



UNIVERSITY OF  
CENTRAL FLORIDA

DEPARTMENT OF  
ELECTRICAL & COMPUTER  
ENGINEERING

**EEL 4515**  
**FUNDAMENTALS OF**  
**DIGITAL COMMUNICATION**  
  
**LABORATORY MANUAL**

Revised  
August 2018

# Table of Content

• Safety Guidelines	
• Lab instructions	
• Troubleshooting hints	
• Experiment 1: Time Division Multiplexing	<b>1</b>
• Experiment 2: Pulse Code Modulation	<b>5</b>
• Experiment 3: Delta Modulation	<b>15</b>
• Experiment 4: Line Coding (Manchester)	<b>21</b>
• Experiment 5: Frequency Shift Keying	<b>25</b>
• Experiment 6: Phase Shift Keying	<b>33</b>
• Experiment 7: Matlab Project	<b>40</b>

# Safety Rules and Operating Procedures

1. Note the location of the Emergency Disconnect (red button near the door) to shut off power in an emergency. Note the location of the nearest telephone (map on bulletin board).
2. Students are allowed in the laboratory only when the instructor is present.
3. Open drinks and food are not allowed near the lab benches.
4. Report any broken equipment or defective parts to the lab instructor. Do not open, remove the cover, or attempt to repair any equipment.
5. When the lab exercise is over, all instruments, except computers, must be turned off. Return substitution boxes to the designated location. Your lab grade will be affected if your laboratory station is not tidy when you leave.
6. University property must not be taken from the laboratory.
7. Do not move instruments from one lab station to another lab station.
8. Do not tamper with or remove security straps, locks, or other security devices. Do not disable or attempt to defeat the security camera.

ANYONE VIOLATING ANY RULES OR REGULATIONS MAY BE DENIED ACCESS TO THESE FACILITIES.

I have read and understand these rules and procedures. I agree to abide by these rules and procedures at all times while using these facilities. I understand that failure to follow these rules and procedures will result in my immediate dismissal from the laboratory and additional disciplinary action may be taken.

---

Signature

Date

---

Lab #

# Laboratory Safety Information

## Introduction

The danger of injury or death from electrical shock, fire, or explosion is present while conducting experiments in this laboratory. To work safely, it is important that you understand the prudent practices necessary to minimize the risks and what to do if there is an accident.

## Electrical Shock

Avoid contact with conductors in energized electrical circuits. Electrocution has been reported at dc voltages as low as 42 volts. 100ma of current passing through the chest is usually fatal. Muscle contractions can prevent the person from moving away while being electrocuted.

Do not touch someone who is being shocked while still in contact with the electrical conductor or you may also be electrocuted? Instead, press the Emergency Disconnect (red button located near the door to the laboratory). This shuts off all power, except the lights.

Make sure your hands are dry. The resistance of dry, unbroken skin is relatively high and thus reduces the risk of shock. Skin that is broken, wet, or damp with sweat has a low resistance.

When working with an energized circuit, work with only your right hand, keeping your left hand away from all conductive material. This reduces the likelihood of an accident that results in current passing through your heart.

Be cautious of rings, watches, and necklaces. Skin beneath a ring or watch is damp, lowering the skin resistance. Shoes covering the feet are much safer than sandals.

If the victim isn't breathing, find someone certified in CPR. Be quick! Some of the staff in the Department Office are certified in CPR. If the victim is unconscious or needs an ambulance, contact the Department Office for help or call 911. If able, the victim should go to the Student Health Services for examination and treatment.

## Fire

Transistors and other components can become extremely hot and cause severe burns if touched. If resistors or other components on your proto-board catch fire, turn off the power supply and notify the instructor. If electronic instruments catch fire, press the Emergency Disconnect (red button). These small electrical fires extinguish quickly after the power is shut off. Avoid using fire extinguishers on electronic instruments.

## Explosions

When using electrolytic capacitors, be careful to observe proper polarity and do not exceed the voltage rating. Electrolytic capacitors can explode and cause injury. A first aid kit is located on the wall near the door. Proceed to Student Health Services, if needed.

# Instructions

## General Policies

1. Arrive on time.
2. Frequent late arrival shall be penalized.
3. When a lab session has to be cancelled or rescheduled, a notification will be posted outside the Lab prior to that session.
4. Each experiment is performed in two lab sessions, a pre-lab and an implementation session, (excluding Experiment 7: Matlab Project, does not require lab attendance).
5. Attendance and participation in all experiments are mandatory for every student.
6. Missing any of the lab sessions without prior permission or legitimate reason may result in losing the experiment grade.
7. Every lab report submitted late without any legitimate reason and notification in advance will be penalized by 10% per weekday. Lab reports that are late for more than FIVE weekdays will not be accepted.
8. Before you come to the lab for an experiment:

Make sure you have familiarized yourself with the material related to the experiment. The needed theoretical background can be found in your textbook and lecture notes. Each experiment also has a theoretical background section.

## Lab Reports

- Lab reports have to be legible, detailed and presentable.
- Students are encouraged to include tables, graphs and figures when presenting their results.
- Answers to questions in each experiment should be clear and detailed.
- Make sufficient comments concerning your observations during the experiment.
- Lab report should include the following:
  - Cover page containing name(s), course number, title and section, experiment number and title, and due date.
  - Experiment objectives in your own words.
  - Experimental procedures
  - Presentation of analytical, simulation and experimental results.
  - Conclusions.
  - Answers to lab questions.
- Put emphasis on conclusions that might be drawn from the comparison of theoretical results and experimental ones.

## Grading Policy

To be uploaded to webcourses or given in the lab by instructor.

# Troubleshooting Hints

1. Be sure that the power is turned on.
2. Be sure that the ground connections are common.
3. Be sure that the circuit you built is identical to that the diagram. (Do a node-by-node check).
4. Be sure that the supply voltages are correct.
5. Be sure that the equipment is set up correctly and you are measuring the correct parameter.

If steps 1 through 5 are correct, then you probably have used a component with the wrong value or one that doesn't work. It is also possible that the equipment does not work (although this is not probable) or the breadboard you are using may have some unwanted paths between nodes. To find your problem you must trace through the voltages in your circuit node by node and compare the signal you have to the signal you expect to have. Then if they are different use your engineering judgment to decide what is causing the difference or ask your lab assistant.

# Lab 1: Time Division Multiplexing

---

## Objective

In this experiment, the aim is to study the Time Division Multiplexing of two band-limited sinusoidal signals.

## Theory

Time-division multiplexing (TDM) is a method of transmitting and receiving independent signals over a common signal path by means of synchronized switches at each end of the transmission line so that each signal appears on the line only a fraction of time in an alternating pattern. It is used when the bit rate of the transmission medium exceeds that of the signal to be transmitted. This form of signal multiplexing was developed in telecommunications for telegraphy systems in the late 19th century, but found its most common application in digital telephony in the second half of the 20th century. In Digital Multiplexing, several low bit-rate signals can be multiplexed or combined, to form one high bit-rate signal, to be transmitted over a high frequency medium. Because the medium is time-shared by various incoming signals, this is a case of TDM. Multiplexing can be done on a bit-by-bit basis (known as digit interleaving), or on a word-by-word basis (known as byte or word-interleaving).

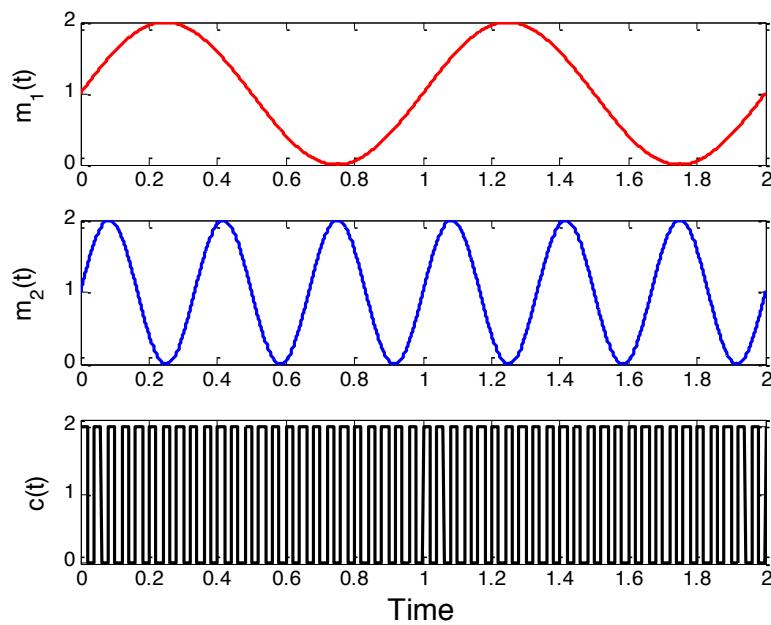


Figure 1.1 (a): Two input sinusoidal signals to be combined in time and a carrier signal

At the receiving terminal, the incoming digit stream must be divided and distributed to the appropriate output channel. For this purpose, the receiving terminal must be able to correctly identify each bit. This requires the receiving system to uniquely synchronize in time with the beginning of each frame, with each slot in a frame and with each bit within a slot. This is accomplished by adding framing and synchronization bits to the data bits. These bits are part of the so-called overhead bits.

Figure 1.1 (a) and (b) shows the inputs signals and their corresponding TDM signal.

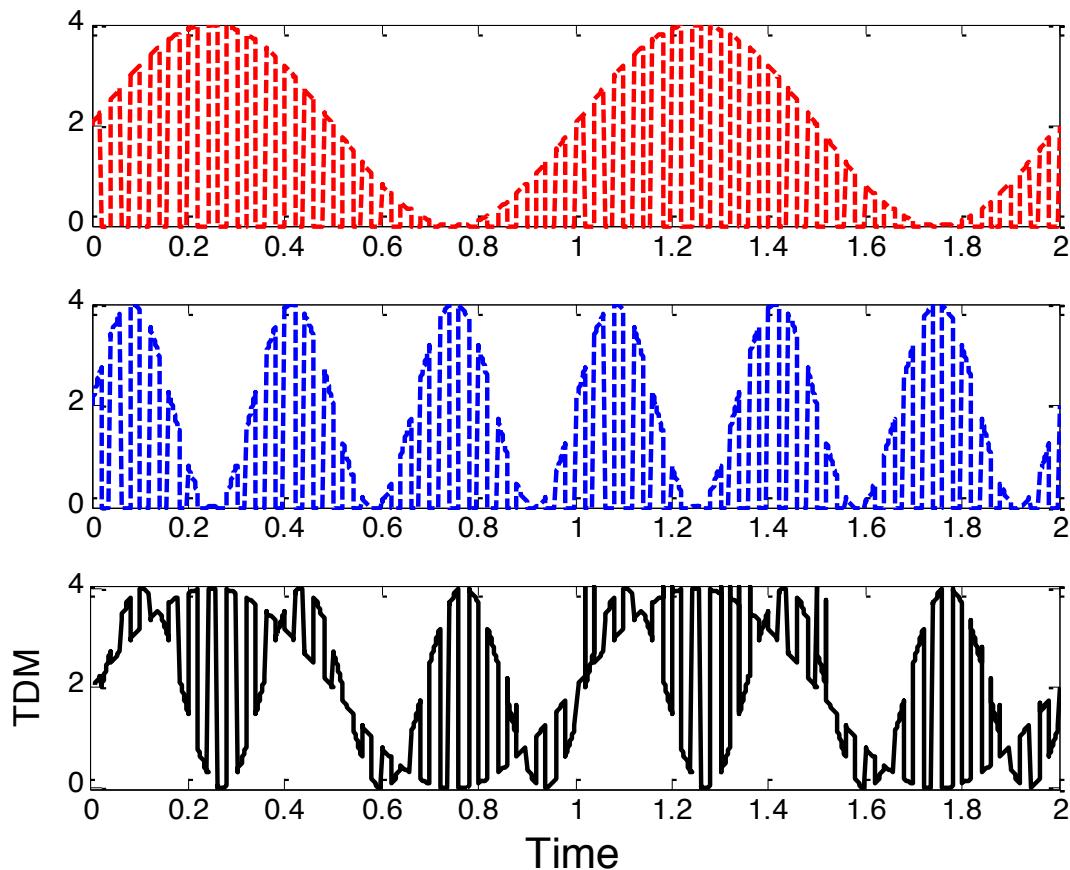


Figure 1.1(b): Demonstration of two signals sampled and multiplexed in time

## Procedure

1. Connections are made as shown in Figure 1.2.
2. Apply a square wave (TTL) carrier signal of 2 kHz (or >2 kHz) of 5V amplitude.
3. Apply  $m_1(t)$  and  $m_2(t)$  whose frequencies are  $f_1$  (200 Hz, with DC offset) and  $f_2$  (400 Hz, with DC offset).
4. Observe TDM waveform at pin number 3 of IC CD4051.

5. Observe the reconstructed message waveforms  $m_1(t)$  and  $m_2(t)$  at pin numbers 13 and 14 of 2<sup>nd</sup> IC CD4051.

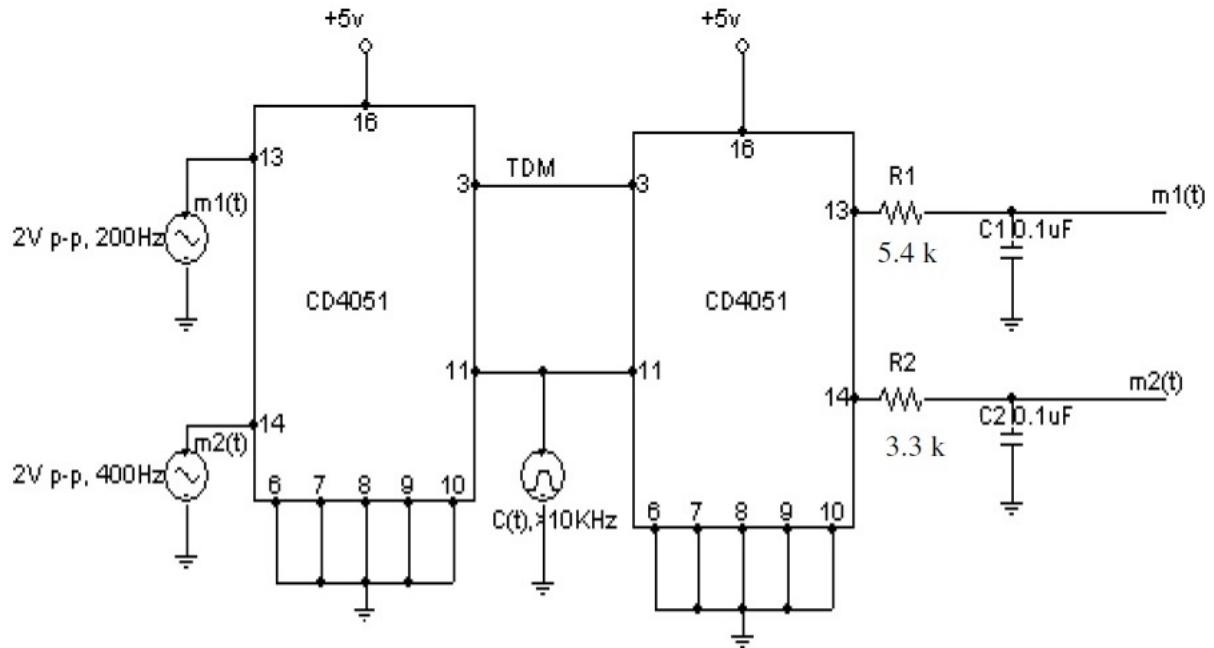


Figure 1.2 Time division multiplexing/demultiplexing circuit diagram

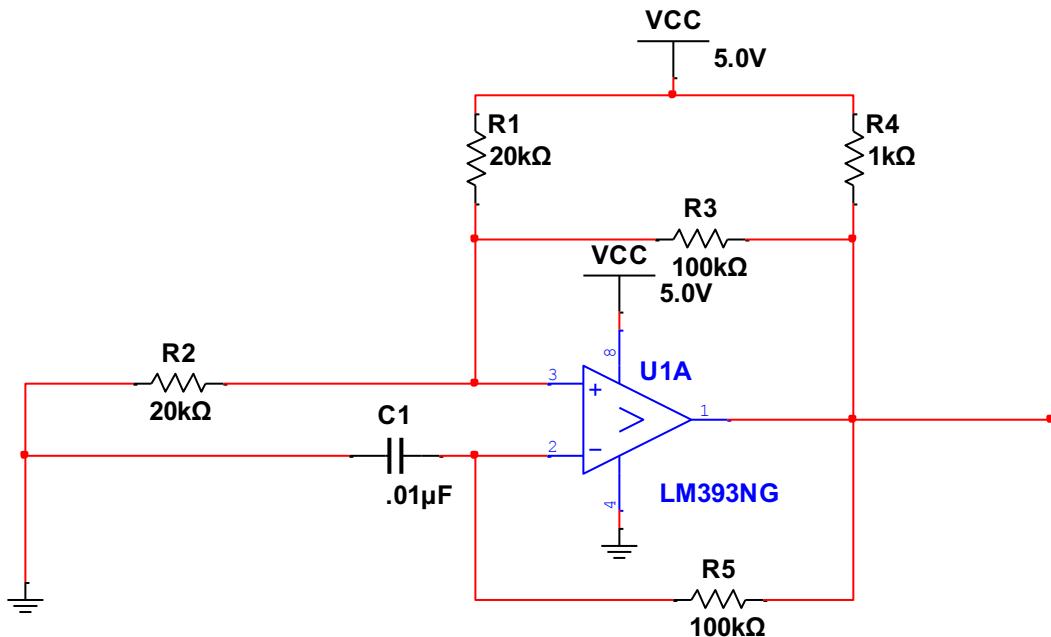


Figure 1.2 (b) Square wave clock generator to be connected to pin 11 to the CD4051 ICs.

## Questions

1. Discuss the role of each element of the TDM system.
2. How to increase the quality of the demultiplexed signals?
3. How to change the frequency of the square wave generated by Figure 1.2(b)?
4. For each one of the 6 signals plotted in Figure 1.1 (a) & (b), approximately sketch its corresponding spectrum (frequency domain).

# Lab 2: Pulse Code Modulation (PCM)

---

## Objective

The objective of this lab is to get familiarized with Pulse Code Modulation. Students are expected to write a Matlab code that will help them to understand the fundamentals of PCM. They will also observe the modulation and demodulation process of PCM using a pre-built hardware board.

## Theory

Pulse Code Modulation (PCM) is a method of converting an analog signal into a digital signal (A/D conversion). Analog signal is characterized by the fact that its amplitude can take on any value over a continuous range. This means that it can take on an infinite number of values. But, a digital signal amplitude can only take finite number of values.

In PCM an analog signal is first sampled at a rate higher than the Nyquist rate, and then the samples are quantized. It is assumed that the analog signal is distributed on an interval denoted by  $[-X_{\max}, X_{\max}]$ , and the number of quantization levels is large. The quantization levels can be equal (uniform PCM) or unequal (Non-uniform PCM). In this lab, we will only consider the uniform PCM. Topics on non-uniform PCM may be found in the following reference.

B. P. Lathi, “Modern Digital and Analog Communication Systems – 3rd edition”, pages 267-271, Oxford University Press, 1998

### ➤ Uniform PCM

In uniform PCM the interval  $[-X_{\max}, X_{\max}]$  of length  $2X_{\max}$  is divided into  $N$  equal subintervals, each of length  $\Delta = (2X_{\max}) / N$ . If  $N$  is a power of 2 or  $N=2^v$ , then  $v$  bits are required for representation of each level.

As we discussed, after quantization, the quantized levels are encoded using  $v$  bits for each quantized level. The encoding scheme that is usually employed is *natural binary coding* (NBC), meaning that the lowest level is mapped into a sequence of all 0's and the highest level is mapped into a sequence of all 1's. All the other levels are mapped in increasing order of the quantized value.

Figure 2.1 shows an analog signal which amplitude lies in the range of  $[-X_{\max}, X_{\max}]$ . The number of levels to which the signal is quantized is  $N=8$ . In this case, each sample requires  $v=3$  bits to represent digitally.

The message is periodically sampled and digitally encoded in PCM. Since the signal is digitally encoded, only certain discrete voltage levels can be represented and there will be some error in encoding a random analog signal. This error is known as the ‘quantization error’. The ratio of the signal power to quantization error power is generally termed as SQNR (Signal to Quantization Noise Ratio).

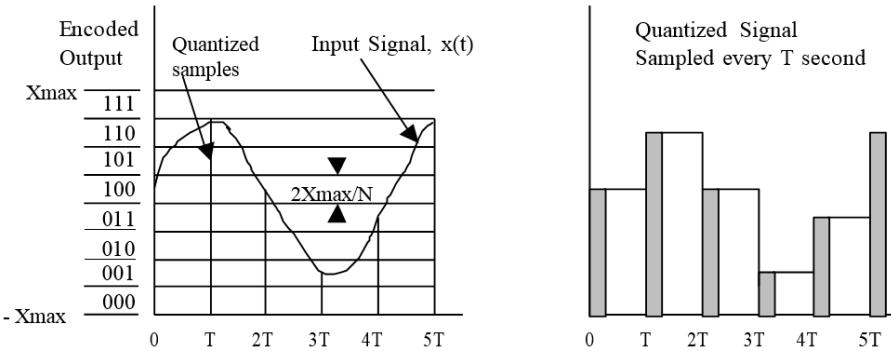


Figure 2.1 Quantization of a sampled analog signal

## Simulation

In this simulation, we will quantize a sinewave signal (message) and encode it to binary bits. We will reconstruct the sinewave from binary bits. We will also calculate the SQNR for the 8-level and 16-level quantization.

For the simulation, generate a sinusoidal signal with amplitude 1, and  $\omega=1$ . Using a uniform PCM scheme, quantize it once to 8 levels and once to 16 levels. Plot the original signal and the quantized signals on the same axis. Compare the resulting SQNRs in the two cases.

We arbitrarily choose the duration of the signal to be 10s. The resulting SQNRs are 18.90 dB for the 8-level PCM and 25.13 dB for 16-level PCM, compare these values with the results that you get from your simulation. The plots are shown in figure 2.2.

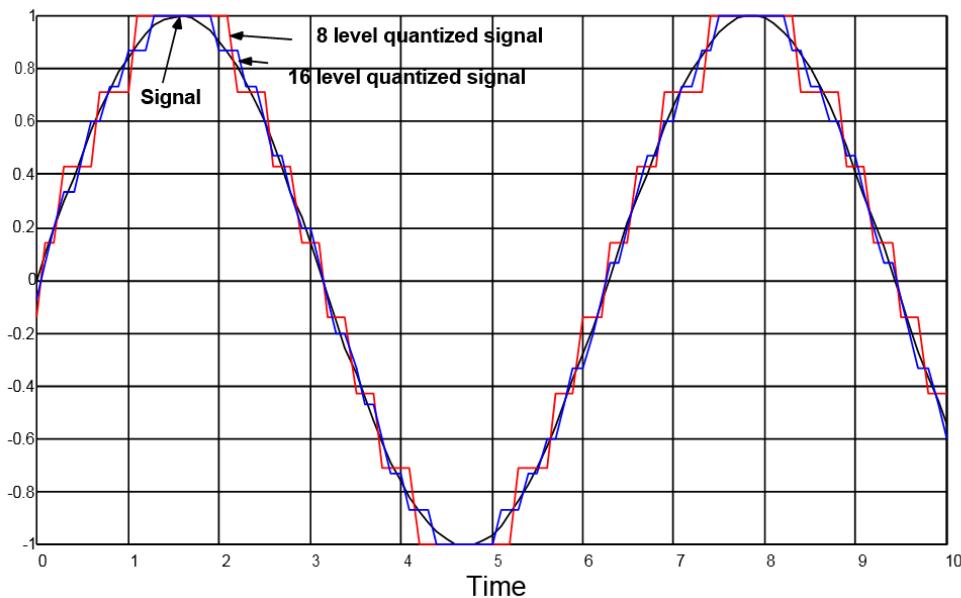


Figure 2.2 Sinusoidal signal before and after quantization.

➤ **Matlab functions that may be needed for this lab:**

Length, abs, max, round, log10, dec2bin, plot, grid, title, X label, Y label, Subplot, Figure

➤ **Simulation procedure:**

The following procedure only gives you an idea of how the PCM modulation and de-modulation are performed. Actual implementation may vary.

1. Construct a time array from 0 to 10 sec with 0.1 intervals. Name it ‘t’.
2. Find the size of the array t. Use ‘length’ function. Name the size ‘m’.
3. Construct signal array ‘a’. Signal is a sinewave with unity amplitude and angular frequency.
4. Assign the number of quantization level n equal to 8.
5. Calculate,  $a_{max} = \max(\text{abs}(a))$ . See figure 2.3(a).
6. Calculate the following as shown in figure

$$b = a + a_{max}$$

$$c = (n-1) \times (b/(2a_{max}))$$

$$d = \text{round}(c)$$

$$a\_quan = 2 * \text{amax} * d / (n-1) - \text{amax}$$

$$a\_error = a - a\_quan$$

7. Calculate

$$S = \sum_{i=1}^m a(i)^2$$

$$N = \sum_{i=1}^m a\_error(i)^2$$

8. Calculate Signal to noise ratio,  $\text{SQNR} = 10 \log_{10}(S/N)$
9. Calculate binary coding to corresponding quantized value using `dec2bin` function.
10. Repeat 4-9 with quantization level  $n=16$ .
11. Plot the input signal and quantized signal as in Figure 2.2. Use black solid line for signal, red solid line for 8 levels quantized and blue solid line for 16 level quantized signals. Use ‘figure’ function for multiple figures (you have to plot another figure in step 12).
12. Plot your variables  $a$ ,  $b$ ,  $c$ ,  $d$ ,  $a\_quant$  and  $a\_error$  for quantization level 8. Show your plots exactly as shown in the Figure 2.3 in a single template. Use ‘subplot’ function for that.
13. Compare the two quantized signals. Which is more close to input signal?
14. Tabulate few sample binary codes for both cases.
15. Reconstruct the original sinewave from the binary codes (both) using the Matlab function ‘`bin2dec`’. How is it different from the original signal?

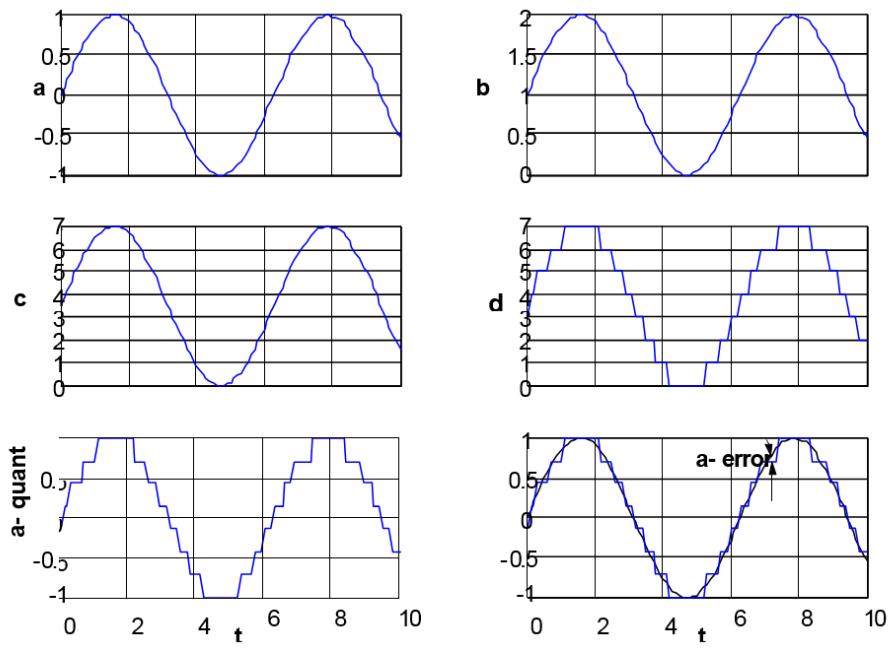


Figure 2.3 Steps of quantization in simulation

## Implementation

### ➤ Synchronization

When the encoded bits are transmitted, how does the receiver know when the bit sequence starts and when it ends?

For this type of encoding it is possible that some synchronization bits are transmitted periodically. The receiver can synchronize itself using these synchronizing bits. Additionally, some guard bits are needed to separate adjacent codes. The total number of bits necessary to transmit to be able to decode a signal is therefore much larger than the number of bits containing information. The pre-built board ‘Module 296f’ will be used for this lab. This board implements PCM with two “0”s as the guard bits, the four information bits, and two more guard “0”s as followed by eight “1” as synchronization bits (Figure 2.8).

The Figure 2.4 shows the complete PCM system. In the transmitter the input message signal  $f(t)$  is sampled, quantized, and encoded to binary bits. The serial bit stream represents the PCM output. At the receiver the PCM bit stream is decoded. After D/A conversion the message signal is reconstructed. Reconstructed signal represents the original message signal with some quantization error.

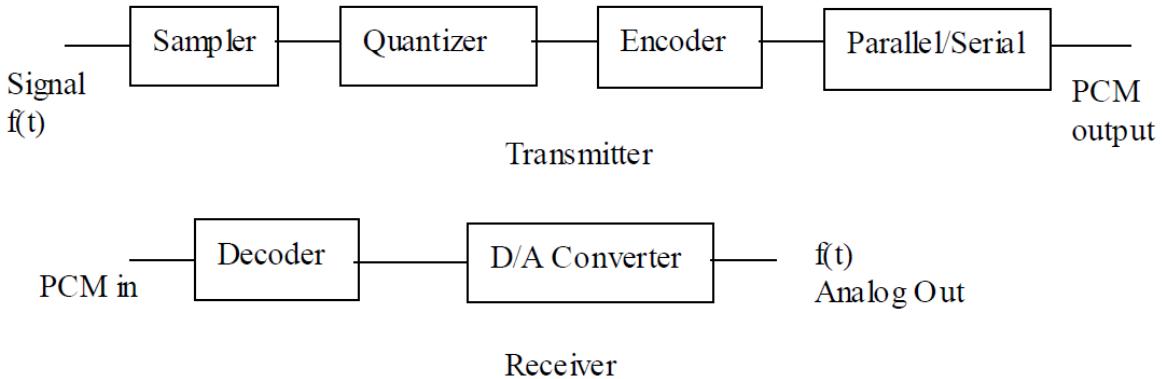


Figure 2.4: PCM system: Transmitter & Receiver

The module 296f encodes a signal by comparing the input signal level to an 8 or 16-step ramp. Each step ramp represents the current state of the binary counter. To encode PCM, the ramp must repeat itself at least at Nyquist frequency, which is higher than the frequency of the message. The message signal and ramp are inputs to a comparator, which has two output states high, and low. When the ramp signal exceeds the message signal, the comparator output swings low and the counter value is latched and is held until the ramp repeats.

### ➤ Decoding

When the data bits are separated from the bit streams (removing the synchronization and guard bits), they are sent to a D/A converter, where each bit is given the appropriate weight in voltage, and the voltage due to each bit is summed.

### ➤ Modulator

As shown in the Figure 2.5, the clock frequency  $f_1$  feeds to the “clk” line of the 4-bit counter and also the shift-left control line of the 8-bit register. The clock  $f_1$  simultaneously increases the counter and shifts left the 8-bit register.

The comparator has  $f(t)/k$  (where  $k$  is an attenuation constant) and a ramp signal as the + and - inputs respectively. The output of the comparator drives the load line of the 4-bit latch. The comparator output is high until the ramp input becomes higher than  $f(t)/k$ . at that point, the last input to the latch stays for the rest of cycle until counter is reset to 0000.

Synchronously with the counter reset the  $f_2$  clock signal goes high loading the output of the latch to the 8-bit register. Note that only the four counter bits are loaded from the latch, the other four are grounded and therefore loaded as 0.

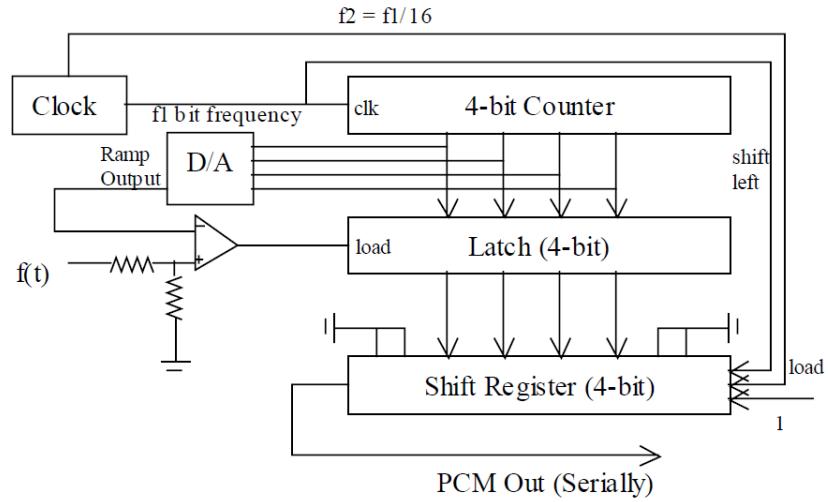


Figure 2.5: PCM Modulator

At this point the cycle starts again. The counter starts counting and the 8-bit register starts shifting out serially the 8 bits loaded in it. Note that every time a bit is shifted out serially, another bit is taken as input in the rightmost bit of the registers from the serial input. In our cases this input is connected to a logic “1” and are the synchronization bits. An example is given in the Figure 2.6 (a) & (b).

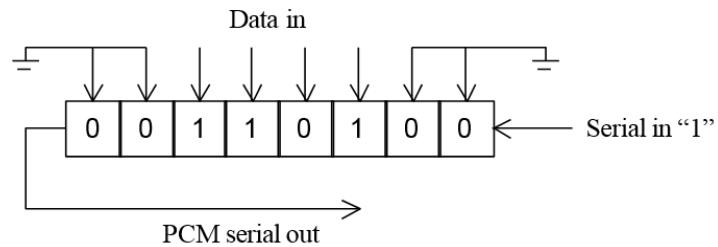


Figure 2.6 (a): Status of the shift register as it shifts the bits

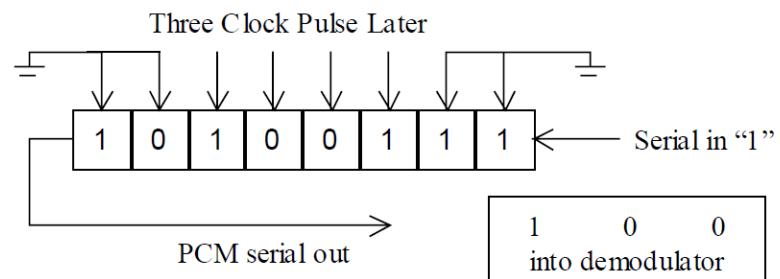


Figure 2.6 (b): Status of the shift register as it shifts the bits

## ➤ Demodulator

The synchronization circuit found in Figure 2.7 recognizes the eight “1” used to synchronize and with the last “1” turns the clock on (same frequency as f1). The 8-bit register on the demodulator will load serially the next eight bits coming from the modulator.

The same clock runs a 3-bit counter with its “carry” line connected to a load control line of latch. The latch will load the 4 counter bits on the 8-bit register when the counter is reset to 0000. Note that it takes nine clock pulses to reset instead of eight. In order to avoid an extra bit to be shifted-in three outputs from the counter are NAND together and tied to the clock. This avoids the 8 bit register to shift in another bit when the counter output is “111”.

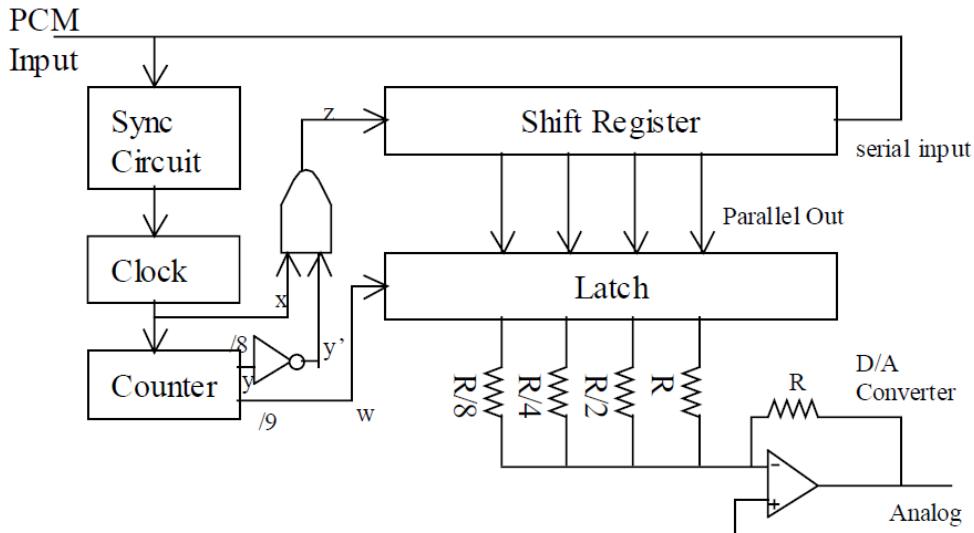


Figure 2.7: PCM Demodulator

## Hardware Procedure

### ➤ Encoding

Plug module 296f into the power supply (wooden adaptors are available). Set the clock control to ‘slow’ and the 3 bit /4 bit switch to ‘4 bit’.

Set the oscilloscope to ‘dc’ and display the ‘ramp output’ signal on the scope. The state of the counter is displayed on the LED’s, and you should see all the binary numbers between 0000 and 1111 (0 and 15, decimal) displayed while you see the voltage step up on the scope. The voltage level displayed on the scope is an analog equivalent of the counter value, determined by the internal A/D converter.

To see the full ramp at once, set the clock control to ‘fast. What are the voltage limits of the ramp at the ‘ramp output’?

Return the clock control to ‘slow’.

Set the analog input to the following DC values and record the binary number indicated as shown on LED display.

Dc Input (V)	Number latched
-4	
-3	
-2	
-1	
0	
1	
2	
3	
4	

Table 2.1: Record the number latched for different dc inputs.

### ➤ Decoding

Set the clock control to ‘fast’. Apply a 4V peak, 0.2Hz sinusoid to the ‘analogue input’.

Set both scope inputs to ‘dc’. Display the ‘PCM output’ along with the ‘analogue input’ on the scope.

Draw one full period of the ‘PCM out’, identifying the sync bits, guard bits, and the information bits.

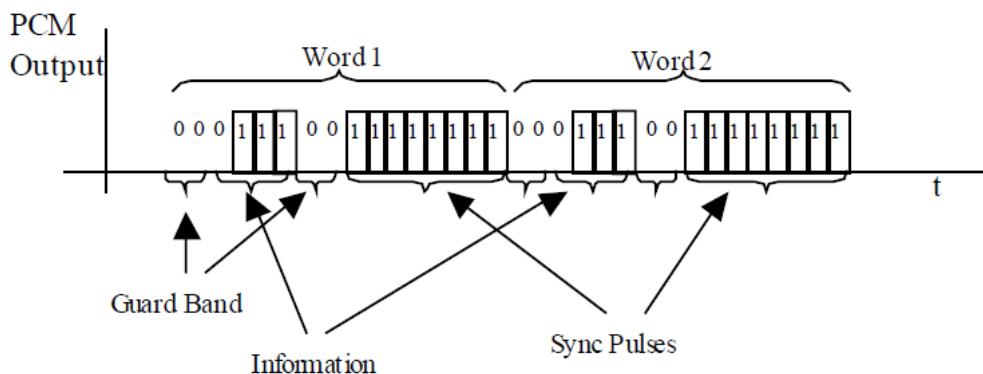


Figure 2.8: PCM Output: What it should look like

Change the input signal to a 2.5V peak, 500Hz sine wave.

Connect ‘PCM output’ to the ‘PCM’ input and make all grounds common.

Display the ‘analogue input’ together with ‘analogue output’ on the scope, where ‘analogue output’ is the demodulated message. Is the input signal accurately reproduced? Explain any differences.

Press the ‘inhibit sync’ button on the decoder clock. What happens? Is synchronization necessary for PCM?

## Questions

Consider again the output signal from step (6) of the decoding portion in section 2.4. What could you do to improve the demodulated signal? What would that cost be? Use bandwidth, data storage, cost or any other relevant factor as your basis.

*Hint: Think about the quantization and its effect.*

# Lab 3: Delta Modulation

## Objective

Introduction of basic concepts of Delta Modulation (DM) system with simulation and hardware implementation.

## Theory

With Delta Modulation (DM) a train of fixed width pulses is transmitted, whose polarity indicates whether the demodulator output should rise or fall at each pulse. The output is caused to rise or fall by a fixed step height at each pulse. A block diagram of DM system is given in Figure 3.1.

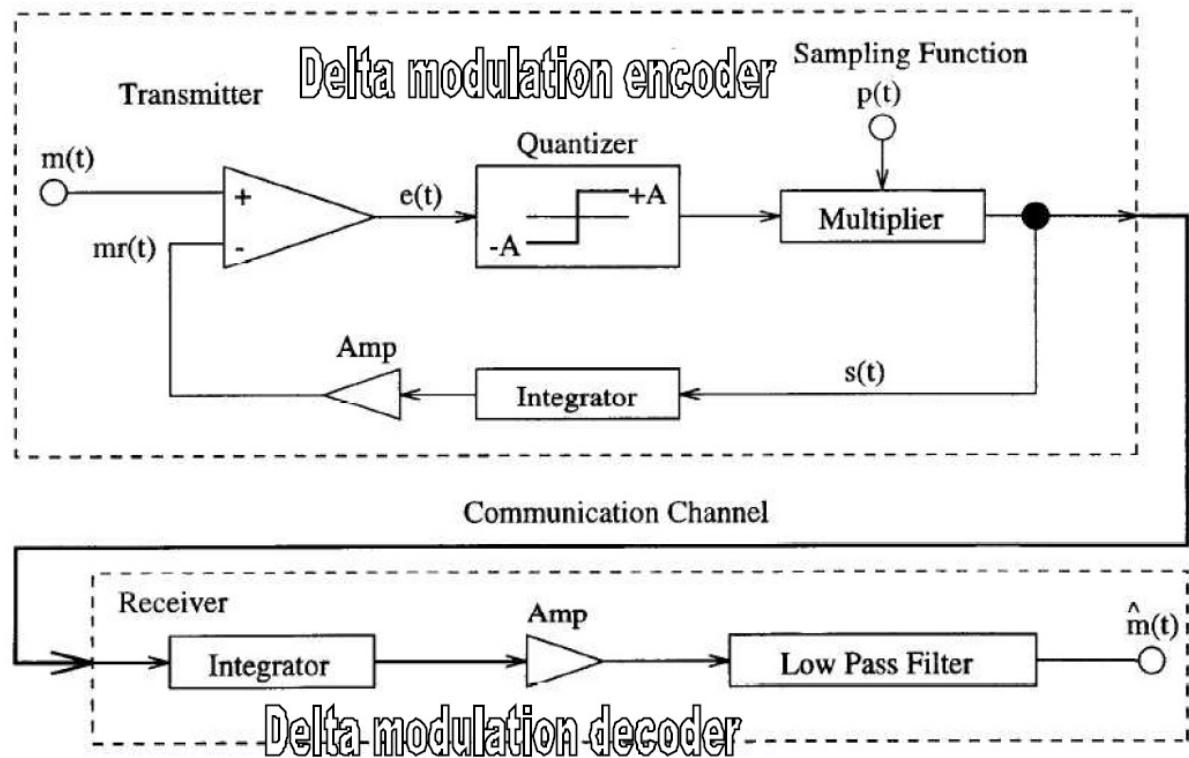


Figure 3.1: Delta modulation system. Upper block is for encoder and lower block is for decoder.

The modulating message signal  $m(t)$  is applied to the non-inverting input of a high gain differential amplifier. The input analog signal  $m(t)$  is compared with the loop estimated signal  $mr(t)$  to generate the error signal  $e(t)$ . Then, this error signal is fed into a two level quantizer. The quantized

error signal is multiplied by the sampling function  $p(t)$  to generate the output  $s(t)$ . The signal  $s(t)$  is integrated and amplified to generate the loop estimate  $mr(t)$ .

In this implementation, the sampling function is a train of narrow pulses with unit amplitude (most practical implementations also). The multiplier becomes a switch and the sampling function becomes the switch enable input. The output  $s(t)$  is a sequence of pulses with amplitude  $+A$  or  $-A$ . It is very important to note that the DM process samples the quantized error function and not the input signal itself and for narrow sampling pulses the estimated signal  $mr(t)$  is a stepwise approximation of  $m(t)$ .

The DM process is a process of analog to digital conversion. Quantization error is present at the output. Two types of quantization noise are present in delta modulation, granular noise and slope overload noise.

Granular noise is due to the use of finite quantization steps to reconstruct (approximate) the input signal. It is similar to the quantization noise observed in pulse code modulation (PCM) systems.

Slope overload noise occurs whenever the slope of the input signal exceeds the DM maximum slope or the input signal changes between samples, by an amount greater than the step size of the DM.

DM approximate a waveform by a linear staircase function, the waveform must change slowly relative to the sampling rate. This requirement implies that waveform must be oversampled, i.e., at least five times the Nyquist rate.

"Oversampling" means that the signal is sampled faster than is necessary. In the case of Delta Modulation this means that the sampling rate will be much higher than the minimum rate of twice the bandwidth. Delta Modulation requires "oversampling" in order to obtain an accurate prediction of the next input. Since each encoded sample contains a relatively small amount of information Delta Modulation systems require higher sampling rates than PCM systems.

## Simulation

The simulation in this lab includes the modulation (encoding) and demodulation (decoding) steps.

1. Construct a time array from 0 to 1/25 seconds with  $ts=1/fs$  interval (sample frequency  $fs=2000$ ). Name it '**tn**'.
2. Set the step size 1/15. Name it '**StepSize**'.

3. Construct a signal '**m**' which is a sine wave with 0.5 amplitude and frequency of 50, time is '**tn**'. The signal '**m**' is our modulating message signal.

4. Call **function** Delta modulation encoder '**s=DM\_Encoder(m, StepSize)**'

With input signal '**m**' and step size "**StepSize**" as the parameter to the function. '**s**' is the output. The following steps describe the procedure to write this function.

4.1 Use '**length**' to find the size of the signal **m**, name it '**xlen**'.

4.2 Set accumulation at start is zero.

4.3 Compare input signal(s) with accumulation (i) from beginning to **end**.

When signal is greater than accumulation, encoder out  $s(i)=+1$ , next accumulation (i+1) equals accumulation (i) plus step size.

When signal less than accumulation,, encoder out  $s(i)=-1$ , next accumulation(i+1) equals accumulation(i) minus step size.

5. Call **function** Delta modulation decoder '**[Si]=DM\_Decoder(StepSize,s)**'

With input signal '**s**' and step size "**StepSize**". Output '**Si**'.

5.1 Use '**length**' find size of signal (**s**), name it '**xlen**'

5.2 Set accumulation at start is zero.

5.3 Compare input signal **s(i)** with zero from beginning to **end**. When **s(i)** is greater than 0, next accumulation(i+1) equals accumulation(i) plus step size.

Else, next accumulation (i+1) equals accumulation (i) minus step size.

5.4 Set output '**Si**' equal equals accumulation (2:i+1)

6. Call **function** low pass filter '**So=LowPassFilter(100, 4\*50/fs, Si);**'

Input signal '**Si**', Output '**So**'.

This function program is given in following.

```
function So=LowPassFilter(fod, cf, Si)
%lowpass filter
%So=LowPassFilter(100, .1, Si);
%
%fod: filter order.
%cf: cut-off frequency.
%So: output.
```

```
b=fir1(fod,cf);
size(b)
So = conv2(Si,b,'same');
```

7. Plot the signal of input '**m**', Decoder output '**Si**' and Lowpassfilter output '**So**' vs time. Use '**subplot**' function plot them in same template.
8. By change the (1) fs to 500, 1000, rerun program. What happen? Why?
9. By change the (2) step size to 1/20, 1/40 rerun program. What happen? Why?
10. Submit a report and the soft copy of your codes.

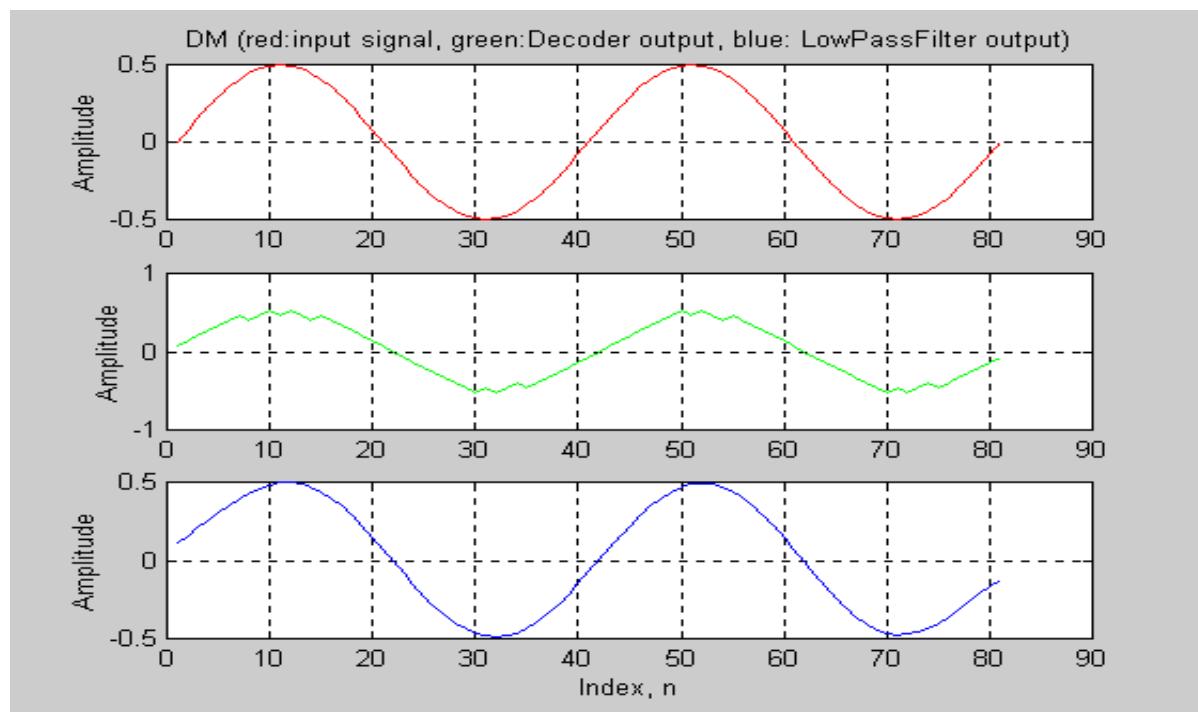


Fig 3.2. Top sinewave signal is given as the input to the modulator. Next signal is the decoder output. The last signal is the low pass filter. So it is faithful reproduction of the original message signal.

## Hardware Experiment

For the hardware experiment we will focus on the encoding as well as estimation process of the Delta Modulation (DM).

The circuit for DM modulation is given in Figure 3.3. Which consists of an LM741 operate amplifier, CD4013 dual ‘D’ flip flop and CD4016 bilateral switch. The LM 741 is operated open loop as a comparator between the input signal  $m(t)$  and the feedback signal  $mr(t)$ . The function of the CD4013 is to hold the value of the quantized error signal constant (+ or – Vcc) during the sampling period. Both the flip-flop and the bilateral switch are enabled by same clock pulse. Propagation delay in the flip-flop may be considered negligible in comparison with the pulse width. A dc integrator is used in the feedback loop.

## Procedure

1. Assemble the DM circuit as given in Figure 3.3.
2. Set the signal generator to generate a square wave with amplitude =10 volt (p-p) with 5V dc and frequency=10 kHz.  
(Hint: Use Channel 1 for the Clock and Channel 2 of the function generator to generate  $(t)$ )  
Note: Clock input should be within 0 to +Vdd. The CD 4013 IC will burn with negative voltage.
3. Generate a 200 Hz, 1.0 Vpp sine wave with 0.5 V dc offset with the other signal generator. Same precaution applies here as written in the previous step.
4. Connect the output of the square wave generator to the clock input of the DM circuit. Connect the sine wave to the message input of the DM circuit.
5. Check the waveforms at the output of the flip-flop and the sampler. Compare the output of the integrator  $mr(t)$  with the input waveform  $m(t)$  by superimposing both signals. Plot all waveforms and measure the step size. Explain your observations. Vary the following parameters and observe the changes in the reconstructed signal  $mr(t)$ . Record the waveforms for each case.
6. Vary the message frequency between 50 Hz and 1 KHz. Vary the message amplitude between 0 and 2Vpp.
7. Vary the sampling frequency between 400Hz and 100 KHz.

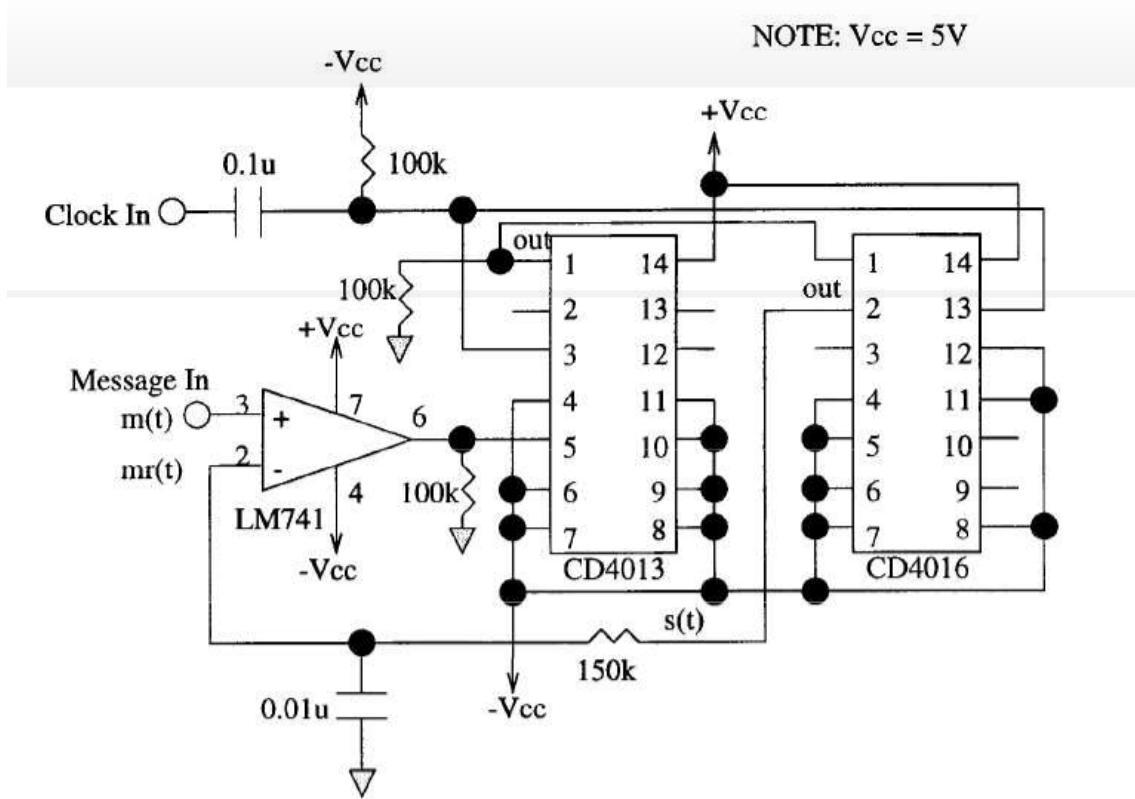


Figure 3.3: Delta Modulation circuit.

## Questions

Delta modulation is special case of Differential PCM where each sample is represented by just 1 bit – explain.

(Hint: See the sections 6.3 and 6.4 of your textbook)

# Lab 3: Manchester Coding

---

## Objective

The purpose of this experiment is to be familiarized with the basics of line coding, i.e., mapping bits to pulses. We focus on Manchester coding in generation using Matlab and observing the spectrum of the Manchester encoded signal on the spectrum analyzer.

## Theory

Digital Line Coding is a special coding system chosen to allow transmission to take place in a communications system. The chosen code or pattern of voltage used to represent binary digits on a transmission medium is called line encoding. Primarily, there are three major categories of line coding: Unipolar, Polar, and Bipolar. As discussed in the lectures, the main purpose of line coding is to enhance the signal transmission. This includes, having a small transmission bandwidth, enhancing the power efficiency for the required data rate, and having a suitable power spectral density with little low frequency content. Some line codes enable the receiver to extract the clock from the received signal. In Fig. 4.1  $p(t)$  is the pulse to be transmitted representing a ‘1’ and  $-p(t)$  to be transmitted representing a ‘0’ bit.

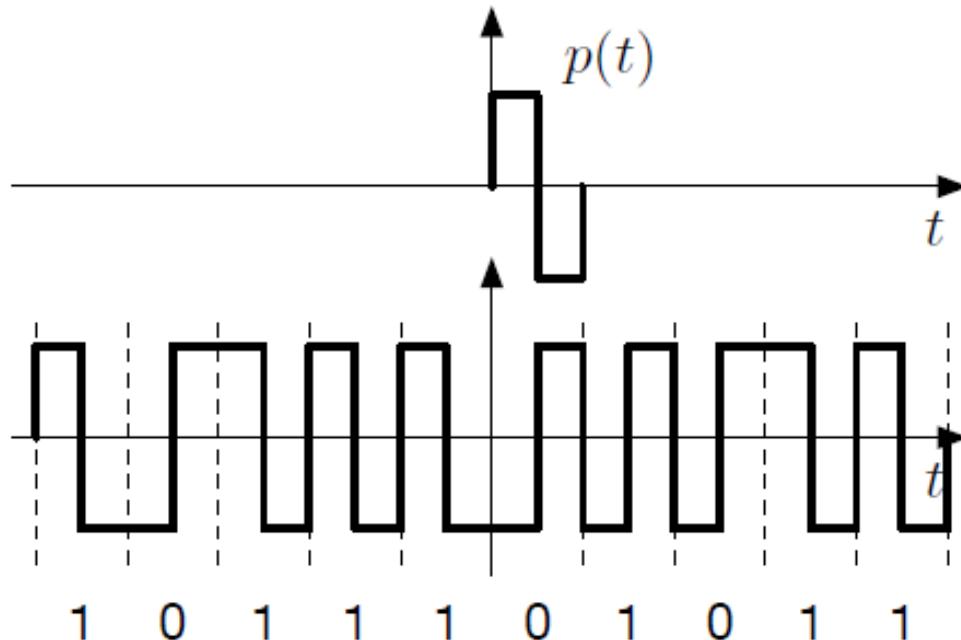


Figure 4.1 Phase splitting encoding (Manchester)

A ‘0’ is expressed by a low-to-high transition, a ‘1’ by high-to-low transition.

## Simulation

Write a Matlab Code to observe the spectrum of a Manchester encoded signal. Make your remarks on the advantages of the spectrum of the Manchester encoded signal.

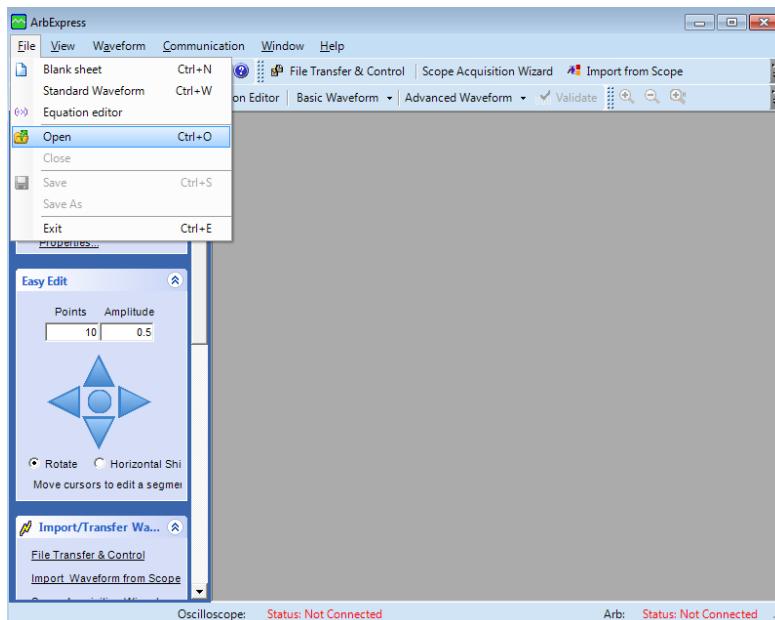
Hints:

- 1- Generate a random binary sequence of length N ( $N > 10000$ ).
- 2- Develop a time signal ‘x’ to represent the random signal in time.
- 3- Encode the signal using Manchester Coding to obtain signal ‘m’.
- 4- Find the Fourier transform of the encoded signal ‘m’ using the *fft()* function to obtain ‘M’.
- 5- Plot the following figures
  - a) Original signal ‘x’ versus time.
  - b) Encoded signal ‘m’ versus time.
  - c) Spectrum of the encoded signal ‘M’ versus frequency.
- 6- Save the generated signal ‘m’ as a comma separated value file, ‘.csv’.

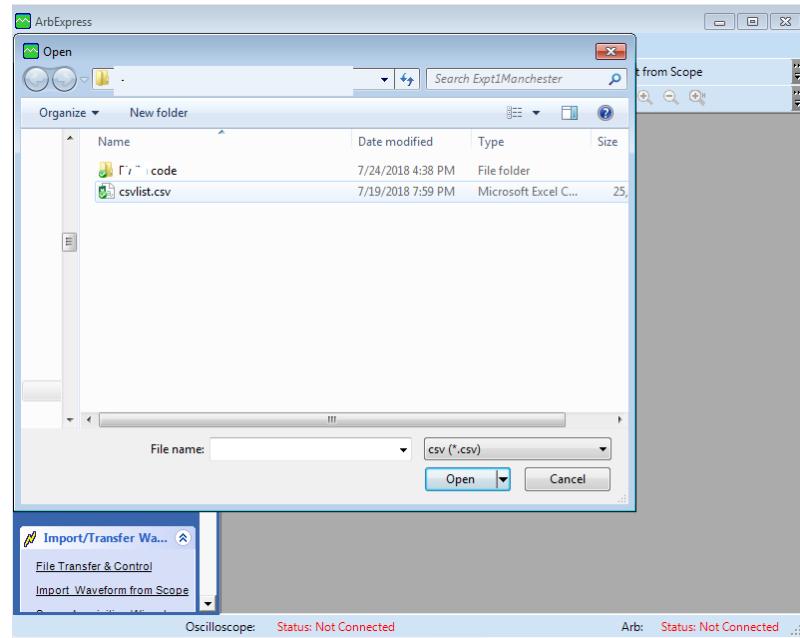
```
csvwrite('csvsignal.csv', m)
```

## Experiment

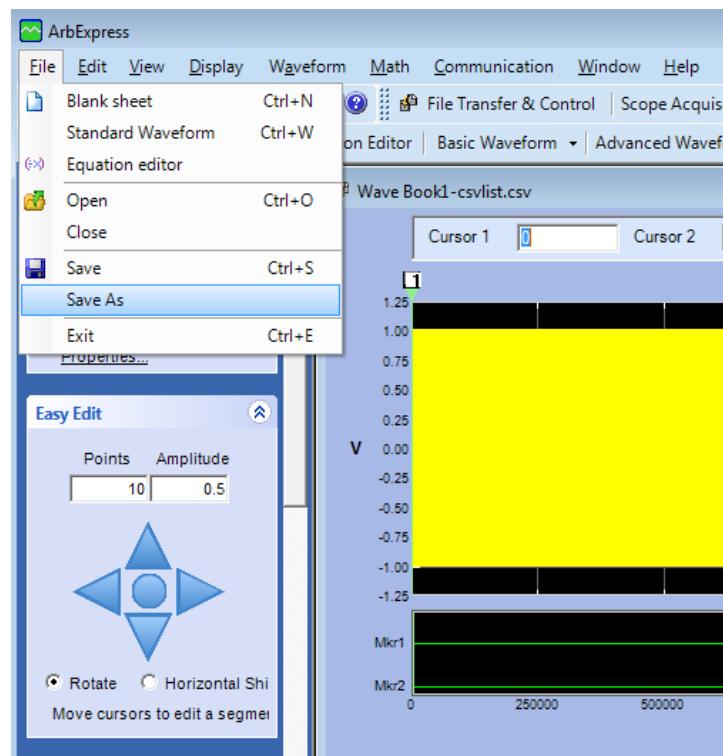
1. One can observe the spectrum of the generated Manchester encoded using the Spectrum Analyzer. In order to generate the Manchester encoded signal, use the saved signal from the Matlab file.

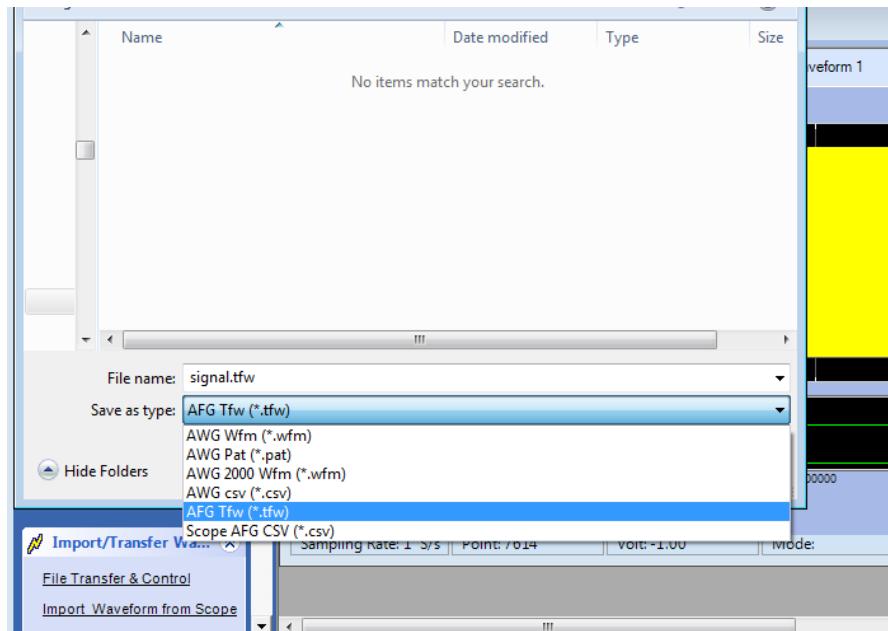


2. Open the ArbExpress software on the PC.
3. Load the saved .csv signal file to the ArbExpress.



4. After observing the Signal on the ArbExpress window, save it as '.tfw' on your USB flash drive.





5. Plug the USB drive in the Tektronix function generator.
6. Select the 'Arbitrary' mode option on the function generator.
7. Load the 'signal.tfw' signal to the function generator.
8. Connect it to the Spectrum Analyzer/Oscilloscope.
9. Save screen shots of the Spectrum and compare it to the Matlab simulation.

## Questions

1. Compare Manchester coding to other line coding schemes studied in the lectures.
2. Compute the power spectral density of the Manchester coded signal and verify it theoretically.
3. What assumption(s) should be satisfied to ideally verify the PSD calculations?

# Lab 5: FREQUENCY SHIFT KEYING

---

## Objective

The objective of this lab is to introduce the concepts of Frequency Shift Keying (FSK) modulation technique through simulation. Students apply the concepts of FSK to design a modem, which uses this modulation scheme.

## Theory

### ➤ FSK: Modulation:

In Frequency Shift Keying (FSK), the instantaneous frequency of the carrier signal is switched between two (or more) values in response to the digital code (e.g. PCM code).

In binary FSK, the binary digital information is modulated to two different frequencies, say  $f_1$  and  $f_2 = f_1 + \Delta f$ .

Thus binary ‘0’ can be expressed by a sinusoidal signal with frequency  $f_1$ , as  $u_1 = \cos 2\pi f_1 t$ . And, binary ‘1’ can be expressed as a sinusoidal signal with frequency  $f_2$ , as  $u_2 = \cos 2\pi f_2 t$ , where  $f_2 = f_1 + \Delta f$ . Figure 5.1 shows a digital signal and the transmitted signal as Binary FSK.

Let us assume that the FSK modulated signal is delayed in the transmission through the channel. The channel also introduces random noise in the received signal. We may assume the random noise to be Additive white Gaussian noise (AWGN).

Consequently, the received signal is,

$$r(t) = \cos(2\pi f_i t + \phi) + AWGN; i = 1, 2$$

,where  $\phi$  = phase delay during transmission.

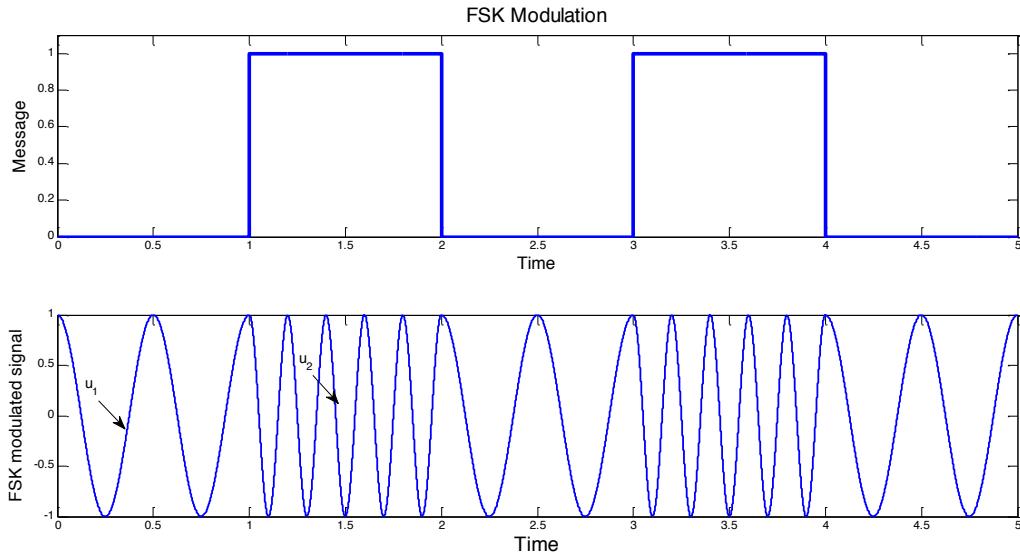


Figure 5.1: Digital signal and its corresponding FSK modulated signal. Notice that binary ‘1’ is represented by a sinusoidal signal with frequency  $f_1$  and binary ‘0’ by signal with frequency  $f_2$

➤ **Detection/Demodulation:**

Demodulation of the binary FSK signal can be done by the non-coherent detection method as shown in the Figure 5.2.

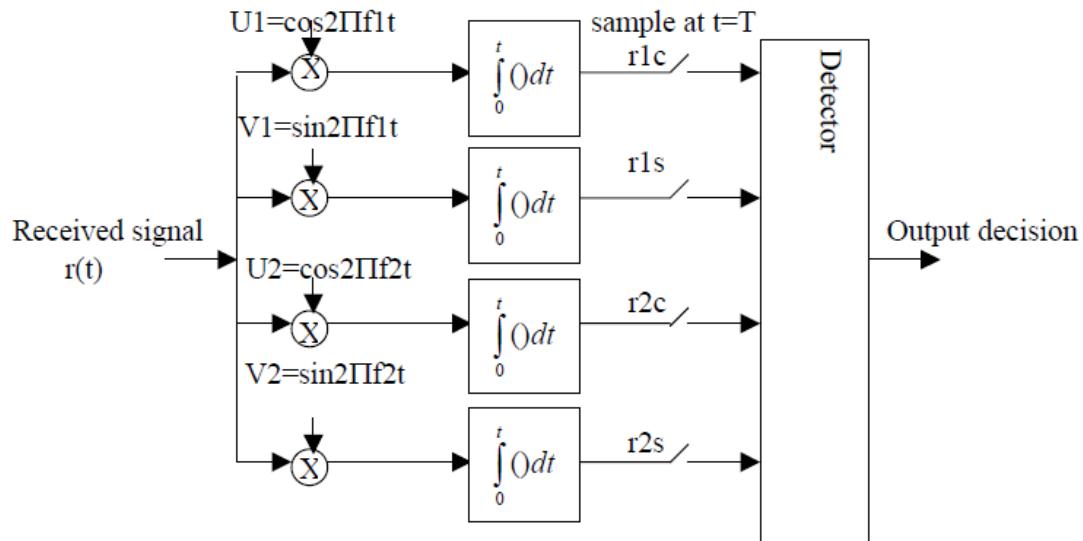


Figure 5.2 FSK Demodulator Block Diagram

The  $r_1$  and  $r_2$  is defined by the following equations,

$$r_1 = \sqrt{r_{1c}^2 + r_{1s}^2}$$

$$r_2 = \sqrt{r_{2c}^2 + r_{2s}^2}$$

The detector decides the received signal by comparing  $r_1$  and  $r_2$ . To be precise if  $r_1 > r_2$  then the detector decides that the received signal is of frequency  $f_1$  (which corresponds to binary ‘0’ at input) and vice versa.

## Simulation

We will use Matlab to do the simulation. We will use the system described in section 5.2 for this simulation. In the simulation, a single bit (either logic ‘0’ or ‘1’) is modulated, channel impairments are added to the modulated signal, and, finally the received signal is non-coherently detected using the system in Figure 5.2 and verified whether detection decision is right or wrong.

### ➤ Simulation Procedure:

1. Define  $Tb=1$ ;  $f1=1000/Tb$ ; and  $f2=f1+1/Tb$ ;
  - a.  $\Phi=\pi/4$ ; %Delay during transmission.
  - b.  $N=5000$ ;
2. Construct time array ‘t’ from zero to  $Tb$  with 5000 points.
3. Define
  - a.  $u_1 = \cos 2\pi f_1 t$
  - b.  $u_2 = \cos 2\pi f_2 t$
  - c.  $v_1 = \sin 2\pi f_1 t$
  - d.  $v_2 = \sin 2\pi f_2 t$
4. Assume  $u_1$  is transmitted. So, received signal is given by  $r(t) = \cos(2\pi f_1 t + \Phi)$
5. Demodulate the signal as
  - a.  $r_{1c} = \sum_{k=1}^N r(k) * u_1(k)$ .
  - b.  $r_{1s} = \sum_{k=1}^N r(k) * v_1(k)$ .
  - c.  $r_{2c} = \sum_{k=1}^N r(k) * u_2(k)$ .
  - d.  $r_{2s} = \sum_{k=1}^N r(k) * v_2(k)$ .
6. Calculate
  - a.  $r_1 = \sqrt{r_{1c}^2 + r_{1s}^2}$
  - b.  $r_2 = \sqrt{r_{2c}^2 + r_{2s}^2}$
7. Compare if  $r_1 > r_2$ , then the output  $y=0$ . Otherwise, the output  $y=1$ .
8. Repeat steps 4 to 7 assuming  $u_2$  was transmitted.
9. Repeat steps 4 to 7 assuming  $u_1$  was transmitted in presence of AWGN such that,

$$r(t) = \cos(2\pi f_1 t + \Phi) + awgn(1,1,1);$$

## FSK modem design

Frequency shift keying (FSK) modulation is introduced in this experiment. It is a commonly used method for transmitting digital data over telephone lines. For FSK, a modulator- demodulator (modem) is needed to translate digital 1's and 0's into their respective frequencies. For modems operating at 1200 bps over commercial telephone channels, the transmit frequencies are 1200 and 2200 Hz.

In this experiment a modem with 1200 bps data rate with mark (correspond to binary '1') and space (correspond to binary '0') frequencies of 1200 Hz and 2200 Hz will be designed.

In this experiment we will use XR-2206 monolithic function generator, and XR-2211 phase-locked loop ICs. General and preliminary descriptions for these ICs are given below. These descriptions are taken from the respective data sheets.

### ➤ **XR-2206**

The XR-2206 is a monolithic function generator integrated circuit capable of producing high quality sine, square, triangle, ramp, and pulse waveforms of high-stability and accuracy. The output waveforms can be both amplitude and frequency modulated by an external voltage. Frequency of operation can be selected externally over a range of 0.01Hz to more than 1MHz.

The circuit is ideally suited for communications, instrumentation, and function generator applications requiring sinusoidal tone, AM, FM, or FSK generation. It has a typical drift specification of 20ppm/ $^{\circ}$ C. The oscillator frequency can be linearly swept over a 2000:1 frequency range with an external control voltage, while maintaining low distortion.

### ➤ **XR-2211**

The XR-2211 is a monolithic phase-locked loop (PLL) system especially designed for data communications applications. It is particularly suited for FSK modem applications. It operates over a wide supply voltage range of 4.5 to 20V and a wide frequency range of 0.01Hz to 300kHz. It can accommodate analog signals between 10mV and 3V, and can interface with conventional DTL, TTL, and ECL logic families.

The circuit consists of a basic PLL for tracking an input signal within the pass band, a quadrature phase detector, which provides carrier detection, and an FSK voltage comparator, which provides FSK demodulation. External components are used to independently set center frequency,

bandwidth, and output delay. An internal voltage reference proportional to the power supply is provided at an output pin

For further detail, students are strongly encouraged to see the data sheets of these ICs in the following web pages before they start designing the modem.

<http://www.exar.com/products/xr2211.pdf>  
<http://www.exar.com/products/xr2206.pdf>

### ➤ Modulator

Frequency of oscillation,  $f_0$  : The XR-2206 can be operated with two separate timing resistors,  $R_1$  and  $R_2$ , connected to pins 7 and 8 respectively. When pin 9 is open circuited or connected to logic '1' only  $R_1$  is active. Similarly, when the voltage level at pin 9 is logic '0' only  $R_2$  is activated.

Therefore, the output frequency can be generated between two frequency  $f_1$  and  $f_2$ , and  $f_1=1/R_1C_0$  and  $f_2=1/R_2C_0$ , let  $C_0=0.022 \mu F$

for a mark frequency,  $f_1$  of 1200 Hz  $R_1=1/(1200*0.022E-6)=37.8$  Select  $R_1=39 K$

For a space frequency,  $f_2$ , of 2200 Hz  $R_2=1/(2200*0.022E-6)=20.7$  Select  $R_2=20 K$

Output amplitude, for a sinewave output, is approximately 60mv per K Ohm of R3. Since the XR-2211 requires an input smaller than 3Vrms a value of 50 K for R3 will be chosen (experimentally, a value of 25 K give the desired results).

Output dc level. The dc levels at pin 2 are approximately the same as the DC bias at pin 3. It is set for  $V_c/2$ .

In applications where minimal distortion is unnecessary, pins 15 and 16 may be left open and a 200 resistor is then connected between pins 13 and 14.

### ➤ Demodulator

The tracking range ( $+- \Delta f$ ) is the range of frequencies over which the phase-locked loop can retain locked with a swept input signal. This range is determined by the formula

$$\Delta f=(R_0/R_4) * f_0$$

$\Delta f$  should be made equal to, or slightly less than, the difference between the mark and space frequencies.

The capture range ( $+/-f_c$ ) is the range of frequencies over which the phase-locked loop can acquire lock. It is always less than the tracking range. The capture range is limited by  $C_2$ , which in conjunction with  $R_4$ , forms the loop filter time constant. In most applications,  $\Delta f_c = (80\%-90\%) \Delta f$ .

The loop damping factor ( $\xi$ ) determines the amount of over shoot, undershoot, or ringing present in the phase locked loop response to a step change in frequency. It is determined with:

$\xi = (1/4) * \sqrt{C_1/C_2}$ . For most modem applications  $\xi = 0.5$ .

#### ➤ Calculations:

The XR-2211 can be used for any FSK decoding application by the choice of five key circuit components  $R_0$ ,  $R_4$ ,  $C_1$ ,  $C_2$ , and  $C_F$ . For a given set of FSK mark and space frequencies,  $f_1$  and  $f_2$ , these key circuit components parameters should be calculated as follows.

Calculate the PLL center frequency,  $f_0$ :  $f_0 = (f_1 + f_2)/2 = (1200 + 2200)/2 = 1700\text{Hz}$

Calculate  $R_0$  from the VCO's center frequency design equation:  $f_0 = 1/R_0 C_1$  let  $C_1 = 0.027 \mu\text{F}$

$R_0 = (1/C_1) * f_0 = 1/(1/0.027\text{E}-6) * 1700 = 21.8$  Select,  $R_0 = 22\text{ K}$

Calculate  $R_4$  to give a  $f$  equal to the mark-space deviation:  $R_4 = R_0 * f_0 / (f_1 - f_2) = 22000 * (1700 / (2200 - 1200)) = 39\text{ K}$

Calculate  $C_2$  to set loop damping. From the design equation and the recommended of  $\xi$ :  $C_2 = C_1/4 = 0.027\text{E}-6/4 = 6.75\text{ nF}$  select  $C_2 = 6.8\text{nF}$

Calculate data filter capacitance,  $C_F$ , from the data filter time constant equation:  $T_F = R_F C_F$  where  $T_F = 0.3/\text{baud rate}$ , and  $R_F = 100\text{K}$

$C_F = 0.3 / (\text{baud rate}) R_F = 0.3 / (1200 * 100\text{E}3) = 2.5\text{nF}$  select  $C_F = 2.4\text{nF}$

## Experimental Procedure

Build the modem according to the Figure 5.3 given. Observe and draw the input, FSK, and output waveforms.

Explain the operation and performance of the modem. Was it necessary to make any adjustments for frequency tuning?

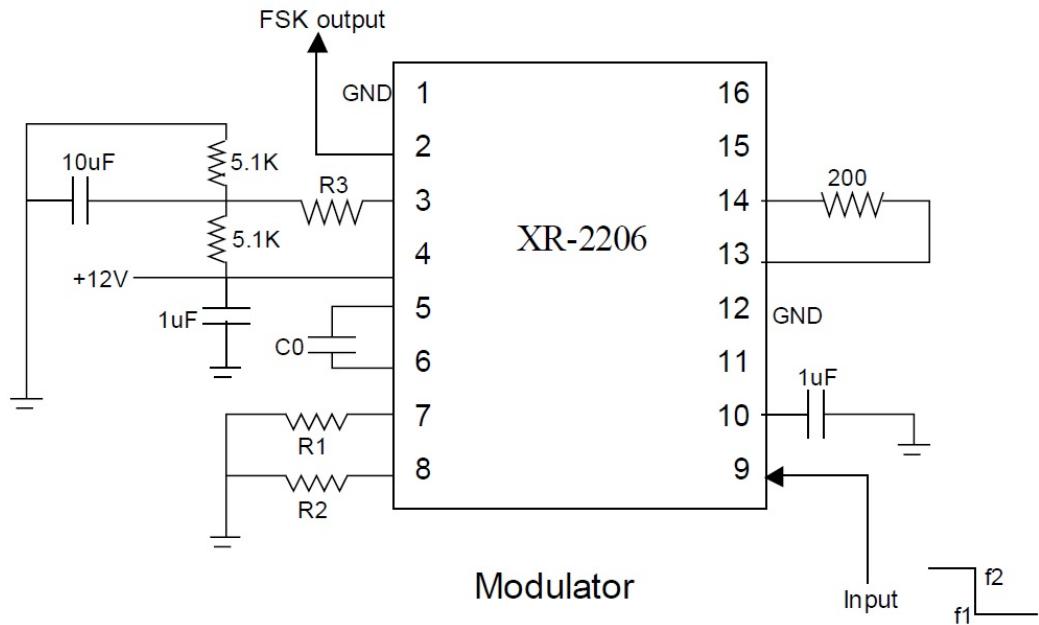


Figure 5.3 (a) FSK modulator

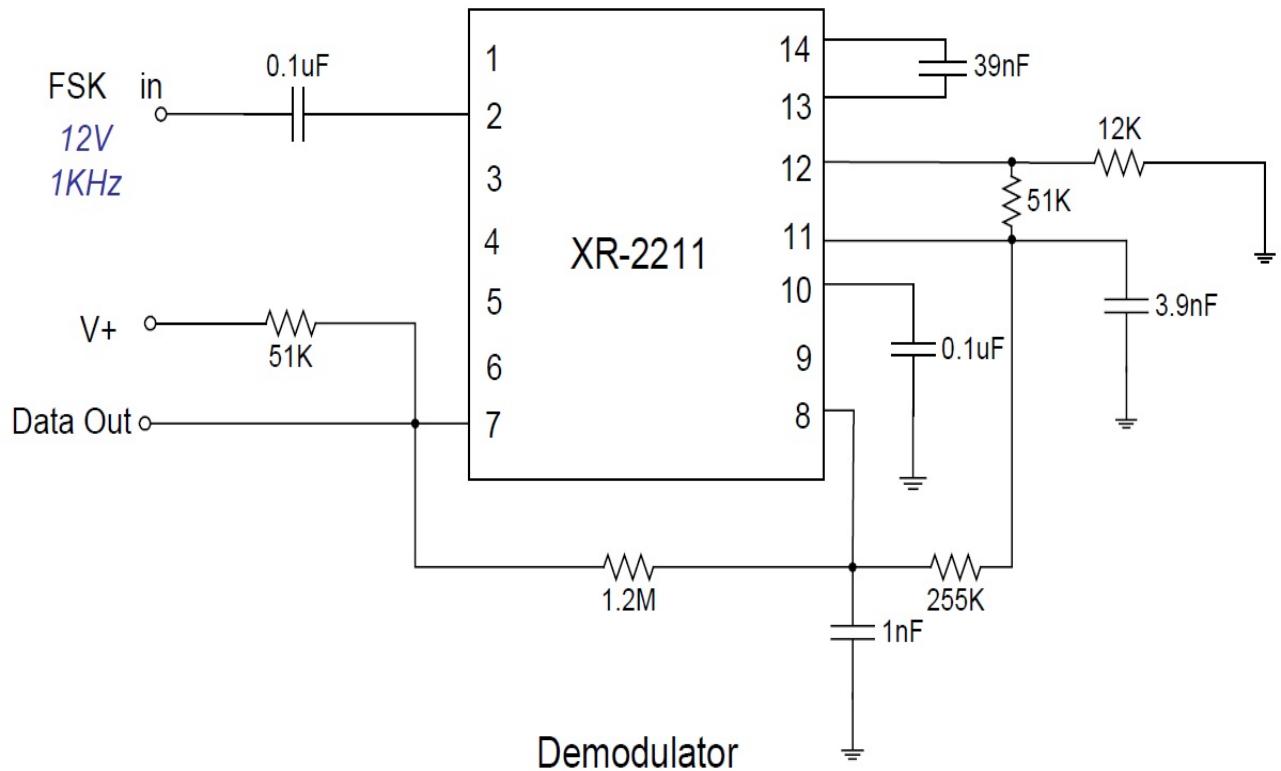


Figure 5.3 (b) FSK modulator

## Questions

Calculate  $R_0$ ,  $R_4$ ,  $C_1$ ,  $C_2$ , and  $C_F$  for a 75 baud FSK demodulator with mark/space frequencies of 1110/1170 Hz.

# Lab 6: Binary Phase Shift Keying

---

## Objective

Understand the principles of Binary Phase Shift Keying (BPSK) digital modulation scheme, its error performance through simulation and hardware implementation of BPSK modulation.

## Theory

PSK was developed during the early days in the deep-space program; PSK is now widely used in both military and commercial communications systems. The general analytic expression for PSK is

$$S_i(t) = \sqrt{\frac{2E}{T_b}} \cos(2\pi f_c t + \phi_i(t))$$

where,  $0 < t < T_b$ ,  $i = 1, \dots, M$ . Note that,  $f_c$  is the carrier frequency,  $T_b$  is the symbol duration,  $E$  is the symbol energy and has  $M$  discrete values. Typically, we choose

$$\phi_i(t) = \frac{2\pi i}{M}$$

For binary PSK (BPSK)  $M$  is 2. In BPSK modulation the data shifts the phase of the waveform  $S_i(t)$  to one of the two states, either zero or  $\pi$ . The signals waveform can be represented in terms of a basis function  $\phi_1(t)$ , as follows

$$S_1(t) = \sqrt{2E} \phi_1(t)$$

$$S_2(t) = -\sqrt{2E} \phi_1(t)$$

$$\text{where, } \phi_1(t) = \sqrt{\frac{2E}{T_b}} \cos(2\pi f_c t)$$

The signal waveform can also be represented as vectors or phasors in a polar plot; the vector length corresponds the signal amplitude, and the vector direction corresponds to the signal direction. In the BPSK case, a binary ‘one’ and ‘zero’ can be represented by the two vectors  $S_1$  and  $S_2$  respectively.  $S_1$  and  $S_2$  are collinear and of opposing directions. Such signal sets are called antipodal signal sets.

There are several ways to demodulate or detect such BPSK symbols. The receiver may perform a differentially coherent detection process, in which the phase of each bit is compared to the phase of the preceding bit. Better performance can be obtained with fully coherent PSK, but that requires an absolute phase reference at each end, and no phase variations in the propagation path.

The received signal from the channel is given by

$$r(t) = S_i(t) + n(t)$$

where,  $S_i$  is either  $S_1$  or  $S_2$ , and  $n(t)$  is the noise signal.

The noise analysis of communication system is customarily based on an idealized form of noise called Additive Gaussian White Noise (AWGN) signal  $n(t)$ . The probability distribution of  $n(t)$  is given by

The noise  $n(t)$  contains all frequency component from  $-\infty$  to  $+\infty$ . The power spectral density of such a noise is shown in Figure 6.1.  $N_0$  is the single-sided noise power spectral density (Watts / Hertz).  $N$  = Noise Power over the bandwidth  $B$ .

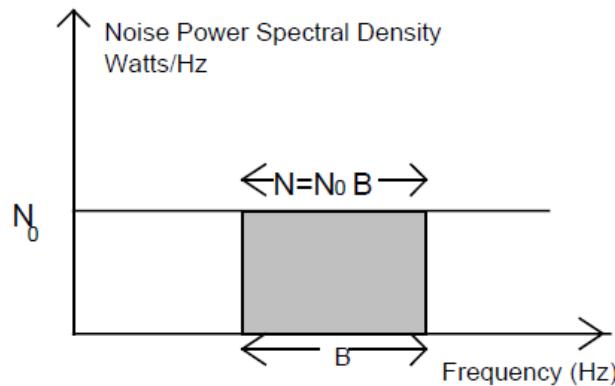


Figure 6.1 Noise Spectral Density

In Figure 6.5, we show the block diagram of a BPSK receiver. If there is no noise in the channel i.e.  $n(t) = 0$ , then for  $r(t) = s_1(t)$  i.e.  $n(t) = 0$ , then for  $r(t) = s_1(t)$ ,  $x_1 = +\sqrt{E_b}$  and  $x_2 = -\sqrt{E_b}$ .

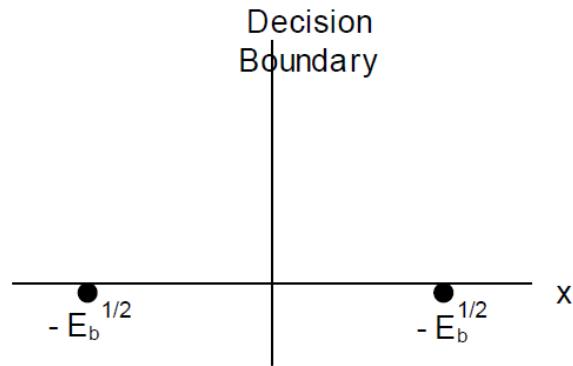


Figure 6.2 Constellation diagram

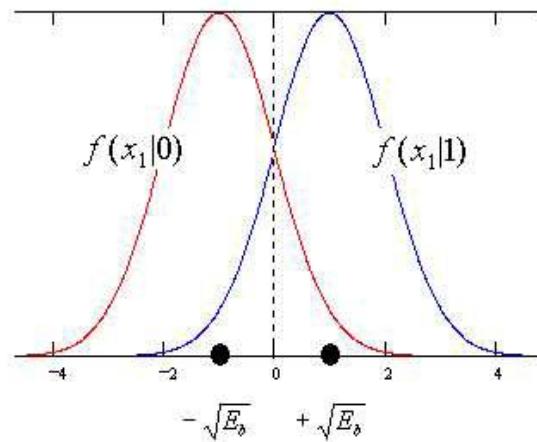


Figure 6.3 Probability distribution of the received BPSK signal

The probability distribution of the received signal is shown Figure 6.3. It has been assumed that the BPSK signals are transmitted through an Additive White Gaussian Noise (AWGN). The  $f(x_1|0)$  curve is the probability distribution of the signal point if only  $S_1$  is transmitted. The  $f(x_1|1)$  curve is the probability distribution of the signal point if only  $S_2$  is transmitted.

The decision boundary of a BPSK receiver affected by AWGN is the point  $X_1=0$  (see Figure 6.3). If the received signal lies on the left of the decision boundary, we will decide that a symbol  $s_1$  was transmitted and vice versa.

For the binary decision making depicted in Figure 6.3, there are two ways an error can occur. An error will occur when  $S_1$  was transmitted, and channel noise results in the received output signal  $r(t)$  being less than the decision boundary. The probability of such occurrence can be found integrating the shaded in the Figure 6.4.

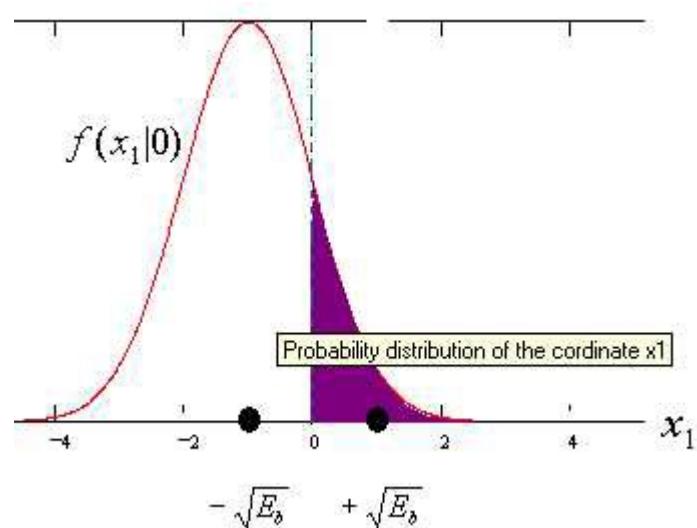


Figure 6.4 Calculating the probability of error

For the receiver shown in Figure 6.5 The probability of error given that a binary ‘zero’ is transmitted and binary ‘one’ is detected can be written by the following mathematical expression

$$P(1 | 0) = \int_0^{\infty} f(x_1 | 0) dx_1 = \int_0^{\infty} \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(x_1 + \sqrt{E_b})^2}{2\sigma^2}\right] dx_1$$

$$\text{Let, } Z = \frac{(x_1 + \sqrt{E_b})}{2\sigma}$$

Then

$$P(1 | 0) = \int_{\sqrt{E_b}/\sigma}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{z^2}{2}\right] dz = \int_{\sqrt{E_b}/\sigma}^{\infty} f(z) dz = Q\left(\frac{\sqrt{E_b}}{\sigma}\right)$$

For  $P(1|0) = P(0|1)$  and  $P(1)=P(0)=0.5$ . The probability of an error in a binary digit is

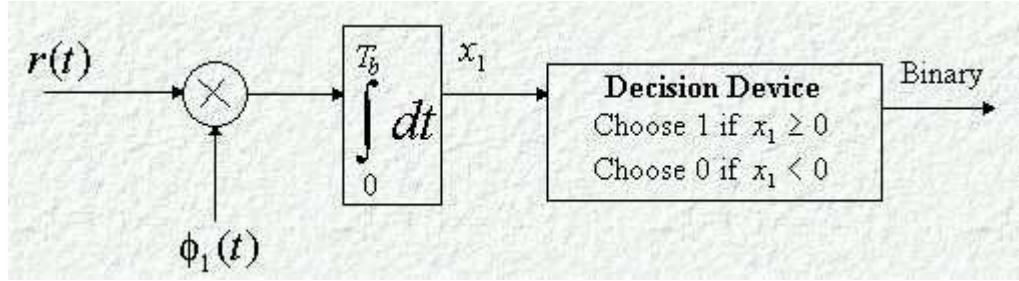


Figure 6.5 BPSK receiver

$$P_e = P(1|0)P(0) + P(0|1)P(1) = P(1|0)[P(0) + P(1)] = P(1|0) = Q\left(\frac{\sqrt{E_b}}{\sigma}\right).$$

The standard deviation of the probability distribution of the coordinate  $x_1$  on the signal space diagram is

$$\sigma = \sqrt{\frac{N_0}{2}}$$

For a given signal-to-noise ratio SNR.

$$\frac{E_b}{N_0} = 10^{\frac{E_b/N_0(dB)}{10}} = 10^{\frac{SNR}{10}}.$$

## Simulation procedure

In the simulation of this lab, we will generate number of BPSK signal and will be added to AWGN and will be detected. The probability of error will be calculated from number of wrong detection. The simulation procedure is given below.

1. Generate a signal to noise ratio (SNR) vector between -15 and 5 (dB). (hint: values are given assuming  $E_b=1$ ).
2. Calculate the SNR ration ' $N_0$ ' as given in the equation above to convert from dB to a ratio.
3. Generate a binary sequence, 'a', of ones and zeros of length  $N > 10000$ .
4. Convert binary sequence 'a' to an antipodal sequence 's' of  $\{-1,1\}$ .
5. Define basis signal  $\phi = \sqrt{2} \cos(2\pi ft)$ .
  - $f=10$  Hz;
  - $t=0:0.01:1$ ;
6. Define  $s1=\phi$  and  $s2=-\phi$
7. Calculate  $\sigma$ .
8. Define a vector of AWGN 'noise\_matrix= randn(length(t),1) \*sigma';
9. For each noise level and every transmitted bit, the received signal is described as

- $R = \text{noise} + s_1$  or  $R = \text{noise} + s_2$ ;
10. Detect the signal using the BPSK receiver given in Fig. 6.5
  11. Compare the number of error at every noise level,
  12. Calculate Error = number of error / N.
  13. Plot Error versus SNR (dB).
  14. Compare Error to the theoretical Probability of error given in the equation using the qfunc(.)

## Hardware Procedure

In the hardware experiment we will build a BPSK modulator. We will generate binary data stream and convert it to a BPSK modulated waveform.

Build the BPSK modulation circuit shown in Figure 6.6.

The data source in Figure 6.6 generates a bit-stream. In this case the bit-stream is a square wave with around 16KHz. Apply a sinewave (5V p-p) as a carrier in your modulator with a frequency of 7-8 times the frequency of the data-stream. A carrier with 120KHz will work in this case.

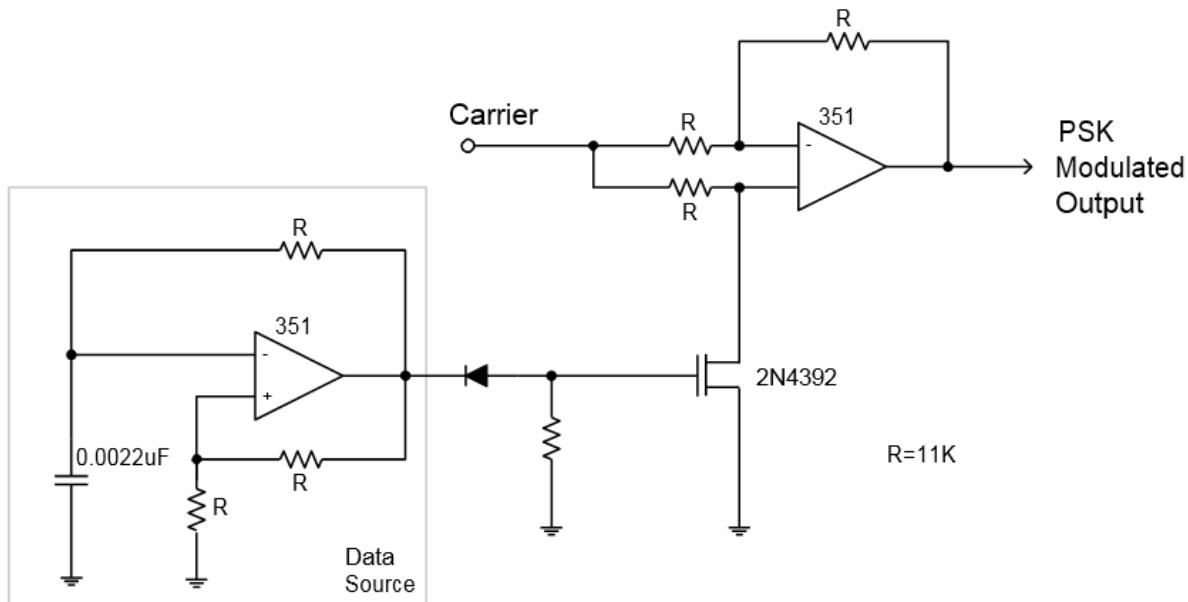
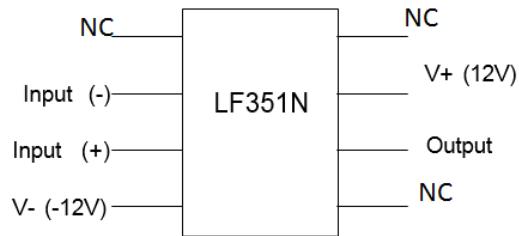
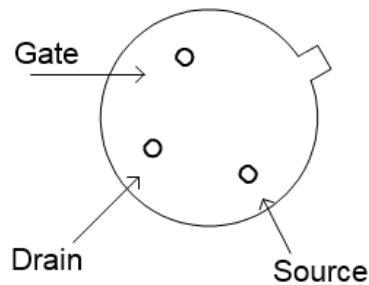


Figure 6.6 BPSK modulator

The 351 in the Figure 6.6 is a wideband operational amplifier. The pin-out for an LF351N is given below.



The 2N4392 in Figure 6.6 is a NMOS transistor. The pin-out is given below. Note that this is the bottom view of the transistor



Bottom View

Plot the carrier, data-stream and BPSK modulated signal

## Questions

How BPSK modulated signal can be detected? Show the demodulation process for BPSK symbols through block diagrams for the case of

- Coherent detection, and
- Non-coherent detection.

# Lab 7: Project

---

## Objective

The aim of this Matlab project is to enhance the student's understanding of, and ability to simulate, digital communication systems. It should also help the student better understand various design aspects of communication systems.

## Disclaimer

The project can be changed or modified at the instructor's discretion.

## Grading Policy

To be announced by the lab instructor.

## Project

The aim of this project is to illustrate the effect of dispersive channel.

- 1- Generate a random sequence of QPSK of 1 million symbols.
- 2- Let the amplitude of QPSK symbols be of the form
  - a.  $A(1+j)$
  - b.  $A(1-j)$
  - c.  $-A(1-j)$
  - d.  $-A(1+j)$
- 3- Let  $A=\sqrt{2}$
- 4- Plot the constellation points of the generated sequence.
- 5- Let the received signal  $y(t)$  be of the form

$$y(t) = x(t) * g(t) + n(t)$$

Where  $g(t)$  represents a 2-ray multipath channel that can be expressed as the following filter

- a.  $Channel = [1 \ 0 \ 0 \ 0.65]$
- 6- Generate AWGN noise of 12 dBm to be added to the transmitted signal.
- 7- Again, plot the constellation points of the following:
  - a. Transmitted signal plus noise.

- b. Transmitted signal after being filtered by channel.
- c. Transmitted signal after channel plus noise.

Bonus –,

- A. Design a channel equalizer and plot the constellation points before and after equalization.
- B. Design a Matched Filter to recover the transmitted symbols and calculate the Symbol Error Rate (SER) with and without equalization.

Hints:

- 1- To filter the signal with the channel use ‘conv’.
- 2- To generate a random sequence of integers use ‘randi’
- 3- To plot constellation points use ‘plot’ and use a dot marker ‘.’
- 4- To get the real part use ‘real’
- 5- To get the imaginary part use ‘imag’.