INTRODUCTION

This experiment investigates three important concepts in the field of binary communications, namely:
A. Digital Modulation Techniques
B. Error-Control Coding
C. Transmission of Coded Signals

Part I illustrates the use of two modulation schemes, Pulse Code Modulation (PCM) and Pulse Amplitude Modulation (PAM). Since this handout is intended to serve only as a guide, an elaborate treatment of the underlying principles is not presented. The reader is referred to [1] (sections 11.1, 12.1, and 12.2) or [2] (sections 5.1 and 6.1) for more details.

Part II demonstrates two error-control codes, Parity Check and Hamming codes. [1] (sections 13.1 and 13.2) or [2] (section 8.3).

Part III is a baseband simulation of a transmitter that uses the modulation schemes of Part I and the error-control codes of Part II. Error statistics are obtained to evaluate the merits and demerits of the transmission schemes.

BACKGROUND THEORY

A. DIGITAL MODULATION TECHNIQUES

Pulse Code Modulation (PCM)

The schematic of a PCM system is illustrated in FIGURE 1. (From the introductory experiment in this laboratory, you may observe that it is identical to that of an Analog to Digital Converter). In principle, PCM involves the quantization of a continuous signal to a discrete set of values, followed by a suitably chosen encoding scheme. The encoder in FIGURE 1 operates on uniformly quantized
signal values and produces an encoded bit stream. Such a system is known as Linear PCM or simply PCM. A detailed discussion of PCM may be found in [1] sections 12.1 and 12.2. In this laboratory, you will observe the performance of a transmitter that employs PCM, in the presence of additive noise. In Part III, error-control codes will be used in conjunction with the modulation scheme, in an effort to improve the transmitter's overall performance. The sample rate for PCM is given by

\[ (T_{\text{sample}})_{\text{PCM}} = q T_b = q[\text{overhead} + n_{\text{delay}}\delta] \]

where \( T_b \) is the bit duration, \( q \) is the number of bits/word, \( n_{\text{delay}} \) is the integral number of delays entered in the program, and \( \delta \) is the program's time delay period.

**FIGURE 1. Pulse Code Modulation.**

**Pulse Amplitude Modulation (PAM)**

The primary aim of a digital communication scheme is to transfer a digital message sequence from a given source to a destination. The main impediments to a successful transfer are finite bandwidth and transmission noise. The bandwidth limits the rate of transfer of data, while transmission noise introduces errors in the transmitted message sequence. The modulation scheme studied in this section provides a way to improve the robustness of the transmission system.

The transmitted signal may be represented at baseband as

\[ x(t) = \sum_k a_k p(t - kD) \quad -\infty < k < +\infty \]

where \( a_k \) represents the value of the \( k^{th} \) sample of the analog signal. \( p(t) \) represents the modulating pulse, whose choice plays a critical role in the design of the transmission scheme. For simplicity, we shall assume \( p(t) \) to be a rectangular pulse, subject to the constraints

\[ p(t) = \begin{cases} 
1 & t = 0 \\
0 & t = \pm D, \pm 2D, \ldots 
\end{cases} \]

Hence, at the receiver, the message can be recovered by sampling \( x(t) \) at \( t = KD, K = 0, \pm 1, \pm 2, \ldots \)

\[ x(KD) = \sum_k a_k p(KD - kD) = a_k \]

Note that \( D \) represents the time assigned to each symbol. In this experiment, we shall employ binary signaling, so that \( D = T_b \), the bit duration. The bit rate is then given by

\[ r_b = 1/T_b \text{ bits/sec} \]
FIGURE 2 illustrates the PAM scheme used in this experiment. If the pulse has a duration of $1/T_b$ followed by a return to the '0' level, the scheme is called a Return to Zero (RZ) format. If the pulse duration is $T_b$, a Non Return to Zero (NRZ) format results. Other PAM formats exist. The reader is referred to [1] (section 11.1) for details.

![PAM and PCM waveforms](image_url)

FIGURE 2. PAM and PCM waveforms.

B. ERROR-CONTROL CODING

Error-Control codes are employed in transmission systems to combat the inherent channel interference, through the addition of redundant information. We shall study two schemes in this laboratory, and observe their performance over a simulated channel.
**Parity Check Code**

The parity of a binary word refers to the number of 1s in the word. If the number of 1s is even, the parity is said to be *even*. Otherwise, the parity is *odd*. In a parity check code, an extra bit $c$ is added to an $N$-1 bit word $\{m_1 \ m_2 \ ... \ \ m_{N-1}\}$ (with message bits $m_i$) resulting in an $N$ bit codeword

$$c \ \ m_1 \ \ m_2 \ ... \ \ m_{N-1}$$

*Odd* or *even* parity is used, such that all transmitted codewords have the same parity. At the receiver, the parity of every codeword is checked, and a discrepancy between the parities of the transmitted and received codewords will indicate a transmission error. Mathematically, the check bit can be calculated for even or odd parity by either

$$c = m_1 \oplus m_2 \oplus m_3 \oplus m_4 \oplus m_5 \oplus m_6 \oplus m_7 \oplus m_8$$

or

$$c = m_1 \oplus m_2 \oplus m_3 \oplus m_4 \oplus m_5 \oplus m_6 \oplus m_7 \oplus m_8$$

**(7,4) Hamming Code**

In a (7,4) Hamming code, 3 check bits are transmitted with every 4 input message bits. The relation between the check bits and the message bits in a non-interlaced code is as follows:

$$\begin{align*}
c_1 \ &= \ m_1 \oplus m_2 \oplus m_3 \\
c_2 \ &= \ m_2 \oplus m_3 \oplus m_4 \\
c_3 \ &= \ m_1 \oplus m_2 \oplus m_4
\end{align*}$$

Each message bit appears in at least two check bit equations as in all Hamming codes. At the receiver, decoding is not as trivial as in the case of a parity check code. Nevertheless, for a (7,4) Hamming code, syndrome decoding offers a simple solution, and is used in this laboratory. For a detailed treatment of parity check and Hamming codes, and their associated decoding schemes, the reader is referred to [1] (sections 13.1 and 13.2) or [4].

**C. TRANSMISSION OF CODED SIGNALS**

This section is essentially a study of a combination of the techniques presented in parts (1) and (2). The choice of modulation scheme in conjunction with a suitable coding technique can affect the overall performance of the transmission scheme in the presence of noise. In this experiment, PAM and PCM schemes are used with the two coding schemes from part (2). For a random input bit stream, error statistics are obtained on the received (and decoded) words, leading to a comparison of the modulation/coding schemes as the S/N ratio changes in the transmission channel.

**EXPERIMENTAL PROCEDURE**

Your report should include brief descriptions of what procedures you used, the data collected, and an analysis of the results. Wherever possible, **graph** the data, and be sure to include **explanations** in answering any questions.
Make sure you follow the guideline under **NOTES** below on page 9. Once in DOS only mode execute the 6/29/05 version of BinComm (NOT the 1998 version). Also note that the analog ports on the DOS PCs use a 5-BNC connector aluminum box with inputs and outputs labeled rather than the BNC-2110 boxes used by National Instruments.

Finally, remember that this lab is an investigation of binary communication techniques, so don't dwell on sampling theory and other concepts covered in the Introductory Experiment.

**PART I**

The experiment configuration for this section is shown in FIGURE 3. Once the equipment is set up, type *bincomm* at the DOS prompt. The program is completely menu driven. Choose the option for **Digital Modulation Techniques**. For the chosen modulation scheme (*Pulse Amplitude Modulation* or *Pulse Code Modulation*), enter appropriate values for the specified parameters. The program is then executed by selecting the appropriate menu item.

Note that the signal-to-noise ratio should be evaluated in terms of the ratio of the average signal power to the average noise power. The noise power may be obtained from the RMS voltage reading on the meter of the noise generator. Indicate any assumptions you make in calculating the average signal power.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Input Function</th>
<th>Input Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>q</td>
<td>Number of bits</td>
<td>1 - 12 or 1 - 8</td>
</tr>
<tr>
<td>n</td>
<td>code delay value</td>
<td>0 - 255</td>
</tr>
<tr>
<td>amp</td>
<td>PCM voltage amplitude</td>
<td>0 - 10</td>
</tr>
<tr>
<td>RZ</td>
<td>Return or Non-return-to-zero</td>
<td>0=RZ, 1=NRZ</td>
</tr>
</tbody>
</table>

Direct your investigation to answer at least the following questions:
1. How does the noise in the channel affect the reconstructed waveform for PAM and PCM? (A qualitative explanation will suffice.) How does the **rate** of degradation in received signal quality compare?
2. How many pulses per sample are sent using PAM? How many for PCM? What does this imply about the bandwidth requirements for each scheme?
3. What is the difference between *RZ* and *NRZ* modulation? What are the advantages and disadvantages of each? Consider factors such as bandwidth, required power, synchronous/asynchronous transmission, etc.
4. How does the in-channel noise immunity (as determined by the quality of the reconstruction) change with varying bit resolution for PAM and PCM?
5. What are the differences in transmission as the sampling rate and the quantization are changed?
6. What are the relative advantages and disadvantages of PAM and PCM?

PART II

From the main menu, choose the option for Error-Control Coding. Then using the menu number, select one of the coding or decoding schemes. The choices are:

   Code Generation
   Even Parity (E), Odd Parity (O) or Hamming Code (H)

   Code Verification
   Even Parity (E), Odd Parity (O) or Hamming Code (H)

Enter any binary word. The output word is displayed continuously.

From the results of your investigation, answer the following questions:
1. Write down two code words that use even parity. Show the calculations used to generate the parity bits. Repeat for odd parity.
2. For each code, form a table indicating the number of errors that can be detected and the number that can be corrected. Do you observe any dependencies in the table (e.g. one possibility excluding another)?
3. The Code Rate ($R_c$) is defined as the number of message bits divided by the total number of bits in the codeword. What is the code rate of the parity code and the (7,4) Hamming code? What does this imply about the efficiency of the two codes?
4. Explain the working of these codes? Your answer should be based on Hamming distance or signal space concepts. Verify by experiment the minimum Hamming distance between valid code words for the Hamming code.
5. What are the advantages and disadvantages of parity check and Hamming codes?

PART III

For this simulation, set up the equipment as shown in FIGURE 4. The input bit stream of 50,176 words for the simulation is generated in the computer using a pseudo random number generator. Choose the option for Transmission of Coded Signals from the main menu, and select a code type. The bit stream is transmitted using NRZ PCM. For each choice, enter the appropriate parameters, and run the program by picking the appropriate menu item to transmit the stream over the noisy channel simulated with the noise generator and voltage summer. The simulation may be aborted at any time by pressing any key. The results of the simulation are then printed on the screen.
Note that the uncorrected (undetected) error statistic does not include errors that are corrected by the Hamming code or detected by the parity check. Although a constant 50,176 words are transmitted, the word length varies from 7 bits for the Hamming to 8 bits for the unencoded and parity check code words.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Input Function</th>
<th>Input Range</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>amp</code></td>
<td>PCM voltage amplitude (volts)</td>
<td>0 - 10</td>
</tr>
<tr>
<td></td>
<td>Code type</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1=Odd-Parity</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2=Hamming, single error correction</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3=Hamming, double error detection</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Output Function</th>
<th>Output Range</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>biterr</code></td>
<td>cumulative bit errors</td>
<td>0 - 401,408</td>
</tr>
<tr>
<td><code>corr</code></td>
<td>single-bit word errors</td>
<td>0 - 50,176</td>
</tr>
<tr>
<td><code>uncorr</code></td>
<td>double-bit word errors</td>
<td>0 - 50,176</td>
</tr>
<tr>
<td><code>uncorr-b</code></td>
<td>more-bit word errors</td>
<td>0 - 50,176</td>
</tr>
</tbody>
</table>

Use the simulation to answer the following problems. In each case 50,176 words are transmitted every simulation run. Run enough simulations to get accurate graphs. For problems 1 to 4, show the details of your calculations for only one S/N ratio for each problem in order to demonstrate you know what your doing. Tabulate the results for the other S/N ratios and graph the results as described. You may want to use a spreadsheet program to perform the repetitive calculations and plotting.

1. The bit error rate (BER) is defined as the number of message bits in error divided by the total number of message bits transmitted. Graph the BER vs. S/N ratio for an uncoded system. Use the parity code option and assume that all of the bits in each word are message bits. The number of bits in error after the simulation is complete is given to you. The BER should range from 1E-2 to 1E-7 since you will be using this same graph for the other problems. You will want to use a log scale for the BER but not for the S/N ratio. Clearly label all parts of the graph.

2. Assume now that an (8,7) parity check code is used to detect errors. Assume also that if an error is detected in a code word, the entire word is retransmitted and the retransmission is perfect (i.e. no errors). Now there are only 7 message bits per word. The words which contain a single bit error are detected and retransmitted. The other words are decoded and contribute to the total number of bit errors; two errors for each of the words with double bit errors and assume three errors for those with three or more bit errors. Since only 7 out of 8 bits in each word are message bits, multiply the total number of errors contained in the double and triple bit error words by .875 to get the total number of message bit errors. Use this procedure to calculate the BER for an (8,7) parity code for different S/N ratios and plot the results on the same graph used for problem 1.
3. Now assume that a (7,4) Hamming code is used to correct channel errors. This time only correction is used and no codeword retransmission is done. The codewords which contain only a single bit error are corrected. Those which contain double bit errors will be erroneously corrected and actually contain 3 bit errors after the decoding is done. Additionally, those words which contain 3 or more errors can be assumed to contain 3 bit errors each without altering the results significantly. Total up the number of bit errors after decoding and multiply this total by 4/7 since there are 4 message bits per code word (the other errors are assumed to be in the check bits which don't contribute to the BER). Once again calculate the BER for various S/N ratios and plot the results on the same graph as in problems 1 and 2.

4. Finally, assume that the (7,4) Hamming code is used to detect errors and that the perfect retransmission channel of problem 2 is used if any errors are detected in a received codeword. This time, only the words which contain 3 or more errors contribute to the error count. Assume that each of these words contain only 3 errors and total up the number of bit errors. Multiply this total by 4/7 to get the number of message bit errors and plot the BER vs. S/N ratio on the same graph as before.

5. Consider the parity code used in problem 2. What S/N ratio gave a BER of 1E-4? A total 50,176 words were sent over the channel. How many message bits is this? How many total bits is this? For a BER of 1E-4 how many words retransmissions can be expected for 50,176 words? How many extra bits will this require (both message and check bits) to be sent? After the retransmission (assumed errorless) of the extra words, what is the total number of bits sent to achieve a BER of 1E-4? Divide this number by the number of message bits we wanted to send. This is the number of bits required to send one message bit over the channel with a BER of 1E-4 when using a parity check code with perfect retransmission.

6. What is the S/N need for a BER of 1E-4 in problem 3. If 50,176 words were sent, how many message bits does this correspond to? What is the total number of bits sent? Divide the two numbers to get a ratio similar to that in problem 5.

7. Repeat problem 5 using the (7,4) Hamming error detecting code.

8. What S/N ratio is required for a BER of 1E-4 for an uncoded system? What is the ratio of message bits to total bits? Compare the relative bandwidth requirements for each code. Compare the S/N ratios required for each code. Give an example of a situation in which the parity code would work best. Do the same for the uncoded and Hamming codes (both correcting and detecting).

NOTE: In many of the above questions, the codewords with multiple (3 or more) errors were not handled rigorously in the calculations. For example, in the (8,7) parity code, words with three errors would actually be detected and be retransmitted. Additionally words with 3 or more errors would contribute only 3 errors to the bit error count. It was assumed that such shortcuts would not effect the calculations very much. Why is this assumption valid?

REFERENCES


NOTES:

The lab PCs may be booted up in one of three modes: DOS only (no Windows running in the background), Windows98, or WindowsXP, depending on the experiment is to be run. For this lab, Binary Communication, the PCs should be booted in **DOS only**.

While the PC is booting up, hit the `<F8>` key immediately after the Dell screen. For DOS only, up arrow to **Safe Mode with Command Prompt** and hit `<Enter>`. For one of the Windows, down arrow to **Return to OS Choices Menu** and hit `<Enter>`. Then arrow to either **Microsoft Windows XP Professional** or **Microsoft Windows** (for 98) and `<Enter>`.

The procedure to boot the PC without Windows running in the background must be used every time the computer is used for real-time control applications in the Computer Applications Lab experiments (Binary Communications Lab). Repeated here as a reference, remember to restart the PC, if it is currently running Windows or turned off. While it is booting up – immediately after the Dell screen, hit the `<F8>` key. Choose **Safe Mode with Command Prompt** in the list. If Windows 98 loads again, you will need to do a restart selecting the DOS mode only option. After booting up, at the DOS prompt type: `CD C:\Cstudio\CAL_LAB`. A `dir` command will give you a list of available directories for the different experiments to which you can move (ex. `cd bincomm`) to run the appropriate executable programs.

When using the oscilloscopes to display waveforms, images on the scope may be save digitally to be included in the experiment reports. The easiest way to save images on the newest oscilloscopes is through the USB port to your thumb drive using the **Save** button after freezing the image on the screen. To do this on the older oscilloscopes insert a floppy in the scopes floppy drive and use the menu options to save the image to the floppy in a TIFF or some other format that can be understood by your PC. Again, the trace first should be frozen on the screen for best results. Generally there will be floppies available in the lab for use in transferring images from the scope to a lab desktop PC with its floppy drive where you can mail them to your RPI mail account.

If floppies are not available, female DB-9-to-female DB-9 null modem serial cables can be found in the lab that will connect the scopes to the PCs for transferring halted scope screen images. This method is recommended only as a last resort because transfers typically take over 3 minutes each while floppy writes take about 3 seconds. After connecting the scope serial port to the PC COM1 port, start **Word 54600 Toolbar** (WindowsXP) or **Start Word Toolbar Addin** (Windows98). Select the **Connect to Scope** icon to open the **Scope Control Properties** window. Under **Search Instruments** click on the **Search** button. Select the 54622D ... **AGILENT TECHNOLOGIES** device when it is found and click on the **Test** button. Select **OK** if everything is working. When there is an image on the screen to be saved, move the Word cursor to the desired insertion position, click on the camera icon (**Get Screen Image**) on the Agilent 54600 toolbar, and select the **OK** button in the **Get Screen Image** window that opens.
APPENDIX

Part III of the procedure requires the analysis of a lot of data obtained for the four types of coding systems being compared: uncoded, parity check code, Hamming code – single error correction, and Hamming code – double error detection. For the given BER requested in questions 5 - 8, it is suggested that the following table be used to help organize the collected data for systematic analysis.

<table>
<thead>
<tr>
<th></th>
<th>Uncoded</th>
<th>Parity Check Code</th>
<th>Hamming Code (single bit correction)</th>
<th>Hamming Code (double bit detection)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BER</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SNR for BER</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total Message Bits Sent</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Totals Bits Sent (message + check bits)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Words Retransmitted</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total # of Extra Bits Needed for Retransmission</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total Bits Sent for BER</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bits Required to Send One Message Bit (Efficiency Factor)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Advantages</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disadvantages</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Possible Uses</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>