DAT330 – Principles of Digital Audio
Cogswell Polytechnical College
Spring 2009

Week 2 – Class Notes

Digital Audio Recording

3. Digital Audio Recording

Pulse-code Modulation

Modulation is a means of encoding information for the purpose of transmission or storage. For example, in the analog realm, AM (amplitude modulation) and FM (frequency modulation) are used for radio broadcast. Because they encode continuous information, they are referred as wave-parameter modulation.

For sampled information pulse modulation is used for representing a signals amplitude at sample time. For example:
Pulse-width modulation (PWM) – the width in time is encoded.
Pulse-position modulation (PPM) – the pulse position in time is encoded.

In both cases, the original signal amplitude is coded and conveyed through constant-amplitude pulses.

However, a signal's amplitude can also be represented directly by pulses. Examples of this encoding are:
Pulse-amplitude modulation (PAM) – the amplitude of the pulse equals the amplitude of the signal at sample time.

Other pulse modulation methods can be numerical, such as:
Pulse-number modulation (PNM) – the modulator generates a string of pulses, where each pulse count represents the amplitude of the signal at sample time.

These previous examples of pulse modulation are not suitable for transmission or recording because of error and bandwidth limitations.

The most common modulation method is pulse-code modulation (PCM). In this form of pulse modulation, the input signal is sampled, quantized, and coded. The measured signal amplitude is represented with a pulse code and thus, binary numbers can be directly used to represent amplitude values.

While PCM might need several pulses per sample and a larger bandwidth per channel, it is a robust signal that can easily be regenerated without loss. Thus, the quality of a PCM transmission depends on the sampling and quantizing process, not on the quality of the channel itself. Multiple PCM signals can be combined and simultaneously conveyed in order to expedite the use of PCM.

The encoding section of a stereo PCM recorder consists of input amplifiers, dither generator, input LPF, S/H circuits, A/D converters, multiplexer, DSP and modulation circuits, and a storage medium such as digital tape or disc.

Let us describe each part of an audio digitalization system

Dither Generator

Dither is a noise signal added to the input audio signal in order to remove or minimize quantization artifacts. The resultant audio signal varies between quantization levels and thus, decorrelates the quantization error from the signal. This action removes the quantization error, and encodes the signal amplitude below the amplitude quantization increment. Although this reduces harmonic distortion caused by the quantization error, dither is perceived as a white noise floor.

Several types of dither signals are commonly used, such as Gaussian, rectangular, and triangular probability density functions. The word length of the quantizer must be sufficiently long for the audio program, and its LSB must be appropriately dithered.

Input Lowpass Filter

To ensure that an input audio signal has frequency contents within the limits of the Nyquist frequency, a bandlimiting input filter must be employed. This ensures that the Nyquist Theorem is observed and that aliasing is prevented. This lowpass filter or anti-aliasing filter must attenuate all frequencies above the half-sampling frequency. Ideally this filter would be a rectangular filter and must not affect the phase of the input signal, and should have a flat frequency response. In reality, rectangular filters are difficult to implement. They have a passband ripple and are have a slope between the cutoff frequency and the stopband. To avoid aliasing, the cutoff frequency must be below the Nyquist frequency, or the sampling frequency must be extended to higher rates.

Analog filter criteria are overshoot, ringing, and phase linearity. Filters are classified according to the mathematical polynomials that describe their characteristics.

Sample and Hold Circuits

This circuit has two important functions: 1) time-sampling the analog waveform a a periodic rate, and 2) holds the analog value of the sample while the A/D converter outputs the corresponding digital word. This is important because the analog value could change after the designated sample time, causing the A/D converter to output incorrect digital words.

The S/H circuit sample rate is determined by an oscillator circuit, or clock, that outputs timing pulses. In this manner, the sample taking and reading are at the same rate.

If for some reason there is a variation in sampling times, the result will be a timing error called jitter. Jitter will add noise and distortion to the sampled signal. In the case of high amplitude, high frequency input signals, jitter tends to be quite significant.

As a result, timing accuracy in the A/D conversion process must be considerable. Jitter in the S/H circuit must be less than 200 picoseconds to allow 16-bit accuracy from a full amplitude., 20 kHz sinewave. As the quantization levels increase, jitter length must decrease. Under these conditions, jitter noise will fall below the quantization noise floor.

Acquisition time is the time between the initiation of the sample command and the actual taking of the sample. This time lag results in a sampled value difference from the current sample. The S/H circuit will hold the captured analog voltage while conversion takes place. The captures voltage must remain constant because any variation greater than a quantization increment can result in an error at the A/D output. Droop is the decrease in hold voltage as the storage capacitor leaks between sample times.

Analog-to Digital Converter

The A/D converter lies at the heart of the encoding side of an audio digitalization system. If any errors are introduced by the A/D converter will be added to the signal and will remain there even until the signal is outputted as an analog signal. The A/D circuit's function is to examine the sampled input signal. Determine the quantization level nearest to the sample's value, and output the binary value corresponding to that level. It must be noted that this entire process must take an entire sampling period (20μs for 48 kHz sampling frequency).

The digital word provided in the A/D conversion must be an accurate representation of the input voltage. For example, in a 16-bit successive approximation converter, each of the 65,536 intervals must be evenly spaced throughout the amplitude range, so that the LSB in the word is meaningful. However, any A/D converter will have an inherent error of ±1/2 LSB due to the quantization process itself.

The conversion time of an A/D converter is the time required to output a digital word, and must be less than one sample period. Accurate conversion from sample to sample can be difficult because of settling time or propagation errors. Meaning that the result of one conversion might influence the next.

Accuracy in the A/D converter is another important consideration. Integral linearity is a way of measuring the accuracy of the converter's output. It basically determines the deviation of an actual bit transition from the ideal transition value, at any level over the range. An *n*-bit converter is not considered a true *n*-bit converter unless it has at least ±1/2 LSB integral linearity.

Differential error linearity error is the difference between the actual step height and the ideal value of the LSB. It is the measured distance between transition voltages. Ideally, all the steps in A/D conversion should be 1 LSB. If a differential linearity error of ±1/2 LSB exists, means that the input voltage must increase or decrease from 1/2 to 1½ LSB before the output transition occurs. If this specification is exceeded some output codes would not exist.

Absolute accuracy error is the difference between the ideal level at which a digital transition occurs and where it actually occurs. This type of error should be less than ±1/2 LSB. Offset voltage, gain error, or noise error can affect this specification.

Code width, or quantum, is the range of analog input values for which a given output code will occurs. Ideally, code width should be 1 LSB. A/D converters can exhibit offset and gain errors. The first output code transition should occur at an analog input value of 1/2 LSB above 0 V. Offset error is the deviation from the actual transition value from the ideal value. Offset error can be unipolar or bipolar.

Gain error is the deviation of the actual analog value at the last transition point from the ideal value. The last output code transition occurs for an analog input value 1½ LSB below the nominal full-scale value.

The maximum analog input signal should be scaled as close as possible to the maximum input conversion range, so that the converter's maximum signal resolution is utilized.

Changes in the dc power supply can affect the A/D converter's accuracy by supplying deviations that change the positive full-scale value. This change results in a proportional change in all code transition values, thus giving a gain error.

16-bit resolution gives excellent audio fidelity, and was the former quality benchmark for most digital audio devices. Current digital audio systems can now store and process more than 16 bits. Some DSP chips process up to 56-bit words, and DVD Audio discs can store 20- or 24-bit words.

Successive Approximation A/D Converter

This type of A/D converter compares an analog input voltage with its interim digital word converted to a second analog voltage, adjusting the interim conversion until the two agree within a given resolution. The algorithm in this device sets the digital word, bit by bit, to match the analog input.

Let's describe the successive approximation process:

For example, take an analog input value of 6.92V and a 8-bit SAR A/D. The MSB in the SAR is set to 1, while the remaining bits are set to 0; thus the word 10000000 is applied to the internal D/A converter. This word places the D/A output at 5V (half its value). Because the input voltage is greater than the D/A output, the comparator will remain high. The first bit is stored at logical 1. The next significant bit is set to 1 and the word 11000000 is applied to the converter with an interim output of 7.5V. This voltage is too high, so the second bit is reset to 0 and stored. The third bit is set to 1, and the word 10100000 is applied to the converter giving an interim output of 6.25V. The third bit remains high.

This process continues until the LSB is stored and the digital word 10110001, representing the converted 6.91V, is output from the A/D converter.

Successive approximation require *n* D/A conversions for every one A/D conversion, where *n* is the number of bits in the output word. Despite the recursion, this conversion method is relatively fast and is cost effective.

Oversampling A/D Converter

In oversampling A/D conversion, the input signal is first passed through a mild analog lowpass filter. The Nyquist frequency is then extended by sampling the filtered signal at a higher sampling rate and then quantized. After this process, a digital lowpass filter decimates the signal in order to reduce the sampling frequency and to prevent aliasing at the new and lower sampling rate.

Consider an oversampling A/D converter. An analog anti-aliasing filter restricts the bandwidth to 1.5 f_s, where f_s is the sampling frequency. The transition band from 0.5 to 1.5 f_s provides good phase response. In this case a 7^{th} order Butterworth filter could be used. The signal is then sampled and held at 2 f_s, and then converted. The digital filter limits the signal to 0.5 f_s. Decimation will cause the sampling frequency of the signal to be undersampled and thus, reduced to f_s. This is done with a linear-phase FIR filter.

High sampling frequencies are useful because they can provide high resolution word lengths of just a couple of bits. Also, oversampling A/D methods reduce the in-band noise by extending the quantization spectrum far outside the audio baseband. In this case, the digital filter that prevents aliasing also removes out-of-band noise components.

Record Processing

After the analog input signal has been converted into binary numbers, several operations must occur before the information is stored or transmitted.

Some audio programs are stored or transmitted with emphasis. Emphasis is a means of reducing the noise floor in a signal. Pre-emphasis equalization boosts high frequencies prior to storage or transmission. At the output, a corresponding de-emphasis attenuates high frequencies.

Digital recording errors are minimized with error detection and correction, otherwise the quality of the audio recording would diminish. To prevent the destruction of a large area of consecutive data by a single large defect, interleaving is used. This measure scatters the data throughout the bit stream so the net effect of the error is distributed when the data is de-interleaved during playback. During playback, parity code data is added. This extra data is redundant data created from the original data to help detect and correct errors.

Multiplexing, used to form a serial bitstream, allows for the transmission of simultaneous data streams to be outputted. A multiplexer converts this parallel data to serial data, a single data stream. The multiple input circuit accepts parallel data words and outputs one bit at a time, serially, to form a continuous bit stream.

However, raw data must be formatted to facilitate its recording or transmission. Time-multiplexed data code is grouped into frames. To avoid ambiguity, each frame is given a synchronization code to delineate frames as they occur in the stream.

Addressing or timing data is ordered sequentially and distributed throughout the recording to distinguish between different sections. Error correction data is also put in the frame. Finally, identification codes are also added to the data stream so that information pertinent to the playback processing is shown. This code might show, for example, the sampling frequency, use of pre-emphasis, a table of contents, timing and track information, and copyright information.

Channel Codes

Digitized audio audio samples are not conveyed directly as binary code, rather, they are represented along with other information as a modulated channel code. Therefore, it is this modulated waveform that is interpreted during playback to recover the binary audio data, and thus reproduce the waveform. Modulation, by delineating the recorded logical states, facilitates the data reading, as well as achieving higher coding efficiency by conveying greater data throughput.

A channel code describes the manner in which information is modulated into a channel signal, stored, transmitted, or demodulated. Information bits are converted into channel bits.

Channel code should be self-clocking to permit synchronization at the receiver, minimize interfering low-frequency content, permit higher data rate transmission or higher density recording, bandlimit spectral energy, be immune to channel noise, and reveal invalid signal conditions.

The decoding clock at the receiver end, must be synchronized in frequency and phase with the clock in the transmitted signal. The synchronization word is marked in each frame f the binary bitstream. Without synchronization of any type, it is impossible to distinguish between individual channel bits. If no timing or decoding information is available, the implicit timing information in the channel bit is lost. Therefore, data must be recorded in such a way that the pulse timing is delineated. Usually, self-clocking codes are used for this purpose.

However, code efficiency must be diminished to achieve self-clocking since clocking increases the number of transitions that increase the channel bit rate. Higher- frequency signals produce robust clocking, but minimize the medium's storage capacity and can degraded over long cable transmissions.

T_{min} is the minimum distance between transitions and determines the highest frequency in the code, often it is too the highest frequency the storage medium can support. Density ratio (DR) is the ration of T_{min} to the length of a single bit period on input information data. T_{max} is the maximum distance between transitions needed to support clocking. A long T_{min} is desirable for bandwidth purposes. A short T_{max} is ideal for clocking purposes.

Time-axis variations, such as jitter, are characterized by phase variations in a signal and are observable as a frequency modulation in a stable waveform. Modulation code has to be tolerant in locating a transition in the code. This tolerance is called the window margin, phase margin, or jitter margin, T_W. It describes the minimum difference between code wavelengths, such that larger clock windows provide better jitter immunity.

Because storing binary code directly on a medium is inefficient, coding methods with low modulation codes can achieve greater density and higher code fidelity. For example, PCM is not suitable for recording. Binary recording stores a binary bit data stream, but the actual modulated recorded signal will be quite different. The channel data will be represented by transitions from one level to another, rather than the amplitude changes themselves. Therefore, we can consider that the events recorded in a digitally encoded signal are the instants in time at which the state of the signal changes.

Efficient coding formats must restrict the DC content in the coded waveform. The DC content is the fraction of time in which the signal is high during a string of 1's or 0's minus the fraction of time it is low. Ideally a DC content of 0 is desired, otherwise the timing synchronization is disrupted, the SNR is reduced by the DC baseline offset, and the servo system used to for tracking and reading in optical systems will show errors.

One way to prevent such errors is to have channel codes with minimal energy at low (avoid clocking and servo errors) and high frequencies (reduced bandwidth).

Simple Codes

Channel codes define the logical states of the input information by conveying them as 1's or 0's. It could be assumed that a direct relationship exists between a high amplitude and a logical 1, and a low amplitude and the logical 0, however, many other relationships are possible. For example,

Return to zero code (RZ): most basic code type. Sends a 1 for each pulse and sends nothing for a 0. The signal level always returns to zero at the end of each bit period.

Nonreturn to zero (NRZ): 1's and 0's are represented as high and low levels. The direction of the transition at the beginning or end of a bit represents a 1 or 0. Does not produce transition signals, thus clocking cannot be extracted and created DC content.

Nonreturn to zero inverted (NRZI): similar to NRZ, but 1's denote amplitude transitions and 0's show no transitions. Transitions occurs in the middle of the bit period.

Binary frequency modulation (FM): two transitions are made for a 1 and one for a 0. FM is a self-clocking code.

Phase encoding or phase modulation (PE or PM): 1's are coded with a negative-going transition and 0's as positive-going. Does not have DC content and is self-clocking.

Modified frequency modulation (MFM): 1's are coded with either positive or negative going transitions in the center of the bit period, for each 1. There is no transition for 0's, rather, a transition occurs at the end of each bit period only if a string of 0's occurs. This code type is self-clocking and may have DC content.

Group Classes

While simple codes code one information into one channel bit, group codes use code tables to convert groups of input words (*m* bits) into patterns of output words (*n* bits). The output pattern selection is based on their desired coding characteristics and uniqueness used in error detection.

Group codes can also be considered run-length limited codes (RLL). The run-length is the length of time between transitions. The transitions spacings can be any multiple of the period, and as a result, the distinction between data, clock transitions, and consecutive 1's is broken. RLL code use a set of rules to convert the information bitstream into a stream of channel bits by defining a particular relationship between them.

Code Applications

After encoding, the data is ready for storage or transmission. For example, in Compact Disc systems the modulation code results in pits. Each pit edge represents a binary 1 channel bit, and spaces represent 0's. The pit edge detection provides the clocking data.