Assignment 2: Speech Compression and

Quantization

EQ2320 Speech Signal Processing

2016-01-23

# 2 The Uniform Scalar Quantizer

### TASKS/QUESTIONS

1. Implement a uniform scalar encoder with a function header

idx = sq\_enc(in, n\_bits, xmax, m)

where in is a vector with the original speech samples, n\_bits is the number of bits available to quantize one sample in the quantizer (i.e., the rate), xmax and m define the range of the quantizer from m-xmax to m+xmax, so that the width of each quantization interval is ∆=2×xmax/*L*, where *L* is the number of quantization intervals and corresponding reconstruction values. m defines the mean (or offset) of the quantizer reconstruction levels. Setting m = 0 defines a “midrise” quantizer, and m = ∆/2 gives a “midtread” quantizer (see the course book sec. 7.2). The function should return the index of the chosen quantization level.

Implement the corresponding decoder function:

outq = sq dec(idx, n bits, xmax, m)

where outq is the corresponding reconstruction value for idx.

The USQ is a highly structured quantizer. The encoder can be implemented essentially by only a scalar division (no multiplications, comparisons or loops are needed), making the computational complexity independent of the bitrate. Make sure your encoder has a computational complexity independent of the bitrate!

1. Run the encoder and decoder on a ramp signal x=-6:0.01:6. Use a 2-bit quantizer with *xmax*=4. Plot the quantizer output as a function of the input. Make sure the output levels are exactly where you expect them to be. Use a quantizer mean m=0. Do a similar plot with m=1.5.

(4 pts)

# 3 Parametric Coding of Speech

## 3.1 Quantizing the Gain

### TASKS/QUESTIONS

1. Provide a plot of the histogram of the gain parameter. Indicate in the plot the range of the quantizer, i.e., mark the outer boundaries m±xmax (also mark m). Note that the pdf has a non-zero mean. (1 pts)
2. Run the vocoder with a uniform scalar gain quantizer according to the design above. Find the rate at which you cannot hear the quantization distortion. (2 pts)
3. Take the logarithm of the gain parameter prior to quantization (does not matter which base). Provide a plot of the histogram of the gain parameter in the log-domain. Indicate the range of this quantizer as above. (1 pts)
4. Run the vocoder with a uniform scalar log-gain quantizer according to the design above. Find the rate at which you cannot hear the quantization distortion. Make sure to modify the decoder accordingly (apply the exp function to the quantized log-gain). (2 pts)
5. Which is better: gain quantization in linear or log domain? (2 pts)

# 3.2 Quantizing the Pitch and Voiced/Unvoiced Decision

Come up with an efficient way to encode the pitch and voiced/unvoiced decision! (2 pts)

# 3.3 Quantizing the LP parameters

For the quantization of LP parameters, we will use a vector quantizer (VQ). You do not need to optimize (train) the VQs; that has been done for you, and the codebooks can be found in the MATLAB variables lsfCB1 and lsfCB2 in the file assignment2.mat. The codebooks constitute a multistage VQ. lsfCB1 is a 10 bit VQ optimized on 10 dimensional LSF vectors. lsfCB2 is a 10 bit second stage residual codebook. What you need to do is to program an encoding function and a corresponding decoding function for a multistage VQ. A suitable calling syntax for these functions can be

codeA=encodefilter(A, cb1,cb2) and

Aq= decodefilter(codeA, cb1,cb2).

Here A is a matrix with filter coefficients stored row-wise, and codeA is a two-column matrix with the corresponding code indices, stored row-wise.

### HINTS/REMARKS

1. To convert between polynomial (a-) coefficients and LSFs see poly2lsf and lsf2poly.
2. poly2lsf requires the polynomial coefficients to correspond to a minimum phase whitening filter. This is guaranteed by the autocorrelation LP analysis. lsf2poly requires that the LSFs correspond to a minimum phase whitening filter. The multistage VQ can output LSFs that do not satisfy this. As a precaution simply sort the LSFs prior to calling lsf2poly. Also check so they are between 0 and π. (4 pts)

# 3.4 Optimizing the Bit Allocation

### TASKS/QUESTIONS

1. Evaluate the SNR for your design above. (2 pts)
2. What number of bits do you suggest for the pitch? For the gain? For the voiced/unvoiced decision? (3 pts)
3. What is the rate in bits per sample of your vocoder with the bit allocation suggested above? In bits per second? (2 pts)
4. Does it make sense to evaluate SNR here? Why or why not? (2 pts)

# 4 Speech Waveform Quantization

## 4.1 Uniform Scalar Quantization of Speech

### TASKS/QUESTIONS

1. Evaluate the optimal *k* for *R* = 3. (2 pts)
2. Run the quantizer at rates 16, 15, 14, . . ., 2, 1, and evaluate the SNR for each rate. Provide a plot of the SNR as a function of rate. (2 pts)
3. Provide a graph of the theoretical SNR in the same plot as the experimental SNR plot. For the theoretical SNR, assume that the number of quantization levels is high, and that overload is negligible. (2 pts)
4. At what rate can you not tell the difference between the original and the quantized signal? (1 pts)
5. Listen to the quantization error signal, *q(n)*! How would you characterize *q(n)* for a system operating at rate *R* = 1. Increase the rate (up to *R* ≈ 12) and describe how the character of *q(n)* changes. (1 pts)
6. OPTIONAL: Is it advantageous to have a reconstruction level in the origin for low rates? Compare (by listening) midrise and midtread quantizers at low bit rates. (2 bonus pts)

# 5 Adaptive Open-Loop DPCM

### TASKS/QUESTIONS

1. You are not given many guidelines here. Give it your best shot and make sure you can motivate your choice of for example

* analysis frame length,
* update length (to keep things simple make analysis and update lengths equal, i.e., no overlapping analysis frames),
* window function (for the analysis of certain parameters),
* number of bits to quantize the gain,
* number of bits to quantize the residual,

Use the VQ as before to quantize the LP parameters (thus, you need not decide prediction order!).

Design the PCM quantizer for the prediction error *d(n)* according to *xmax = kσd*. Optimize *k* for *R* = 3 (*R* meaning the rate of the residual quantizer), by experimenting, so that it sounds good, i.e., do not optimize SNR theoretically.

(4 pts)

1. Run your system at R = 3. How would you characterize the reconstructed speech? What does the quantization error sound like? (1 pts)
2. What shape does the quantization error spectrum have? Plot a DFT based spectrum of the error for a voiced frame. What does theory say? (2 pts)
3. Measure the SNR of your system. Compare with the SNR of PCM at the same rate. Comments? (2 pts)
4. What is the total rate of your coder in bits per sample? In bits per second? (2 pts)
5. Is it better to use the quantized LP coefficients in the encoder filter than to use the unquantized LP coefficients? (2 pts)