HOMEWORK ASSIGNMENT #3

Due Tues. Oct. 19, 1999 (in class)

1. *Practical Bit Allocation:*
With regards to bit allocation for transform coder outputs, we proved that the constrained optimization

$$\min_{\{R_k\}} \sum_{k=0}^{N-1} \sigma_{y_k}^2 2^{-2R_k} \quad \text{s.t.} \quad R = \frac{1}{N}\sum_{k=0}^{N-1} R_k$$

(where $\sigma_{y_k}^2$ is the variance of the k^{th} transform output, R_k is the bit rate allocated for transmission of this output, and R is the average bit rate over all outputs) led to the optimal bit allocation rule

$$R_\ell^{\text{opt}} \;=\; R + \frac{1}{2}\log_2\left(\frac{\sigma_{y_\ell}^2}{\left(\prod_{k=0}^{N-1}\sigma_{y_k}^2\right)^{1/N}}\right).$$

Recognizing that the equation above may yield impractical (e.g, negative or non-integer) values for R_ℓ^{opt}, we discussed a practical bit allocation strategy where, one by one, R_ℓ are fixed at practical values and the remaining $\{R_k\}$ are re-optimized. Specifically, consider the following algorithm:

$\mathcal{K}_a = \{\}$, `%set of allocated output indices`
$\mathcal{K}_u = \{0, 1, 2, \ldots, N-1\}$, `%set of unallocated output indices`
`while` $\mathcal{K}_u \neq \{\}$,
  `calculate quasi-optimal` $\{R_k^{\text{qua}} : k \in \mathcal{K}_u\}$,
  `set` $k_* = \arg\min_{k\in\mathcal{K}_u} R_k^{\text{qua}}$,
  `round` $R_{k_*}^{\text{qua}}$ `to nearest non-negative integer, saving as practical` R_{k_*}
  `remove` k_* `from` $\mathcal{K}_u$ `and add` k_* `to` $\mathcal{K}_a$,
`end.`

The step "`calculate quasi-optimal` $\{R_k^{\text{qua}} : k \in \mathcal{K}_u\}$" requires solving the following constrained optimization problem.

$$\min_{\{R_k : k\in\mathcal{K}_u\}} \sum_{k\in\mathcal{K}_u} \sigma_{y_k}^2 2^{-2R_k} \quad \text{s.t.} \quad R = \frac{1}{N}\sum_{k=0}^{N-1} R_k.$$

Prove that the solution is given by

$$R_\ell^{\text{qua}} \;=\; \frac{NR - \sum_{k\in\mathcal{K}_a} R_k}{\text{size}(\mathcal{K}_u)} \;+\; \frac{1}{2}\log_2\left(\frac{\sigma_{y_\ell}^2}{\left(\prod_{k\in\mathcal{K}_u}\sigma_{y_k}^2\right)^{1/\,\text{size}(\mathcal{K}_u)}}\right) \qquad \text{for} \quad \ell \in \mathcal{K}_u.$$

(Don't be intimidated—this is a lot easier than it might look!)

2. *Adaptive Transform Coding:*
   In this problem you will implement the adaptive transform coder in Fig. 1.

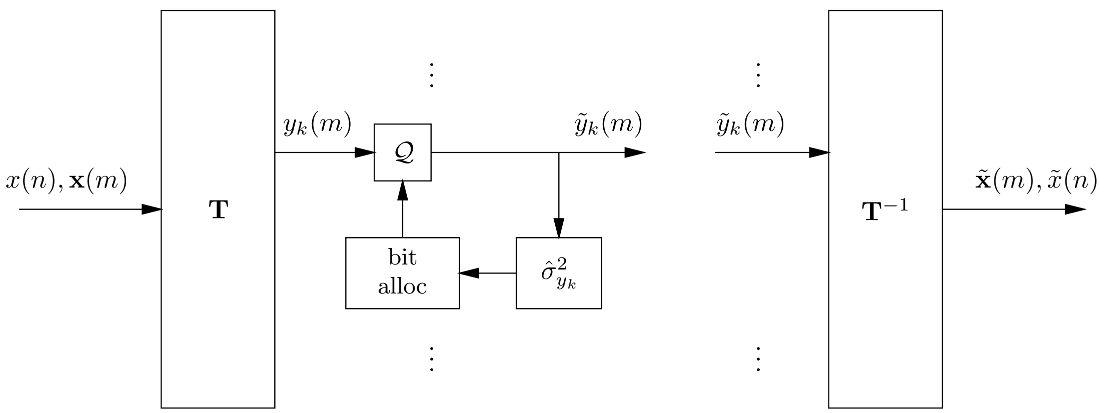

Figure 1: An Adaptive Transform Coder

The input $x(n)$ will be an "autoregressive" (AR) process generated by filtering zero-mean white Gaussian noise $v(n)$ ($\sigma_v^2 = 1$) through linear system $H(z) = B(z)/A(z) = 1/(1 - 0.8z^{-1})$.

The quantizers will be uniform with $L_k(m) = 2^{R_k(m)}$ levels, where $R_k(m)$ is calculated using the method of Problem 1, but with $\sigma^2_{y_k}$ replaced by the backward variance estimate $\hat{\sigma}^2_{y_k}(m)$:

$$\hat{\sigma}^2_{y_k}(m) \;=\; (1-\alpha)\tilde{y}_k^2(m-1) + \alpha\, \hat{\sigma}^2_{y_k}(m-1), \qquad k = 0, \ldots, N-1.$$

Assume $\alpha = 0.95$, transform size $N = 16$, average bit rate $R = 4$ bits/sample, and quantizer design factor $\phi_{y_k} = 3$. (You should not be generating any random data until part (e) below!)

(a) Plot the input power spectrum $S_x(e^{j\omega})$. (Hint: Realize $x(n) = \sum_{i=0}^{\infty} h_i v(n-i)$, where $\{h_0, h_1, \ldots\}$ is the impulse response of $H(z)$. Use the Matlab command `impz` to find a truncated approximation of $\{h_i\}$.)

(b) What is the asymptotic reconstruction error variance $\sigma_r^2\big|_{\text{TC},N} = \text{var}(\tilde{x}(n) - x(n))$ for the optimal infinite-dimensional transform and optimal bit allocation?

(c) What is the reconstruction error variance $\sigma_r^2\big|_{\text{TC},N}$ when using the optimal $N \times N$ transform and optimal bit allocation? (Hint: Use `toeplitz` to construct the autocorrelation matrix and `eig` to compute the eigendecomposition.)

(d) For transform $\mathbf{T}$, prove that $(\sigma^2_{y_0}, \ldots, \sigma^2_{y_{N-1}})^t = \text{diag}(\mathbf{T}\mathbf{R}_x\mathbf{T}^t)$, where diag$(\cdot)$ extracts the main diagonal of a matrix.
   What is the reconstruction error variance $\sigma_r^2\big|_{\text{TC},N}$ when using the DCT and optimal bit allocation? (Hint: Construct $\mathbf{T}$ with `dctmtx`.)

(e) For $M = 1000$, generate an MN-length realization of $x(n)$ and implement the adaptive TC scheme of Fig. 1 using a DCT. (Hint: use `filter` to create $x(n)$, initialize $\hat{\sigma}^2_{y_k}(0) = \sigma_x^2 \;\; \forall k$, and use `[R_srt,indx]=sort(R_opt)` in the bit allocation procedure.)
   One on plot, display the optimal bit allocations for the two branches $k = 0$ and $k = N-1$ together with the practical bit allocations for the same branches (see Fig. 2 for an example).
   What is the mean-squared reconstruction error $\mathcal{E}_{\text{TC}} = \frac{1}{MN} \sum_{n=0}^{MN} |\tilde{x}(n) - x(n)|^2$?

(f) For the same input sequence $x(n)$, compute $\mathcal{E}_{\text{PCM}}$ for the PCM system in Fig. 3. Assume uniform quantization with $L = 2^R$ levels and quantizer design factor $\phi_x = 3$. (See previous homework solutions for efficient ways of doing this.)

(g) Discuss the differences between the various values of $\sigma_r^2\big|_{\text{TC},N}$ and $\mathcal{E}$ computed in parts (b)-(f).

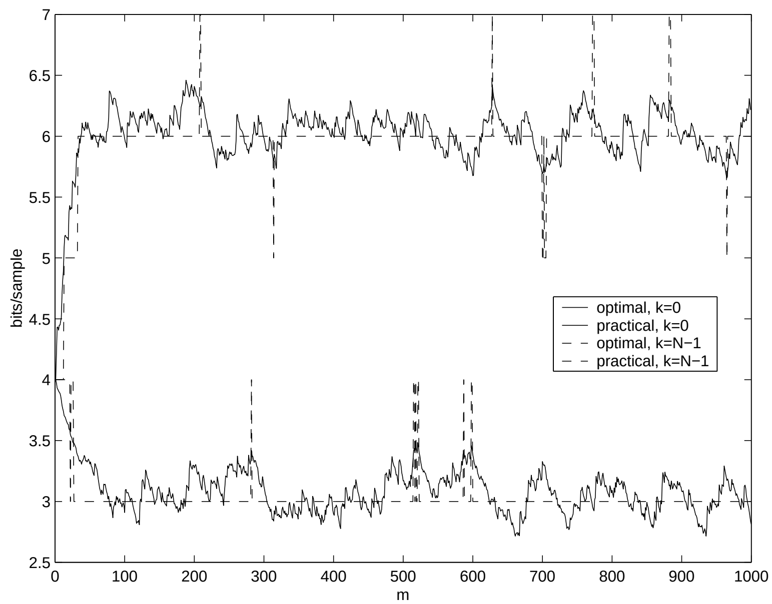

Figure 2: Optimal and practical bit allocations for output branches $k = 0$ and $k = N-1$ versus input block m.

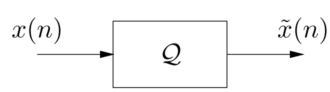

Figure 3: PCM system.

3. <u>*Suboptimal Transforms:*</u>
Now we'll compare the performance of various transforms as a function of transform dimension.

(a) Consider the AR input process generated by passing zero-mean white Gaussian noise through the filter

$$H(z) \;=\; \frac{1}{A(z)} \;=\; \frac{1}{1 - 0.8z^{-1} + 0.4z^{-2}}.$$

Assuming optimal bit allocation, plot theoretical TC gain over PCM for transform dimensions $N = 1, 2, 4, 8, 16, 32, 64$ and the following transforms: KLT, DCT, real-DFT, DHT. Superimpose asymptotic ($N \to \infty$) TC gain in the form of a dashed line. See Fig. 4 for an example. (Hint: create appropriate matrix $\mathbf{T}$, then use 2(d).)

(b) Repeat for $A(z) = 1 + 0.7z^{-1} + 0.2z^{-2}$.

(c) Discuss all relevant features of the two plots.

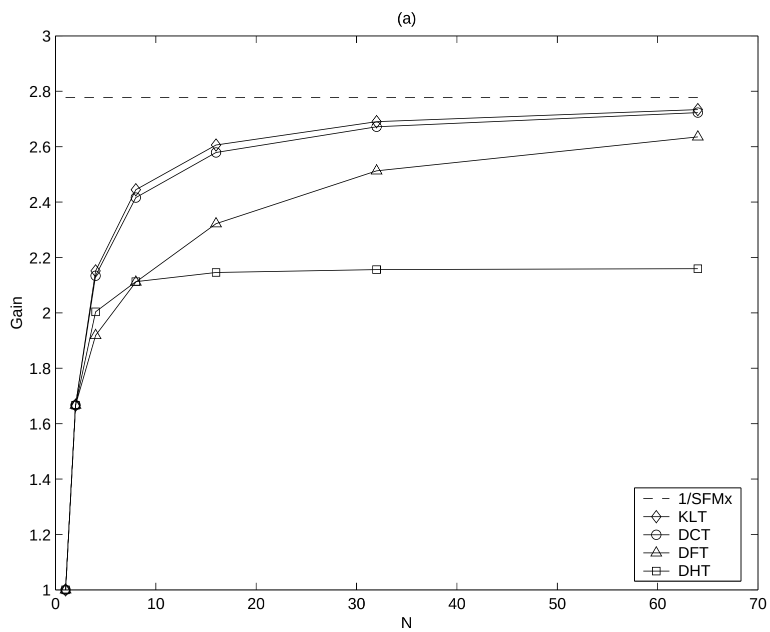

Figure 4: Example of TC-gain-over-PCM versus transform dimension N for various transforms and a lowpass source.