DAT330 – Principles of Digital Audio
Cogswell Polytechnical College
Spring 2009

Week 4 – Class Notes

Error Correction

5. Error Correction

Sources of Error

Error can be generated at any stage of the digital audio signal chain. Common error sources are caused by quantization error, converter nonlinearities, jitter, and peak shift. However, error can be minimized if care is taken during the design of circuits and the manufacture stage.

Because error can be most severe during the storage and transmission process, error correction techniques will focus most on these media. Errors occurring in the recording or the storage medium, resulting in corrupted data, will manifest themselves as a defective audio signal. A common form of digital error is a defect in the digital medium that causes a momentary reduction in signal strength, or dropout. This form of error can be caused by a manufactured effect in the medium, or by a defect introduced during use.

If such defects in the media are read, signal transitions will be misrepresented and as a result, the data will be misread. Invalid or lost data can provoke clicks or pops because the DAC output will suddenly jump to a new amplitude.

The severity of an error depends on the nature of the error. For example an error in the LSB of a PCM word can be unnoticeable, while an error in the MSB can create drastic amplitude changes.

We can classify errors into two groups:

1. Random-bit errors – errors that have no relation to each other. Occur in a single manner and can be easily corrected.
2. Burst errors – large and disruptive error. Caused by manufacturing defects, foreign particles in the medium, noise spikes, crosstalk, or transmission channel problems.

Because error-correction systems need to be able to correct both types of errors simultaneously, it is important that they have an adequate burst length, or the maximum number of adjacent erroneous bits that can be corrected. Also, it is important that the correction techniques employed are optimized for the particular medium or application at hand. Finally, the modulation code used to represent the data influences the particular error-correction design.

Data can be degraded by several factors during transmission, such as bandwidth limitations, intersymbol interference, attenuation, ringing, reflection, and noise.

The integrity of data can be quantified by means of the bit-error rate (BER). This number is the number of bits received with error divided by the total number of bits. For example, a BER of 10^{-5} specifies one bit of error for 10^{5} bits received.

While the BER shows the total number of errors in a data stream, it does not give their distribution. For example, a BER value can be the same for a single large burst error, as for several smaller bursts. Because of this, the BER is not a good indicator for a reliable channel subject to error.

Alternatively, the block-error rate (BLER) measures the number of blocks or frames of data that have at least one occurrence of data error. The BLER is measured as an error count per unit of time. While this measure of error does not provide a literal number of errors, it does give their severity.

The burst error-length (BERL) counts the number of consecutive block in error, specified as counts per second. This measure does not provide the number of bits in error, rather, it counts the occasions of error.

Objectives of Error Correction

The errors introduced in the storage or transmission of data can be corrected, so as to provide an accurate recovery, by providing data redundancy and proper error detection-correction coding. Theoretically, it is possible to provide perfect error-correction (every error is detected and corrected). However, this would create an unreasonably high amount of data overhead because of the amount of redundant data required. Thus, an efficient error-correction system minimizes the amount of redundant data and processing, as well as having a low error rate.

An error-correction system has to provide the following operations:

1. Use redundancy to permit data to be checked for validity.
2. Use redundancy to replace erroneous data with newly calculated valid data.
3. Apply error concealment techniques, if the error is large or there is insufficient data for correction, that substitute approximately correct data for invalid data.

In the worst case scenario, when no error concealment is possible, the digital audio system must mute the output signal. Otherwise, the circuitry attempts to decode the incorrect data and will produce incorrect sounds.

Error Detection

All error-detection and correction techniques are based on the redundancy of data. This data is derived from existing data, and thus conveys no additional information. As a general rule, the greater the likelihood of errors, the greater the redundancy required.

Redundancy is necessary to validate data, otherwise there is no way to check its validity at the receiving end. It can be seen that additional information is needed to reliably detect errors in the received data. This additional data is generated by the original data, an as such, it is subject to the same error-creating conditions. Therefore, an error-detection system must properly encode the stored or transmitted information, so when data is corrupted, the error can be detected.

Single-Bit Parity

Parity, in general, can be defined as the state or condition of being equal. In computation, parity is a function whose being ever or odd provides a way of checking a set of binary values. Parity bits can be formed in the following manner: if the number of 1's in a data word is even (or zero), the parity bit is a 0; if the number os 1's is odd, the parity bit is a 1. Parity bits are added together with modulo-2 addition, such that an 8-bit word is made into a 9-bit word with an even parity bit.

The validity of the received data word is tested by using the parity bit (the data bits are added together to calculate the parity of the received data).

If the received parity bit and the calculated parity bit are not equal, an error has occurred. Notice that the single-bit detector has no correction ability. Probability dictates that the error is in the data word and not in the parity bit itself, however the reverse could be true and a parity bit error could be detected.

Single-bit parity detectors are not reliable and are not suitable for burst errors. Thus, this detection method is no adequate for digital audio systems.

Cyclic Redundancy Check Code

This error detection method is preferred in audio applications since it is able to detect burst errors in the recording medium or transmission channel. CRCC is a cyclic clock code that generates a parity check word by adding together the bits in a data word. The resulting check word is appended to the data word to form the codeword for transmission or storage. Any disagreement between the received checksum and that formed from the received data would indicate the probability that an error has occurred.

In simple terms, each data block is divided by an arbitrary and constant number. The remainder is appended to the stored or transmitted data block. In the reproduction stage, division is performed on the received word. If the remainder is a zero, the signal has no errors.

In practice, given a k-bit data word with m (where $m = n - k$) bits of CCRC, a codeword of n bits is formed. The error detectability of CCRC code can be summarized as follows:

1. Burst errors less than or equal to m bits are always detectable.
2. Detection probability of burst errors of $m + 1$ bits is $1 - 2^{-m+1}$.
3. Detection probability of burst errors longer than $m + 1$ bits is $1 - 2^{-m}$.
4. Random errors up to three consecutive bits can be detected.

CCRC is quite reliable, however the medium determines the design of the CCRC and the rest of the error-correction system.

Error-Correction Codes

Data redundancy (duplicating data) and channel coding (error correcting sequences longer than the original data word) can indeed be useful when correcting errors, however they are inefficient because of the data overhead they require. A more effective approach is to use error correction codes. While they use data redundancy to form the code, less redundancy is required because the codes are designed with a particular application in mind. Algorithms are responsible for the detection and correction decoder. It must be noted that detection and correction codes cannot work unless one or more of the possible data words are defined as being disallowed in the code. The more words that are disallowed, the more robust the detection and correction.

Two approaches are used: block codes that use algebraic methods, and convolutional codes using probabilistic methods.

Block Codes

This type of encoders assemble a number of data words to form a block and, operating over that clock, generate one or more parity words and append them to the block. During the decoding process, an algorithm forms a syndrome word (an error word) that detects errors and, given the redundancy, corrects them.

These algorithms are used in digital audio applications. Error correction is further enhanced by interleaving consecutive words. Block codes base their parity calculation on an entire block of information to form a parity word. Parity can be formed using single-bit or cyclic code. In this way, greater redundancy is achieved and correction is improved.

For enhanced performance, two parity words can be generated to protect the data block. If any two words are erroneous and marked with pointers, the code provides correction. Similarly, if any two words are marks with erasure, the code can use the two syndromes to correct the data.

Block codes, because of of their discrete nature, are more suitable for editing recorded data.

Hamming Codes

Hamming codes are a special subset of block codes that create syndromes that point to the location of the error. Multiple parity bits are formed for each data word, with unique encoding. For example, three parity check bits can be added to a 4-bit data word; seven bits are transmitted. The three parity bits are uniquely formed by modulo addition of a particular combination of three data bits. The total seven bits provide 128 different encoding possibilities, but only 16 are used; the remaining 112 patterns denote an error.

This particular code requires that for any word to change into another one, at least three bit changes are required. The dissimilarity in data corresponds to the code's error-correction ability. Thus, any n-bit error leaves the received word closer to the correct word than any other. The number of bits that one legal word must change to become another legal word is known as the Hamming or minimum distance. The hamming distance defines the potential error correction of a code. As this distance increases, so does the code's correction ability.

Block codes are notated in terms of the input relative to the output data. Data is grouped in symbols, with the smallest symbol being 1-bit long. Thus, *k* symbols (input) are used to create a larger *n*-bit symbol (output), notated as (*n, k*). The parity symbols are generated by *n – k.*

Convolutional Codes

Also known as recurrent codes do not partition the data into blocks, instead the message digits *k* are taken few at a time and used to generate coded digits *n,* formed not only from those *k* message digits, but from previous *k* digits as well, saved in delay memories. In this manner, the coded output contains a history of the previous input data. Such codes are called (*n, k*) convolutional code. It uses (N – 1) message blocks with *k* digits.

Encoding is performed and codewords are transmitted or stored. Upon retrieval, the correction decoder uses syndromes to check codeword errors. The delay memories required in the encoder and decoder are implemented by shift registers. The amount of delay determines the code's constraint length, which is analogous to the block length of a block code.

Convolutional codes are inexpensive to implement, and perform well under high error conditions. However, they do posses the disadvantage of error propagation. If an error cannot be fully corrected, it will generate syndromes reflecting this error and can introduce errors in subsequent decoding.

Convolutional codes are often used in broadcasting.

Interleaving

When an error is sustained, such as a burst error, large amounts of continuous data (including redundant data) can be lost. In this case, error correction is impossible. A way in which this can be avoided is by interleaving or dispersing data through the data stream prior to storage and transmission. For example, if a burst error occurs, upon playback the bitstream is de-interleaved; thus, the data is returned to its original sequence and the errors are distributed through the bitstream. The valid data and redundancy surround the damaged data, and the correction algorithm is able to better reconstruct the damaged data.

Without interleaving, the amount of redundancy is dictated by the size of the largest correctable burst error. Interleaving, thus provides the advantage of having the largest error that can occur in a block to the size of the interleaved sections. As such, the amount of redundancy is determined by the size of the interleaved section, as opposed to the burst size.

The interleaving and de-interleaving process may yield only one erroneous word per block, which facilitates the functioning of block checksums. Thus, interleaving increases the correctability of burst-errors in block codes. Bit-interleaving, having the same purpose as block interleaving, allows burst errors to be handled as shorter burst errors and random errors. Any interleaving requires a memory buffer of sufficient length to hold the distributed data during interleaving and de-interleaving.

Cross Interleaving

If burst errors and random errors occur simultaneously, interleaving might not be adequate to deal with them. Event though burst errors are scattered, random errors in a given word tend to overload the correction algorithm. This problem can be solved by producing two correction codes, separated by an interleave and delay. This process is known as cross interleaving. If the blocks are arranged as a matrix, the code is called a product or crossword code. Cross-Interleave Code (CIC) is comprised of two or more block codes assembled in a convolutional structure. This method is effective because the syndromes generated from one code can be used as an error detector, while the other code can be used as a corrector code.

Product codes are used in DVD's and a variation of CIC codes, the Cross-Interleaved Reed-Solomon code (CIRC), is used in CD's.

Reed-Solomon Codes

Reed-Solomon (RS) codes are a subset of Hamming codes that provide multiple-error correction. When combined with cross-interleaving and error detection pointers such as CCRC, they are effective for digital audio applications because they can correct both burst and random errors.

Other applications of RS codes are direct broadcast satellite, digital radio, and digital television applications.

Cross-Interleaved Reed-Solomon Codes

CIRC code is a quadrature erasure, or double error, correction code used in the CD format. CIRC applies RS codes sequentially, with an interleaving process between C2 and C1 encoding. Encoding carries data through the C2 encoder, then the C1 encoder. Decoding will reverse this process. C2 is a (28,24) code, while C1 is (32,28) code.

By combining interleaving and parity checks, the data is more resistant to errors generated during storage, the data is encoded before being stored in the disc and decoded upon playback.

The block-error rate (BLER) measures the number of data frames that have at least one occurrence of uncorrected bytes at the C1 error-correction decoder input (E11 + E21 + E31); thus it is a measure of both correctable and uncorrectable errors at the decoding stage. E11, E21, E31, E12, E22, and E32 are error counts that asses the integrity of the stored data that passes through a CIRC algorithm. The first digit of the error count specifies the number of erroneous bytes (or symbols), while the second digit specifies at which decoder (C1 or C2) they occur. As a brief note, E22 and E32 errors are the worst type of errors since they are considered uncorrectable. They usually indicate a localized damage to a disc, produced by either the manufacture process or through usage.

The BLER is a good measure of the discs data quality. High BLER counts indicate many random-bit errors produced by poor pit geometry.

Product Codes

Product codes arranged in a two-dimensional manner, or a matrix, are separated by block interleaving. The code that is the first to encode and the last to decode is called the outer C2 code. The code that is second to encode and the first to decode is called the inner C1 code. Product codes use this method of crossing two error-correction codes. The C2 code adds parity *P* to the block rows, while the C1 code adds *Q* parity to the block columns.

During decoding, *Q* parity is used by the C1 decoder to correct random-bit errors, and burst errors are flagged and passed through de-interleaving to the C2 decodes. *P* parity and flags are used by the C2 decoder to correct errors in the inner codewords.

Error Concealment

While it is theoretically possible to achieve perfect error detection and correction, in practice it is not convenient due to its large data overhead. A practical error-correction method must provide a balance of those limitations against the probability of uncorrected errors, while allowing severe errors to remain uncorrected. In subsequent processing, an error concealment system compensates for those errors and ensures that they are not audible.

There are two types of uncorrectable errors output from the correction algorithm. The first error can be detected, but not corrected by the algorithm. However, it can be concealed with a proper concealment method. The second type are errors that are not detected and miscorrected. These errors cannot be concealed and might produce artifacts in the audio output. They are caused by simultaneous burst and random errors, and thus have to be minimized.

An error-correction system must reduce the amount of undetected errors, while applying error-concealment techniques to resolve uncorrectable errors.

Interpolation

Interpolation, in its simplest form, holds the previous sample value and repeats it to cover the missing or incorrect sample. This is called zero-order or previous-value interpolation.

First-order interpolation, or linear-order, replaces the erroneous value with a new sample derived from the mean value of the previous and subsequent samples. Many digital audio systems use a combination of zero- and first-order interpolation.

Other higher-order interpolation can be used as well; in this case, *n*th-order polynomials are used to calculate the data.

Muting

Muting sets the value of the missing or uncorrected words to zero. Silence, in this case, is preferable to the unpredictable sounds resulting from decoding erroneous data. Muting is used to prevent audible clicks produced by uncorrected errors, and it is a preferable method when severe data damage or player malfunction occurs. Muting can create a momentary, usually imperceptible, increase in distortion, however muting algorithms can gradually attenuate the output's signal amplitude prior to a mute and gradually restore the amplitude afterward.

Concealment strategies are assisted when audio channels are independently processed. For example, muting one channel rather than both.

Duplication

Digital audio recording provides an advantage over analog recording/duplication in which subsequent copies of a recording will not degrade as easily. Digital audio can be lossless, but its success depends on the effectiveness of the error-correction system. Although error-correction techniques can provide completely correct data, error-concealment does not because it can possibly introduce audible errors into the copied data. Thus, subsequent digital copies can contained accumulated errors not found in the original. Therefore, errors must be corrected at their origin.

Digital audio data can be reliably copied. Both the original file and its copies are identical. However, they do not have control over the media in which they are copied to or reproduced.