Chapter 5  MATLAB EXERCISE

Experimenting with Quantization Error in MATLAB

Imagine this scenario: You record something at 16 bits, and you don’t set your microphone level high enough before recording. Consequently, your maximum recorded amplitude is only about 25% of the maximum possible. You then try to fix the problem by normalizing the audio clip so that it takes 100% of the dynamic range.

What is the average quantization error on the scale of the bit depth when you quantize at 16 bits (apart from the normalization)?

How is this error magnified by the normalization process?

To verify your conclusions, write a MATLAB program (or sequence of command-line statements) that simulates the above scenario. Do this two ways:

Experiment 1
- By evaluating and summing sine waves, generate a one second chord of the notes A, C, and E (starting at the A below middle C).
- Reduce the amplitude of the sine wave to 25% of the maximum possible. Call this yLowAmp.
- Quantize the chord to 16 bits. Call this y16.
- Get the error for each sample on the scale of 16 bits (in a vector the same length as the vector of sample values). Call this errQuant16.
- Graph the error vector from errQuant16. Display just the first 500 values.
- Compute the average magnitude error on the scale of 16 bits. Compare it to what you predicted the error would be.
- Normalize the quantized sound to 100%.
- Compute the error vector again. Call this errQuant16Norm.
- Graph the error vector.
- Compute the average magnitude error of errQuant16Norm. Call this avgErr16.

Experiment 2
- Use the same chord that you generated in Experiment 1.
- Reduce the amplitude of the sine wave to 25%.
- Quantize the chord to 24 bits.
- Compute the error on the scale of 24 bits (in a vector the same length as the vector of sample values). Call this errQuant24.
• Graph the error vector.
• Compute the average magnitude error on the scale of 24 bits. Compare it to what you would predict the error to be.
• Normalize the sound to 100%.
• Compute the error vector again. Call this errQuant24Norm.
• Graph the error vector errQuant24Norm.
• Reduce the bit depth from 24 to 16 bits.
• Compute the error vector again. Call this err24to16.
• Graph the error vector err24to16.
• Compute the average magnitude error from err24to16.
• Get the sum of errQuant24Norm and err24to16. Call this totalError24Downsampled.
• Get the average magnitude error of totalError24Downsampled. Call this avgErr24.
• Compare avgErr16 to avgErr24 and draw conclusions.

Experiment 3
• Use the chord that you generated in Experiment 1, at 25% of maximum amplitude. Call this chord yLowAmp.
• Quantize the chord to 16 bits. Call this y16.
• Convert the quantized values to 32 bit floating point. Call this y32.
• Normalize to 100% amplitude. Call this norm32.
• Reduce to 16 bits again. Call this norm32Requantized16.
• Compute an error vector representing the difference between the unquantized, unreduced, original chord (on a scale of 16 bits) and the quantized, normalized, requantized norm32Requantized16.
• Graph the error vector norm32Requantized16.
• Compute the average magnitude error from norm32Requantized16 on a scale of 16 bits.
• Compare avgErr16 to norm32Requantized16 and draw conclusions.