between the noiseless and noisy eye-diagrams?

3.2 SIGNAL SPECTRUM AND PULSE SPECTRUM

- Plot the (two-sided) spectrum of the pulse shape $p(t)$ by using the FFT-size of 2048
- Plot the spectrum of the noiseless PAM signal $x(t)$ by using the FFT-size of 2048
- Plot the spectrum of the noisy PAM signal by using the FFT-size of 2048

Compare the pulse spectrum and the PAM signal spectra. Any similarities in spectrum shape or bandwidth?

3.3 MODIFYING THE SYSTEM PARAMETER

In the following, we test the above code (tasks 2 and 3) by modifying a single parameter at the time. Make sure that other parameters are adjusted to their original values (i.e., 4-PAM signal, SNR=20dB, α=0.3) before testing with the desired parameter.

TIP: you can get the same "random" noise realization for each test, if you fix the so called random number generator number (seed) in the beginning of the code, e.g., `rng(41307)` (help rng)

- Test the system by using 8-PAM. Compared to 4-PAM, what was the effect on
    - Pulse shape?
    - Eye diagram?
    - Spectrum and bandwidth?

```
% To get 8-PAM signal, you only have to redefine the alphabet-variable
alphabet = [-7 -5 -3 -1 1 3 5 7]; % Alphabet for 8-PAM (or alphabet=-7:2:7)
```

- Test the system by using SNR=15dB and SNR=10dB (remember to return to 4-PAM before proceeding). Compared to SNR=20dB, what was the effect on
    - Pulse shape?
    - Eye diagram?
    - Spectrum and bandwidth?
- Test the system by using α=0.05 and α=0.95 (again, remember to use 4-PAM and SNR=20dB before proceeding). Compared to α=0.3, what was the effect on
    - Pulse shape?
    - Eye diagram?
    - Spectrum and bandwidth?

Make sure that you understand how different parameters affect the signal!