Xi'an Jiaotong-Liverpool University

西交利物浦大学

EEE411 Advanced Signal Processing: Matlab Individual Assignment II

Due on Friday December 15th 2017, 17:00

Siyi Wang

Assignment Regulations

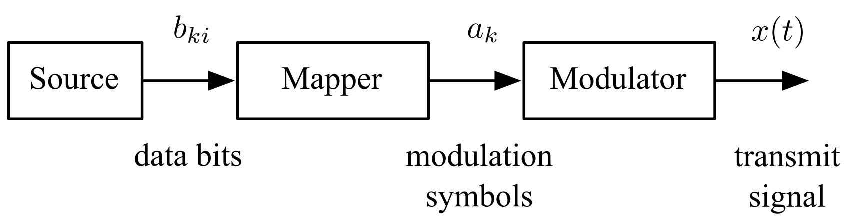
- This is an individual assignment accounting for 15% of the overall assessment for this module. Each student MUST submit a **hard** copy of the report (double-sided printing) into the collecting box before the deadline. A soft copy of the report in SINGLE PDF format should also be submitted on ICE, which however ONLY serves as a backup when the hard copy is missing.
- A coversheet can be created in your choice but the following information must be included: Your student ID number, full name, email address and name of the module.
- Write a report which should contain a concise description of your results and observations to each question.
- All the formula, derivations, completed MATLAB scripts and functions that are NOT provided on ICE, with original highlighted text format in MATLAB editor and output waveforms should be part of your reports. You can find a sample answer on the last two pages of this document.
- Toolbox Functions and GUI Tools are **NOT ALLOWED** to use.
- You may refer to textbooks and lecture notes to discover approaches to problems, however the report should be your own work.
- Where you do make use of other references, please cite them in your work and provide a list of references in IEEE style. Students are reminded to refer and adhere to plagiarism policy and regulations set by XJTLU.
- Assignments may be accepted up to 5 days after the deadline has passed; a late penalty of 5% will apply for each day late without an extension being granted. Submissions over 5 days late will not be marked. Emailed submissions will NOT be accepted without exceptional circumstances.

Practical 3–5: Digital Transmission System with Matched Filter

This practical deals with the simulation of a digital communication system with MATLAB. You will implement a transmitter and a receiver of a baseband system, and investigate the effect of noise to the system.

Digital modulator

The figure shows a block diagram of a digital transmitter.



Problem 1 (*6 points*)

Consider 4ASK with the modulation alphabet $\mathcal{A} = \{-4, -2, +2, +4\}$, and a mapping from bit pairs $b_{k1}b_{k2}$ to modulation symbols a_k, as given in the following table (the index k indicates the time):

$b_{k1}b_{k2}$	a_k
00	$+4$
$\cdots$	$+2$
11	-2
$\cdots$	$\cdots$

Fill in the gaps. (There is no unique solution.)
Write a MATLAB function that implements this mapping with a bit pair as the input and the corresponding modulation symbol as the output. Next, write a MATLAB function that maps a sequence of data bits to the corresponding sequence of modulation symbols.

Problem 2 (*4 points*)

Assume the transmit pulse $p(t) = \texttt{rect}\left(\frac{t-T_s/2}{T_s}\right)$ and the symbol period $T_s = 2$ msec. Sketch the transmit pulse by hand (so that you know what to expect in MATLAB).
Plot the transmit pulse with MATLAB using a time resolution of 10 sample points per symbol period; pick a reasonable time interval. (You may use ***rect.m*** from the ICE.)

Problem 3 (*4 points*)

Assume a sequence of K modulation symbols: $a_1, a_2, ..., a_K$. The corresponding transmit signal is given by

$$x(t) = \sum_{k=1}^{K} a_k \cdot p(t - kT_s).$$

Sketch by hand the transmit signal for a sequence a_1, a_2, a_3, a_4 of four chosen modulation symbols.

Problem 4 (*8 points*)

Use the same modulation symbols as in the previous question. Write a MATLAB script to plot the signal $x_1(t) = a_1 \cdot p(t - T_s)$ with MATLAB.
You may use the following code:

```
t = -2*Ts:T0:6*Ts;
p = @(t) rect( (t-Ts/2)/Ts );
x1 = a1 * p(t-1*Ts);
plot(t,x1);
```

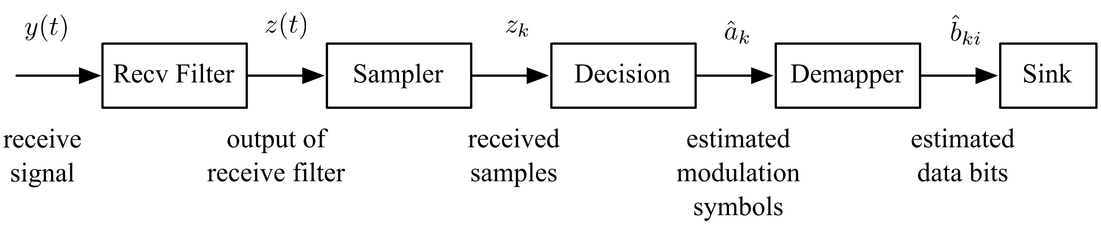
In a similar way, plot the signals $x_2(t), x_3(t)$, and $x_4(t)$ in the same figure.

Problem 5 (*8 points*)

Write a MATLAB function that implements the modulator, *i.e.* takes a sequence of modulation symbols as the input and produces the transmit signal as the output.
Test your function by comparing it to the result of the previous two questions.

Digital demodulator

The figure shows a block diagram of a digital receiver.



Problem 6 (*4 points*)

Assume that the receive filter is a matched filter, *i.e.,* the impulse response of the receive filter is $h(t) = p(T_s - t)$. Sketch by hand the impulse response of the receive filter (with the correct time shift).

Problem 7 (*4 points*)

For the time being, assume noise-free transmission. In this case, the received signal is $y(t) = x(t)$. The output signal of the receive filter is $z(t) = y(t) * h(t)$. Show analytically (by a short calculation) that

$$z(t) = \sum_{k=1}^{K} a_k \cdot [p(t - kT_s) * h(t)].$$

Which properties do you use to show that?

Problem 8 (*4 points*)

Use MATLAB to compute and plot the signal $g(t) = p(t - kTs) * h(t)$ for $k = 1$. Discuss if the result is expected.

Problem 9 (*10 points*)

Assume the transmit signal $x_1(t)$ from Problem 4. Compute and plot the corresponding signal $z(t)$.
You may use the following code:

```
tp = 0:T0:(Ts-T0);
h = p(tp);
z = T0/Ts * conv(y,h);
tz = T0 * (1:(length(z)));
plot(tz, z)
```

Relate your result to $g(t)$ from the previous question.
Repeat this with $x_2(t)$. Is the result as expected? Now repeat this with the full transmit signal $x(t)$ and interpret the result.

Problem 10 (*8 points*)

Based on the results of the previous problem, discuss what are the optimal sampling times for $z(t)$, to determine the samples z_k. Write a MATLAB function (or script) that obtains the samples z_k from $z(t)$.

Problem 11 (*12 points*)

Assume now transmission over an AWGN channel: $y(t) = x(t) + w(t)$. With MATLAB you can obtain a noisy receive signal with the following commands:

```
varnoise = 1;
y = x + sqrt(varnoise) * randn(size(x));
```

The value of ***varnoise*** denotes the variance of the additive white Gaussian noise. Write a MATLAB script that plots (in subplots) the transmit signal, the noisy receive signal, the signal at the matched filter output, and prints out the samples z_k.
Do experiments with various noise variance, and discuss the effect of noise on $y(t), z(t)$ and z_k. Is it easier to estimate the transmitted modulation symbol a_k from $y(t)$ or from $z(t)$?

Problem 12 (*6 points*)

Consider now the decision block. Determine the decision regions for each modulation symbol. Then write a MATLAB function that implements the decision function, *i.e.,* that maps z_k to $\hat{a}_k$.

Problem 13 (*6 points*)

Write a MATLAB function that implements the Demapper. (This is very similar to your Mapper, written before.)

Digital transmission system

In the previous questions we have investigated the individual components required for a baseband digital communication system. We now put all the components together.

Problem 14 (*10 points*)

Write a MATLAB script that implements the transmitter, the channel and the receiver.
The start is a sequence of data bits (you may choose them or pick them randomly), and the end is the sequence of estimated data bits.

Problem 15 (*6 points*)

Test your script for very low noise and for higher noise. Does the system react as expected?

Sample Question

Use function **`cal_roots.m`** on ICE to solve the following quadratic equations:

(a) $x^2 + 5x + 6 = 0$,

(b) $x^2 + 4x + 4 = 0$,

(c) $x^2 + 2x + 5 = 0$,

(d) $2x + 5 = 0$.

Sample Solution

For a general quadratic equation $ax^2 + bx + c = 0$, the roots of the equation can be found in the following:

1. $a = 0$, It is not a quadratic equation.
2. $a \neq 0, \Delta = b^2 - 4ac = 0, x = -\dfrac{b}{2a}$.
3. $a \neq 0, \Delta = b^2 - 4ac \neq 0, x = \dfrac{-b \pm \sqrt{\Delta}}{2a}$.

The listing below shows a MATLAB function that was used to solve the given quadratic equations.

Listing 1: MATLAB function **`cal_roots.m`**

```matlab
function cal_roots(a,b,c)
% cal_roots solves a quadratic function ax^2+bx+c=0
if a==0
    error('a = 0 -- Not a quadratic equation');
% b^2-4*a*c==0
elseif abs(b^2-4*a*c)<1e-10
    x = -b/(2*a)
else
    x1 = (-b+sqrt(b^2-4*a*c))/(2*a)
    x2 = (-b-sqrt(b^2-4*a*c))/(2*a)
end;
```

Below are the MATLAB outputs from Command Window while invoking the function **`cal_roots.m`**.

```
>> cal_roots(1,5,6)

x1 =

    -2


x2 =

    -3

>> cal_roots(1,4,4)
```

```
x =

    -2

>> cal_roots(1,2,5)

x1 =

  -1.0000 + 2.0000i


x2 =

  -1.0000 - 2.0000i

>> cal_roots(0,2,5)
Error using cal_roots (line 4)
a = 0 -- Not a quadratic equation
```