ECE 405 - SOURCE CODING - INVESTIGATION 18
INTRODUCTION TO COMPANDING

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.

From the last two Investigations we know how to calculate quantized PCM values and we've seen how PCM codes can be obtained with analog-to-digital converters. The main objective of this Investigation is to calculate signal-to-noise ratios (SNRs) for uniform PCM codes like the ones we've been working with and for compressed PCM codes we'll be introducing. Our big result will be that we can obtain good SNRs with fewer bits when we use compression.

1. We begin with a review problem. From Investigation 12 we know that when the roundoff (quantization) error q of an ADC with resolution Δ is uniformly distributed then

$$\text{Average Noise Power } = E\left[Q^2\right] = \frac{\Delta^2}{12}$$

Make use of this result to find the average power of the roundoff error for a bipolar ADC with

$$\Delta = \frac{2m_{max}}{2^n} = \frac{2m_{max}}{\text{Number of Quantization Levels}}$$

when

a. $m_{max} = 5$ and n=8
b. $m_{max} = 10$ and n=12
c. $m_{max} = 10$ and 0.8 of the samples are 12 bits and 0.2 of them are 10 bits

2. Clearly we want the noise created by the roundoff errors of our ADCs to be as small as possible. But that by itself is not enough. What's really important - as in the case of analog signals - is the signal-to-noise ratio $(SNR)_o$ at the outputs of our ADCs as follows

$$(SNR)_o = \frac{\text{Average Power of the Signal m(t)}}{\text{Average Power of the Noise}}$$

a. Why is knowing the signal-to-noise ratio more important than just knowing the average power of the noise
b. Find the $(SNR)_o$ in dB - find $10\log_{10}(SNR)_o$ - for an ADC with $(SNR)_o$=1000
c. How does increasing the number of bits n of an ADC by one affect the $(SNR)_o$

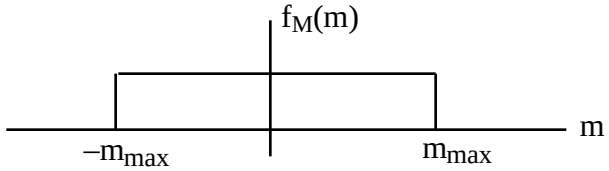
3. The objective of this problem is to find the $(SNR)_o$ at the output of an n bit ADC when $m(t)$ is equal to the following sinusoid

$$m(t) = m_p \cos(2\pi f t)$$

a. Find the average power of $m(t)$
b. Make use of your result in part (a) to show that the $(SNR)_o$ in dB is given by

$$10\log_{10}(SNR)_o = 6n + 1.76 \text{ dB}$$

4. The objective of this problem is to find the $(SNR)_o$ at the output of an ADC when the input m(t) is uniformly distributed over the range $(-m_{\max}, m_{\max})$ with probability density function as follows



   a. Show that $E\left[m^2(t)\right] = \frac{m_{\max}^2}{3}$. Hint - $E[g(x)] = \int g(x) f_X(x) dx$
   b. Make use of your result in part (a) to show that $(SNR)_o = 2^{2n}$
   c. Make use of your result in part (b) to show that $(SNR)_o$ dB $= 6n$ dB

5. How does the $(SNR)_o$ of $m_1(t)$ compare with that of $m_2(t)$ if $m_1(t)$ is always smaller than $m_2(t)$ assuming both signals have the same $m_{\max}$ and the same n

6. From Problem (5) we see that the smaller the average power of $m(t)$ the smaller the $(SNR)_o$ for a given number of bits n. So this presents a dilemma - at least for voice transmission over the telephone. Since most speech is relatively small in amplitude we need at least 30dB of dynamic range for the lowest voice signals in addition to an additional 40dB of dynamic range to accommodate louder speakers. Which gives us a total $(SNR)_o$ of 70dB or 12 bits - which is 4 more bits than we would like to use.

   The way we're able to avoid having to transmit 12 bits for voice communication is by "rounding off" the larger amplitude signals. We're able to do this without adversely affecting the overall signal-to-noise ratio $(SNR)_o$ because the amount of noise contributed by rounding off the larger signals is relatively small because there are relatively few large amplitude voice signals.

   This is all accomplished by **compressing** the PCM code at the transmitter to 8 bits and then **expanding** it back to 12 bits at the receiver - but with lower resolution. This whole process is referred to as **companding**. Chips that do this as well as other "chores" are referred to as **codecs** (coders/decoders). The compression standard used in Europe is the A-law characteristic and that in the United States, Canada and Japan the very similar μ-law as follows

$$m_c(m_s) = \text{sgn}(m_s) m_{\max} \frac{\ln\left(1 + \mu \left|\frac{m_s}{m_{\max}}\right|\right)}{\ln(1+\mu)}$$

   where $\text{sgn}(m_s) =$ sign of m_s. To implement this compression

   (1) First calculate $m_c(m_s)$
   (2) And then convert $m_c(m_s)$ to 8 bit signed binary

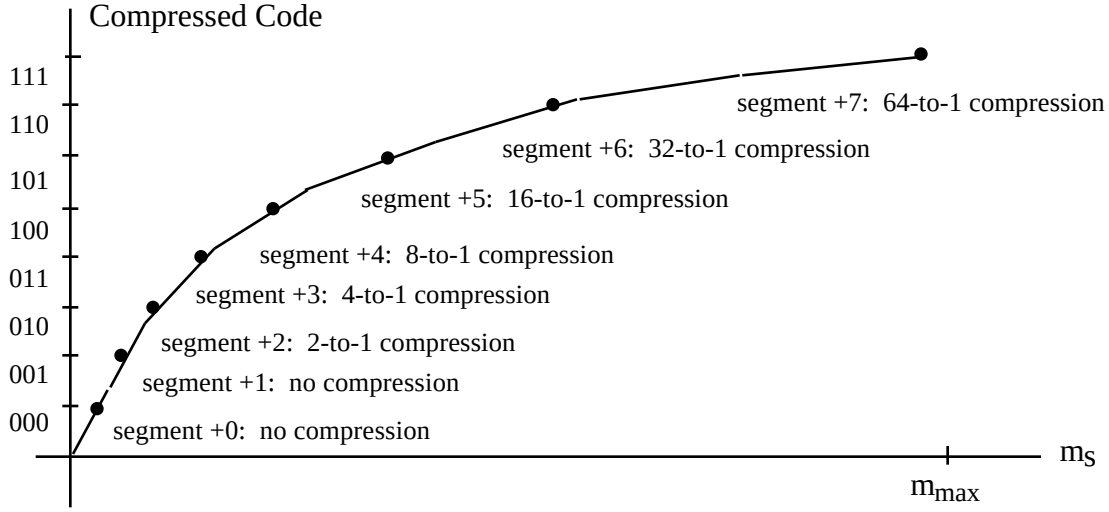
   a. Calculate $m_c(0.65)$ for $\mu = 255$ with $m_{\max} = 10$. And then convert $m_c(0.65)$ to 8-bit signed binary
   b. Convert your 8-bit signed binary for $m_s = 0.65$ back to decimal

c. Find the difference between $m_s = 0.65$ and the 8-bit μ-law value

d. Repeat parts (a)-(c) for $m_s = 8.35$

e. Verify that there's a bigger difference between $m_s = 8.35$ and the 8-bit representation of $m_c(8.35)$ than between the smaller sample $m_s = 0.65$ and the 8-bit representation of $m_c(0.65)$

7. A common way to implement μ-law compression with $\mu = 255$ is to first *approximate* the μ-law curve with a series of eight straight line segments as shown in the following graph for positive sample values m_s. Note in particular that each segment is uniformly divided into $2^4 = 16$ codes



And then implement the 12-bit to 8-bit compression with a ROM lookup table as follows

a. Complete the following table

Segment	12 bits From ADC	Compressed PCM
+0	s0000000ABCD	s000ABCD
+1	s0000001ABCD	s001ABCD
+2	s000001ABCDx	s010ABCD
+3	s00001ABCDxx	s011ABCD
.	.	.
.	.	.

with a row for each segment as shown on the graph specifying the relationship between the 12-bit signed binary output of the ADC and the 8-bit compressed code at the output of the ROM with

(1) MSB = s = sign of the sample (0 for + and 1 for -)

(2) Next 3 bits = segment number = location of the first 1 after the sign bit as shown in the Table. Note in particular that the counting is done from the right

(3) LSB 4 bits = ABCD = the 4 bits after the first 1

b. Make use of your Table in part (a) to find the $\mu = 255$ compressed PCM code for 0011 0111 0101

c. Find the $\mu = 255$ compressed PCM code for $m_s = 0.65$ with $m_{\max} = 10$ by first converting m_s to 12-bit signed binary and then making use of your Table in part (a) to find the compressed PCM signal

d. Compare your compressed signals in part (c) and Problem (6b)

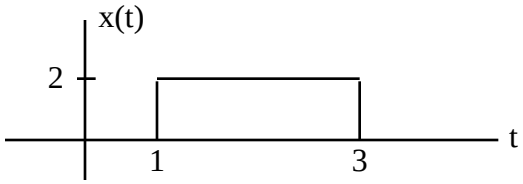
e. Repeat for parts (b)-(d) for $m_s = 8.35$

8. Fourier Review - Find and sketch the Fourier Transforms of

a. $x(t) = 10\cos(2\pi 1000t) + 5\cos(2\pi 2000t)$

b. $x(t) = 5\delta(t)$

9. Math Review - Given the following signal



a. Sketch $x(t+1)$

b. Sketch $x(t)x(t+1)$

10. Make use of Mathcad or Matlab to obtain a graph of three periods of $x(t) = 5\cos(2\pi 1000t)$