

Homework No. 7

Tomás Balderas Contreras
Instituto Nacional de Astrofísica, Óptica y Electrónica
Course: Fundamentals of Digital Signal Processing

May 5, 2003

1 Computer Projects

1. Design a digital Chebyshev-I filter with passband frequency $\omega_p = 0.2$, stopband frequency $\omega_s = 0.3$, passband ripple $R_p = 0.2$ dB and stopband attenuation $A = -60$ dB. Plot impulse response, magnitude response, phase response and Pole/Zero pattern. Plot the structure of the filter.

Solution: The MATLAB program which was written to solve this problem is explained next, step by step and with care.

At first we initialize the program's variables with the parameters we are given. This is done by the following piece of code

```
% Set normalized passband frequency
Wp = 0.2;
% Set normalized stopband frequency
Ws = 0.3;
% Set passband ripple (dB)
Rp = 0.2;
% Set minimum stopband attenuation
A = 60;
```

Now we actually design the filter by using the M-files `cheb1ord` and `cheby1` which determine the lowest order of the filter's system function and the system function itself, respectively. The calls to these functions are shown next:

```
% Calculate the order of Chebyshev I filter
[n, Wn] = cheb1ord(Wp, Ws, Rp, A);

% Design Chebyshev I filter
[b, a] = cheby1(n, Rp, Wn);
```

In the rest of the code we compute the Pole/Zero pattern from the system function's coefficients, compute the frequency response needed to determine magnitude and phase response and compute unit sample response. The code is as follows:

```
% Compute frequency response
[H w] = freqz(b, a, 512);

% Compute Pole/Zero pattern
[z, p, k] = tf2zp(b, a);

% Compute impulse response
```

```
[h, t] = impz(b, a, 120);

% Shows information
disp('Zeros: '); disp(z);
disp('Poles: '); disp(p);
```

The sequence of statements which plots all of the information we have computed so far is the next part of the program, as you can see in the following part of the program:

```
% Plot impulse response
subplot(2, 2, 2);
stem(t, h);
xlabel('n'); ylabel('h(n)');

% Plot magnitude response
subplot(2, 2, 3);
plot(w/pi, 20*log10(abs(H))); grid;
xlabel('\omega/\pi'); ylabel('Gain, dB');

% Computes and plots phase response
subplot(2, 2, 4);
plot(w/pi, angle(H)); grid;
xlabel('\omega/\pi'); ylabel('Phase, radians')
```

Figure 1 depicts the plots obtained by running this program in MATLAB. The exact location of singularities in the Z -plane were also computed and are shown next:

```
Zeros:
-1.0305
-1.0229 + 0.0198i
-1.0229 - 0.0198i
-1.0045 + 0.0295i
-1.0045 - 0.0295i
-0.9848 + 0.0251i
-0.9848 - 0.0251i
-0.9726 + 0.0097i
-0.9726 - 0.0097i

Poles:
0.7834 + 0.5801i
0.7834 - 0.5801i
0.7832 + 0.4971i
0.7832 - 0.4971i
0.8083 + 0.3665i
0.8083 - 0.3665i
0.8494
0.8371 + 0.1955i
0.8371 - 0.1955i
```

Finally we use the `zp2sos` function to determine the coefficients of the second order sections used to implement the filter as a cascade combination of them. Figure 2 shows a block diagram of the filter's implementation.

```
% Computes second-order section representation
[sos, g] = zp2sos(z, p, k);

disp('Coefficients for second-order sections: '); disp(sos);
disp('Gain: '); disp(g);
```

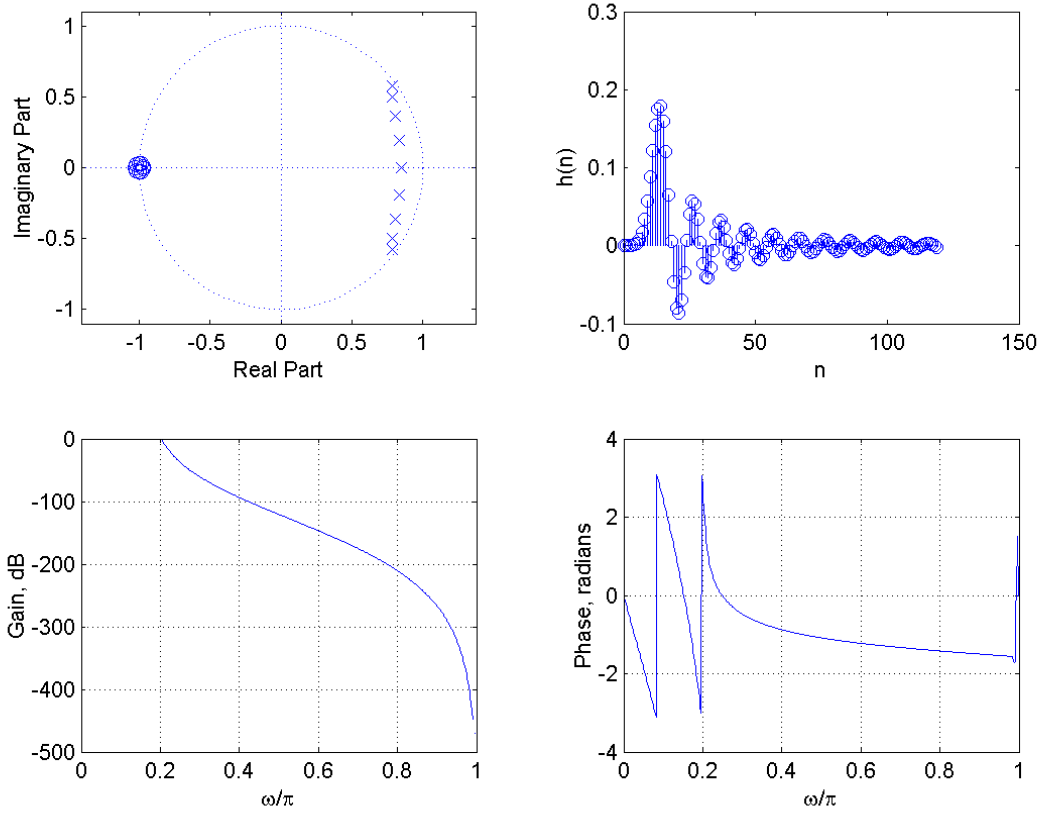


Figure 1: Pole/Zero pattern, impulse response, magnitude response and phase response for the Chebyshev-I filter.

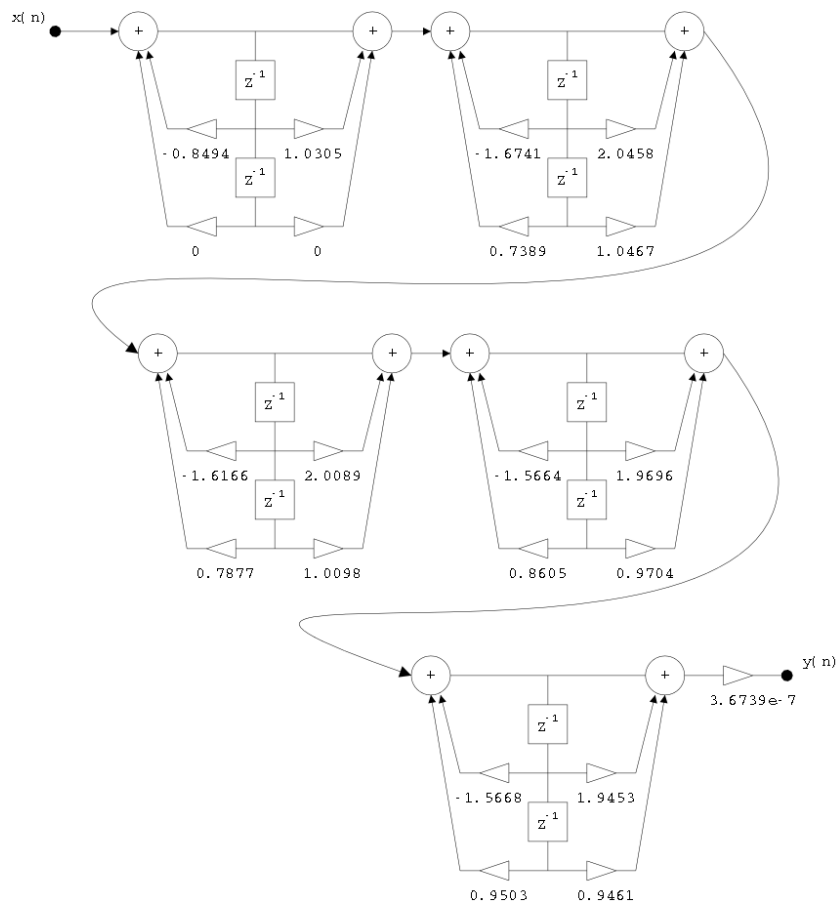


Figure 2: Block diagram for the Chebyshev-I filter.

2. Design the elliptic digital filter with the same specification. Compare results.

Solution: In this case we use the same program as above, but instead of calling to `cheb1ord` and `cheby1` we will call to functions `ellipord` and `ellip` to compute the order of the filter and build the system function, respectively. This sequence of calls is as follows:

```
% Calculate the order of Elliptic filter
[n, Wn] = ellipord(Wp, Ws, Rp, A);

% Design Elliptic filter
[b, a] = ellip(n, Rp, A, Wn);
```

Figures 3 and 4 depict plots for the responses and Pole/Zero pattern and the block diagram of the implementation of the elliptic filter, respectively. The explicit values of the singularities computed by the program are given now:

```
Zeros:
-0.4508 + 0.8926i
-0.4508 - 0.8926i
 0.6074 + 0.7944i
 0.6074 - 0.7944i
 0.4252 + 0.9051i
 0.4252 - 0.9051i
```

```
Poles:
 0.7674 + 0.5769i
 0.7674 - 0.5769i
 0.7472 + 0.4349i
 0.7472 - 0.4349i
 0.7444 + 0.1682i
 0.7444 - 0.1682i
```

We notice that the elliptic option has some advantages over the Chebyshev-I one, in terms of the number of multipliers and memory elements needed to implement it. Since the elliptic filter requires less components than the Chebyshev-I this is a great advantage.

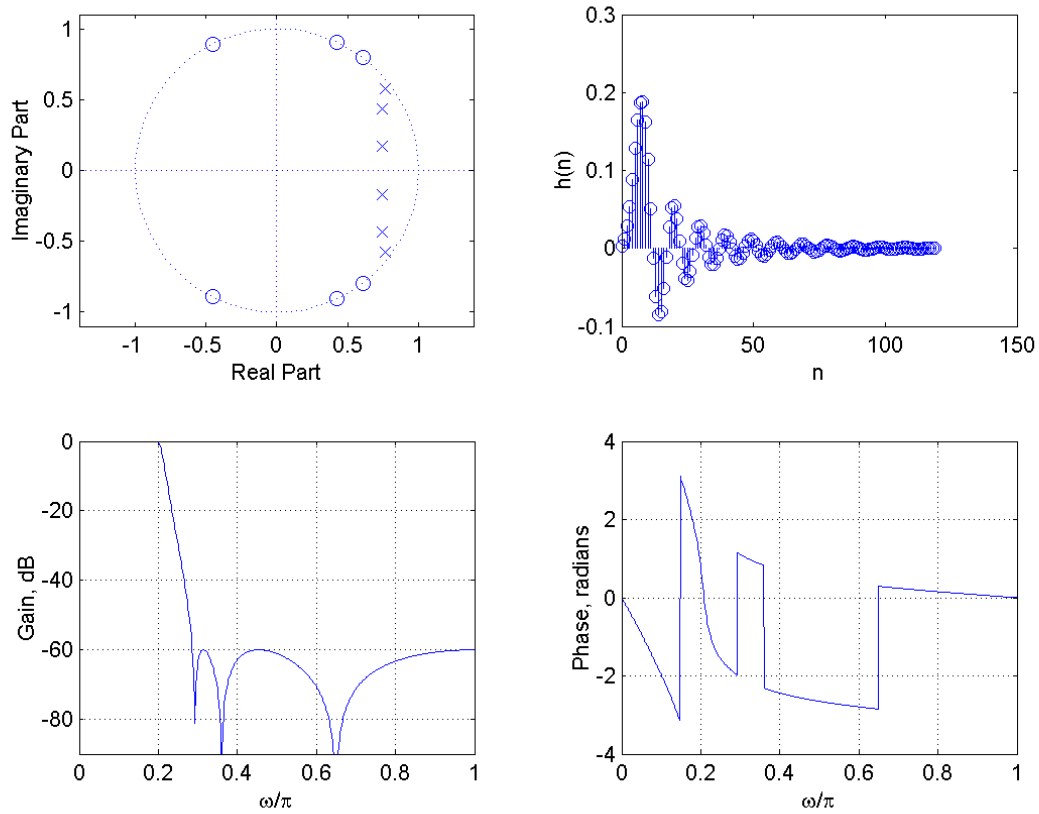


Figure 3: Pole/Zero pattern, impulse response, magnitude response and phase response for the Elliptic filter.

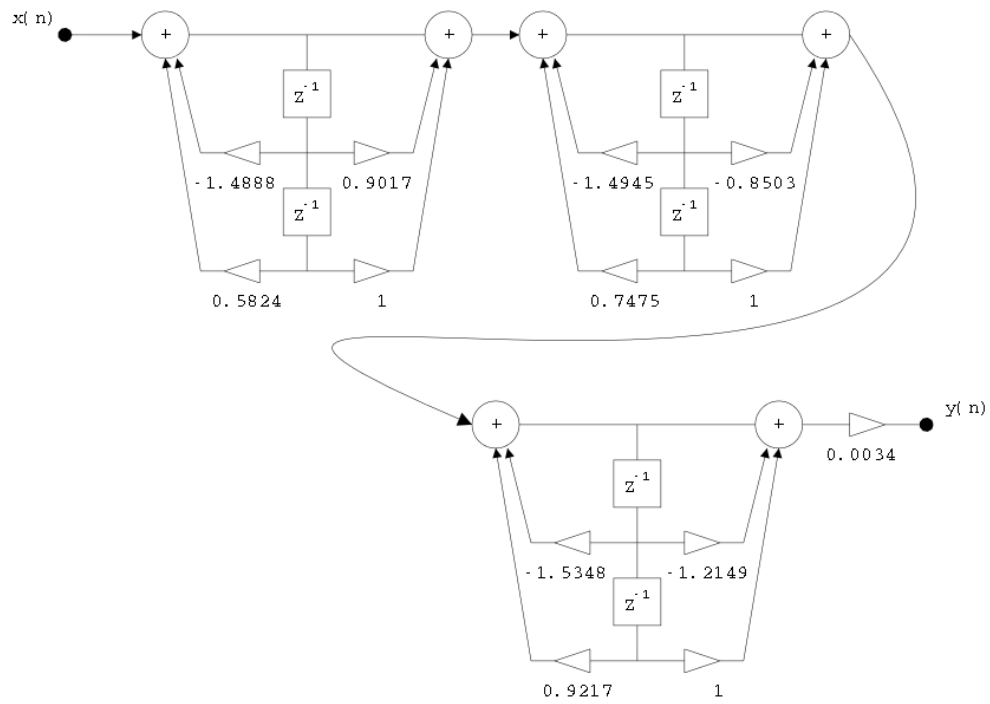


Figure 4: Block diagram for the Elliptic filter.