ECE 405 - SOURCE CODING - INVESTIGATION 16
INTRODUCTION TO PULSE CODE MODULATION

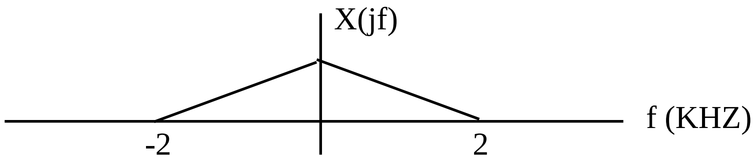
FALL 2005 **A.P. FELZER**

To do "well" on this investigation you must not only get the right answers but must also do neat, complete and concise writeups that make obvious what each problem is, how you're solving the problem and what your answer is. You also need to include drawings of all circuits as well as appropriate graphs and tables.

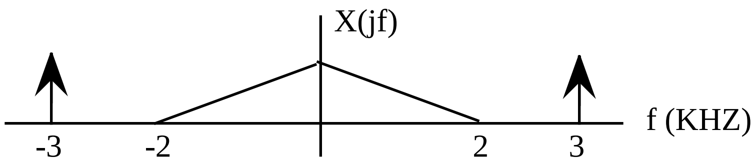
Pulse mode techniques as discussed in the last Investigation have their applications but they don't give us any real advantage over analog communications. The real advantage of digital communications comes when we digitize the samples - convert the sample values to binary numbers - and then transmit the 1's and 0's. This often requires a larger bandwidth but has the big advantage that we don't have to worry about the exact values of the received pulses - only if they are those for 1's or those for 0's. We call this pulse code modulation or PCM.

The objective of this Investigation is to show how sample values m_s can be converted to binary numbers with a finite number of bits and then take a look at the noise generated from the inevitable rounding. In the next Investigation we'll look at circuits that implement analog-to-digital and digital-to-analog conversions.

1. Let us begin with a review. How fast do we need to sample each of the following signals to be able to recreate it from its samples
   a. $x(t) = 2\cos(2\pi 1000t)$
   b. $x(t) = 2\cos(2\pi 1000t) + 3\cos(2\pi 2000t)$
   c. $x(t) = 2\cos(2\pi 1000t)\cos(2\pi 2000t)$
   d.



   e.



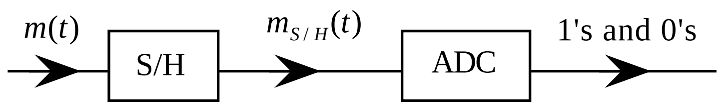
2. Suppose we sample the following signal $x(t)$ at $f_s = 3500$ samples/sec

$$x(t) = 2\cos(2\pi 1000t)$$

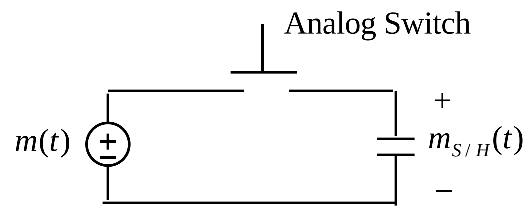
   a. What is the sample time T_s - the time between samples
   b. Find the first 5 samples starting at $t = 0$

3. What happens when we don't sample fast enough

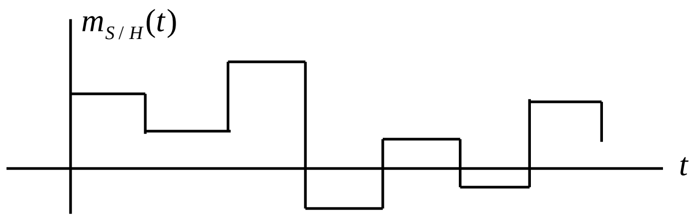
4. As we said above **Pulse Code Modulation (PCM)** is a digital communication system where we sample a signal every T_s seconds and then convert the samples to 1's and 0's for transmission. The sampling is done with a circuit called a **Sample and Hold (S/H)** and the conversion to binary with an **Analog-to-Digital Converter (ADC)** as follows



The S/H circuit, in particular, is a circuit like the following



with outputs $x_{S/H}(t)$ as follows



that hold the value of the sample for the ADC to convert it to binary.

a. Explain how our simple sample-and-hold circuit works - how it's able to sample and then hold $x(t)$
b. Sketch a timing diagram for $0 \le t \le 2$ msec that includes the clock controlling the analog switch and $x_{S/H}(t)$ when $x(t) = 2\cos(2\pi 1000t)$ sampled at $f_s = 5000$ samples/sec

5. Now once $m(t)$ is sampled the ADC converts it to binary. The objective of this and the next several problems is to review the math of this conversion. We begin with a review of converting positive binary integers to decimal numbers as illustrated in the following example

$$1101_2 = 1(2^3) + 1(2^2) + 0(2^1) + 1(2^0) = 13_{10}$$

in which we simply multiply each binary digit by its place value or weight. Use this algorithm to convert the following binary numbers to decimal

a. 10111_2
b. 11010_2

6. This problem reviews how to convert positive decimal numbers to binary by repeatedly dividing by 2 as indicated in the following example that converts 23_{10} to binary as follows

$$23_{10} = 2\;\; 11 + 1 = 2^1\;\; 11 + 1\;\; 2^0$$
$$2^1\;\; 11 = 2^1(2\;\; 5 + 1) = 2^2\;\; 5 + 1\;\; 2^1$$
$$2^2\;\; 5 = 2^2(2\;\; 2 + 1) = 2^3\;\; 2 + 1\;\; 2^2$$
$$2^3\;\; 2 = 2^3(2\;\; 1 + 0) = 2^4\;\; 1 + 0\;\; 2^3$$
$$2^4\;\; 1 = 2^4(2\;\; 0 + 1) = 2^5\;\; 0 + 1\;\; 2^4$$

And so we have

$$23_{10} = 1\;\; 2^4 + 0\;\; 2^3 + 1\;\; 2^2 + 1\;\; 2^1 + 1\;\; 2^0 = 10111_2$$

More compactly we have

Divide By Two	**Dividend**	**Remainder**
23/2	11	1
11/2	5	1
5/2	2	1
2/2	1	0
1/2	0	1

Memorize this algorithm. Then use it to

a. Convert 95_{10} to binary
b. Convert 43_{10} to binary
c. Check your results in parts (a) and (b) with a calculator

7. The objective of this problem is to show how we can deal with the fact that a real system has only a finite number of bits for representing samples. Suppose in particular that we want to convert the sample m_s in the range

$$0 \le m_s \le m_{\max}$$

to n-bit binary. Then we proceed as follows

(1) Calculate the resolution $\Delta = \frac{m_{\max}}{2^n}$

(2) Calculate $m_\Delta = \frac{m_s}{\Delta}$ rounded off to the nearest integer

(3) Convert m_Δ to binary

Assuming $m_{\max} = 10$

a. Convert $m_s = 7.34$ to 8-bit binary
b. Convert $m_s = 7.34$ to 10-bit binary

8. So far so good but we need a way to deal with negative samples. The simplest way to do this is with signed binary with the MSB representing the sign of the sample as follows

MSB	*Sign*
0	+
1	–

Express the following signed binary numbers in decimal

a. $A = 01101101_{SB}$
b. $B = 11101101_{SB}$

9. Convert each of the following decimal numbers to 12-bit signed binary

a. $m_\Delta = 734$
b. $m_\Delta = -734$

10. Generalizing on our algorithm in Problem (7) we have the following general method for converting samples m_s in the range

$$-m_{\max} \le m_s \le m_{\max}$$

to n-bit signed binary is as follows

(1) Calculate the resolution $\Delta = \frac{2m_{\max}}{2^n} = \frac{m_{\max}}{2^{n-1}}$

(2) Calculate $m_\Delta = \frac{m_s}{\Delta}$ rounded off to the nearest integer

(3) Convert m_Δ to signed binary

Assuming $m_{\max} = 10$

a. Convert $m_s = 7.34$ to 8-bit signed binary
b. Convert $-m_s = 7.34$ to 8-bit signed binary
c. Find m_s in decimal if in 8-bit signed binary $m_s = 01011101_{SB}$

11. What's the value of the transmitted signal if $m_\Delta = 0011$ and $\Delta = 0.01$

12. Whenever we convert a sample value to binary there will usually be roundoff or truncation error as given by

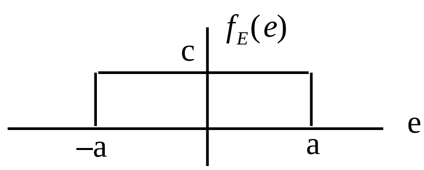
Conversion error = Sample value – (Binary value) Δ

a. What's the roundoff error if $m_s = 223.3$ and $\Delta = 0.02$
b. What's the roundoff error if $m_s = 223.8$ and again $\Delta = 0.02$
c. Suppose we convert samples to n-bit signed binary with a resolution of Δ. Explain why the roundoff error e is in the range

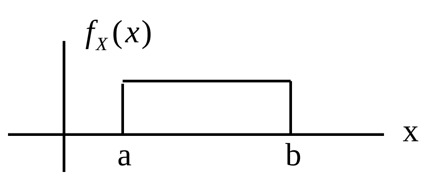
$$-\frac{\Delta}{2} \le e \le \frac{\Delta}{2}$$

d. What happens to the roundoff error when we add an extra bit

13. Assuming that roundoff error as discussed in the last problem has a uniform probability density as follows



a. Find a and c if samples $-5 \le m_s \le 5$ are converted to 8-bit signed binary numbers
b. Find the average of the square of the roundoff in part (a). Hint - make use of the result from ECE 315 that if a random variable X has a uniform probability density as follows



then its variance is given by $Var[x] = \frac{(a-b)^2}{12}$

c. Make use of your result in part (b) to determine what happens to the noise power as the number of bits is increased

14. Another common way to include negative numbers in PCM codes is with **offset binary**. Suppose in particular that as in Problem (6)

$$-m_{\max} \le m_s \le m_{\max}$$

Then m_s is converted to offset binary as follows

(1) First add $m_{\max}$ to m_s to obtain $m_s + m_{\max}$

(2) Then convert $m_s + m_{\max}$ to binary with $\Delta = \frac{2m_{\max}}{2^n}$

a. Convert $m_s = 6.3$ to 8-bit offset binary when $m_{\max} = 10$
b. Convert $m_s = -6.3$ to 8-bit offset binary when $m_{\max} = 10$

15. From ECE 204 we know that 2's complement is another way to represent binary numbers. What's the advantage of 2's complement numbers.