ECE431 Lab 4: DSP in Binary Baseband Communications

by Takis Zourntos

January 27, 2004

1 Introduction

This lab is brief! *Hallelujah!* You should have extra time to work on your design projects! In this lab, you’ll investigate the application of DSP to yet another important area: digital communications. We consider baseband communications, i.e., communications without the use of a modulating carrier signal, so that transmitted signals may contain dc components.

**Tip 1** Why is it that everybody’s psychic tells them that they were on Atlantis in a previous life? Well, we now know why Atlantis sunk: there were just too many people on it.

After the immense success of its zooming spectrum analyzer, your company, CuteDSPs Inc., targets the digital communications market. The management feels that with you as newly appointed head of engineering, it may confidently move into an entirely new area. The responsibility is great and the stress is high, but the finesse with which you handled your previous work infuses you with enthusiasm.

2 The Communication System

Your first task is to design a DSP-based component for the Decoder block of the communication system shown in Figure 1. You can access the Simulink model of this system by entering lab4 at the MATLAB prompt. Run the script lab4init before running a simulation. Please note that the simulation will not function properly before you complete the design of the Decoder block.

The binary data being transmitted takes the form

\[ \mathbf{bd} = \{b_0, 0, b_1, 0, \ldots, b_{N-1}, 0\} \]  \hspace{1cm} (1)

![Figure 1: A Binary Baseband Communication System.](image)
in which the symbols \( b_i, i = 1, \ldots, n \), can take on the values +1 or -1. The 0s act as separators. After running \texttt{labbinit}, you can have a look at the \texttt{bd} workspace variable. One element (a symbol or a separator) of \texttt{bd} is sent to the Encoder block every \( T_{\text{sym}} \) seconds (in this lab, \( T_{\text{sym}} = 1 \)).

The Encoder transmits a pulse train (with pulse width \( \frac{T}{L} \) seconds, where \( T = \frac{T_{\text{sym}}}{L} \), in which \( L \) is a positive integer of your choosing) of length \( T_{\text{sym}} \) for each symbol, \( b_i \). If \( b_i = +1 \), the pulse train is positive-going, otherwise, if \( b_i = -1 \), the pulse train is negative-going. A separator (0) is encoded as a constant zero output for the entire \( T_{\text{sym}} \) duration. The encoder output is illustrated in Figure 2, along with other signals in the communication system. We employ a very simple additive noise channel for this problem. Double-click on the channel model then double-click on the “Band-Limited White Noise” block to adjust the noise power.

The first stage of the Decoder block is essentially a discrete-time integrate-and-dump system. That is, while a symbol is transmitted, it operates as an accumulator, “integrating” \( y[n] \). While a separator is transmitted, it \textit{resets}, setting its internal state(s) to zero. The Decoder is illustrated in Figure 3. Your task is to design the sub-block, \( A_i \), according to the timing diagram in Figure 2. If you open the Decoder in \texttt{Simulink}, you will discover an empty block for you to develop.

The output of the Decoder is stored in the workspace variable, \texttt{rd}. Owing to various delays, \texttt{rd} is at best a \textit{shifted} version of \texttt{bd}. The performance of the system is measured in term of bit-error rate, \( \beta \), defined as

\[
\beta = \frac{\text{No. of incorrect (symbol) bits in } \texttt{rd}}{\text{No. of symbols (not including separators)}}
\]  \hspace{1cm} (2)

3 Your Tasks and General Guidelines

Your tasks for this lab are:
Figure 3: Detailed view of the Decoder block. You are asked to design A using the elements in the Simulink lab4toolbox.

1. Using the lab4toolbox Simulink toolbox provided, design a resetting discrete-time integrator which fits the timing diagram in Figure 2. Please document its operation, and include mathematical expressions which govern it. Note: your Resetting Integrator should operate at a sampling period of $T$, defined above. Try to explain the advantage of our encoding process and use of integration in decoding as far as robustness to noise is concerned (doing the next few parts will help you answer this).

2. For $L = 5$, show how $\beta$ increases with increasing noise power. Simulate your system for the following noise powers: 0.01 : 0.01 : 0.1, (MATLAB notation for 0.01 to 0.1 in steps of 0.01) Note that you set $L$ by directly editing the file lab4init.m. Plot $\beta$ versus noise power; what do you conclude? (Make sure you run the experiment enough times to get a sufficient number of errors (at least 100) for each noise power, and then average appropriately to get $\beta$)

3. Fix the noise power at 0.04. Simulate the system for $L = 1 : 1 : 4$. Find $\beta$ in each case, and plot it versus $L$. What do you notice?

4. Thanks for your attention. Have a great life, and avoid participating in the brain drain, if at all possible!

Tip 2 (Lifelong Learning) As you probably know, you can’t let this final year of studies be the end of your education, even though by now you probably want it to be. To stay sharp, we all need to practice our fundamentals and commit to lifelong learning. This really amounts to growth— if you are not expanding as a person, you are regressing. Ommmmmmm...