

# Shaping Filter Design for a 155Mbps 64-CAP ATM Transceiver

Farzad Etemadi, Hung-Kang Liu,  
Nader Bagherzadeh, Fadi J. Kurdahi

December 12, 1996

## 1 Introduction

The purpose of modulation in digital transmission systems is to change the frequency characteristics of the transmitted signal to a form which could be accepted by the channel. Transmission of a lowpass signal through a bandpass channel results in an undesired signal distortion and loss of energy. A modulator is thus necessary to match the signal and the channel characteristics. The Carrierless AM/PM(CAP) modulation technique [1, 2] uses a pair of *In-Phase(I)* and *Quadrature(Q)* filters to modulate a stream of data symbols into a bandpass signal which could satisfy the channel requirements. At the receiving end, the demodulator shaping filter retrieves the original lowpass signal which could be used to get the data symbols back.

In addition to the frequency domain constraints, the shaping filters must also satisfy a time domain constraint called the *Nyquist Criterion* in order to avoid the *Inter Symbol Interference(ISI)*. Finally, to maximize the Signal to Noise Ratio(SNR) at the receiver, the transmitter/receiver filters should form a *matched* pair. Thus, any data transmission filter should satisfy the frequency domain specifications, ISI constraint, and matched filter requirement at the same time.

Design of the shaping filters of a 155Mbps transceiver is the subject of this report. The filter design problem is discussed first, then the design algorithm for Nyquist and matched filters is presented. Finally, some implementation issues will be discussed and the directions for future work will be given.

## 2 Shaping Filter Design

The in-phase and quadrature filters of a CAP modulator system are given by [2]:

$$\begin{aligned} p(t) &= g(t)\cos(2\pi f_c t) \\ \tilde{p}(t) &= g(t)\sin(2\pi f_c t) \end{aligned} \quad (1)$$

where  $p(t)$  is the in-phase filter,  $\tilde{p}(t)$  is the quadrature filter, and  $f_c$  is the carrier frequency.  $g(t)$  is a lowpass pulse, and as equation(1) shows, the bandpass in-phase and quadrature filters are obtained by modulating  $g(t)$  with the carrier frequency  $f_c$ . The lowpass pulse  $g(t)$  will be designed first, and then the I(Q) filter will be obtained by pointwise multiplication of  $g(t)$  by the cosine(sine) function.

In order to avoid ISI, the cascaded impulse response of the transmitter, the channel, and the receiver should be a *Nyquist Pulse*. A Nyquist pulse has the property that its value is zero at the nonzero integer multiples of the symbol period. If we denote this impulse response by  $h(n)$ , then:

$$h(n) = \begin{cases} C & n = n_0 \\ 0 & n = n_0 \pm kM, \quad k = 1, 2, \dots \end{cases} \quad (2)$$

where  $M$  is the upsampling factor and  $C$  is a constant. It can be shown that the frequency domain translation of (2) is [3]:

$$\sum_{k=0}^{M-1} H(e^{j(\omega+2\pi k/M)}) = 1 \quad (3)$$

If we assume that the channel is perfect, then:

$$h(n) = h_T(n) * h_R(n) \quad (4)$$

where  $h_T$  and  $h_R$  are the impulse responses of the transmitter and receiver, respectively. The maximum SNR can be achieved if  $h_T$  and  $h_R$  form a matched pair, that is:

$$H_R(\omega) = H_T^*(\omega) \quad (5)$$

$$h_R(n) = h_T\left(\frac{N-1}{2} - n\right) \quad (6)$$

in which  $N$  is the length of  $h(n)$ .  $N$  should be an odd integer, since we assume that  $h_T$  and  $h_R$  are of the same order, and therefore  $h(n)$  must be of even order. From (6) it is evident that the matched filters are mirror images of each other.

The filter design problem can be formulated in two steps:

**Nyquist Filter Design** Design a Nyquist pulse  $h(n)$  satisfying (2) and the frequency domain specifications (i.e., stopband attenuation and the edge frequencies).

**Matched Filter Design** Split  $h(n)$  into two equal order filters  $h_T(n)$  and  $h_R(n)$  satisfying (4) and (6). This step is called the *Spectral Factorization* problem.

The perfect channel assumption of (4) is not realistic. However, by using an equalizer, we could approximate the ideal channel. In the next two sections we will discuss these two steps.

### 3 Nyquist Filter Design

The ideal Nyquist pulse has a rectangular frequency response bandlimited to the half of the symbol rate and it can be proved that this bandwidth is the minimum bandwidth required for the given symbol rate [3]. The ideal Nyquist pulse is not realizable in practice and therefore it is always approximated by using some *excess bandwidth*. The deviation from the ideal Nyquist pulse is indicated by the *roll-off* factor  $\alpha$ .  $\alpha = 0$  corresponds to the ideal Nyquist pulse, while  $\alpha = 1$  corresponds to a 100% excess bandwidth.

One of the most commonly used Nyquist filters is the so-called *Raised-Cosine* family [3]. Closed-form formulas exist for this type of Nyquist filter which could be easily used to design a Nyquist filter for a given  $\alpha$ . Moreover, the resulting Nyquist filter could be easily split into two matched *Square-root Raised-Cosine* filters and no spectral factorization is needed for this type of filters. However, this type of filter is not optimal in order, and for small values of  $\alpha$ , the resulting raised-cosine filter could be too long.

Optimal low excess bandwidth Nyquist filters can be designed by modifying the well-known Parks-McClellan algorithm [6] for optimal equiripple FIR filter design. The modifications are necessary to satisfy the ISI constraint. Several methods exist for FIR filter design. However, it is known that linear programming is a method of choice for FIR filters with both time and frequency domain constraints, such as Nyquist filters [5]. One problem

with the linear programming approach is that the filter constraints are normally applied to a dense grid of frequencies in the stopband region, and this will result in a numerically ill-conditioned problem. The algorithm proposed by Samueli [4] eliminates redundant constraints from the program and thus avoids possible numerical problems. This algorithm was used in our work and the results will be discussed next.

An ideal Nyquist filter with an upsampling factor of  $M$  can be described as:

$$H_{Ideal}(e^{j\omega}) = \begin{cases} 1 & 0 \leq \omega < \pi/M \\ 0 & \pi/M < \omega \leq \pi \end{cases} \quad (7)$$

With a roll-off factor  $\alpha$ , the stopband edge will be equal to :

$$\omega_s = (1 + \alpha)\pi/M \quad (8)$$

In a general equiripple FIR filter design the passband edge  $\omega_p$ , the stopband edge  $\omega_s$ , the passband ripple  $\delta_p$ , and the stopband ripple  $\delta_s$  are independently specified. However, in our application, the critical parameters are  $\omega_s$ , which determines the bandwidth required for transmission, and  $\delta_s$ , which determines the out-of-band signal attenuation.  $\omega_p$  and  $\delta_p$  only determine the envelope of the transmitted spectrum which is generally not of much importance. Moreover, the ISI constraint (3) implies that  $\delta_p$  and  $\delta_s$  are not independent and are related through:

$$\delta_p \leq (M - 1)\delta_s \quad (9)$$

The frequency response of an  $N$ -tap ( $N$  odd) linear phase FIR filter is given by:

$$H(e^{j\omega}) = h(0) + \sum_{n=1}^{\frac{N-1}{2}} 2h(n)\cos\omega n \quad (10)$$

For the case of linear phase Nyquist filters, the central coefficient is always  $1/M$ , therefore:

$$H(e^{j\omega}) = 1/M + \sum_{n \notin Z}^{\frac{N-1}{2}} 2h(n)\cos\omega n \quad (11)$$

where  $Z$  is the set of indices corresponding to the zero-valued coefficients of the impulse response. As mentioned before, the frequency response of the

filter will be constrained in the stopband only. These constraints are:

$$\sum_{n \notin Z}^{\frac{N-1}{2}} 2h(n)\cos\omega_k n \geq -1/M \quad (12)$$

$$-\sum_{n \notin Z}^{\frac{N-1}{2}} 2h(n)\cos\omega_k n + \delta \geq 1/M \quad (13)$$

where  $\omega_s \leq \omega_k \leq \pi$ . Equation (12) makes the frequency response positive and thus factorable, while equation (13) limits its maximum value to  $\delta$  in the stopband. In this linear program, the variables are  $\delta$  and the non-zero coefficients of  $h(n)$ .  $\delta$  is the objective function which should be minimized subject to (12) and (13). The minimum number of frequencies  $\omega_k$  which is required to solve this program is equal to the number of variables. The selection of these frequencies is based on the fact that the maximum total number of the extremal frequencies of an optimal equiripple FIR filter is known and is equal to  $N_e = (N + 1)/2$  [6]. By knowing that the length of the stopband region is  $(M - 1)$  times the length of the passband region (see (7)), Samuelli approximates the number of the extremal frequencies in the stopband to be  $(M - 1)$  times of the number of the passband extrema. Therefore, the number of the stopband extrema is :

$$N_s \simeq N_e(M - 1)/M = (N + 1)(M - 1)/2M \quad (14)$$

The important point in this result is that if we limit the value of N to be of form:

$$N = 2Mk - 1 \quad k = 1, 2, \dots \quad (15)$$

then  $N_s$  will be equal to the number of the non-zero coefficients of  $h(n)$ . Therefore, if we apply the constraints of (12) and (13) at each extremum frequency and  $\omega_s$ , then the number of the constraints will be equal to the number of the variables, and the linear program could be easily solved. The location of the extremal frequencies of the stopband is not know *a priori*, but they will be located using an algorithm similar to the Remez exchange algorithm. That is, the real extrema will be located by an iterative procedure in which the extrema of one iteration will be used as the starting point in the next iteration.

In summary, the Nyquist filter design algorithm can be described as:

1. Find  $\omega_s$  using (8).
2. Select a value for  $N$  using (15).
3. Initialize the passband extrema frequencies to an equally spaced set of  $N_s$  frequencies in the interval  $[\omega_s, \pi]$ . The value of  $N_s$  is given by (14).
4. Minimize  $\delta$  subject to (12) and (13).
5. Goto step 7 if the difference between the new value and the old value of  $\delta$  is smaller than a given threshold.
6. Find the passband extrema of  $H(e^{j\omega})$ . Goto step 4.
7. Stop if the value of  $\delta$  is acceptable, otherwise increase  $N$  according to (15) and goto step 3.

Samueli uses Newton's algorithm for finding the exact locations of the extrema frequencies. We used a simple zero-crossing algorithm instead and the results were satisfactory in all cases. The extrema frequencies of  $H(e^{j\omega})$  were found as the midpoints of the zero-crossings of the derivatives of equation (11), calculated on a grid of  $6N$  points in the stopband.

This algorithm was implemented in MATLAB and The results for  $\alpha = 0.15$ ,  $M = 4$ , and  $N=103$  are shown in Figure 1. As this figure shows, the algorithm converges to an equiripple Nyquist filter with -64 dB attenuation in the stopband. The impulse response of this filter is shown in Figure 2. This filter can be later split into two filters of length 52 with -30 dB attenuation in the stopband.

As a comparison, the same specifications were used in the Filter Design System (FDS) software [7] to design a square-root raised-cosine filter. The minimum tap length for this specification was 89. This result indicates that our design is an optimal filter.

## 4 Matched Filter Design

The next step of the design is the spectral factorization of the Nyquist filter into two filters  $h_T$  and  $h_R$ . This could be done by finding all the zeros of the Nyquist filter and then using half of these zeros for  $h_T$  and the other half for  $h_R$ . Since the filter is of order 102, the resulting root-finding algorithm could

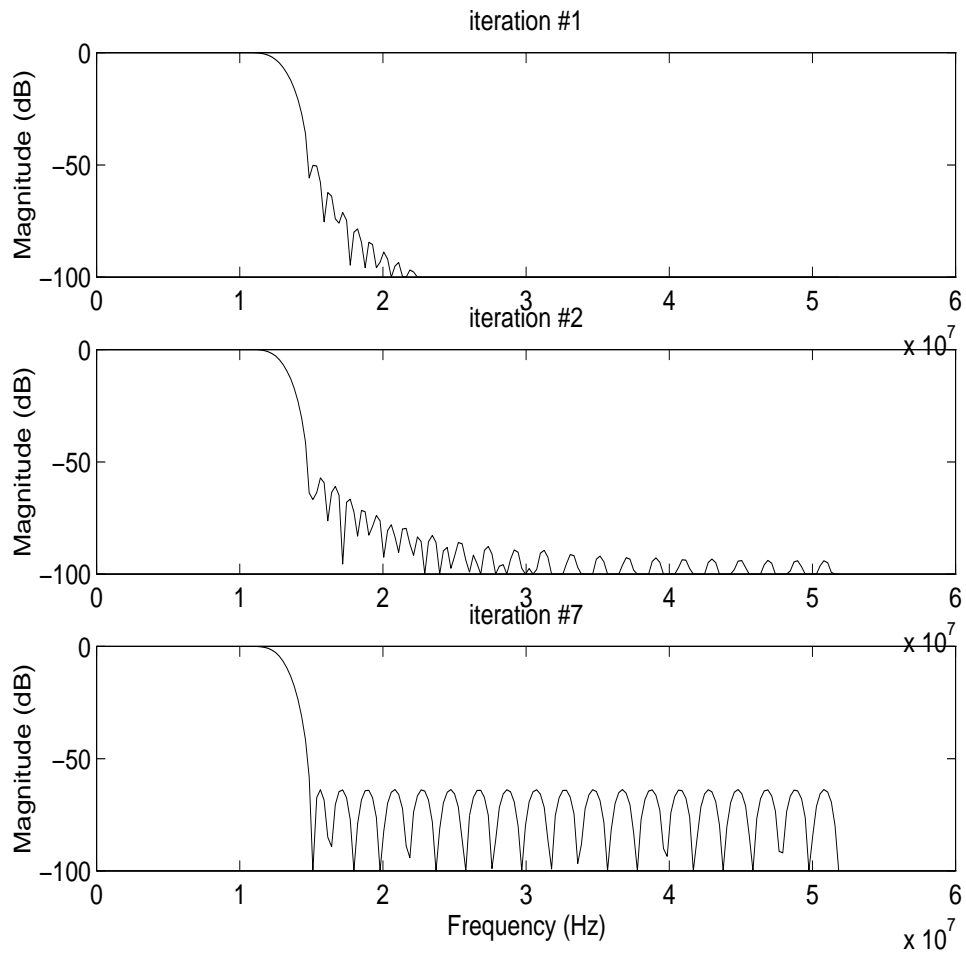


Figure 1: Nyquist Filter Design Iterations

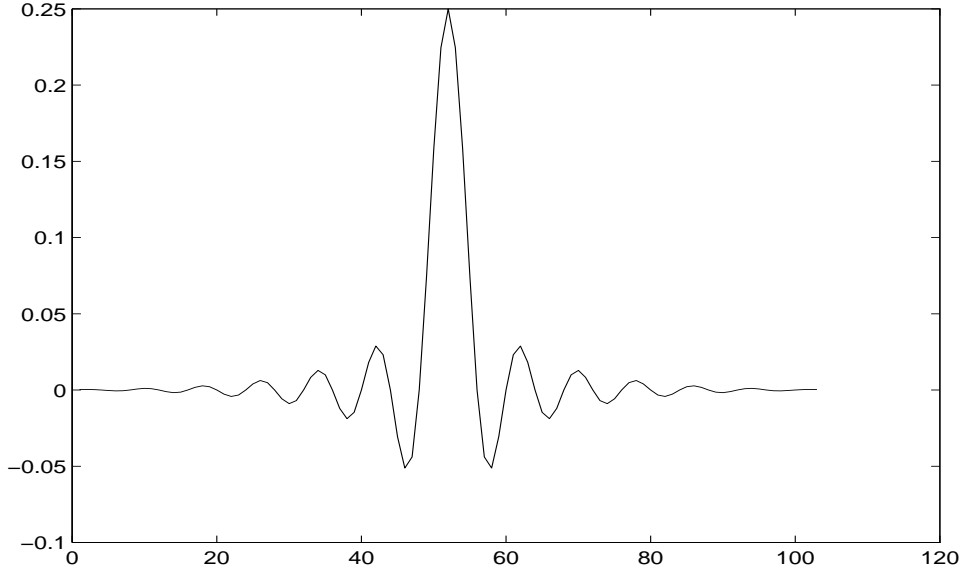


Figure 2: Nyquist Filter Impulse Response

be ill-conditioned [8]. In our design, the Nyquist filter zeros were the same using both MATLAB and FDS, which verified their validity. However, some problems arose using the convolution function of MATLAB for multiplying the polynomials. The first problem was that the coefficients of  $h_T$  and  $h_R$  were found to be complex valued. The reason was that the complex roots were not the exact conjugates of each other, and these small deviations were accumulated together. This problem was solved by grouping together each pair of conjugate roots into a real valued second order polynomial, and then convolving these real polynomials.

The other problem was that for some particular groupings of zeros, the result was totally wrong. The reason was that the convolution function is generally ill-conditioned for high-order polynomials. It was found out that the minimum and maximum phase decomposition would result to a valid decomposition. Furthermore, the minimum and maximum phase filters are matched, and therefore they satisfy the matched filter constraint (5). For these reasons, we factored the Nyquist filter into its minimum and maximum phase components. This was done by selecting all the inside unit circle zeros and every other unit circle zero for  $h_T$ , and all others for  $h_R$ .

Figure 3 shows the impulse response of the original Nyquist filter, and

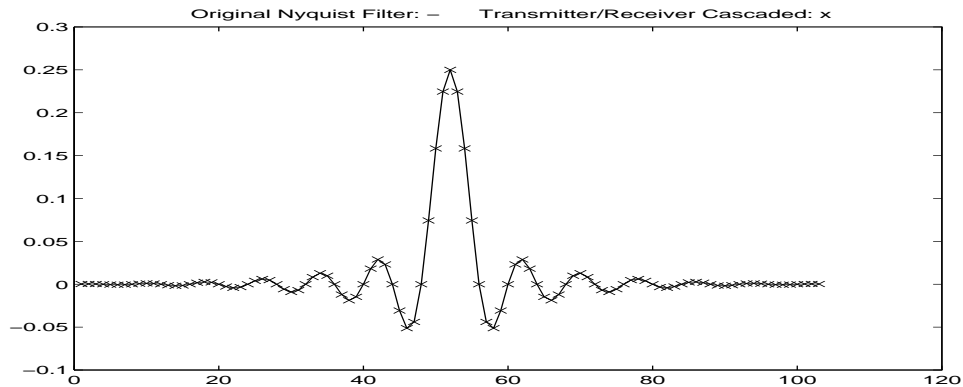


Figure 3: Nyquist Pulse Factorization

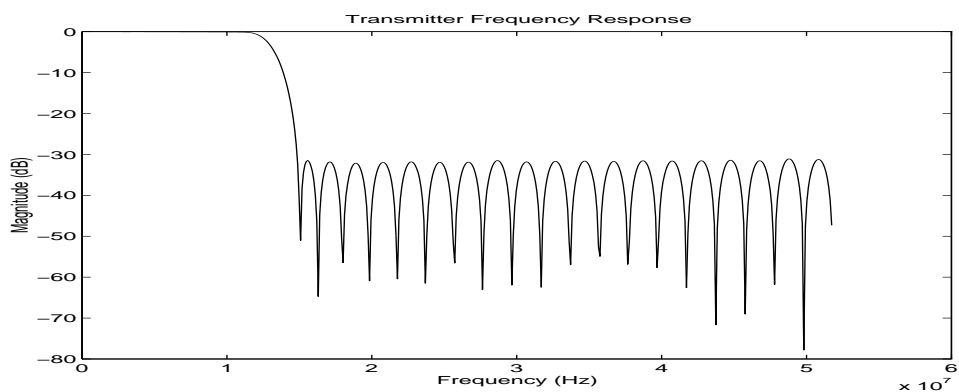


Figure 4: Transmitter Frequency Response

the impulse response of the cascaded Transmitter/Receiver filters, which verifies the factorization. The individual frequency responses of these filters are shown in figures 4 and 5. Finally, the impulse responses are shown in figure 6, which shows the mirror symmetry property of these matched filters.

## 5 Implementation Issues

The low-pass filters designed in previous section will be converted to the bandpass filters by pointwise multiplication of the coefficients by the cosine and sine functions. In the case of the 52Mbps transceiver described in [1, 9], this multiplication is equivalent to multiplying by  $\{-1,0,1,0\}$  which results

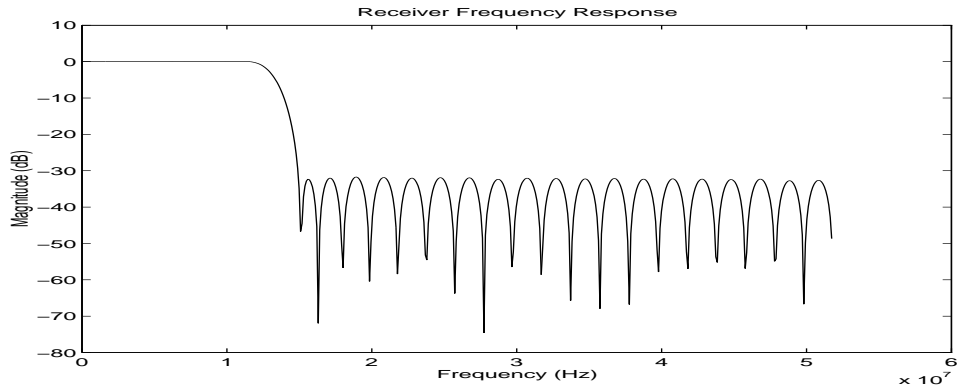


Figure 5: Receiver Frequency Response

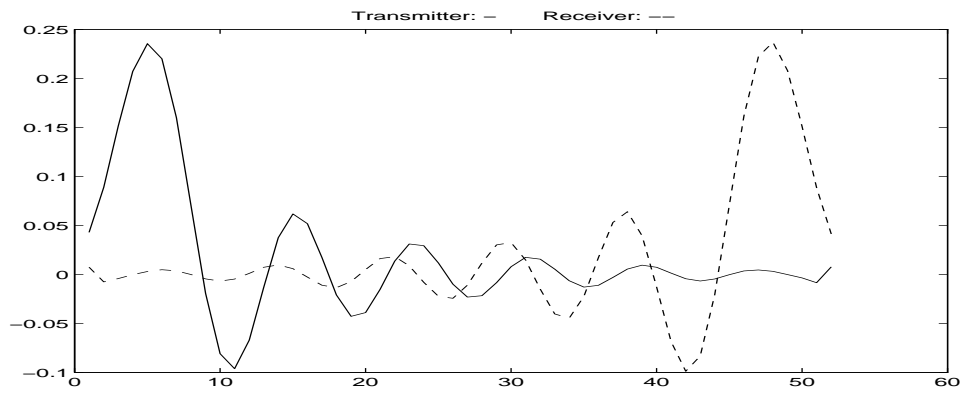


Figure 6: Transmitter/Receiver Impulse Response

in significant hardware savings. In the 155Mbps transceiver, however, this simplification is not possible. Even without this simplification, the CAP scheme is still more hardware efficient than QAM [9]. The use of CSD codes for filter coefficients is also an important factor in hardware simplification.

Another important problem is that the minimum and maximum phase filters are not linear phase. Therefore their impulse response is not symmetric and it is not possible to reduce the number of taps by using the folded structure of the symmetric linear phase FIR filters. Our future work will concentrate on the design of symmetric transmitter/receiver filters.

The addition of the symmetry constraints will reduce the degrees of freedom in the filter design problem, thus making the optimization more difficult. Moreover, these constraints are nonlinear, and nonlinear optimization methods should be used. It is expected for the symmetric filters to be of higher order. Therefore a cost trade-off exists between a high-order symmetric filter and a low-order non-symmetric filter. The final choice will depend on the solution of the nonlinear optimization problem.

## 6 Conclusion

Design of the transmitter and receiver filters of a 155Mbps transceiver was discussed in this report. It was shown that the optimal equiripple filters are much more efficient than the raised-cosine filters. The final filters have an asymmetric impulse responses which are not as efficient as the symmetric filters, and our future work will concentrate on the design of the symmetric filters.

## References

- [1] Hung-Kang Liu, *Design and Implementation of a 52 Mbps ATM Transceiver*. UCI, Fall 1996.
- [2] J. J. Werner. *Tutorial on Carrierless AM/PM-Part I*. AT&T Bell Laboratories, 1992.
- [3] Edward A. Lee, David G. Messerschmitt. *Digital Communication*. Boston, Kluwer Academic Publishers, 1994.

- [4] Henry Samuelli, “On the Design of Optimal Equiripple FIR Digital Filters for Data Transmission Applications”, *IEEE Transactions on Circuits and Systems*, Vol. 35, No. 12, December 1988.
- [5] Lawrence R. Rabiner, Bernard Gold. *Theory and Application of Digital Signal Processing*. Englewood Cliffs, Prentice-Hall, 1975.
- [6] Alan V. Oppenheim, Ronald W. Schaffer. *Discrete-time Signal Processing*. Englewood Cliffs, Prentice Hall, 1989.
- [7] ALTA GROUP of Cadence Design Systems, Inc. *Signal Processing Worksystem Filter Design System*. March 1995.
- [8] C.E. Schmidt, L. R. Rabiner, “A Study of Techniques for Finding the Zeros of Linear Phase FIR Digital Filters”, *IEEE Transactions on Acoustic, Speech, and Signal Processing*, February 1977.
- [9] Jeffrey Scott Putnam. *Analysis and Design of High-Speed Local Area Network Transceivers*. UCLA, 1995.

## Appendix: MATLAB Source Code

```
% Nyquist Filter Design Program
% -----
% Farzad Etemadi, UCI, Fall 1996.

% Initialization
% -----

a=0.15;          % Excess bandwidth
m=4;            % Up-sampling factor
ws=(1+a)*pi/m;  % Stopband discrete frequency
fsamp=103.68e6; % Sampling frequency
fmod=15e6;      % Modulation frequency

k=13;          % Integer used for filter order calculation
n=2*m*k-1;     % Filter order (must be of 2mk-1 form)
ns=k*(m-1);    % Number of stopband extrema
maxiter=100;
Threshold=1e-7;
npoints=200;

f=zeros(ns+1,1); f(ns+1,1)=1; % Objective function

b=ones(ns+1,1)/m;          % Right-hand side vector of the inequality
for j=1:2:ns+1-rem(ns,2),
    b(j,1)=-b(j,1);
end;

w=zeros(ns+1,1); % Vector of extremum frequencies
d=(pi-ws)/ns;    % Spacing between extrema
w(1)=ws;
for j=2:ns+1,
    w(j)=w(j-1)+d; % Equally spaced extremum frequencies
end;

h=zeros(n,1);

% main loop
```

```

% -----
iter=0;
OldDelta=1; delta=2;
while (iter<maxiter) & (abs(delta-OldDelta)>Threshold),
    ineqm=formineqm(ns,m,w); % initialize A
    x=lp(f,ineqm,b); % solve linear programming
    OldDelta=delta;
    delta=x(ns+1);
    h((n+1)/2)=1/m; % form impulse response
    for j=1:ns,
        h(j+fix((j-1)/(m-1))+(n+1)/2)=x(j);
    end;
    h(1:(n-1)/2)=h(n:-1:(n+3)/2);
    [d wd]=der(ws,h,6*n);
    zc=[];
    lzc=0;
    for j=1:length(d)-1,
        if d(j)*d(j+1)<=0
            lzc=lzc+1;
            zc(lzc)=(wd(j)+wd(j+1))/2;
        end;
    end;
    zc=sort(zc);
    w(2:ns)=zc(1:lzc);
    clc;
    iter=iter+1
    delta
end;

freqplot(h,m,npoints,0,pi,fsamp);
-----
function A=formineqm(nstop,up,we)

% A=formineqm(nstop,up,we)
% Forms the A matrix for the AX<b equation. nstop is the number of the
% stopband frequencies, up is the upsampling factor and we is the extremum
% frequency vector.

A=zeros(nstop+1); % Inequality matrix

```

```

for j=1:2:nstop+1-rem(nstop,2),
    A(j,nstop+1)=-1;
end;

for r=1:nstop+1,
    for p=1:nstop,
        A(r,p)=2*cos(we(r)*(p+fix((p-1)/(up-1))));
    end;
end;

for j=2:2:nstop+rem(nstop,2),
    A(j,:)=-A(j,:);
end;
-----
function [dv, wd]=der(ws,h,np)
% Calculates the derivative of the frequency response in the stopband region.
% ws is the stopband edge frequency, h is the impulse response, and np is
% the number of points used to calculate the derivative.
wd=zeros(np,1);
d=(pi-ws)/(np+1);
for j=1:np,
    wd(j)=ws+d*j;
end;

N=length(h);
dv=zeros(np,1);
for j=1:np,
    for p=1:(N-1)/2,
        dv(j)=dv(j)+p*h(p+(N+1)/2)*sin(wd(j)*p);
    end;
end;
-----
% Spectral Factorization Routine
% -----
load filtdata;

hr=roots(h);
nr=length(hr);
h3=hr(3:2:101);

```

```

[ah3 ind]=sort(angle(h3));
h3=h3(ind);
h4=h3;

h1=zeros(25,0);
h2=zeros(25,0);

h1(1:19)=h3(13:2:49);
h2(1:19)=h3(14:2:50);

h3=h3(1:12);

h1(20)=h3(2) ; h1(21)=h3(4);
h1(22)=h3(5) ; h1(23)=h3(8);
h1(24)=h3(10) ; h1(25)=h3(12);

h2(20)=h3(1) ; h2(21)=h3(3);
h2(22)=h3(6) ; h2(23)=h3(7);
h2(24)=h3(9) ; h2(25)=h3(11);

hh=[1 1];
for j=1:length(h1),
    sot=[1 -2*real(h1(j)) abs(h1(j))^2];
    hh=conv(hh,sot);
end;

tran=hh/sum(hh);
tran=tran';

hh=[1 1];
for j=1:length(h2),
    sot=[1 -2*real(h2(j)) abs(h2(j))^2];
    hh=conv(hh,sot);
end;

rec=hh/sum(hh);
rec=rec';

hc=conv(tran,rec);

```

```

figure(1);
ind=1:103;
plot(ind,h,'-',ind,hc,'x');
title('Original Nyquist Filter: - Transmitter/Receiver Cascaded: x');

f=(103.68e6)*0.5*(0:511)/512;
figure(2);
tm=20*log10(freqz(tran,1,512));
plot(f,tm);
title('Transmitter Frequency Response');
xlabel('Frequency (Hz)');
ylabel('Magnitude (dB)');

figure(3);
tm=20*log10(freqz(rec,1,512));
plot(f,tm);
xlabel('Frequency (Hz)');
title('Receiver Frequency Response');
xlabel('Frequency (Hz)');
ylabel('Magnitude (dB)');

% Passband Conversion
% -----
wm=2*pi*fmod/fsamp;
trani=zeros(length(tran),1);
tranq=zeros(length(tran),1);
for j=1:length(tran),
    trani(j)=tran(j)*cos(wm*(j-1));
    tranq(j)=tran(j)*sin(wm*(j-1));
end;

figure(4);
ti=20*log10(freqz(trani,1,512));
tq=20*log10(freqz(tranq,1,512));
plot(f,ti,'-',f,tq,'--');
xlabel('Frequency (Hz)');
title('Transmitter: I Channel(-) Q Channel(--)' );
xlabel('Frequency (Hz)');

```

```

ylabel('Magnitude (dB)');

figure(5);
n=1:length(tran);
plot(n,tranI,'-',n,tranQ,'--');
title('Transmitter:   I Channel(-)           Q Channel(--)'');

wm=2*pi*fmod/fsamp;
reci=zeros(length(rec),1);
recq=zeros(length(rec),1);
for j=1:length(rec),
    reci(j)=rec(j)*cos(wm*(j-1));
    recq(j)=rec(j)*sin(wm*(j-1));
end;

figure(6);
ti=20*log10(freqz(reci,1,512));
tq=20*log10(freqz(recq,1,512));
plot(f,ti,'-', f,tq,'--');
xlabel('Frequency (Hz)');
title('Receiver:   I Channel(-)           Q Channel(--)'');
xlabel('Frequency (Hz)');
ylabel('Magnitude (dB)');

figure(7);
n=1:length(rec);
plot(n,reci,'-',n,recq,'--');
title('Receiver:   I Channel(-)           Q Channel(--)'');
-----
function freqplot(h,m,np,w1,w2,fs)

w=zeros(np,1);
d=(w2-w1)/(np-1);
w(1)=w1;
for j=2:np,
    w(j)=d+w(j-1);
end;

N=length(h);

```

```

freq=ones(np,1)/m;
for j=1:np,
    for n=1:(N-1)/2,
        freq(j)=freq(j)+2*h(n+(N+1)/2)*cos(w(j)*n);
    end;
end;

for j=1:np,
    if abs(freq(j))<1e-5,
        freq(j)=1e-5;
    end;
end;

fr=20*log10(freq);
plot(w*fs/(2*pi),fr);

xlabel('Frequency (Hz)');
ylabel('Magnitude (dB)');

```