SOLUTIONS

E 91 **Digital Signalbehandling**, 2E1340

Final Examination 2003–08–29, 14.00–19.00

1. a) Use Theorem 2.6 in the Complementary Reading compendium directly, or use the inverse transform to find the polyphase components in the time-domain.

$$\begin{aligned} h(n) &= \tfrac{1}{8}\left(\tfrac{1}{2}\right)^{n} u(n) + \tfrac{1}{4}\left(\tfrac{1}{2}\right)^{n-1} u(n-1) + \tfrac{1}{8}\left(\tfrac{1}{2}\right)^{n-2} u(n-2) \\ &= \tfrac{9}{8}\left(\tfrac{1}{2}\right)^{n} u(n) - \delta(n) - \tfrac{1}{4}\delta(n-1) \end{aligned}$$

and

$$\begin{aligned} p_0(n) &= h(2n+0) = \tfrac{9}{8}\left(\tfrac{1}{2}\right)^{2n} u(2n) - \delta(2n) && = \tfrac{9}{8}\left(\tfrac{1}{4}\right)^{n} u(n) - \delta(n) \\ p_1(n) &= h(2n+1) = \tfrac{9}{8}\left(\tfrac{1}{2}\right)^{2n+1} u(2n+1) - \tfrac{1}{4}\delta(2n) && = \tfrac{9}{16}\left(\tfrac{1}{4}\right)^{n} u(n) - \tfrac{1}{4}\delta(n) \end{aligned}$$

$$\begin{aligned} P_0(z) &= \frac{\frac{9}{8}}{1-\frac{1}{4}z^{-1}} - 1 = \frac{0.125 + 0.25z^{-1}}{1-0.25z^{-1}} \\ P_1(z) &= \frac{\frac{9}{16}}{1-\frac{1}{4}z^{-1}} - \frac{1}{4} = \frac{0.3125 + 0.0625z^{-1}}{1-0.25z^{-1}} \end{aligned}$$

b) Write the filters as difference equations and compare the number of operations.
Direct implementation:
Let

$$v(n) = \begin{cases} x(n/2), & n = 0, \pm 2, \pm 4, \ldots \\ 0, & \text{otherwise} \end{cases},$$

then

$$y(n) = 0.125 \cdot v(n) + 0.25 \cdot v(n-1) + 0.125 \cdot v(n-2) + 0.25 \cdot y(n-1).$$

Counting the number of operations gives: $N_{\text{mul}} = 4, \quad N_{\text{add}} = 3$ for each output value.
Polyphase implementation:

$$\begin{aligned} y(2n) &= 0.125 \cdot x(n) + 0.25 \cdot x(n-1) + 0.25 \cdot y(2n-2)) \\ y(2n+1) &= 0.3125 \cdot x(n) + 0.0625 \cdot x(n-1) + 0.25 \cdot y(2n-1) \end{aligned}$$

Thus, both even and odd output samples require $N_{\text{mul}} = 3, \quad N_{\text{add}} = 2$. In this case, the polyphase implementation saves 1 multiplication and 1 addition.

2. a) In the left figure, the quantization errors are uniformly distributed over $[-2^{-9}, 2^{-9}] \approx [-0.002, 0.002]$ which corresponds well with the standard approximations for round-off to fixed point numbers with 8 bits + 1 sign bit resolution. In the right figure, most of the errors are also uniformly distributed over the same range, but a few errors are larger, which indicates overflow. This lab group must have set the signal amplitude slightly larger than what could be handled by the A/D converter.

b) The Upper implementation is better since it preserves as large part of the original frequency content of the signal as possible. In the lower figure, the signal is decimated first, which means that all frequencies above 1/D are destroyed by aliasing (or removed by $H_D(f)$).

c) In the middle figure, the two peaks of the original signal have been smeared out. This is a typical bias effect of using a too short time window. In the lower figure, there are many random variations up and down from the true spectrum, i.e. the variance is large (the estimate was averaged over too few data segments).

3. The program can be described as a filter $h(n)$ that calculates the average of the last 5 input values followed by a downsampling by 5, see figure 1. The transfer function of the filter is

$$H(z) = \frac{1}{5}\sum_{k=0}^{4} z^{-k} = \frac{1-z^{-5}}{5(1-z^{-1})} = z^{-2}\frac{z^{2.5}-z^{-2.5}}{5(z^{0.5}-z^{-0.5})}$$

Which gives $H(f) = \frac{\sin(5\pi f)}{5\sin(\pi f)} e^{-j4\pi f}$. Combined with the downsampling, this leads to

$$Y(f) = \frac{1}{5}\sum_{k=0}^{4} Z(\frac{f-k}{5}) = \frac{1}{25}\sum_{k=0}^{4} X(\frac{f-k}{5})\frac{\sin(\pi(f-k))}{\sin(\frac{\pi(f-k)}{5})} e^{-j4\pi(f-k)/5}$$

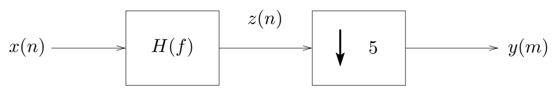


Figure 1:

4. The MATLAB code implements the filters shown in figure 2. The first implementation results in one round-off term that enters the filter at the input and two round-off terms directly at the output. The variance of the round-off noise at the output is thus $\sigma^2_{\varepsilon,1} = (\sum h^2(n) + 2)\sigma^2_e$.

$$\begin{gathered} h(n) = 0.3 \cdot 0.6^n \cdot u(n) + 0.1 \cdot 0.6^{n-1} u(n-1) = \tfrac{7}{15} \cdot 0.6^n u(n) - \tfrac{1}{6} \\ \sigma^2_{\varepsilon,1} = (\sum h^2(n) + 2)\sigma^2_e \approx 2.29\sigma^2_e \end{gathered}$$

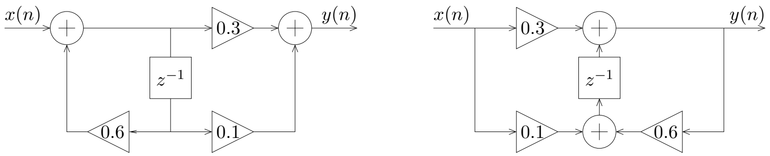
The second implementation gives three noise terms, all entering at the second stage of the filter. The variance of the round-off noise is $\sigma^2_{\varepsilon,2} = 3\sum h_2^2(n)\sigma^2_e$, where $H_2(z) = \frac{1}{1-0.6z^{-1}}$.

$$\begin{gathered} h_2(n) = 0.6^n \cdot u(n) \\ \sigma^2_{\varepsilon,2} = 3\sum h_2^2(n)\sigma^2_e = 3 \cdot \tfrac{1}{1-0.36}\sigma^2_e \approx 4.69\sigma^2_e \end{gathered}$$

We conclude that the first implementation gives the smallest round-off noise.

5. a)

$$\begin{aligned} \varepsilon(\tau) &= \sum_{n=\infty}^{\infty} |y(n) - x(nT_s - \tau)|^2 \\ &= \underbrace{\sum_{n=\infty}^{\infty} |y(n)|^2}_{\text{constant!}} + \underbrace{\sum_{n=\infty}^{\infty} |x(nT_s-\tau)|^2}_{\text{constant!?, see below}} - 2\sum_{n=\infty}^{\infty} y(n)x(nT_s-\tau) = \text{constant} - 2z(\tau)\ , \end{aligned}$$



(a) `my_function1` (b) `my_function2`

Figure 2: Filter structure from MATLAB code

(The sum over $x(nT_s - \tau)$ is actually only guaranteed to be constant if τ is an integer). This shows that maximizing $z(\tau)$ is (more or less) equivalent to minimizing $\varepsilon(\tau)$.

b) Let $w(n) = x_d(-n)$ (which gives $W(f) = X_d^*(f)$). Then,

$$z_d(n) = y(n) * w(n)$$

Also, if $v(t) = x(t - \tau)$, then $y(n) = v(nT_s)$. Since the signals are band limited, the sampling will not introduce any aliasing and consequently

$$Z_d(f) = Y(f)W(f) = V(f/T_s)X_d^*(f) = X(f/T_s)e^{-j2\pi f\frac{\tau_0}{T_s}}X^*(f/T_s) = |X(f/T_s)|^2 e^{-j2\pi f\frac{\tau_0}{T_s}}$$

c) If $z_d(n)$ is non-zero in the interval $n = M, M+1, \ldots, M+N-1$, then

$$\begin{aligned} Z_d(k) = \sum_{n=0}^{N-1} z_d((n)_{\bmod (N)})e^{-j2\pi kn/N} &= \sum_{n=M}^{M+N-1} z_d(n)e^{-j2\pi kn/N} \\ &= \sum_{n=M}^{M+N-1} z_d(n)e^{-j2\pi f}\rfloor_{f=k/N} = Z_d(f)\rfloor_{f=k/N} = \left|X\left(\frac{k}{NT_s}\right)\right|^2 e^{-j2\pi k\frac{\tau_0}{NT_s}} \end{aligned}$$

d) If $|X(k/NT_s)|^2$ is constant for a range of k values, then $Z_d(k) = \text{const}\ e^{-j2\pi k\nu}$ where we introduced the variable $\nu = \tau_0/NT_s$. Note that this looks exactly like an ordinary *time series* with frequency ν. This means that we can use any standard frequency estimation method, for example a periodogram, Bartlett, Blackman-Tukey or model based methods such as AR modeling, Pisarenko or MUSIC to estimate the "frequency" $\hat{\nu}$ and then obtain an estimate of τ_0 from $\hat{\tau}_0 = \hat{\nu}NT_s$.

Note that calculating the standard periodogram is more or less equivalent to calculating the inverse transform $z_d(n)$, so there is no point to consider this frequency domain formulation if we don't use zero-padding to get a finer grid of values when estimating ν.