Laboratory Exercises 3, Digital Filters 2010

Amir Eghbali

April 16, 2010

This laboratory exercise gives some insights into the design and properties of multirate digital signal processing structures and specifically, transmultiplexers (TMUXs). The MATLAB commands and programs that are required for the exercise are listed below. Note that the file **Lab3.zip** containts some other MATLAB and data files you will need during the laboraotory. To make an efficient use of time during the laboratory hours, it is **necessary to do the preparatory exercises before the lab!**.

1 Optional Sections

Some exercises, i.e., Sections 5.4 and 5.5, are **OPTIONAL** and it is not necessary to complete them. Furthermore, you can also ignore the filters in the file **"Filters3.mat"**.

2 Some MATLAB Commands

This section provides some MATLAB commands. For details of how to use them, we refer to the built-in help in MATLAB.

Elementary matrices and matrix manipulation (elmat)
linspace - Linearly spaced vector.

Data analysis and Fourier transforms (datafun)
max - Largest component.
sum - Sum of elements.
format - Set output format.

Elementary math functions (elfun)
abs - Absolute value.
log10 - Common (base 10) logarithm.

Handle graphics (graphics)
figure - Create figure window.
gca - Get handle to current axes.
set - Set object properties.

Two dimensional graphs (graph2d)
axes - Create axes in arbitrary positions.

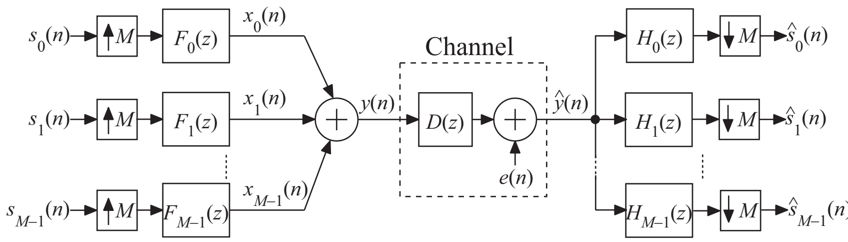

Figure 1: P-channel transmultiplexer.

axis - Control axis scaling and appearance.
grid - Grid lines.
hold - Hold current graph.
plot - Linear plot.
stem - Discrete sequence plot.
impz - Discrete sequence plot.
subplot - Create axes in tiled positions.
title - Graph title.
xlabel - X-axis label.
ylabel - Y-axis label.
zoom - Zoom in and out on a 2-D plot.

M-files in MATLABs Signal Processing Toolbox.
freqz - Digital filter frequency response.
zplane - Z-plane zero-pole plot.
conv - Convolution of two signals.
quant - Quantization of a vector.
upsample - Upsampling a vector.
downsample - Downsampling a vector.
load - Load the data stored in a .mat file.

3 Introduction to Transmultiplexers

A TMUX converts time multiplexed signals into a frequency multiplexed version and back. Suppose we have a series of symbol streams $s_k(n)$, either generated by different users or parts of a signal generated by one user, and we want to transmit these signals through a channel. As shown in Fig. 1, we can pass the signals through a series of transmitter (or pulse shaping) filters $F_k(z)$ to produce the signals

$$x_k(n) = \sum_i s_k(i) f_k(n - iM). \tag{1}$$

The term pulse shaping comes from the fact that the filters $F_k(z)$ take symbols of $s_k(n)$ and put pulses $f_k(n)$ around them. Here, we have M users transmitted through one channel which can be described by a linear time invariant (LTI) filter $D(z)$ followed by additive noise $e(n)$. At the receiver side, the filters $H_k(z)$ separate the signals and only a downsampling by M is needed to get the original

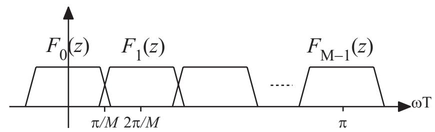

Figure 2: M-channel transmultiplexer filters.

symbol streams. In this system, M signals are multiplexed into one channel and ignoring the effects of the channel, the input-output relationship can be written as

$$\hat{S}_i(z) = \frac{1}{M} \sum_{k=0}^{M-1} S_k(z) T_{ki}(z), \tag{2}$$

where the transfer function

$$T_{ki}(z^M) = \sum_{l=0}^{M-1} F_k(zW^l) H_i(zW^l), \quad W = e^{-j\frac{2\pi}{M}} \tag{3}$$

relates the output signal $\hat{s}_i(z)$ to the input signal $s_k(z)$. Typical characteristics of the filters $F_k(z)$ and $H_k(z)$ are shown in Fig. 2.

3.1 Filter Design

If an LTI filter $g(n)$ is placed between an upsampler and a downsampler of ratio M, the overall system is equivalent to the decimated version of the filter impulse response which becomes $g(nM)$ (refer to Section 4). In this case, designing the transmit/receive filters in each branch, so that the decimated version of $F_k(z)H_m(z)$ becomes a pure delay if $k = m$ (refer to Section 4) and zero otherwise, the TMUX is a perfect reconstruction (PR) system. In a PR system, $\hat{s}_k(n) = \alpha s_k(n - \beta)$ which means that the output is a scaled (by α) and delayed (by β) version of the input. The PR properties are independent of filter lengths, causality of filters, etc. Additionally, to design a PR system, there is a need to use nonlinear programming techniques so that requirements on the zeroth polyphase components of the cascaded filters are satisfied. In other words, the filters designed using firpm command (as in laboratory 1) can not be used since we need specific requirements on the transition bands. The file **"Filters4.mat"** contains a set of filters designed using the firpm function whereas the files **"Filters1.mat"**, **"Filters2.mat"**, and **"Filters3.mat"** contain filters designed using nonlinear optimization.

4 Preparatory Exercises

1. Based on Fig. 1, the TMUX structure for $M = 2$ can be drawn as in Fig. 3. Derive the transfer functions $T_{00}(z)$ and $T_{01}(z)$.

2. Based on Fig. 2, draw the typical characteristics of the filters for a TMUX with $M = 2$.

3. Show that if an LTI filter $g(n)$ is placed between an upsampler and a downsampler of ratio 2, (refer to Fig. 4), the overall system is equivalent

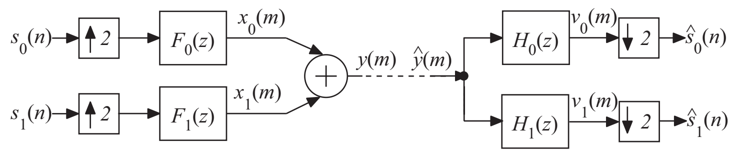

Figure 3: 2-channel transmultiplexer.

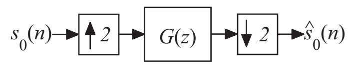

Figure 4: LTI filter placed between an upsampler and downsampler.

to the decimated version of the filter impulse response which becomes $g(2n)$. Which polyphase component does $g(2n)$ correspond to? How do you derive the zeroth polyphase component from an impulse response?

4. What does a pure delay mean and what is the impulse response of such an LTI filter?

5. In a linear-phase FIR filter of order N_{order}, determine the number of samples by which the output will be delayed with respect to the input.

6. In a system with W_b bits and assuming that rounding is used, what is the quantization step?

5 Laboratory Exercises

5.1 Individual Filter Characteristics

1. For the file **"Filters2.mat"**, plot the impulse response of the filters **f0** and **f1**. Which type of linear phase filters (Type I, II, III, IV) are they? What orders do they have?

2. For the file **"Filters2.mat"**, plot the frequency response of the filters **f0** and **f1** in the frequency range $[0, \pi]$. What type of characteristics do these filters have?

3. Assuming the general frequency specification of lowpass and highpass filters as

$$\begin{aligned} 1-\delta_c \leq |H_{LP}(e^{j\omega T})| \leq 1+\delta_c, \quad & \omega T \in [0, \omega_c T], \\ |H_{LP}(e^{j\omega T})| \leq \delta_s, \quad & \omega T \in [\omega_s T, \pi], \end{aligned} \tag{4}$$

$$\begin{aligned} 1-\delta_c \leq |H_{HP}(e^{j\omega T})| \leq 1+\delta_c, \quad & \omega T \in [\omega_c T, \pi], \\ |H_{HP}(e^{j\omega T})| \leq \delta_s, \quad & \omega T \in [0, \omega_s T], \end{aligned} \tag{5}$$

find the values of δ_c and δ_s for the filters **f0** and **f1** in files **"Filters1.mat"**, **"Filters2.mat"**, and **"Filters3.mat"**. Assume $\omega_c T = 0.45\pi$ and $\omega_s T = 0.55\pi$ for the lowpass filter and $\omega_s T = 0.45\pi$ and $\omega_c T = 0.55\pi$ for the highpass filter. What is Typical of these filters? Which design method do you think has been used?

4. In file **"Filters2.mat"**, plot the pole-zero locations of the filters **f0** and **f1**. Relate the pole-zero locations to the frequency response of the filters.

5.2 Overall Filter Characteristics

1. For the file **"Filters2.mat"**, plot the frequency response for the cascade of **f0** and **h0**; the cascade of **f1** and **h1**; and the cascade of **f0** and **h1**. (Note: $\omega T \in [0, \pi]$.)

2. For the files **"Filters1.mat"**, **"Filters2.mat"**, and **"Filters3.mat"**, plot the frequency response of the zeroth polyphase component for the cascade of **f0** and **h0**; the cascade of **f1** and **h1**; and the cascade of **f0** and **h1**. (Note: $\omega T \in [0, \pi]$.)

3. Assuming that the specifications of an allpass transfer function is given by
$$1 - \delta_c \leq |H_{AP}(e^{j\omega T})| \leq 1 + \delta_c, \quad \omega T \in [0, \pi], \tag{6}$$
determine the values of δ_c for the zeroth polyphase component of the cascade of **f0** and **h0** as well the cascade of the **f1** and **h1** in the files **"Filters1.mat"**, **"Filters2.mat"**, and **"Filters3.mat"**.

4. In the files **"Filters2.mat"** and **"Filters4.mat"**, plot the term $|F_0(e^{j\omega T})|^2 + |F_1(e^{j\omega T})|^2$. What differences do you see? (Note: $\omega T \in [0, \pi]$.)

5.3 TMUX Implementation

1. The file **"Inputs.mat"** contains complex vectors **s0** and **s1** which are the 16-QAM modulated representations of text messages **m0** and **m1**. The TMUX will transmit the complex vectors **s0** and **s1** through the channel. To see how each text message is mapped onto the 16-QAM constellation, run the function **"constellationplot.m"**.

2. Using the commands **conv**, **upsample**, and **downsample**, implement the TMUX of Fig. 3. To compare the transmit and receive sequences i.e., $s_k(n)$ and $\hat{s}_k(n)$, in the time domain, you need to compensate for the delay of the cascade of the filters in the TMUX.

3. To compensate for the delay, ignore a number of samples (say N_d) at the beginning of vectors $\hat{\mathbf{s}}_0$ and $\hat{\mathbf{s}}_1$. To get the correct value for N_d, divide your answer in Part 5 of Section 4 by 2.

4. To represent the 16-QAM modulated output **s0r** as text message **m0r**, you can use the following command:
m0r= Sym2Msg(**s0r**,16)

5. Simulate your TMUX with the filters in files **"Filters1.mat"**, **"Filters2.mat"**, and **"Filters3.mat"**.

6. Compare the number of errors (difference between **m0** and **m0r** as well as the difference between **m1** and **m1r**) and the signal-to-noise ratio (SNR) in your demodulated results, i.e., **s0r** and **s1r**, with respect to the input

signals **s0** and **s1**. Assuming the transmitted and received signals $s_0(n)$ and $\hat{s}_0(n)$ for $n = 1, 2, \ldots, N$, the SNR can be computed as

$$SNR = 10\log_{10} \frac{\sum_{n=1}^{N} |s_0(n)|^2}{\sum_{n=1}^{N} |\hat{s}_0(n) - s_0(n)|^2} \quad dB. \tag{7}$$

What conclusions can you draw about the error and the SNR considering the values of δ_c measured in Part 3 of Section 5.2?

7. To get an insight on how the TMUX works, plot the frequency responses of $S_0(z^2)$, $S_1(z^2)$, $X_0(z)$, $X_1(z)$, $Y(z)$, $V_0(z)$, $V_1(z)$, $\hat{S}_0(z)$, and $\hat{S}_1(z)$ in Fig. 3. How do you relate these frequency responses to the operations of filtering, upsampling, and downsampling?

5.4 Finite Word Length Effects (OPTIONAL)

1. Quantize the coefficients of the filters **f0** and **f1** in file **"Filters2.mat"** with 16, 8, and 4 bits. How are the frequency response and the pole-zero locations affected?

2. Using the filters in file **"Filters2.mat"**, simulate your TMUX with the filters quantized. Compare the error and SNR with those achieved in Section 5.3. What conclusions can you draw?

3. Using the filters in file **"Filters2.mat"**, quantize the outputs of all the filters with 16, 8, and 4 bits. (No quantization in coefficients.)

4. Simulate your TMUX with quantized outputs and compare the error and SNR with those achieved in Section 5.3. What conclusions can you draw?

5.5 Effects of Channel Noise (OPTIONAL)

1. Add an additive white Gaussian noise $e(n)$ to the signal $y(n)$ in your implemented TMUX. To add a noise such that the SNR becomes $40dB$, use the following command
   y = awgn(**y**,40,'measured',1234,'db');

2. For the filters in file **"Filters2.mat"**, simulate your TMUX for SNR values of 5, 10, 20, and 40. Compare the error and SNR and note your conclusions. (Assume no quantization in coefficients and filter outputs.)