Project Report

CD Digital Audio

Gerbrand De Laender
Jan Mariën
Samuel Rimbaut
Arne Vandenberghe

academy year 2019–2020

1 Task Division

Task	Student
Subcode	Samuel
Encoding	Samuel
Decoding	Arne
AudioCD function 1-3, 8	Jan, Gerbrand
AudioCD function 4-7,9,10	Samuel, Gerbrand
AudioCD question 1,4,6	Samuel
AudioCD question 3	theoretical Samuel, simulation Gerbrand
AudioCD question 5	Arne, Gerbrand

2 Audio CD Subcode

1. What is the datarate of the subcode in subcode blocks/s? And in bit/s?

> The subcode block has a datarate of 75 blocks/s and is 98 symbols large. 2 of these are used for syncing purposes and thus do not contribute to the data rate. This means that the bitrate is given by 75x96x8bit/s = 57.6kbit/s. If we only counts the channels R− >W the answer becomes 43.2kbit/s.

2. The subcode consists of 8 channels (P,Q,R,S,T,U,V and W), complete Figure 3 with the state of the channel P flag bit.

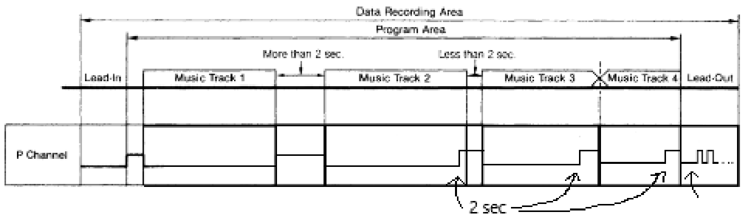


Figure 1: P-flag

3. How does the CD player know the number of tracks, the starting positions of the tracks and the total playing time of the audio CD? How is this data protected from errors?

> As can be seen in figure 1, the start of a track is indicated by the P-flag. The total playing time and the number of tracks in encoded in the Q-channel. This data is protected from errors with a CRC. Furthermore at the end there is also an EFM encoder which adds extra redundancy to the code (8 to 14 bits).

3 Encoding

1. Let α be a primitive element of GF(2^8). Consider the primitive polynomial given by: p(D) = $D^8 + D^4 + D^3 + D^2 + 1$. Show that p(D) is a primitive polynomial with root α for the field GF(2^8).

A primitive polynomial is the minimal polynomial (i.e. an irreducible polynomial) of a primitive element in the defined galois field. As such it may not be divisible by any polynomial with coëfficients in GF(2) that is of order 4 or lower. (Order 5-8 will be found automatically paired with a polynomial of order 3 or lower if there are any. These polynomials are of the form:

$$divisor(x) = ax^4 + bx^3 + cx^2 + dx + e \neq 0, 1 \tag{1}$$

with a,b,c,d,e $\in$ GF(2). Note that this yields 30 polynomials that should be verified. This is done in matlab yielding that indeed p(x) is irreducible.
Next we must verify that α is a root of p(x). This also implies that $\alpha^{(2^k)}$, k = 1..7 are roots of p(x). This leads to:

$$p(x) = \prod_{k=0}^{7} (x - \alpha^{2^k}) \tag{2}$$

Doing this in matlab indeed verifies that p(x) is a primitive polynomial of α.

2. Construct the generator polynomials $g(x) = g_0 + g_1 x + g_2 x^2 + ... + g_{n-k} x^{n-k}$for the C1 Reed-Solomon code described in the standard. Give the coeficients g_i for i = 0..n - k (in GF(2^8)). What is the value of m_0?

The generator polynomial can be found in the standard by means of the check matrix. This yields the roots: $1, \alpha, \alpha^2, \alpha^3$ such that the generator polynomial is given by eq. (3). This means that $m_0 = 0$.

$$g(x) = (x - 1)(x - \alpha)(x - \alpha^2)(x - \alpha^3) \tag{3}$$

Solving this for the coefficients in $GF(2^8)$ yields equation (4).

$$g(x) = x^4 + \alpha^{75} x^3 + \alpha^{249} x^2 + \alpha^{78} x + \alpha^6 \tag{4}$$

3. Give the corresponding check polynomial h(x). It is not necessary to write down the complete polynomial, giving the first and last 5 terms is sufficient.

h(x) can be found from the property

$$x^{n-1} - 1 = g(x)h(x) \tag{5}$$

with n = 256
Such that

$$\begin{aligned} h(x) = \quad & x^{251} + \alpha^{75}x^{250} + \alpha^{163}x^{249} + \alpha^{131}x^{248} + \alpha^{143}x^{247} + ... \\ & + \alpha^{125}x^{4} + \alpha^{116}x^{3} + \alpha^{151}x^{2} + \alpha^{66}x + \alpha^{249} \end{aligned} \tag{6}$$

4. What is the minimal Hamming distance for this code. Use the fact that a Reed-Solomon code is an MDS code.

An MDS code is a code for which the number of codewords = q^{n-d+1} (singleton bound) with q = 2^8. We also know that the number of codewords = q^k since this has to equal the number of information words. As such k = n-d+1 or d = n-k+1 = 5.

5. Explain why a Reed-Solomon code is -in general- more appropriate for data protection than a binary BCH code.

RS-codes encode per symbol. This means that they are more suitable for protection against burst errors. Furthermore one may already note that using the CIRC encoding scheme will lead to good performance both in terms of burst errors and random (spread out) errors.

4 Decoding

1. How many erroneous bits can the C1 Reed-Solomon code always correct? What's the maximum number of correctable bit errors?

> The code can correct up to two symbols. Therefore, the code can correct up to a maximum of 16 bits. However, the amount of bits that the code can always correct is only 2, because there are only 2 bit errors needed to introduce 2 symbol errors.

2. Give the relationship between the bit error rate P_b and the symbol error rate P_s for GF(2^8).

> $P_b = (\text{average amount of bit errors/symbol})P_s = \frac{2^{K-1}}{2^K-1}P_s = 0.502P_s$

5 Audio encoding in CDs

1. Explain the function of every component of the CIRC structure in figures 12 and 13 of the standard ('Delay of 2 frames', 'Interleaving sequence', 'C2 encoder', 'Delay lines of unequal length', 'C1 encoder' and 'Delay of 1 frame'). Which types of errors (random, short burst, long burst) does C1 protect against? And C2?

> We find it most interesting to explain the function of each blocks with regards to the decoding schematic. This is included in figure 2.
> First note that when the incoming symbols are demodulated the even symbols are delayed by 1 frame and some symbols are inverted. The symbols were inverted initially to prevent sending the 0-word.
> Next the C1 decoder is applied. This decoder can correct random errors and short burst errors (a symbol is 8 bits such that 9 subsequent erronous bits never cause more than 2 errors). It however serves another important purpose. It also works as an error detector rather than a corrector. Detected errors get erasure flags assigned. This will serve as extra information that can be used by the C2 decoder.
> Next the delay lines of unequal length serve to severely rearrange the symbols before they are fed to the C2 input. This is such that the erasure flags of the C1 decoder can be redistributed as the C2 decoder can correctly decode with up to 4 erasure flags. Note that this step significantly increases the burst error correctability of the CIRC encoding scheme as burst errors are essentially converted into random errors!
> After this the symbols are again scrambled in pairs of 2 (per word) and some are also delayed again. This initially seems to serve little purpose as all decoding is already done. However this step will increase the effectiveness of the interpolator by again seperator the flagged symbols! The interpolator works better the closer the symbols are to the 'to be interpolated' symbol (closer in time means more reliable, also see later question about interpolator).

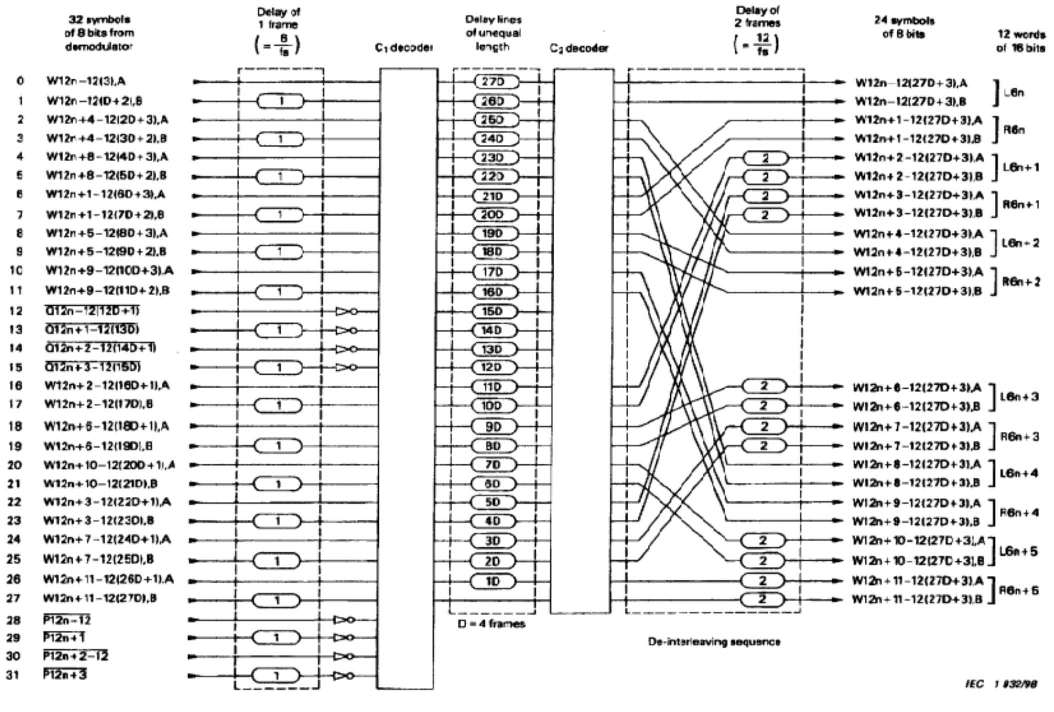


Figure 13 – CIRC decoder

Figure 2: structure of the CIRC decoder

2. Implement the functions. (added to keep the question numbering correct)
3. What is the maximum duration of a burst (in bits) that can always be corrected if we assume that C2 can correct up to 4 erasures ? Translate this to the maximum width of a scratch if the scanning velocity of the laser is 1.3 m/s. (Hint: the audio samplerate is 44.1 kHz.) Compare this with simulation results of your Matlab code and explain.

For this assume that a long burst of symbols at the input of the decoder is wrong, whilst all the other symbols are correct. Note that due to the delays, the symbols at the input of the C2 decoder will be delayed by 4 frames per symbol. If we assume that 4 symbols may be erasures (errors at C1 will set erasure flags) then the max. length of a burst error is 16 frames in total. Since a frame is 32 symbols of 8 bits long, the max. burst error is given by Eq. (7).

$$L_{burst,bits} = 32 \text{ symbols/frame x } 8 \text{ bits/symbol x } 16 \text{ frames} = 4096 \text{ bits} \quad (7)$$

This can now be translated into a scratch length. This is an interesting parameter as scratches on CDs are common and do indeed cause burst errors. Taking into account that the sample rate is 44.1kHz, samples are 16 bits words that come in pairs of 2 (stereo) and that the code rate is given by $\frac{24}{32}$, we can now calculate the number of bits that must be read per second (after encoding):

$$\text{no. bits/s} = 44.1\text{kHz x 16bits x 2 audio channels x } \frac{32}{24} = 1.88 \text{ Mbits/s} \tag{8}$$

This means that if the laser moves over the CD at 1.3 m/s, the (average) size of 1 bit on the CD should be:

$$\text{bit size} = \frac{1.3 \text{ m/s}}{1.88 \text{ Mbits/s}} = 0.69\ \mu\text{m/bit} \tag{9}$$

Or that the max scratch length is given by:

$$L_{scratch} = 0.69\ \mu\text{m/bit x 4096 bits } = 2.82 \text{ mm} \tag{10}$$

The simulation results are in line with those expectations. As shown in Figure 3, the CIRC decoder can correctly decode the signal for scratch lengths up to around 4000 bits, since no erasure flags are present at the output of the decoder (and there are no undetected errors).

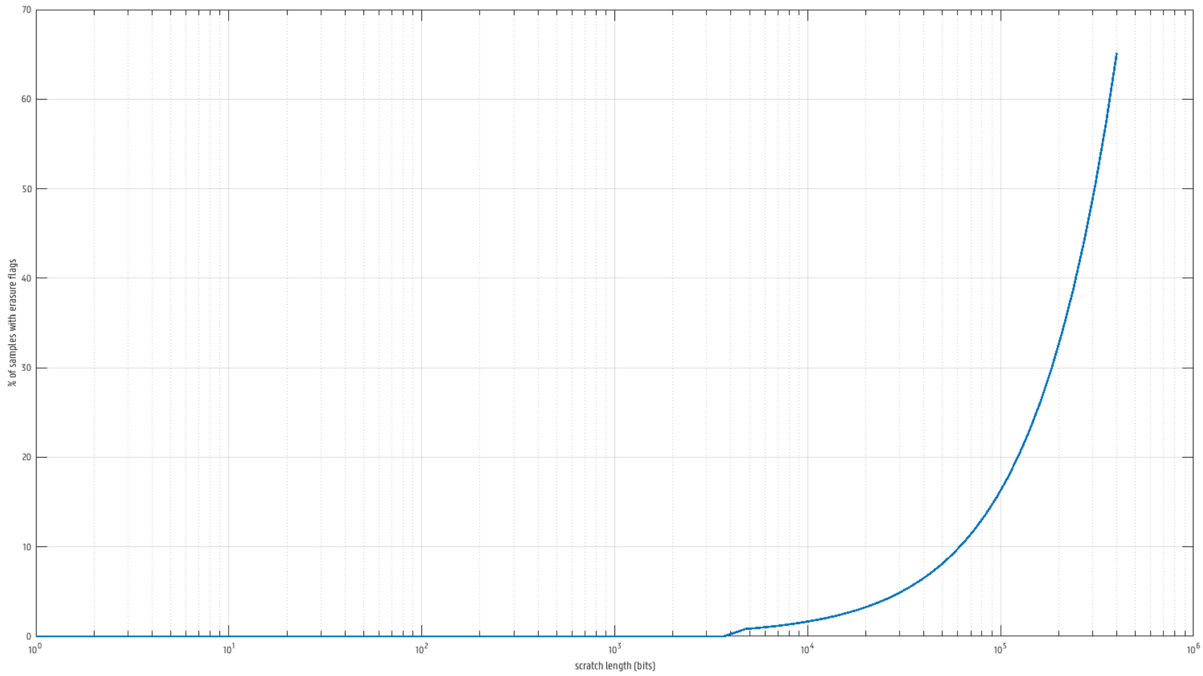


Figure 3: Percentage of samples with erasure flags versus scratch length (in bits).

4. At the output of the CIRC decoder, an interpolator masks the unrepairable errors using linear interpolation of a maximum of 8 subsequent samples. Describe, qualitatively, why this works and what the relation with the sample frequency of the system is.

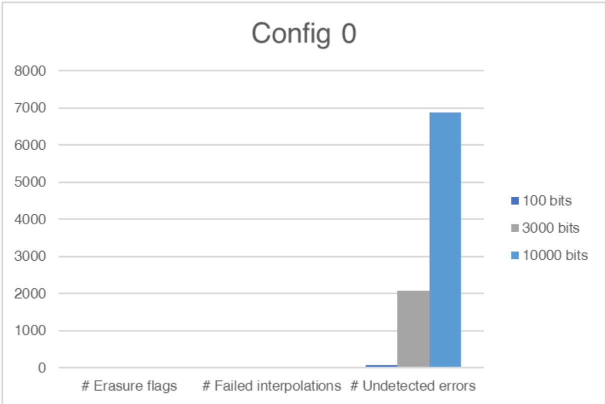
Interpolation can do a good job because sound is a continuous wave. This means that as long as the sample frequency is high enough (higher than 2x20kHz for sound, prevents aliasing) one will find that interpolation is a good way of correcting lost samples.
In essence the sampling is higher than the nyquist sampling frequency such that some lost samples are tolerable without losing the ability to reconstruct the original signal. This means that even if an error of longer than the max burst length occurs, the audio quality can still be kept high. Ofcourse the scratchlength can become too large even for the interpolator to still work.

5. Compare the performance of the 4 configurations (see AudioCD class) under the following circumstances:

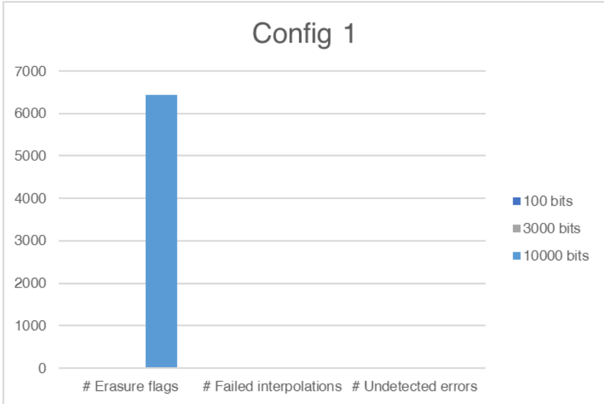
    - A scratch of 100, 3000 and 10000 databits wide, repeated with a period of 600000 bits (the scratch is encountered at every rotation of the disc).

First we note that the file 'Hallelujah_22050.wav' was used for these tests. The used file does not have an effect on the performance of the different configurations, but the number of samples is of dependent on this. (total no. samples = 417024)

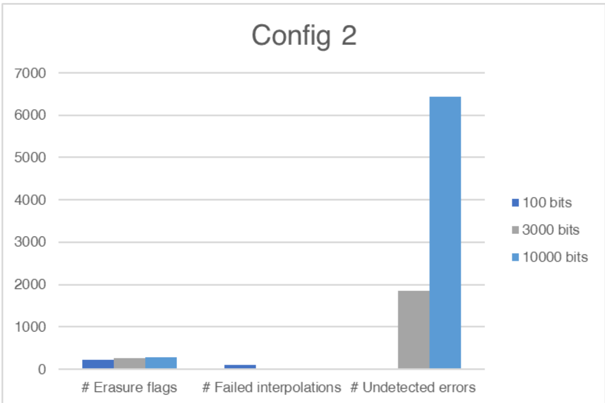
Figure 4 shows that configuration one is the best option. It has no configuration flags or failed interpolations up to a scratch lengths of 10000 databits wide. At 10000 databits, configuration 1 is still able to correctly adapt all errors by interpolating the erased symbols. Configurations 2 and 3 are able to detect errors, but only if the databits scratch width remains small. The table also shows that the amount of erasure flags and failed interpolations of configurations 2 and 3 are less significant, since the amount of undetected errors is nearly the same amount as the amount of undetected errors by configuration 0. Which means that configurations 2 and 3 are unable to correctly find possible errors in the first place.
This lies within expectations as in configurations 2 and 3 do not have any incorporated measures to protect against burst errors.



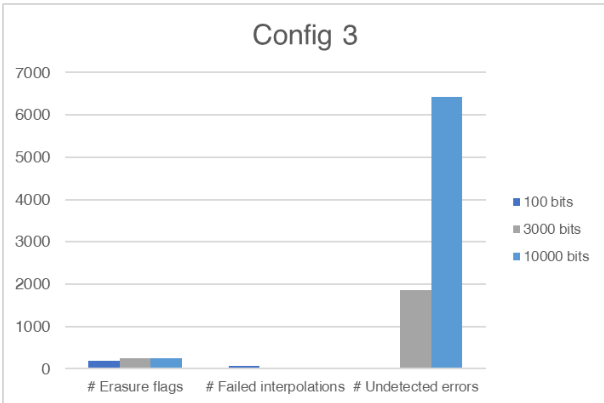
(a) Performance of configuration 0



(b) Performance of configuration 1



(c) Performance of configuration 2



(d) Performance of configuration 3

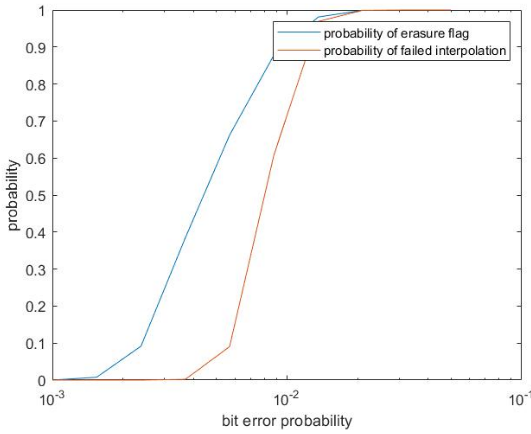
Figure 4

- Random bit errors with a bit error probability p between 0.05 and 0.001 generated by the Matlab command logspace(-1-log10(2),-3,10). Plot the probability that an audio sample has been tagged as an erasure (before interpolation) and the probability that an audio sample could not be interpolated, for configurations 1, 2 and 3 (configuration 0 performs no error correction). Note: think about the scale (linear or logarithmic) you use for the axes!

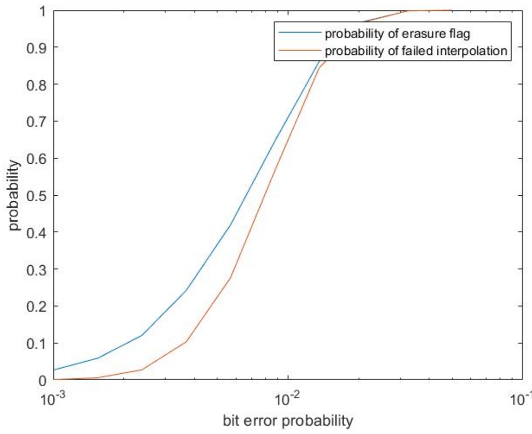
Figure 5 implies that configuration 1 has the worst performance. This might be due to the reshuffling causing more harm than good for high random error probability. After all, reshuffling can cause previously flagged elements to be decoded correctly by c2 but also cause previously correct symbols to be flagged by the C2 decoder. Ofcourse the no. interpolation failures is further from the no. erasure flags in config. 1 because of the last reshuffling step.

Configuration 3 also expectedly performs better than configuration 2. This is because in configuration 2, when the C1 errors fails to decode, the C2 will also fail (no shuffling). Thus the effective error correcting capability of config. 3 is larger than that of config. 2.

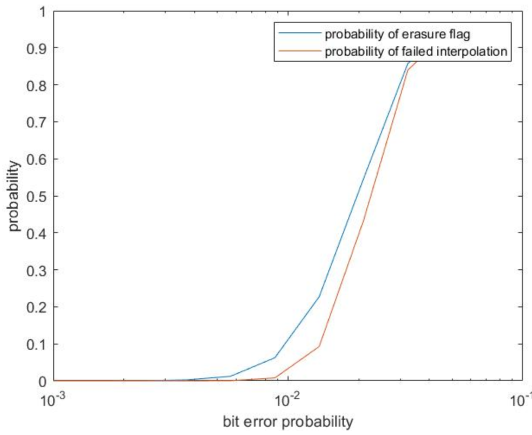
Finally also note that the erasure flag probability goes up to 100%! This is because if there are sufficient errors within 1 frame, the whole frame is flagged. This does not mean that all symbols are incorrect. It just means that the coding scheme does no better than using no scheme at all.



(a) Probabilities with configuration 1



(b) Probabilities with configuration 2



(c) Probabilities with configuration 3

Figure 5

6. After CIRC encoding, the databits are modulated with 'EFM' before being written to the disk. What is EFM and why is it used? Could the EFM demodulator give extra information to the CIRC decoder to improve its error correcting capability?

> The main use of EFM is that CD's carry information by means of grooves into the CD surface. What is actually encoded are transitions of bits by means of these grooves. The EFM step ensures that transitions are far enough apart i.e.

the 1's are seperated by at least 2 zeros such that they can be encoded onto the CD. This encoding of transitions is called Non-return-to-zero (inverted) or NRZ(I) for short. This is also shown in figure 6 as completion.
If the read EFM symbol on the CD does not correspond to a symbol from the lookup table, one could conclude that an error has occured. However this is probably corrected (inherently) by the EFM demodulator as the amount of possible 14-bit patterns that have sufficiently far apart spaced 1's is very limited and this requirement is related directly to the physical restrictions of the read-out system.
For this reason the EFM demodulator cannot give extra information to the CIRC structure.

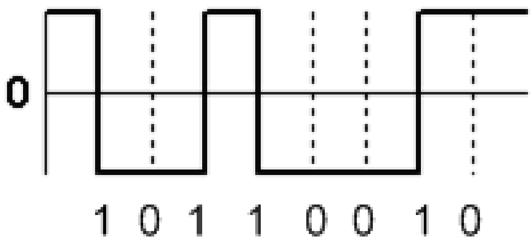


Figure 6: example of NRZI as used on a CD surface