ECE300 – Pset 5

Jonathan Lam

November 30, 2020

1. *Describe when and why we use raised cosine pulses, citing the necessary conditions, at least one theorem and drawing at least one figure.*

   Raised cosine pulses are used to achieve the Nyquist zero-ISI criterion, which states that zero ISI is possible iff the sum of samples of the Fourier transform (with period $2\pi/T_s$, where T_s is the symbol period) is a constant, regardless of the starting point of the sampling. The assumption made is that the channel is a perfect low-pass filter (i.e., constant in the passband), which is a good-enough approximation for most cases. They are used because they are more closely realizable than sinc pulses (the tails in the time-domain representation dies off quicker).

   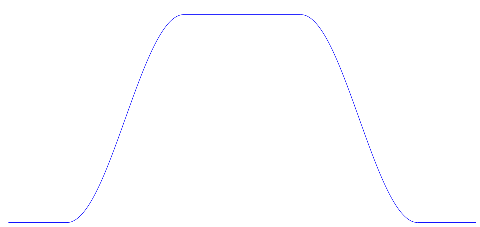

   Figure 1: Sample raised cosine with $T_s = 1$ and $\alpha = 0.5$

   Raised cosine:

$$X_{rc}(\omega) = \begin{cases} T, & 0 \le |\omega| \le \pi\frac{1-\alpha}{T} \\ \frac{T}{2}\left(1 + \cos\left(\frac{T}{2\alpha}\left(|\omega| - \pi\frac{1-\alpha}{T}\right)\right)\right), & \pi\frac{1-\alpha}{T} \le |\omega| \le \pi\frac{1+\alpha}{T} \\ 0, & \pi\frac{1+\alpha}{T} \le |\omega| \end{cases}$$

2. *True raised cosine pulses are not realizable – why not? What is done in practice to approximate the raised cosine?*

   They are not realizable because they are not compact in frequency, and are thus infinite in time, and thus also noncausal. (The time-domain representation involves the sinc, which is infinite.) This can be estimated by truncating the signal (to make it compact in time) and delaying it (to make it causal).

3. *Find the entropy of a geometrically distributed random variable (that is, a variable with probability mass function $f(m) = p(1-p)^{m-1}$ for positive integer m and some fixed probability p) as a function of p.*

$$
\begin{aligned}
H(X) &= \mathbb{E}[I(X)] = \mathbb{E}[-\log f(x)] \\
&= \sum_{x=1}^{\infty} -\log\left(p(1-p)^{x-1}\right) p(1-p)^{x-1} \\
&= \sum_{x=1}^{\infty} -\left[\log p + (x-1)\log(1-p)\right] p(1-p)^{x-1} \\
&= \sum_{x=1}^{\infty} \left[-p(\log p - \log(1-p))\right](1-p)^{x-1} + \sum_{x=1}^{\infty} \left[-p\log(1-p)\right] x(1-p)^{x-1} \\
&= p\log\left(\frac{1-p}{p}\right) \sum_{x'=0}^{\infty} (1-p)^{x'} + p\log\left(\frac{1}{1-p}\right) \sum_{x=1}^{\infty} x(1-p)^{x-1}
\end{aligned}
$$

Infinite sum formulas:

$$
\sum_{x=0}^{\infty} r^x = \frac{1}{1-r} \quad (\text{for } |r| < 1)
$$

$$
\frac{d}{dr}\left[\sum_{x=0}^{\infty} r^x\right] = \sum_{x=1}^{\infty} x r^{x-1} = \frac{1}{(1-r)^2} = \frac{d}{dr}\left[\frac{1}{1-r}\right] \quad (\text{for } |r| < 1)
$$

Substituting:

$$
\begin{aligned}
H(X) &= p\log\left(\frac{1-p}{p}\right)\frac{1}{1-(1-p)} + p\log\left(\frac{1}{1-p}\right)\frac{1}{(1-(1-p))^2} \\
&= \log\left(\frac{1-p}{p}\right) + \frac{1}{p}\log\left(\frac{1}{1-p}\right) \\
&= \log(1-p) - \log(p) - \frac{1}{p}\log(1-p) \\
&= \frac{p-1}{p}\log(1-p) - \log p \\
&= \log\left(\frac{(1-p)^{1-\frac{1}{p}}}{p}\right)
\end{aligned}
$$

4. *Suppose a source has alphabet $\{a_1, a_2, \ldots, a_8\}$ with corresponding output probabilities $\left\{\frac{1}{16}, \frac{1}{16}, \frac{1}{4}, \frac{1}{32}, \frac{3}{32}, \frac{1}{8}, \frac{1}{16}, \frac{5}{16}\right\}$. Determine the entropy of the*

source.

$$
\begin{aligned}
H(X) &= \sum_{i=1}^{8} -p_i \log p_i \\
&= \frac{4}{16} + \frac{4}{16} + \frac{2}{4} + \frac{5}{32} + \left(\frac{-3\log 3}{32} + \frac{15}{32} \right) + \frac{3}{8} + \frac{4}{16} + \left(\frac{-5\log 5}{16} + \frac{20}{16} \right) \\
&= \frac{7}{2} - \frac{3\log 3 + 10\log 5}{32} \\
&= \frac{7}{2} - \frac{\log 263671875}{32} \approx 2.626
\end{aligned}
$$

5. *Design a Huffman code for the above source. Validate that it satisfies the Kraft inequality, and that the average length satisfies the Huffman code entropy inequalities.*

   See Figure 2 for a visualization of the designed Huffman coding tree. Table 1 shows the equivalent mappings as a table.

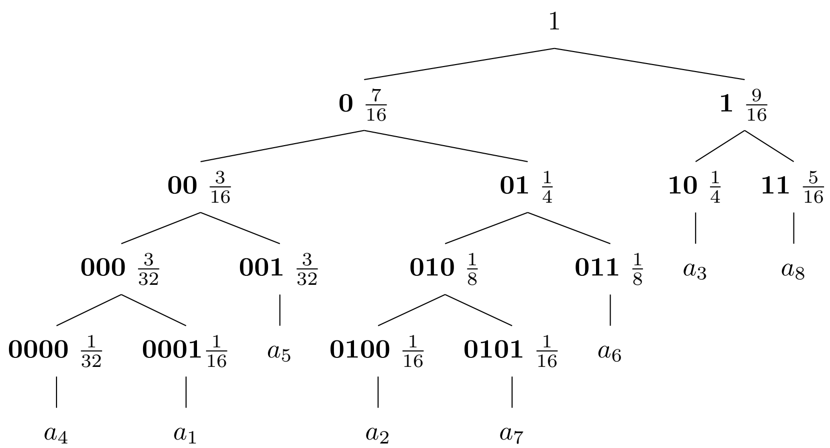

Figure 2: Huffman code tree. Bolded numbers are the codewords (codeword prefixes), and the numbers alongside them are their probabilities. Corresponding symbols are indicated at the leaves.

Kraft inequality (where l_i is the length of each word):

$$\sum_{i=1}^{N} 2^{-l_i} \leq 1$$

In this example:

$$\sum_{i=1}^{8} 2^{-l_i} = (4)2^{-4} + (2)2^{-3} + (2)2^{-2} = \frac{4}{16} + \frac{2}{8} + \frac{2}{4} = 1$$

Symbol	Code	Probability
a_1	0001	1/16
a_2	0100	1/16
a_3	10	1/4
a_4	0000	1/32
a_5	001	3/32
a_6	011	1/8
a_7	0101	1/16
a_8	11	5/16

Table 1: Huffman code symbol-to-code mapping

thus the Kraft inequality is satisfied. For a Huffman code, the average length satisfies:

$$H(X) \leq \bar{l} \leq H(X) + 1$$

In this example:

$$\begin{aligned} \bar{l} &= \sum_{i=1}^{8} l_{a_i} p_i \\ &= \frac{4}{16} + \frac{4}{16} + \frac{2}{4} + \frac{4}{32} + \frac{9}{32} + \frac{3}{8} + \frac{4}{16} + \frac{10}{16} = \frac{85}{32} \approx 2.656 \end{aligned}$$

This satisfies the above inequality, i.e., $2.626 \leq 2.656 \leq 3.626$.

6. *Find the transmission power (in watts) necessary for an AWGN channel with* $N_0 = 108\text{J}$ *and transmission bandwidth* $B = 1\text{MHz}$ *to achieve a channel capacity of* 1Mbps*.*

   The channel capacity assuming AWGN (in bits/sec) is

$$C = B \log\left(1 + \frac{P}{P_N}\right)$$

   Solving for P:

$$\begin{aligned} P_N &= N_0 B \\ P &= N_0 B \left(2^{C/W} - 1\right) \\ &= (108\text{J})(10^6\text{Hz}) \left(2^{(10^6\text{Hz})/(10^6\text{Hz})} - 1\right) \\ &= 1.08 \times 10^8\text{W} \end{aligned}$$

7. *The next several questions will refer to the* (n, k) *linear block code with generator matrix*

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix}$$

What are n and k?

$n = 7$, $k = 3$

8. *Make a list of every data vector and associated codeword.*

$$XG = C$$

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

There are 8 possible datawords and codewords. Each row of X and the corresponding row in C represent a dataword and codeword pair, respectively.

9. *What is the minimum Hamming distance of the code, d_{min}? How many errors can this code correct?*

In a linear code, $d_{min} = w_{min}$ (minimum distance of a code is equal to the minimum weight of that code). It is not hard to see that the minimum weight of this code is 3. Thus this code can detect any error with no more than 2 bit errors, and correct any case with a single bit error.

10. *There exists a systematic code with the same codewords as this code – how can you know that from looking at the list? It should be straightforward to create the generator matrix, G_S.*

If we reduce G to reduced row-echelon form using elementary row operations, we preserve the row space (and thus the same outputted codewords). Since the RREF of G has an identity matrix for the first three columns (i.e., the first three columns are linearly independent), this becomes a systematic code with the same row space (codewords). (Visually, looking at the list of codewords, we see that we have a linear code where each of the datawords forms the first three bits of some codeword, so it should be possible to construct a systematic code.)

To find G_S, swap the first two rows, add (XOR) the third row to (with) the second row.

$$G_S = \left(\begin{array}{ccc|cccc} 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{array}\right) = \left(I_3 \,|\, P\right)$$

(Alternatively, the three row in G_S are the three codewords that begin with 001, 010, and 001, which can be thought of as a basis for the codewords/row space.)

11. *Find the parity check matrix of G_S, H.*

$$H = \left(P^T \mid I_{n-k}\right) = \left(\begin{array}{ccc|cccc} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right)$$

12. *Does H work as a parity check matrix for G?*

    (The following matrix multiplications were performed in MATLAB.)

$$GH^T = G_S H^T = \mathbf{0}_{3\times 4}$$

Thus H works as a parity check for both G and G_S. This is a nice property given that they have the same codewords.