HOMEWORK ASSIGNMENT #5

Due Thurs. Dec. 2, 1999 (in class)

In this homework you will implement a MPEG Layer 1 encoding/decoding system for mono signals in MATLAB. (Actually, about 75% of the work has already been done for you!) The following constants are assumed:

F_s	44100 Hz	sampling frequency
N_s	32	number of subbands
M_h	513	subband prototype filter length
M_{fft}	512	FFT length
L_{fr}	12	number of $N_s \times 1$ subband blocks per frame
B_{hdr}	32	bits reserved for frame header
B_{alloc}	4	bits used for coding bit allocation info

Encoding: The following is a summary of the steps involved to encode each frame of audio data.

1. Calculate L_{fr} blocks of N_s subband samples. I suggest transforming the entire input record at once using the supplied routine `pqf_anal.m`, e.g.

   `S = pqf_anal(x,h_mpeg(1:512),32),`

   though you may use your own PQF code from HW#4 if you are confident that it works.

2. Calculate M_{fft} FFT magnitudes of Hann-windowed input data. *Note: The FFT window must be centered on the samples constituting the current frame of input data.* To figure out which samples these are, realize that the first frame is built from subband blocks 0 to $L_{\text{fr}}-1$ and thus the following input samples:

subband block #	samples used to calculate subband block
$m = 0$	$\{x(0), x(-1), \ldots, x(-M_h+1)\}$
$m = 1$	$\{x(N_s), x(N_s-1), \ldots, x(N_s-M_h+1)\}$
$\vdots$	$\vdots$
$m = L_{\text{fr}}-1$	$\{x((L_{\text{fr}}-1)N_s), x((L_{\text{fr}}-1)N_s-1), \ldots, x((L_{\text{fr}}-1)N_s-M_h+1)\}$

   The second frame is built from:

subband block #	samples used to calculate subband block
$m = L_{\text{fr}}$	$\{x(L_{\text{fr}}N_s), x(L_{\text{fr}}N_s-1), \ldots, x(L_{\text{fr}}N_s-M_h+1)\}$
$m = L_{\text{fr}}+1$	$\{x((L_{\text{fr}}+1)N_s), x((L_{\text{fr}}+1)N_s-1), \ldots, x((L_{\text{fr}}+1)N_s-M_h+1)\}$
$\vdots$	$\vdots$
$m = 2L_{\text{fr}}-1$	$\{x((2L_{\text{fr}}-1)N_s), x((2L_{\text{fr}}-1)N_s-1), \ldots, x((2L_{\text{fr}}-1)N_s-M_h+1)\}$

   From above, you should be able to figure out which input samples to window for FFT analysis.

3. Scale the subband data: A single scalefactor from Table 1 is used to divide the L_{fr} subband samples in each branch. The scalefactor is the *smallest* one making the magnitude of all L_{fr} scaled samples less than one. The function `mpeg_scale.m` loads the values in Table 1.

   Note that 6-bit numbers representing scalefactor table indices (rather than scalefactors themselves) will be transmitted along with each frame of quantized subband data.

4. Estimate the signal level (L_{sb}) and minimum masking threshold (L_{mask}) in each subband. The function `calc_mask.m` accomplishes this for you. (Type "`help calc_mask`" for instructions.) From these values, the signal-to-mask ratio (SMR) in the i^{th} subband can be computed:

$$\text{SMR}(i) \;=\; \text{L}_{\text{sb}}(i) - \text{L}_{\text{mask}}(i).$$

5. Allocate Bits: An iterative process is used to allocate bits for quantization of scaled subband samples. First, a few definitions...

   - The number of bits *available* for coding each frame is

$$B_{\text{avail}} \;=\; 1000\,R\,\frac{\text{bits}}{\text{sec}} \times \frac{1}{F_s}\frac{\text{sec}}{\text{samp}} \times L_{\text{fr}} N_s \frac{\text{samp}}{\text{frame}},$$

     where R is MPEG stream rate in kilobits-per-second (kbps).

   - The total number of bits *used* in coding a frame is:

$$B_{\text{tot}} \;=\; B_{\text{hdr}} + \sum_{i=0}^{N-1} \Big(L_{\text{fr}} B_{\text{data}}(i) + B_{\text{alloc}} + B_{\text{scf}}(i) \Big)$$

     where $B_{\text{hdr}}(= 32)$ is the number of bits reserved for the frame header, $B_{\text{data}}(i)$ is the number of bits per subband data sample in the i^{th} branch, with allowed range

$$B_{\text{data}}(i) \in \{0, 2, 3, 4, \ldots, 15\},$$

     $B_{\text{alloc}}(= 4)$ is the number of bits required for coding these bit allocations, and B_{scf} is the number of bits used to code the scale factor for the i^{th} branch:

$$B_{\text{scf}}(i) = \begin{cases} 6 & B_{\text{data}}(i) > 0, \\ 0 & B_{\text{data}}(i) = 0. \end{cases}$$

     Note that $B_{\text{data}}(i)$ is not allowed to take on values $= 1$ or > 15, and that the scalefactor subband i is not coded when the data for subband i is not coded.

   - Psychoacoustic research suggests that quantization noise is not perceived when mask-to-noise ratio (MNR) is greater than one (i.e., MNR is positive on a dB scale). MNR is defined as

$$\text{MNR}(i) \;=\; \text{SNR}_{\text{q}}\big(B_{\text{data}}(i)\big) - \text{SMR}(i)$$

     where $\text{SNR}_{\text{q}}\big(B\big)$ is the ratio of signal-power to quantization-noise-power for a B-bit quantizer. The MPEG standard specifies $\text{SNR}_{\text{q}}\big(B\big)$ as in Table 2 (which can be loaded via via the function `mpeg_SNR.m`).

   The MPEG-suggested bit allocation procedure is in accordance with the old saying: "the squeaky wheel gets the grease."

```
initialize B_data(i) = 0 and B_scf(i) = 0 for all i,
repeat until no more bits can be allocated:
   calculate {MNR(i)} using current {B_data(i)}.
   find i* = arg min_{i∈{i:B_data(i)<15}} MNR(i),
   if B_data(i*) = 0,
     set B_data(i*) = 2 and B_scf(i*) = 6 as long as it keeps B_tot ≤ B_avail,
   else
     set B_data(i*) = B_data(i*) + 1 as long as it keeps B_tot ≤ B_avail,
   end.
end.
```

6. Quantize the subband samples. The L_{fr} scaled subband samples in the i^{th} branch are quantized using a midtread uniform quantizer with $2^{B_{\text{data}}(i)} - 1$ levels. Since the scaled data has maximum input magnitude of 1, the following rule can be used to implement the quantizer:

$$\tilde{s}_i(m) \;=\; \left\lfloor \frac{1}{2}\Big(\big(2^{B_{\text{data}}(i)} - 1\big)s_i(m) - 1\Big) \right\rfloor .$$

   Note that $\{\tilde{s}_i(m)\}$ take on integer values in the range $-2^{B_{\text{data}}(i)-1} \le \tilde{s}_i(m) \le 2^{B_{\text{data}}(i)-1} - 1$.

   The following pieces of data are then included with each MPEG frame:

   (a) N_s bit allocations: $\{B_{\text{data}}(i)\}$,

   (b) up to N_s scalefactor indices.

   (c) up to $L_{\text{fr}} \times N_s$ quantized subband data values: $\{\tilde{s}_i(m)\}$,

<u>*Decoding:*</u> Now follows a summary of the steps involved in decoding an MPEG frame.

1. Convert the integer-valued quantized data $\{\tilde{s}_i(m)\}$ to floating-point values using the bit allocation info $\{B_{\text{data}}(i)\}$ as follows:

$$\hat{s}_i(m) \;=\; \frac{2\big(\tilde{s}_i(m) + 1\big)}{2^{B_{\text{data}}(i)} - 1}.$$

2. Un-scale the resulting subband samples using the transmitted indices and Table 1 (or `mpeg_scale.m`).

3. Calculate the time-domain output. For this, I suggest first unquantizing/unscaling all frames of subband data using the two steps above, and then reconstructing the entire output record using the supplied routine `pqf_synth.m`, e.g.,

   `x = pqf_synth(Su,h_mpeg(1:512))`,

   though you may use your own PQF code from HW#4 if you are confident that it works.

<u>***Your tasks:***</u>

1. Implement the MPEG Layer 1 coding/decoding scheme outlined above. Include the the ability to display:

   (a) Graphs comparing $\mathrm{L}_{\text{sb}}(i) - \mathrm{SNR}_{\text{q}}(B_{\text{data}}(i))$ to $\mathrm{L}_{\text{mask}}(i)$ and showing $\mathrm{MNR}(i)$ (as in Fig. 1) for each frame. (*Hint:* try the `bar` command with the "grouped" option.)

(b) Bit allocations $B_{\text{data}}(1)$, $B_{\text{data}}(8)$, and $B_{\text{data}}(28)$ versus frame number (as in Fig. 2),

(c) Input delayed by $M_h - 1$, reconstructed output, and their difference (as in Fig. 3).

2. Process the file `sco_29.wav` at $R = 96$ kbps. Of the displays listed above, plot (a) for the second frame, then (b) and (c).

3. Listen to the MPEG versions of the file `sco_160.wav` coded using $R = 192$, 160, 128, and 96 kbps and compare each to the uncoded version. At what point can you hear a difference?

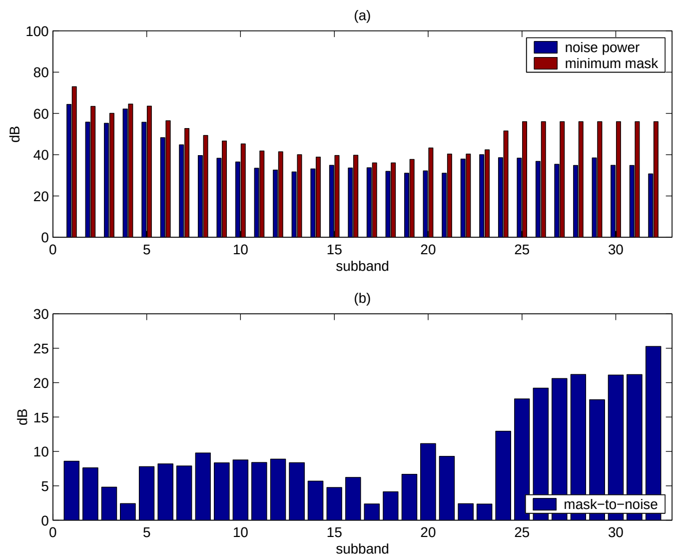

Figure 1: Example of (a) $\mathrm{L_{sb}}(i) - \mathrm{SNR_q}(B_{\text{data}}(i))$ and $\mathrm{L_{mask}}(i)$, and (b) $\mathrm{MNR}(i)$ for subband index i.

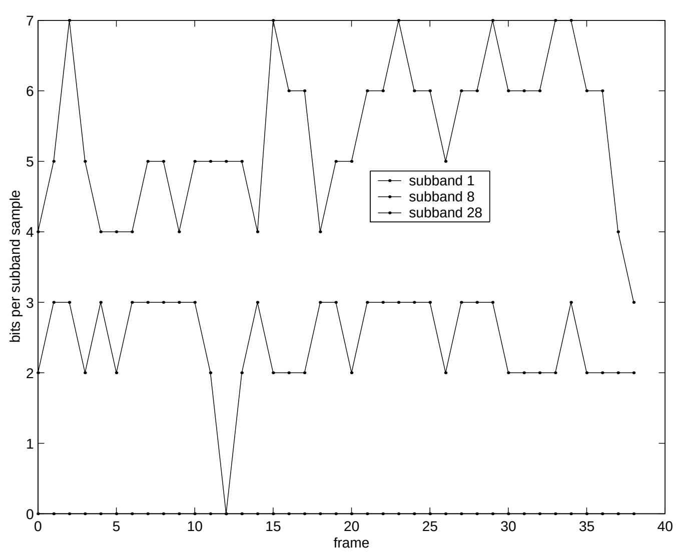

Figure 2: Example bit allocations for three subbands.

2.00000000000000
1.58740105196820
1.25992104989487
1.00000000000000
0.79370052598410
0.62996052494744
0.50000000000000
0.39685026299205
0.31498026247372
0.25000000000000
0.19842513149602
0.15749013123686
0.12500000000000
0.09921256574801
0.07874506561843
0.06250000000000
0.04960628287401
0.03937253280921
0.03125000000000
0.02480314143700
0.01968626640461
0.01562500000000
0.01240157071850
0.00984313320230
0.00781250000000
0.00620078535925
0.00492156660115
0.00390625000000
0.00310039267963
0.00246078330058
0.00195312500000
0.00155019633981
0.00123039165029
0.00097656250000
0.00077509816991
0.00061519582514
0.00048828125000
0.00038754908495
0.00030759791257
0.00024414062500
0.00019377454248
0.00015379895629
0.00012207031250
0.00009688727124
0.00007689947814
0.00006103515625
0.00004844363562
0.00003844973907
0.00003051757813
0.00002422181781
0.00001922486954
0.00001525878906
0.00001211090890
0.00000961243477
0.00000762939453
0.00000605545445
0.00000480621738
0.00000381469727
0.00000302772723
0.00000240310869
0.00000190734863
0.00000151386361
0.00000120155435

Table 1: MPEG scalefactors.

bits	SNR_q
0	0
1	—
2	7
3	16
4	25.28
5	31.59
6	37.75
7	43.84
8	49.89
9	55.93
10	61.96
11	67.98
12	74.01
13	80.03
14	86.05
15	92.01

Table 2: MPEG-specified quantizer SNR versus bits/sample.

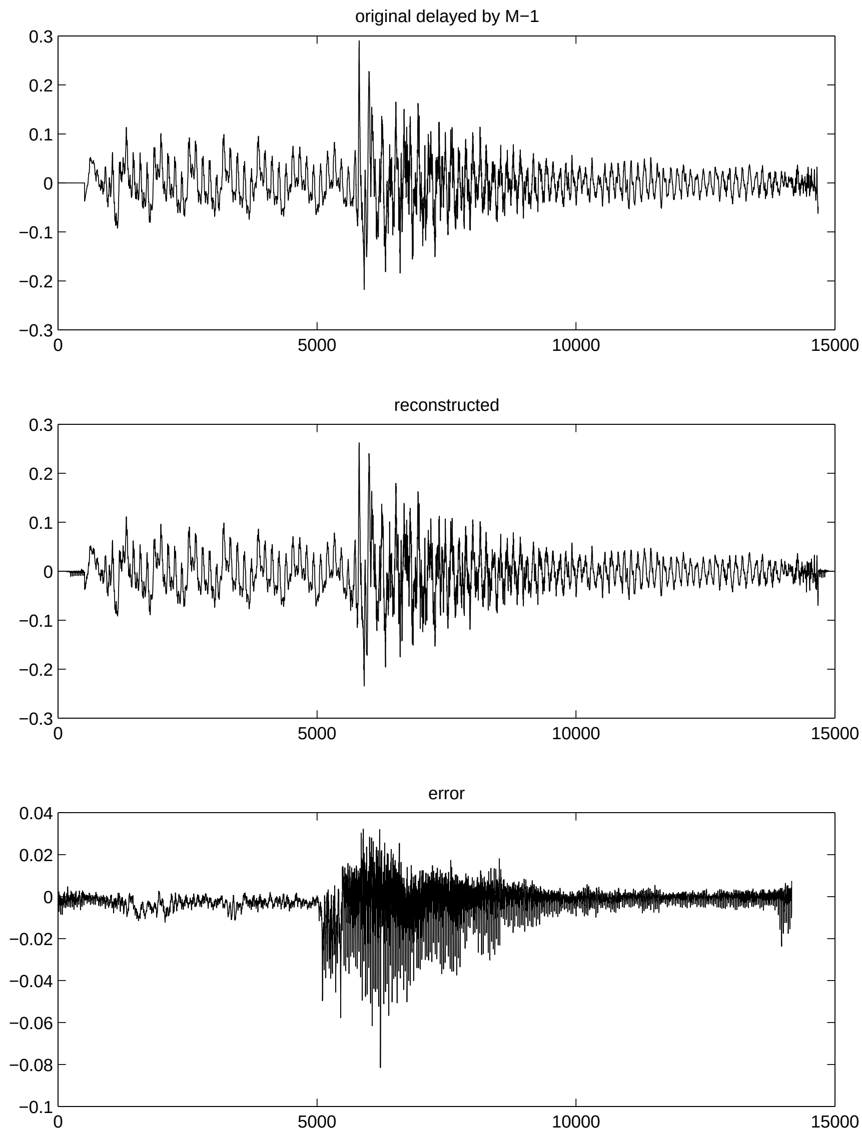

Figure 3: Example of input delayed by $M_h - 1$, reconstructed output, and their difference.