ELEX 3525 : Data Communications
2018 Fall Term

Solutions to Assignment 3

Corrected answer to Question 1.2 to be the value of N required to detect (rather than correct) 1 error.

Question 1

1. For $k = 1$ the codeword transmitted is N 1's or N 0's. So the minimum distance is N. For $k = 2$ there would be N copies of the first bit and N copies of the second bit. The distance could be N or $2N$ so the minimum distance is N. By extension, we see that the minimum distance happens when two codewords differ by only one data bit and in this case the minimum distance will be N. So, in general, for an (Nk, k) repetition code the minimum distance will be N.

2. A code with minimum distance $d_{\min}$ can detect $d_{\min} - 1$ errors. Since $d_{\min} = N$, setting $d_{\min} - 1 = N - 1 = 1$ we find $N \geq 2$ to detect 1 error.

   The code rate would be the number of data bits per transmitted bit which is $\frac{1}{N}$. The rate for a single-error-detecting repetition code would be $\frac{1}{2} = 0.5$.

3. Solving for $d_{\min}$ with $t = 3 \leq \lfloor \frac{d_{\min}-1}{2} \rfloor$ we find $d_{\min} \geq 7$. So to correct 3 errors a repetition code would have to repeat each bit 7 times and the code rate would be $\frac{1}{7} \approx 0.14$.

Question 2

Looking at the received bits:

$$
\begin{array}{cccc|c}
0 & 0 & 1 & 1 & \underline{1} \\
0 & 1 & 0 & 1 & 0 \\
1 & 1 & 1 & 1 & 0 \\
0 & 1 & 0 & 0 & 1 \\
\hline
\underline{0} & 1 & 0 & 1 & 0
\end{array}
$$

(a) The first row and the first column's parity bits (underlined) are not correct so there must be an error.

(b) If there was only one error it must be in the bit that is in the first row and the first column – the top left bit (circled).

(c) This code is only guaranteed to correct one error since it would be impossible to detect two or four errors in any row or column (the parity would be unaffected). However, the code can correct those multi-bit error patterns where all errors lie on the diagonal.

(d) When the number of data bits is $k = m^2$ we would need to transmit m^2 data bits and $2m + 1$ parity bits so the code rate would be:

$$\frac{m^2}{m^2 + 2m + 1}$$

(e) For $k = 64$, $m = \sqrt{64} = 8$ and the code rate is:

$$\frac{8^2}{8^2 + 2 \cdot 8 + 1} = \frac{64}{64 + 17} \approx 0.79$$

Although such horizontal-and-vertical parity check codes[1] are more efficient than repetition codes, the most efficient single-error-correcting codes are Hamming codes. These are also easy to implement. For an integer value m a Hamming code has a codeword size of $n = 2^m - 1$, $k = 2^m - 1 - m$ data bits and $n - k = m$ parity bits. For example, for $m = 6$ each codeword has $n = 63$ bits and $k = 64 - 6 - 1 = 57$ data bits for a rate of $\frac{57}{63} \approx 0.90$.

Question 3

(a) At 100 kb/s the bit duration is 10 μs and a 40 μs-long noise impulse would affect four consecutive bits. The interleaver must spread this out over at least four codewords so the block interleaver depth (number of rows) must be at least 4. The interleaver width should be the FEC codeword size of $n = 256$ bits.

Each interleaved block requires $4 \times 256 \times 10 = 10.24$ ms to transmit which is less than the period of the noise and so there will be only one

[1] Sometimes known as transverse and longitudinal parity checks since they were originally used to protect information stored on tape.

noise impulse per interleaver block. If the period were shorter than this then we would have an average of more than one bit error per codeword and we would need to use a more powerful FEC code.

(b) The delay added due to interleaving can be defined as the time from when the first bit is input to the interleaver to when that same bit is output from the de-interleaver.

As computed above, it takes 10.24 ms to fill the block interleaver at the transmitter before transmission can begin. Then it takes 10.24 ms to transmit the contents of the interleaver and fill the de-interleaver memory – which can happen while the data is being received. At this time the de-interleaver at the receiver is full and the first bit of the received data can be output. Thus the additional delay due to interleaving is $2 \times 10.24 = 20.48$ ms.

Question 4

(a) The sequence of $2^n - 1$ values of the shift registers (SR's) can be computed by hand and are shown below for the two feedback structures.

	External Feedback		Internal Feedback	
step	SR	output	SR	output
1	1111	1	1111	1
2	0111	1	1110	0
3	0011	1	0111	1
4	0001	1	1010	0
5	1000	0	0101	1
6	0100	0	1011	1
7	0010	0	1100	0
8	1001	1	0110	0
9	1100	0	0011	1
10	0110	0	1000	0
11	1011	1	0100	0
12	0101	1	0010	0
13	1010	0	0001	1
14	1101	1	1001	1
15	1110	0	1101	1
1	1111	1	1111	1

(b) Yes, each sequence has 8 1's and 7 0's.

(c) Yes, in both cases the period is 15 (the shift register contents return to all-1's after 15 bits).

(d) By inspection, the two sequences are the same but in reverse order.

You could also use a script to compute the shift register values. For example, using Matlab (Octave):

```
sre=[1,1,1,1];
sri=sre;
for i=[1:16]
   fprintf("%10d %d%d%d%d %d%d%d%d\n",[i,sre,sri])
   sre=[xor(sre(3),sre(4)),sre(1),sre(2),sre(3)];
   sri=[sri(4),sri(1),sri(2),xor(sri(3),sri(4))];
end
```

which outputs:

```
         1 1111 1111
         2 0111 1110
         3 0011 0111
         4 0001 1010
         5 1000 0101
         6 0100 1011
         7 0010 1100
         8 1001 0110
         9 1100 0011
        10 0110 1000
        11 1011 0100
        12 0101 0010
        13 1010 0001
        14 1101 1001
        15 1110 1101
        16 1111 1111
```

or using Python:

```
sre=[1,1,1,1]
sri=sre
for i in range(1,17):
    print("{:10} {}{}{}{} {}{}{}{}"
          .format(*([i]+sre+sri)))
    sre=[sre[2]^sre[3],sre[0],sre[1],sre[2]]
    sri=[sri[3],sri[0],sri[1],sri[2]^sri[3]]
```

which produces the same output as above.