TI-ADWI Exercises

by

John Rohde

AARHUS
UNIVERSITET

Dept. of Engineering, Aarhus School of Engineering
August 27, 2013

Contents

Chapter 1

Introduction

This document contains the exercises for the TI-ADWI course together with a short introduction to the Matlab framework developed to support the exercises.

Chapter 2

Exercise Framework definition

A simplistic Matlab framework is offered in order to facilitate the process of building structured simulation models. A listing of the core m-file is found in listing 2.1. The framework is made up of the core file and a set of m-files, each defining a logical part of the simulation model. According to the listing of the core file, the model consists of the following logical parts:

(a) Message definition
(b) Encoder
(c) Modulator
(d) Channel
(e) Demodulator
(f) Decoder

A full listing of the associated m-files can be found in appendix A.

```
% System model:

[TxMessage MessageSampleRate BitsPerMessage]=ADWI_Message();
[TxEncodedData]=ADWI_Encoder(TxMessage,MessageSampleRate,BitsPerMessage);
[TxSignal]=ADWI_Modulator(TxEncodedData,MessageSampleRate,BitsPerMessage);
[RxSignal]=ADWI_Channel(TxSignal,MessageSampleRate,BitsPerMessage);
[RxEncodedData]=ADWI_Demodulator(RxSignal,MessageSampleRate,BitsPerMessage);
[RxMessage]=ADWI_Decoder(RxEncodedData,MessageSampleRate,BitsPerMessage);

% Playback reconstructed message:
wavplay(RxMessage,MessageSampleRate);
%Various manipulations and presentations

% Bit error rate (BER):

BitErrorVector=xor(TxEncodedData,RxEncodedData);
BER=sum(BitErrorVector)/length(TxEncodedData)
```

Listing 2.1: Matlab Framework: ADWI_Framework.m

Chapter 3

Exercise 1

The purpose of this exercise is twofold: (a) to get acquainted with the Matlab exercise framework and (b) to obtain a deeper understanding of the radio channel and associated descriptive parameters. Remember to use the baseband signal model whenever possible: eliminating the carrier results in simpler calculations/simulations.

3.1 Part I: Propagation of sound waves

In this part of the exercise, the propagation of sound waves is analyzed. An acoustic channel is defined by a delay profile determined experimentally. The delay profile is stated in table 3.1. Please observe, that all amplitude weights are real-valued (no imaginary part).

Delay (ms)	Amplitude weight
300	0.4
350	0.2
400	0.1
450	0.1
800	0.04

Table 3.1: Description of acoustic channel

(a) Determine the channel transfer function $H_c(f)$ associated with this delay profile. Plot $|H_c(f)|$.
(b) Calculate the RMS delay spread (σ_T) and the coherence bandwidth (B_0).
(c) Adapt the Matlab framework so it can simulate the effect of this channel. Run the simulation with SoundMessage.wav as input (default setting for simulation framework).
(d) Comment on the results. Compare results from (a), (b) and (c).

3.2 Part II: Radio channel impact on AM signal

Now assume that the sound file from the previous section is amplitude modulated onto a 2.4 GHz carrier frequency (f_c. The amplitude modulation is obtained as described in equation 3.1.

$$s_{tx} = \big(1 + m(t)\big)\cos(2\pi f_c) \tag{3.1}$$

Here, m(t) is the sound signal. The channel characteristics are given in table 3.2.

Delay (ns)	Amplitude weight
500	0.5
1400	0.5
1800	0.4
2300	0.5
5000	0.8
8000	0.6

Table 3.2: Description of 2.4 GHz radio channel

(e) Determine the channel tranfer function $H_c(f)$ associated with this delay profile. Plot $|H_c(f)|$.
(f) Calculate the RMS delay spread (σ_T) and the coherence bandwidth (B_0).
(g) Adapt the Matlab framework so it can simulate the effect of this channel. Run the simulation.
(h) Comment on the results. Is there agreement between the result from (e), (f) and (g)
(i) Add AWGN to the simulation. Run the simulation at different SNR's. Comment on the results.

3.3 Part III: Radio channel impact on coherent BPSK signal

Repeat Part II (g), (h) and (i) with a BPSK modulation instead. Discuss the differences. What happens, if the delay characteristics are as stated in table 3.3

Delay (ns)	Amplitude weight
50	0.5
140	0.5
180	0.4
230	0.5
500	0.8
800	0.6

Table 3.3: Description of 2.4 GHz radio channel

Chapter 4

Exercise 2

The purpose of this exercise is to get a basic understanding of how a fixed zero-forcing equalizer works. Your objective is to implement a zero-forcing equalizer for the channel delay profile stated in table 4.1.

Delay (μs)	Amplitude weight
0	1
32	1
64	1
96	1
128	1
160	1

Table 4.1: Description of delay profile used for equalizer design

We reuse the AM modulation and carrier frequency from Exercise 1, part II.
(a) Find the baseband signal at the receiver input by applying the channel effect to the data (wave-file). Discuss the effect of the channel by examining the resulting baseband signal.
(b) Implement a zero-forcing equalizer based on the channel profile. Show that this equalizer is working by using it on the AM modulated signal.
(c) What happens when you add noise to the channel and use the equalizer on the noisy signal? Experiment with different SNR's. How does this compare to the answer from exercise 1 (part II)?
(d) Try to introduce some minor adjustments to the delay profile. Test the equalizer from (b) on data passed through the adjusted channel (note: we do not change the equalizer!).

Chapter 5

Exercise 3

Implement a convolutional encoder with the following specifications:

- Code rate=1/2 (=R_C), $C_1 = S_1 \oplus S_2 \oplus S_3$, $C_2 = S_1 \oplus S_3$ (k=1,n=K=3).

(a) Construct a random binary data sequence consisting of 10000 bits. Run the sequence through the convolutional encoder.

(b) Implement a Viterbi decoder based on hard decision and test it on the encoded sequence from (a).

(c) Introduce random bit errors in the encoded bit stream corresponding to BER=10^{-4} and check the performance of the Viterbi decoder from (b). Comment on the result.

Now, assume that errors in the decoded bit stream are unacceptable.

(d) Find an approximate value for the maximum BER tolerated when using the Viterbi decoder from (b). Hint: simple trial varying the BER is indeed a possibility.

Chapter 6

Exercise 4

The purpose of this exercise is to examine a selection of non-linear effects in wireless transceivers: spectral regrowth caused by the Tx amplifier, signal-space diagram constellation point distortion caused by I/Q gain imbalance, phase noise and Tx amplifier non-linearity and intermodulation distortion in the receiver frontend. Due to the focus on analog effects, usage of functions from the Matlab Communication Toolbox is accepted - and encouraged.

The model presented in equation 6.1 is used throughout the exercise to describe a non-linear amplifier element.

$$v_{\text{out}}(t) = G_{\text{V}}\, v_{\text{in}}(t) + \frac{\sqrt{2}}{P_{\text{IIP2}}}\, v_{\text{in}}^2 - \frac{2}{3}.\frac{G_{\text{V}}}{P_{\text{IIP3}}}\, v_{\text{in}}^3 \tag{6.1}$$

6.1 Spectral regrowth and EVM

Consider two transmitters characterized by:

- Tx 1: P_1=P_0, BPSK ($\pm 180°$), R_b=1.0 Mbit/s, no symbol shaping.
- Tx 2: P_2=P_0, BFSK (± 150 kHz), R_b=1.0 Mbit/s, no symbol shaping.

Generate a 1000 bit long random binary data sequence and apply both transmitter models to this data sequence. Assume a 10 MHz carrier frequency for both transmitters for simplicity. Apply the nonlinear model presented in equation 6.1 with G_v=20 dB, P_{IIP2}=20 dBm and P_{IIP3}=10 dBm to the resulting signals.

(a) Plot the power spectrum for the output of both transmitters for P_0=-20 dBm and P_0=10 dBm. Comment on the results.

b Determine the first sidelobe to main lobe power ratio for P_0=-20 dBm to 10 dBm for both transmitters. Plot the results in the same graph. Comment on the results.

(c) Determine the EVM for Tx 1 output spectrum with the non-linear model applied for P_0=-20 dBm and P_0=10 dBm. Comment on the results.

6.2 Receiver intermodulation

Consider three Bluetooth signals represented as continous carriers to simplify matters. The signals are defined by their frequency and signal power:

- Signal 1: f_1 , P_1=-60 dBm
- Signal 2: f_2 , $P_2 = P_I$=-30 dBm
- Signal 3: f_3 , $P_3 = P_I$=-30 dBm

A receiver is receiving all three signals: the wanted signal is signal 1 and the other two are considered to be unwanted interferers. Due to the non-linear behaviour of the LNA in the receiver, 3^{rd}-order intermodulation products are present together with the wanted signal at the output of the LNA.

(d) Determine the relative power of the 3^{rd}-order intermodulation product on the wanted signal frequency (in dB, relative to P_1).

Now, consider that you have to design the LNA and determine the P_{IIP3} requirement based on the test case outlined above with the three signals.

e What is the minimum value of P_{IIP3} that can be accepted, if the relative power in (a) has to be kept below 10 dB?

Appendices

Appendix A

Appendix A

```
% System model:

[TxMessage MessageSampleRate BitsPerMessage]=ADWI_Message();
[TxEncodedData]=ADWI_Encoder(TxMessage,MessageSampleRate,BitsPerMessage);
[TxSignal]=ADWI_Modulator(TxEncodedData,MessageSampleRate,BitsPerMessage);
[RxSignal]=ADWI_Channel(TxSignal,MessageSampleRate,BitsPerMessage);
[RxEncodedData]=ADWI_Demodulator(RxSignal,MessageSampleRate,BitsPerMessage);
[RxMessage]=ADWI_Decoder(RxEncodedData,MessageSampleRate,BitsPerMessage);

% Playback reconstructed message:
wavplay(RxMessage,MessageSampleRate);
%Various manipulations and presentations

% Bit error rate (BER):

BitErrorVector=xor(TxEncodedData,RxEncodedData);
BER=sum(BitErrorVector)/length(TxEncodedData)
```

Listing A.1: Matlab Framework: ADWI_Framework.m

```
function [x,y,z] = ADWI_Message()
% Actual Message data definition. Here,
% data can be read or otherwise generated.
%
%Output:
%x : Vector containing message data
%y: Message Sample Rate
%z: No. of bits per message

[x y z]=wavread('SoundMessage.wav');
```

Listing A.2: Matlab Framework Message: ADWI_Message.m

```
function [x] = ADWI_Encoder(message,fsample,bitspersample)
%   Encoder definition. Output is delivered directly to modulator
%   so the block should include message encoding, encryption
%   and channel coding. Output is a serialized bit stream.
%
%   Output:
%   x : Encoded bit stream

x=message;
```

Listing A.3: Matlab Framework Encoder: ADWI_Encoder.m

```
function [x] = ADWI_Modulator(txencodeddata,fsample,bitspersample)
%
%   Modulator definition. Output is delivered directly to channel
%   so the block should include all analog signal manipulation: pulse
%   shaping, modulation and filtering. Output is an analog signal.
```

```
6  %
7  %   Output:
8  %   x : Transmitter output
9  %
10
11 x=txencodeddata;
```

Listing A.4: Matlab Framework Modulator: ADWI_Modulator.m

```
1  function [x] = ADWI_Framework_Channel(txsignal,fsample,bitspersample)
2  %
3  %   Channel definition. Output is delivered to receiver
4  %   so the block should include all channel effects: noise,
5  %   interference and fading. Output is an analog signal.
6  %
7  %   Output:
8  %   x : Channel output.
9  %
10
11 x=txsignal;
```

Listing A.5: Matlab Framework Channel: ADWI_Channel.m

```
1  function [x] = ADWI_Demodulator(rxsignal,fsample,bitspersample)
2  %
3  %   Demodulator definition. Output is delivered directly to deocoder.
4  %   The block ouputs a serialized bit stream representing an estimate of
5  %   the TxEncodedData.
6  %
7  %   Output:
8  %   x : Decoder output
9  %
10
11 x=rxsignal;
```

Listing A.6: Matlab Framework Demodulator: ADWI_Demodulator.m

```
1  function [x] = ADWI_Decoder(RxEncodedData,fsample,bitspersample)
2  %
3  %   Decoder definition. Output is a reconstruction of the original message
4  %   signal. The block should include: channel decoding and source
5  %   decryption and decoding.
6  %
7  %   Output:
8  %   x : Reconstructed Message
9  %
10
11 x=RxEncodedData;
```

Listing A.7: Matlab Framework Decoder: ADWI_Decoder.m

includewhatevername...

Bibliography