

CHAPTER 15

Finite Wordlength Effects

Tutorial Problems

1. MATLAB script:

```
% P1501: Binary representation conversion
clc; close all;
%% Part a: Decimal to Binary
xa_dec = [121;-48;53;-27;-347];
fprintf('Decimal\t Binary\n')
for ii = 1:length(xa_dec)
    if xa_dec(ii)<0
        xa_bin{ii} = ['1',dec2bin(abs(xa_dec(ii)))];
    else
        xa_bin{ii} = ['0',dec2bin(abs(xa_dec(ii)))];
    end
    fprintf('%d',xa_dec(ii))
    fprintf('\t%s\n',xa_bin{ii})
end
%% Part b: Binary to Decimal
xb_bin = {'1011011','10101','01001','00101','1100110'};
fprintf('Binary\t Decimal\n')
for jj = 1:length(xb_bin)
    temp = xb_bin{jj};
    if temp(1) == '0'
        xb_dec(jj) = bin2dec(temp(2:end));
    elseif temp(1) == '1'
        xb_dec(jj) = -bin2dec(temp(2:end));
    else
        error('Invalid binary string');
    end
end
```

```

        error('bad input')
    end
    fprintf('%s',xb_bin{jj})
    fprintf('\t%d\n',xb_dec(jj))
end

```

2. MATLAB script:

```

% P1502: Binary representation conversion
clc; close all;
xd = [0.12345;-0.54321;0.90645;0.45388623;-0.237649];
L = 8; N = length(xd);
xd_sign = sign(xd);
%% Part a: Sign-Magnitude Representation
xb_sm = zeros(N,L);
ind = (xd < 0);
xb_sm(ind,1) = 1;
xd_abs = abs(xd);
for ii = 1:L-1
    xb_sm(:,ii+1) = floor(2*xd_abs);
    xd_abs = 2*xd_abs - xb_sm(:,ii+1);
end
disp('Sign-Magnitude Representation is:')
xb_sm
%% Part b: Two's-Complement Representation
xb_tc = zeros(N,L);
ind = (xd < 0);
xd_temp = xd; xd_temp(ind) = 2 + xd_temp(ind);
xb_tc(:,1) = floor(xd_temp);
xd_temp = xd_temp - xb_tc(:,1);
for ii = 1:L-1
    xb_tc(:,ii+1) = floor(xd_temp*2);
    xd_temp = xd_temp*2 - xb_tc(:,ii+1);
end
disp('Two''s-Complement Representation is:')
xb_tc

```

3. (a) See plot below.

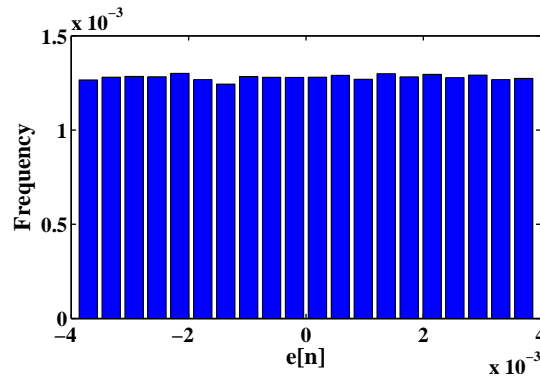


FIGURE 15.1: Plot of the histogram of $e[n]$ using 20 bins.

- (b) See plot below.

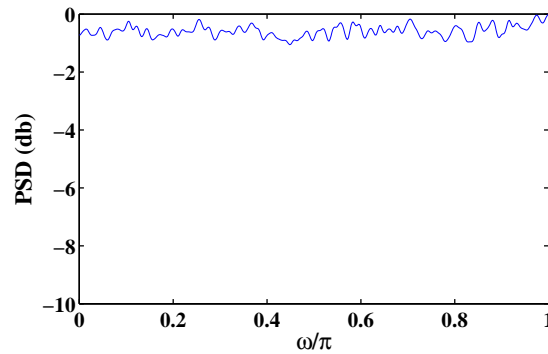


FIGURE 15.2: PSD of $e[n]$ using the `psdwelch` function for $B + 1 = 8$.

- (c) See plot below.

MATLAB script:

```
% P1503: Quantization noise distribution analysis
clc; close all;
N = 1e5;
n = 1:N;
xn = (sin(n/11) + sin(n/31) + cos(n/67))*0.33;
%% Part a
L = 8; % L = 2:2:16;
```

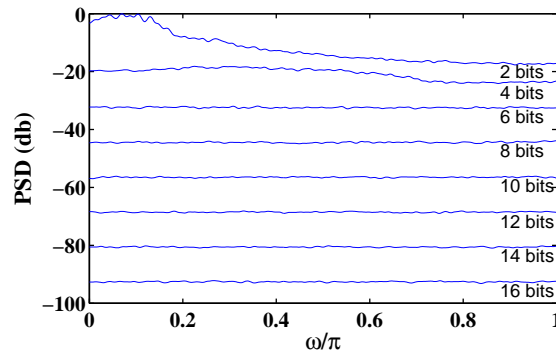


FIGURE 15.3: PSD of $e[n]$ using the `psdwelch` function for $B + 1 = 2, 4, 6, 8, 10, 12, 14$, and 16 .

```
[beq,E,B] = dec2beqR(xn,L);
en = beq - xn;
[eo pe] = epdf(en,20);
hfa = figconfg('P1503a','small');
bar(eo,pe/numel(en),'facecolor','b')
xlabel('e[n]','fontsize',LFS);
ylabel('Frequency','fontsize',LFS);
%% Part b
Se = psdwelch(en(:),250,1024);
w = linspace(0,1,512)*pi;
Se_mag = abs(Se); Se_db = 10*log10(Se_mag/max(Se_mag));
hfb = figconfg('P1503b','small');
plot(w/pi,Se_db); ylim([-10 0])
xlabel('\omega/\pi','fontsize',LFS);
ylabel('PSD (db)','fontsize',LFS);
%% Part c
L = 2:2:16;
hfc = figconfg('P1503c','small');
w = linspace(0,1,512)*pi;
for ii = 1:length(L)
    [beq,E,B] = dec2beqR(xn,L(ii));
    en = beq - xn;
    Se = psdwelch(en(:),250,1024);
    Se_mag = abs(Se);
    if ii == 1
```

```

        Se_max = max(Se_mag);
    end
    Se_db = 10*log10(Se_mag/max(Se_max));
    figure(hfc)
    plot(w/pi,Se_db); hold on
    text(w(450)/pi,Se_db(450)-3,[num2str(L(ii)),' bits'],'fontsize',LFS-4)
    xlabel('\omega/\pi','fontsize',LFS);
    ylabel('PSD (db)','fontsize',LFS);
end

```

4. (a) Solution:

$$f_{e_1}(x) = f_e(x) * f_e(x) = \int_{-\infty}^{\infty} f_e(y)f_e(x-y)dy$$

If $-\Delta \leq x \leq 0$,

$$f_{e_1}(x) = \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}+x} \frac{1}{\Delta^2} dy = \frac{1}{\Delta^2}(\Delta + x)$$

If $0 \leq x \leq \Delta$,

$$f_{e_1}(x) = \int_{x-\frac{\Delta}{2}}^{\frac{\Delta}{2}} \frac{1}{\Delta^2} dy = \frac{1}{\Delta^2}(\Delta - x)$$

Hence,

$$f_{e_1}(x) = \begin{cases} \frac{1}{\Delta^2}(\Delta - |x|), & |x| \leq \Delta \\ 0, & \text{otherwise} \end{cases}$$

(b) See plot below.

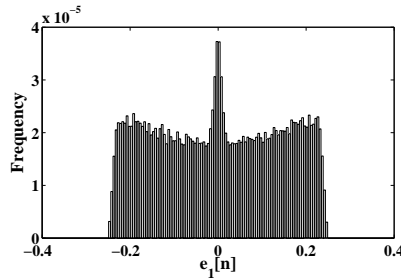
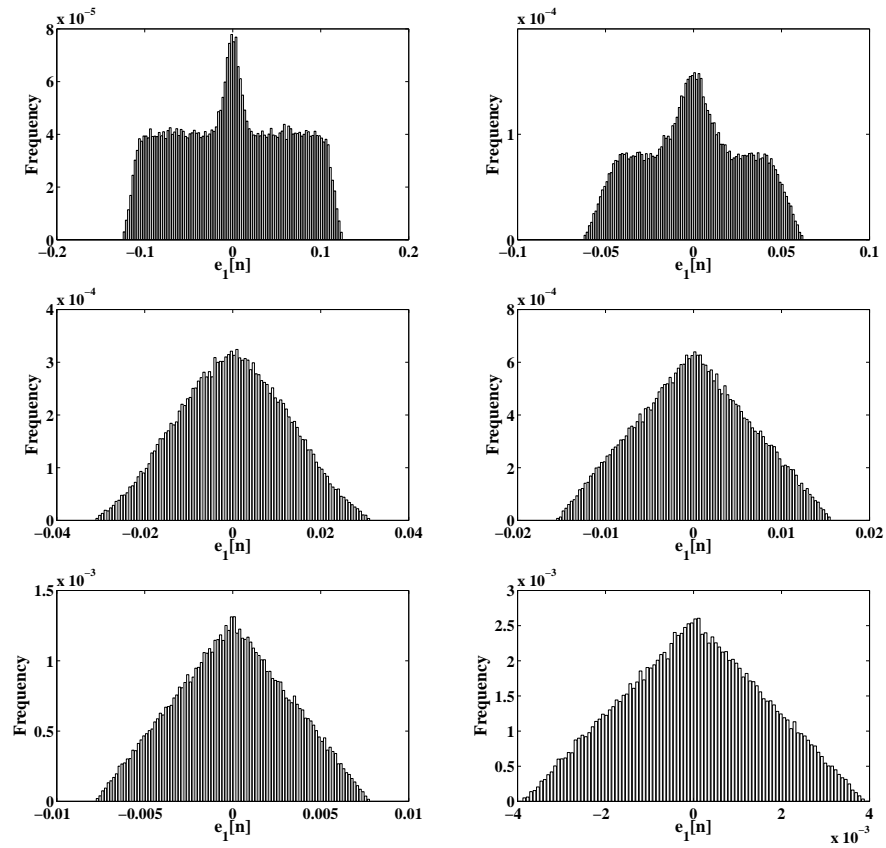


FIGURE 15.4: Plot of the histogram of $e_1[n]$ for $B + 1 = 2$ bits.

FIGURE 15.5: Plot of the histogram of $e_1[n]$ for $B + 1 = 3, 4, 5, 6, 7$, and 8 bits.

(c) See plots below.

MATLAB script:

```
% P1504: Quantization noise distribution analysis
clc; close all;
N = 1e5;
n = 1:N;
xn = (sin(n/11) + sin(n/31) + cos(n/67))*0.33;
%% Part b
L = 8; % L = 2:8;
[beq,E,B] = dec2beqR(xn,L);
en = beq - xn;
e1n = (en + [0 en(1:end-1)])/2;
```

```

[e1o pe1] = epdf(e1n,100);
hfa = figconfg('P1503a','small');
bar(e1o,pe1/numel(e1n),'facecolor','w')
xlabel('e_1[n]','fontsize',LFS);
ylabel('Frequency','fontsize',LFS);

```

5. (a) MATLAB function:

```

function [H,bins,eavg,evar] = QNmodel(x,B)
% Compute quantization error statistics of the input sequence
xm = abs(x(:));
E = max(max(0,fix(log2(xm+eps)+1))); % Integer bits
L = B + E + 1;
[beq,~,~] = dec2beqR(x,L);
en = beq - x;
[bins H] = epdf(en,50);
eavg = mean(en);
evar = var(en);

```

(b) See plot below.

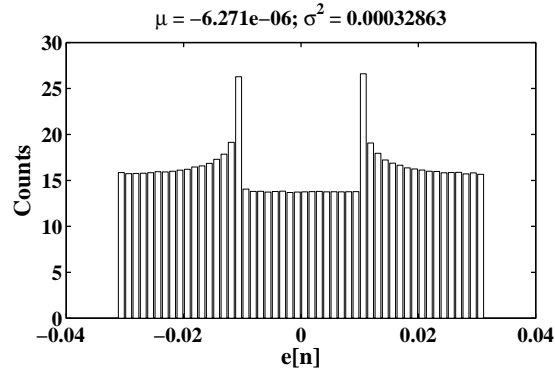


FIGURE 15.6: Plot of the histogram of the quantization error.

(c) tba.

MATLAB script:

```

% P1505: Quantization noise distribution analysis
clc; close all;
N = 1e5;
n = 1:N;

```

```

xn = 0.99*cos(n/17);
%% Part b
B = 4;
[pe, eo, eavg, evar] = QNmodel(xn, B);
hfa = figconfg('P1505a', 'small');
bar(eo, pe, 'facecolor', 'w');
xlabel('e[n]', 'fontsize', LFS);
ylabel('Counts', 'fontsize', LFS);
title(['\mu = ', num2str(eavg), ', \sigma^2 = ', num2str(evar)])

```

6. MATLAB script:

```

% P1506: Variance-Gain
clc; close all;
b1 = [1 3 4]; a1 = [1 0.6 0.08];
b2 = fliplr(b1)/b1(3); a2 = fliplr(a1)/a1(3);
K = b1(3)/a1(3);
[r p k] = residuez(conv(b1, b2), conv(a1, a2));
disp('VG is')
K*(k+r(3)+r(4))
%% Verification:
hn = filter(b1, a1, [1 zeros(1, 999)]);
sum(abs(hn).^2)

```

7. (a) Proof:

We first prove (15.35a). Given that

$$f(x) = \frac{1}{\sigma_x \sqrt{2\pi}} e^{-\frac{x^2}{2\sigma_x^2}}, \quad \Delta = \frac{\eta \sigma_x}{2^B}$$

The granular noise variance is

$$\begin{aligned}
 \sigma_g^2 &= \frac{\Delta^2}{12} \times 2 \int_0^{\eta \sigma_x} f(x) dx \stackrel{u=x/\sigma_x}{=} \frac{\Delta^2}{12} \times 2 \int_0^\eta f(u) du \\
 &= \frac{\Delta^2}{12} \times 2 \left(\int_0^\infty f(u) du - \int_\eta^\infty f(u) du \right) \\
 &= \left(\frac{\eta \sigma_x}{2^B} \right)^2 \frac{2}{12} \left[\frac{1}{2} - \Phi(\eta) \right] \\
 &= 2\sigma_x^2 \frac{\eta^2 2^{-2B}}{12} \left[\frac{1}{2} - \Phi(\eta) \right]
 \end{aligned}$$

We now prove (15.25b). The overload noise variance is

$$\begin{aligned}\sigma_o^2 &= 2 \left[\int_{\eta\sigma_x}^{\infty} x^2 f(x) dx - 2\eta\sigma_x \int_{\eta\sigma_x}^{\infty} x f(x) dx + \eta^2 \sigma_x^2 \int_{\eta\sigma_x}^{\infty} f(x) dx \right] \\ &= 2\sigma_x^2 \left[\int_{\eta}^{\infty} u^2 f(u) du - 2\eta \int_{\eta}^{\infty} u f(u) du + \eta^2 \int_{\eta}^{\infty} f(u) du \right]\end{aligned}$$

where

$$\begin{aligned}\int_{\eta}^{\infty} u^2 f(u) du &= \int_{\eta}^{\infty} u^2 \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}} du = - \int_{\eta}^{\infty} u \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}} d\left(-\frac{u^2}{2}\right) \\ &= - \left[u \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}} \right]_{\eta}^{\infty} - \int_{\eta}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}} du \\ &= \Phi(\eta) + \eta \frac{1}{\sqrt{2\pi}} e^{-\frac{\eta^2}{2}} \\ -2\eta \int_{\eta}^{\infty} u f(u) du &= -2\eta \int_{\eta}^{\infty} u \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}} du \\ &= 2\eta \int_{\eta}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}} d\left(-\frac{u^2}{2}\right) = 2\eta \left(\frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}} \right)_{\eta}^{\infty} \\ &= -2\eta \frac{1}{\sqrt{2\pi}} e^{-\frac{\eta^2}{2}} \\ \eta^2 \int_{\eta}^{\infty} f(u) du &= \eta^2 \Phi(\eta)\end{aligned}$$

Hence,

$$\sigma_o^2 = 2\sigma_x^2 \left[(\eta^2 + 1)\Phi(\eta) - \eta \frac{1}{\sqrt{2\pi}} e^{-\frac{\eta^2}{2}} \right]$$

(b) See plot below.

MATLAB script:

```
% P1507: SQNR
clc; close all;
L = 4:2:16; B = L - 1; B = B(:);
eta = logspace(-1,3,101);
sig2_g = bsxfun(@times,2.^(-2*B),eta.^2/6.*(-1/2+normcdf(eta,0,1)));
sig2_o = 2*(eta.^2+1).*(1-normcdf(eta,0,1))-2*eta.*normpdf(eta,0,1);
```

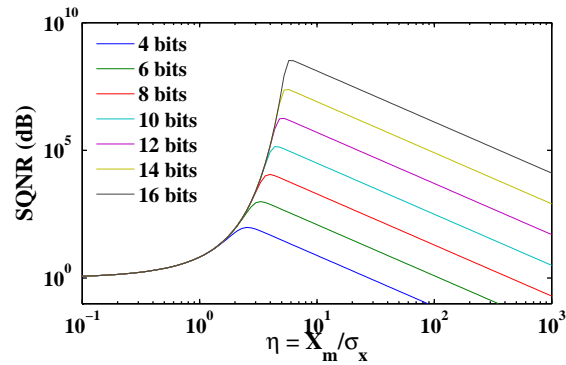


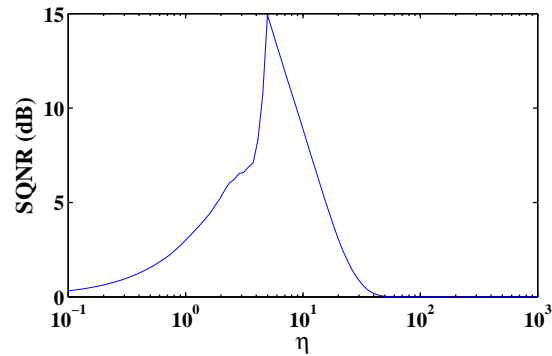
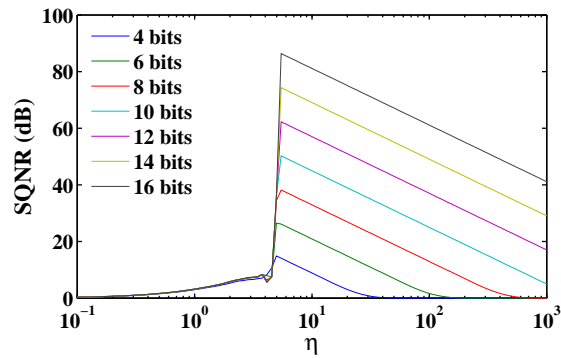
FIGURE 15.7: Plot of the noise variance as a function of η for $B + 1 = 4, 6, 8, 10, 12, 14, 16$.

```
SQNR = 1./bsxfun(@plus,sig2_g,sig2_o);
hfa = figconf('P1507a','small');
loglog(eta,SQNR)
xlabel('\eta = X_m/\sigma_x','fontsize',LFS);
ylabel('SQNR (dB)','fontsize',LFS);
ylim([10^(-1) 10^10])
hl = legend('4 bits','6 bits','8 bits','10 bits','12 bits','14 bits',...
    '16 bits','location','northwest');
set(hl,'box','off')
```

8. (a) Comments:
See script output.
- (b) See plot below.
- (c) See plot below.

MATLAB script:

```
% P1508: Quantizer performance investigation
clc; close all;
%% Part a
N = 1e5;
sigx2 = 1;
randn('seed',0)
xn = randn(N,1)*sqrt(sigx2);
L = 4;
x1 = -1; xL = 1-0.5^(L-1);
```

FIGURE 15.8: Plot of the noise variance as a function of η for $B + 1 = 4$.FIGURE 15.9: Plot of the noise variance as a function of η for $B + 1 = 6, 8, 10, 12, 14$, and 16 .

```

ind = (xn < xL & xn > x1);
xng = xn(ind);
[xqg,E,B] = dec2beqR(xng,L);
sigg2 = mean((xng-xqg).^2);
xno = xn(~ind);
xqo = -ones(size(xno));
ind2 = (xno > x1);
xqo(ind2) = xL;
sigo2 = mean((xno-xqo).^2);
SQNRa = sigx2/(sigg2+sigo2);
%% Part b
eta = logspace(-1,3,100); N2 = length(eta);

```

```

sigx = 1./eta; sigx2 = sigx.^2;
randn('seed',0)
X = bsxfun(@times,randn(N,1),sigx);
SQNRb = zeros(1,N2);
L = 4;
x1 = -1; xL = 1-0.5^(L-1);
for ii = 1:N2
    ind = (X(:,ii) < xL & X(:,ii) > x1);
    xng = X(ind,ii);
    [xqg,E,B] = dec2beqR(xng,L);
    sigg2 = mean((xng-xqg).^2);
    xno = X(~ind,ii);
    xqo = -ones(size(xno));
    ind2 = (xno > x1);
    xqo(ind2) = xL;
    sigo2 = mean((xno-xqo).^2);
    if isnan(sigo2)
        sigo2 = 0;
    end
    SQNRb(ii) = sigx2(ii)/(sigg2+sigo2);
end
SQNRb_db = 10*log10(SQNRb/min(SQNRb));
hfb = figconfig('P1508b','small');
semilogx(eta,SQNRb_db)
xlabel('\eta','fontsize',LFS);
ylabel('SQNR (dB)','fontsize',LFS);
%% Part c
eta = logspace(-1,3,100); N2 = length(eta);
sigx = 1./eta; sigx2 = sigx.^2;
randn('seed',0)
X = bsxfun(@times,randn(N,1),sigx);
L = 4:2:16; NL = length(L);
SQNRc = zeros(NL,N2);
x1 = -1; xL = 1-0.5^(L-1);
for jj = 1:NL
    for ii = 1:N2
        ind = (X(:,ii) < xL(jj) & X(:,ii) > x1);
        xng = X(ind,ii);
        [xqg,E,B] = dec2beqR(xng,L(jj));
        sigg2 = mean((xng-xqg).^2);
    end
end

```

```

xno = X(~ind,ii);
xqo = -ones(size(xno));
ind2 = (xno > x1);
xqo(ind2) = xL(jj);
sigo2 = mean((xno-xqo).^2);
if isnan(sigo2)
    sigo2 = 0;
end
SQNRc(jj,ii) = sigx2(ii)/(sigg2+sigo2);
end
end
SQNRc_db = 10*log10(SQNRc/min(SQNRc(:)));
hfc = figconf('P1508c','small');
semilogx(eta,SQNRc_db)
xlabel('\eta','fontsize',LFS);
ylabel('SQNR (dB)','fontsize',LFS);
hl = legend('4 bits','6 bits','8 bits','10 bits','12 bits','14 bits',...
    '16 bits','location','northwest');
set(hl,'box','off')

```

9. (a) Proof:

Since $|z| = 1$, we have $z = e^{j\omega}$,

$$dz = je^{j\omega}d\omega \implies d\omega = \frac{1}{j}z^{-1}dz$$

Hence, change the variable ω by z , we have

$$VG = \frac{1}{2\pi} \oint_{UC} H(z)H(z^{-1})\frac{1}{j}z^{-1}dz = \frac{1}{2\pi j} \oint_{UC} H(z)H(z^{-1})z^{-1}dz$$

Compared to the inverse z -transform formula, that is

$$x[n] = Z^{-1}\{X(z)\} = \frac{1}{2\pi j} \oint_{UC} X(z)z^{n-1}dz$$

we can prove (15.38) that is

$$VG = \frac{1}{2\pi j} \oint_{UC} H(z)H(z^{-1})z^{-1}dz = Z^{-1}[H(z)H(z^{-1})]|_{n=0}$$

(b) tba

10. Proof:

Consider Equation (15.59b), that is

$$S_{e_f}(e^{j\omega}) = |H_e(e^{j\omega})|^2 S_e(e^{j\omega}) = \sigma_e^2 [2 \sin(\omega/2)]^2$$

Applying inverse Fourier transform and set $n = 0$, we have the left hand side equals

$$E[e_f^2[n]]$$

The right hand side equals

$$\begin{aligned} \frac{\sigma_e^2}{2\pi} \int_{-\pi}^{\pi} [2 \sin(\omega/2)]^2 d\omega &= \frac{\sigma_e^2}{\pi} \int_{-\pi}^{\pi} (1 - \cos \omega) d\omega \\ &= \frac{\sigma_e^2}{\pi} (\omega - \sin \omega) \Big|_{-\pi}^{\pi} \\ &= \frac{\sigma_e^2}{\pi} [(\pi - 0) - (-\pi - 0)] \\ &= 2\sigma_e^2 = 2E[e^2[n]] \end{aligned}$$

11. tba

12. Proof:

$$S_{e_f}(e^{j\omega}) = |H_e(e^{j\omega})|^2 S_e(e^{j\omega}) = \sigma_e^2 [2 \sin(\omega/2)]^4$$

$$\begin{aligned} E[e_o^2[n]] &= \frac{\sigma_e^2}{2\pi} \int_{-\pi/D}^{\pi/D} [2 \sin(\omega/2)]^4 d\omega = \frac{\sigma_e^2}{2\pi} \int_{-\pi/D}^{\pi/D} (2 - 2 \cos \omega)^2 d\omega \\ &= \frac{4\sigma_e^2}{2\pi} \int_{-\pi/D}^{\pi/D} (1 - 2 \cos \omega + \cos^2 \omega) d\omega = \frac{4\sigma_e^2}{2\pi} \int_{-\pi/D}^{\pi/D} \left(1 - 2 \cos \omega + \frac{1 + \cos 2\omega}{2}\right) d\omega \\ &= \frac{4\sigma_e^2}{2\pi} \left[\frac{3\pi}{D} - 4 \sin \frac{\pi}{D} + \frac{1}{2} \sin \frac{2\pi}{D} \right] = \frac{4\sigma_e^2}{2\pi} \left[\frac{3\pi}{D} - 4 \sin \frac{\pi}{D} + \sin \frac{\pi}{D} \cos \frac{\pi}{D} \right] \end{aligned}$$

Note the approximations, that is

$$\sin \theta \approx \theta - \theta^3/6, \quad \cos \theta \approx 1 - \theta^2/2$$

Plug the above approximations into the equation, we have

$$E[e_o^2[n]] \approx \frac{4\sigma_e^2}{2\pi} \left[\frac{3\pi}{D} - 4\left(\frac{\pi}{D} - \frac{\pi^3}{6D^3}\right) + \left(\frac{\pi}{D} - \frac{\pi^3}{6D^3}\right)\left(1 - \frac{\pi^2}{D^2}\right) \right] = \frac{\sigma_e^2 \pi^4}{6D^5}$$

Hence, we conclude

$$\text{SQNR}_D = 10 \ln \frac{\sigma_x^2}{\sigma_e^2} + 10 \ln \frac{6}{\pi^4} + r \ln 2^5 = \text{SQNR}_{\text{NR}} - 12.10 + 15.05r$$

13. (a) Proof:

$$\frac{\frac{\partial D(z)}{\partial a_k} \Big|_{z=p_i}}{\frac{\partial D(z)}{\partial p_i} \Big|_{z=p_i}} = \frac{\partial D(z)}{\partial a_k} \Big|_{z=p_i} \frac{\partial p_i}{\partial D(z)} \Big|_{z=p_i} = \frac{\partial p_i}{\partial a_k}$$

(b) Proof:

$$\frac{\partial D(z)}{\partial a_k} \Big|_{z=p_i} = z^{-k} \Big|_{z=p_i} = p_i^{-k}$$

$$\begin{aligned} \frac{\partial D(z)}{\partial p_i} \Big|_{z=p_i} &= (-z^{-1}) \prod_{j=1, j \neq i}^N (1 - p_j z^{-1}) \Big|_{z=p_i} = (-p_i^{-1}) \prod_{j=1, j \neq i}^N (1 - p_j p_i^{-1}) \\ &= (-p_i^{-N}) \prod_{j=1, j \neq i}^N (p_i - p_j) \end{aligned}$$

Thus,

$$\frac{\partial p_i}{\partial a_k} = \frac{p_i^{-k}}{(-p_i^{-N}) \prod_{j=1, j \neq i}^N (p_i - p_j)} = -\frac{p_i^{N-k}}{\prod_{j=1, j \neq i}^N (p_i - p_j)}$$

Hence,

$$\Delta p_i = -\sum_{k=1}^N \frac{p_i^{N-k}}{\prod_{j=1, j \neq i}^N (p_i - p_j)} \Delta a_k$$

14. (a) See plots below.

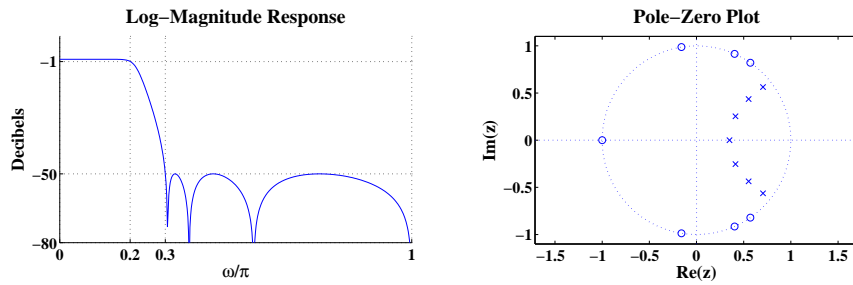


FIGURE 15.10: Plot of the magnitude response and pole-zero diagram of the filter.

(b) See plots below.

(c) See plots below.

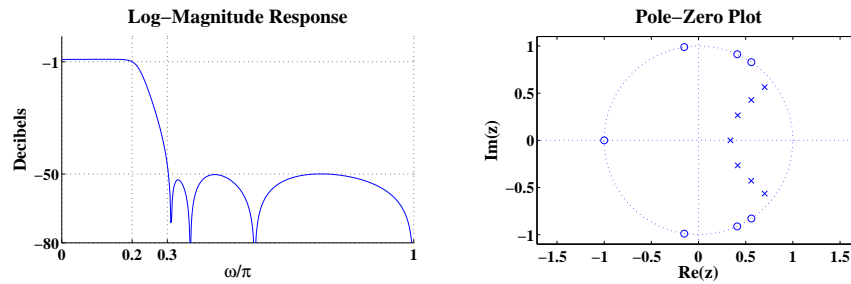


FIGURE 15.11: Plot of the magnitude response and pole-zero diagram of the filter when direct form coefficients are quantized to $L = 16$ bits.

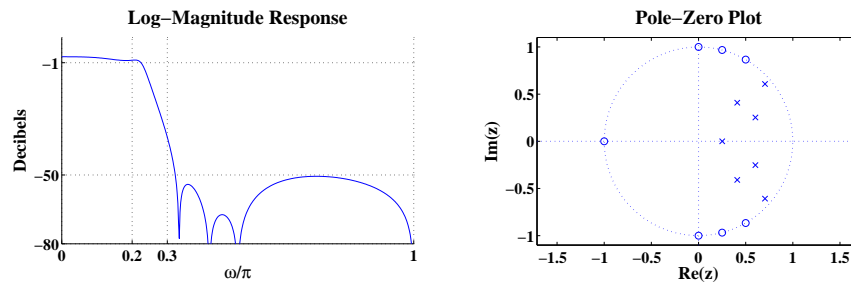


FIGURE 15.12: Plot of the magnitude response and pole-zero diagram of the filter when direct form coefficients are quantized to $L = 12$ bits.

(d) See plots below.

(e) tba.

MATLAB script:

```
% P1514: Digital lowpass filter design by Chebyshev II
close all; clc;
%% Filter Design
omegap = 0.2*pi; omegas = 0.3*pi; As = 50; Ap = 1; % Specification
[N,omegac] = cheb2ord(omegap/pi,omegas/pi,Ap,As); % Order define
[B,A] = cheby2(N,As,omegac); % coefficients
% [sos G] = tf2sos(B,A); % cascade form
%% Filter Coefficient Quantization of Direct Form
L = 16; % Part b
% L = 12; % Part c
% L = 8; % Part d L = 8
[BAhat,E1,B1] = dec2beqR([B;A],L);
```

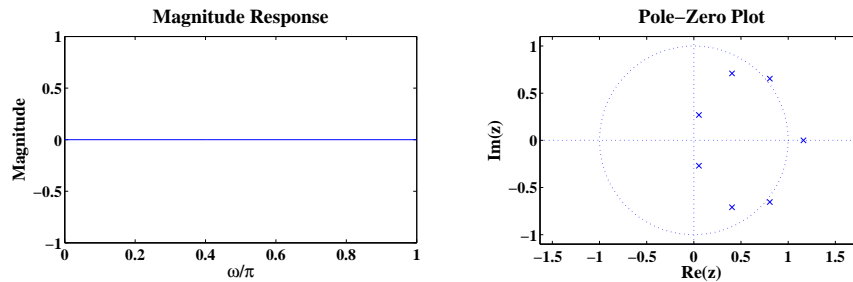



FIGURE 15.13: Plot of the magnitude response and pole-zero diagram of the filter when direct form coefficients are quantized to $L = 8$ bits.

```

Bhat = BAhat(1,:); Ahat = BAhat(2,:);
%% Plotting Parameters and Filter Responses
BP = B; AP = A; % Plot infinite precision
% BP = Bhat; AP = Ahat; % Plot finite precision
om = linspace(0,1,501)*pi;
H = freqz(BP,AP,om); Hmag = abs(H); Hdb = 20*log10(Hmag);
%% Design Plots
hfa = figconfg('P1514a','small'); % Log-Magnitude Response in dB
plot(om/pi,Hdb,'b','linewidth',1); axis([0,1,-80,10]);
xlabel('\omega/\pi','fontsize',LFS);
ylabel('Decibels','fontsize',LFS);
title('Log-Magnitude Response','fontsize',TFS);
set(gca,'xtick',[0,omegap,omegas,pi]/pi);
set(gca,'ytick',[-80,-As,-Ap]); grid; box off;

% hfa = figconfg('P1514a','small'); % Magnitude Response
% plot(om/pi,Hmag,'b','linewidth',1); axis([0,1,-1,1]);
% xlabel('\omega/\pi','fontsize',LFS);
% ylabel('Magnitude','fontsize',LFS);
% title('Magnitude Response','fontsize',TFS);

hfb = figconfg('P1514b','small'); % Pole-Zero Plot
zplane(BP,AP);
xlabel('Re(z)','fontsize',LFS);
ylabel('Im(z)','fontsize',LFS);
title('Pole-Zero Plot','fontsize',TFS);

```

15. (a) See plot below.

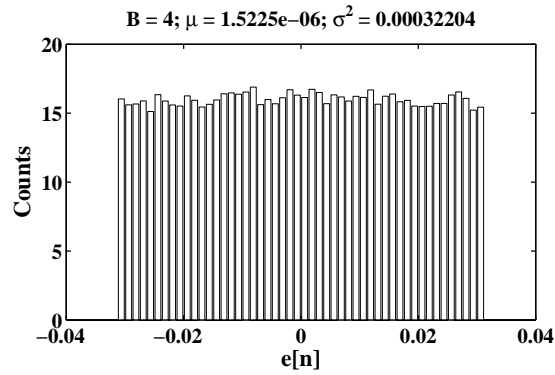


FIGURE 15.14: Plot of the histogram of the resulting error sequence when $ax[n]$ is quantized to $B = 4$.

(b) See plot below.

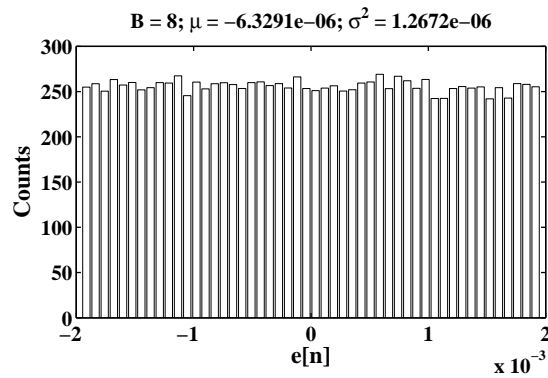


FIGURE 15.15: Plot of the histogram of the resulting error sequence when $ax[n]$ is quantized to $B = 8$.

(c) See plot below.

(d) tba.

MATLAB script:

```
% P1515: Quantization noise distribution analysis
clc; close all;
N = 1e5;
n = 1:N;
xn = (cos(n/11) + sin(n/17) + cos(n/31))/3;
```

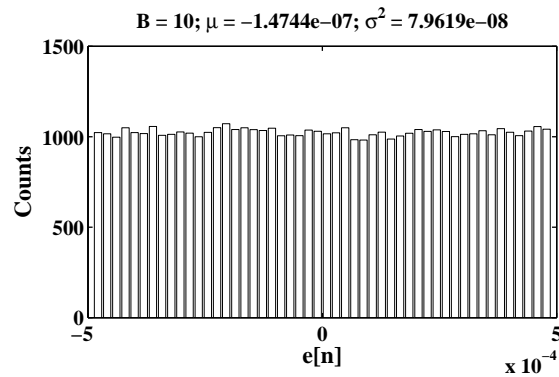


FIGURE 15.16: Plot of the histogram of the resulting error sequence when $ax[n]$ is quantized to $B = 10$.

```

a = 0.9375;
yn = a*xn;
B = 4; % Part a
% B = 8; % Part b
% B = 10; % Part c
[pe, eo, eavg, evar] = QNmodel(yn, B);
hfa = figconf('P1515a', 'small');
bar(eo, pe, 'facecolor', 'w');
xlabel('e[n]', 'fontsize', LFS);
ylabel('Counts', 'fontsize', LFS);
title(['B = ', num2str(B), '; \mu = ', num2str(eavg), ...
      ', \sigma^2 = ', num2str(evar)])

```

16. (a) Comments:
See script for detail.
- (b) See plot below.
- (c) See plot below.
- (d) Comments:
See the script output.

MATLAB script:

```

% P1516:
clc; close all;
a = -0.375; N = 1e5; n = 1:N; S = 1 - abs(a);

```

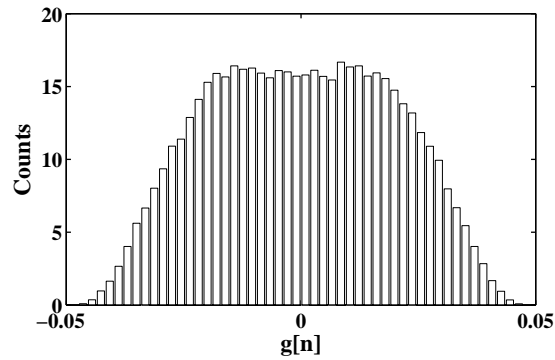


FIGURE 15.17: Plot of the histogram of the resulting error sequence $g[n]$ for $B = 4$.

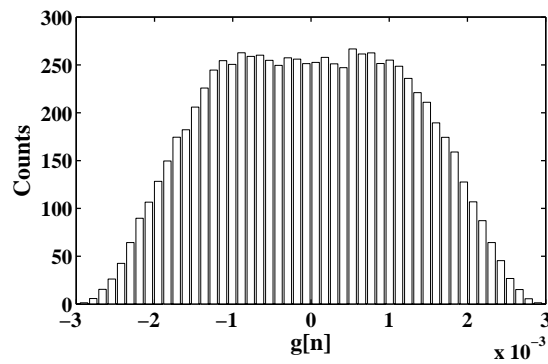


FIGURE 15.18: Plot of the histogram of the resulting error sequence $g[n]$ for $B = 8$.

```

B = 4; % Part a & b
% B = 8; % Part c
xn = cos(n/7);
[xnq,E1,B1] = dec2beqR(S*xn,B+1);
yn = filter(1,[1 a],xnq); % Part a
delta = 2^(-B);
rand('seed',0);
en = rand(1,N)*delta-delta/2;
ynhat = filter(1,[1 a],xnq+en);
gn = ynhat - yn;
[eo pe] = epdf(gn,50);

```

```

SNR = var(yn)/var(gn);
SNR_ref = 2^(2*B+2)*(1-abs(a))^2; % Part d
%% Plot
hfa = figconfg('P1516a','small');
bar(eo,pe,'facecolor','w')
xlabel('g[n]','fontsize',LFS);
ylabel('Counts','fontsize',LFS);

```

17. (a) Solution:

The output display oscillation and the amplitude is 0.125 and frequency is one sample.

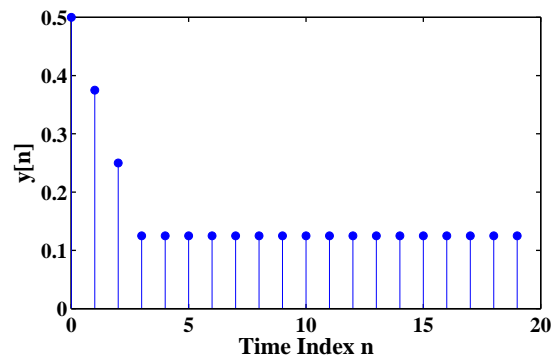


FIGURE 15.19: Plot of the first 20 samples of $y[n]$ when two's-complement overflow is used in the addition.

(b) Solution:

The output display oscillation and the amplitude is 0.125 and frequency is one sample.

MATLAB script:

```

% P1517:
clc; close all;
a0 = 0.5; a1 = 0.625; N = 20; B = 3;
yn = zeros(1,N);
yn(1) = a0;
%% Part a:
for ii = 2:20
    [yn(ii) E1 B1] = dec2beqR(yn(ii-1)*a1,B+1);
    if yn(ii) < -1 || yn(ii) >= 1

```

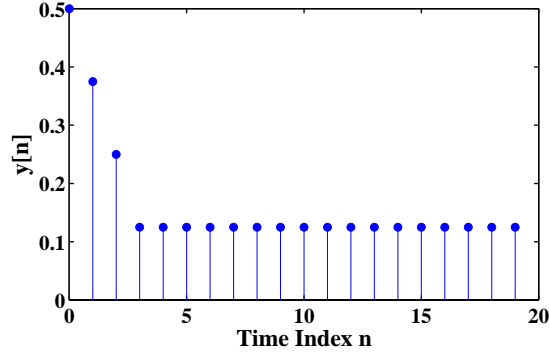


FIGURE 15.20: Plot of the first 20 samples of $y[n]$ when saturation characteristics is used in the addition.

```

        yn(ii) = -sign(yn(ii))*(abs(yn(ii))-1));
    end
end
%% Part b:
% for ii = 2:20
%     [yn(ii) E1 B1] = dec2beqR(yn(ii-1)*a1,B+1);
%     if yn(ii) < -1 || yn(ii) >= 1
%         yn(ii) = sign(yn(ii));
%     end
% end
%% Plot
hfa = figconfig('P1517a','small');
stem(0:N-1,yn,'filled')
xlabel('Time Index n','fontsize',LFS);
ylabel('y[n]','fontsize',LFS);

```

18. (a) Proof:

We first prove the sufficiency. Given

$$|x[n]| < \frac{1}{N}$$

we have

$$|X(k)| = \left| \sum_{n=0}^{N-1} x[n] W_N^{kn} \right| \leq \sum_{n=0}^{N-1} |x[n] W_N^{kn}| = \sum_{n=0}^{N-1} |x[n]| < \sum_{n=0}^{N-1} \frac{1}{N} = 1$$

We next prove the necessity. Given

$$|X(k)| < 1$$

If $|x[n]| = a \geq \frac{1}{N}$, we can assign that

$$x[n] = aW_N^{-kn}$$

Hence,

$$X(k) = Na \geq 1$$

which proves the necessity.

(b) Proof:

$$E[|X(k)|^2] = E\left[\frac{1}{N^2} \left| \sum_{n=0}^{N-1} x[n]W_N^{kn} \right|^2\right] = \frac{1}{N^2} E\left[\left| \sum_{n=0}^{N-1} x[n]W_N^{kn} \right|^2\right]$$

Hence, the signal power decreases by $1/N^2$, while the noise power remains the same, that proves that SQNR decreases by $1/N^2$.

19. Proof:

$$E[x[n]] = 0$$

$$\sigma_x^2 = E[x^2[n]] = \int_{-1/N}^{1/N} x^2 \frac{N}{2} dx = \frac{1}{3N^2}$$

$$E[|X(k)|^2] = E\left(\sum_{n=0}^{N-1} x[n]W_N^{nk} \sum_{m=0}^{N-1} x[m]W_N^{-mk}\right)$$

If $m \neq n$, we have

$$E[x[n]x[m]] = E[x[n]]E[x[m]] = 0$$

Hence,

$$E[|X(k)|^2] = E\left(\sum_{n=0}^{N-1} x^2[n]\right) = \sum_{n=0}^{N-1} E[x^2[n]] = N \frac{1}{3N^2} = \frac{1}{3N}$$

Basic Problems

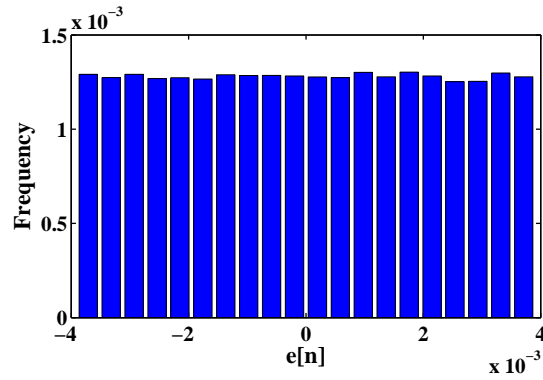
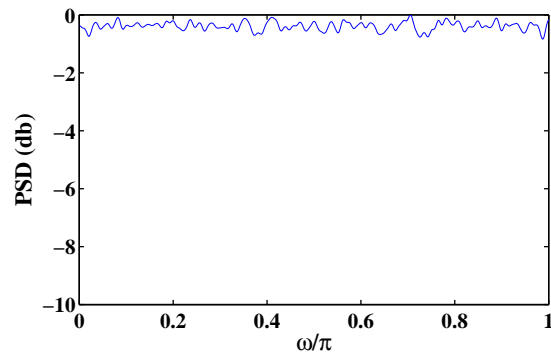
20. MATLAB script:

```
% P1520: Binary representation conversion
clc; close all;
xd = [0.12345;-0.54321;0.90645;0.45388623;-0.237649];
L = 10; N = length(xd);
xd_sign = sign(xd);
%% Part a: Sign-Magnitude Representation
xb_sm = zeros(N,L);
ind = (xd < 0);
xb_sm(ind,1) = 1;
xd_abs = abs(xd);
for ii = 1:L-1
    xb_sm(:,ii+1) = floor(2*xd_abs);
    xd_abs = 2*xd_abs - xb_sm(:,ii+1);
end
disp('Sign-Magnitude Representation is:')
xb_sm
%% Part b: Two's-Complement Representation
xb_tc = zeros(N,L);
ind = (xd < 0);
xd_temp = xd; xd_temp(ind) = 2 + xd_temp(ind);
xb_tc(:,1) = floor(xd_temp);
xd_temp = xd_temp - xb_tc(:,1);
for ii = 1:L-1
    xb_tc(:,ii+1) = floor(xd_temp*2);
    xd_temp = xd_temp*2 - xb_tc(:,ii+1);
end
disp('Two's-Complement Representation is:')
xb_tc
```

21. (a) See plot below.
 (b) See plot below.
 (c) See plot below.

MATLAB script:

```
% P1521: Quantization noise distribution analysis
clc; close all;
```


FIGURE 15.21: Plot of the histogram of $e[n]$ using 20 bins.FIGURE 15.22: PSD of $e[n]$ using the `psd welch` function for $B + 1 = 8$.

```

N = 1e5;
n = 1:N;
rand('seed',0)
xn = 2*rand(1,N)-1;
%% Part a
L = 8; % L = 2:2:16;
[beq,E,B] = dec2beqR(xn,L);
en = beq - xn;
[eo,pe] = epdf(en,20);
hfa = figconf('P1521a','small');
bar(eo,pe/numel(en),'facecolor','b')
xlabel('e[n]','fontsize',LFS);
ylabel('Frequency','fontsize',LFS);

```

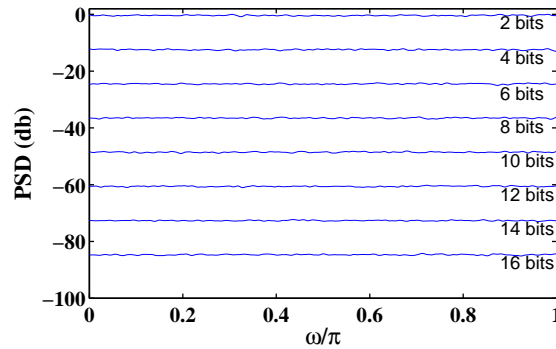


FIGURE 15.23: PSD of $e[n]$ using the `psdwelch` function for $B + 1 = 2, 4, 6, 8, 10, 12, 14$, and 16 .

```
%% Part b
Se = psdwelch(en(:),250,1024);
w = linspace(0,1,512)*pi;
Se_mag = abs(Se); Se_db = 10*log10(Se_mag/max(Se_mag));
hfb = figconf('P1521b','small');
plot(w/pi,Se_db); ylim([-10 0])
xlabel('\omega/\pi','fontsize',LFS);
ylabel('PSD (db)','fontsize',LFS);
%% Part c
L = 2:2:16;
hfc = figconf('P1521c','small');
w = linspace(0,1,512)*pi;
for ii = 1:length(L)
    [beq,E,B] = dec2beqR(xn,L(ii));
    en = beq - xn;
    Se = psdwelch(en(:),250,1024);
    Se_mag = abs(Se);
    if ii == 1
        Se_max = max(Se_mag);
    end
    Se_db = 10*log10(Se_mag/max(Se_max));
    figure(hfc)
    plot(w/pi,Se_db); hold on
    text(w(450)/pi,Se_db(450)-3,[num2str(L(ii)),' bits'],'fontsize',LFS-4)
    xlabel('\omega/\pi','fontsize',LFS);
```

```

        ylabel('PSD (db)', 'fontsize', LFS);
    end
    ylim([-100 2])

```

22. (a) Solution:

$$\begin{aligned}
 H(z) &= \frac{(z-1)(z+1)}{(z-re^{j\theta})(z-re^{-j\theta})} = \frac{z^2-1}{z^2-2r\cos\theta z+r^2} \\
 &= 1 + \frac{2r\cos\theta z - r^2 - 1}{z^2-2r\cos\theta z+r^2} \\
 &= R_0 + \frac{R_1}{z-re^{j\theta}} + \frac{R_2}{z-re^{-j\theta}}
 \end{aligned}$$

where

$$R_0 = 1, \quad R_1 = \frac{2r^2 \cos \theta e^{j\theta} - r^2 - 1}{2jr \sin \theta}, \quad R_2 = -\frac{2r^2 \cos \theta e^{-j\theta} - r^2 - 1}{2jr \sin \theta}$$

Hence,

$$VG = R_0^2 + \sum_{i=1}^2 \sum_{j=1}^2 \frac{R_i R_j^*}{1 - p_i p_j^*}$$

(b) Solution:

$$VG = 10.5263$$

(c) Repeat (b) for $r = 0.99$ and $\theta = \pi/2$.

Solution:

$$VG = 100.5025$$

(d) Comments:

See script output for details.

MATLAB script:

```

% P1522:
close all; clc;
%% Part b
r = 0.9; theta = pi/4;
%% Part c
% r = 0.99; theta = pi/2;

p1 = r*exp(1j*theta); p2 = r*exp(-1j*theta);
R0 = 1;

```

```

R1 = (2*r^2*cos(theta)*exp(1j*theta)-r^2-1)/2j/r/sin(theta);
R2 = -(2*r^2*cos(theta)*exp(-1j*theta)-r^2-1)/2j/r/sin(theta);

R = [R1;R2]; P = [p1;p2];
RR = R*R'; PP = P*P';
VG = R0^2 + sum(RR(:)./(1-PP(:)));

%% Part d
hn = filter([1 0 -1],[1 -2*r*cos(theta) r^2],[1 zeros(1,1000)]);
VG_ref = sum(abs(hn).^2);

```

23. (a) Comments:
See script output.
- (b) See plot below.

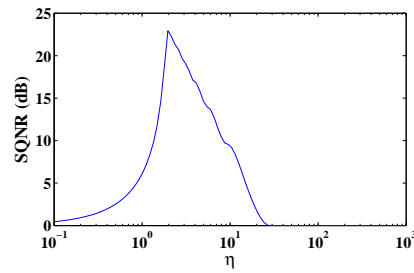


FIGURE 15.24: Plot of the noise variance as a function of η for $B + 1 = 4$ bits.

- (c) See plot below.

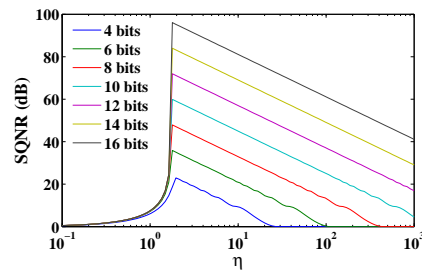


FIGURE 15.25: Plot of the noise variance as a function of η for $B + 1 = 6, 8, 10, 12, 14$, and 16 bits.

MATLAB script:

```

% P1523: Quantizer performance investigation
clc; close all;
%% Part a
N = 1e5;
A = sqrt(3);
sigx2 = A^2/3;
rand('seed',0)
xn = rand(N,1)*2*A-A;
L = 4;
x1 = -1; xL = 1-0.5^(L-1);
ind = (xn < xL & xn > x1);
xng = xn(ind);
[xqg,E,B] = dec2beqR(xng,L);
sigg2 = mean((xng-xqg).^2);
xno = xn(~ind);
xqo = -ones(size(xno));
ind2 = (xno > x1);
xqo(ind2) = xL;
sigo2 = mean((xno-xqo).^2);
SQNRa = sigx2/(sigg2+sigo2);
%% Part b
eta = logspace(-1,3,100); N2 = length(eta);
sigx = 1./eta; sigx2 = sigx.^2;
A = sqrt(3*sigx2);
randn('seed',0)
X = bsxfun(@times,rand(N,1),A);
SQNRb = zeros(1,N2);
L = 4;
x1 = -1; xL = 1-0.5^(L-1);
for ii = 1:N2
    ind = (X(:,ii) < xL & X(:,ii) > x1);
    xng = X(ind,ii);
    [xqg,E,B] = dec2beqR(xng,L);
    sigg2 = mean((xng-xqg).^2);
    xno = X(~ind,ii);
    xqo = -ones(size(xno));
    ind2 = (xno > x1);
    xqo(ind2) = xL;
    sigo2 = mean((xno-xqo).^2);
    if isnan(sigo2)

```

```

        sigo2 = 0;
    end
    SQNRb(ii) = sigx2(ii)/(sigg2+sigo2);
end
SQNRb_db = 10*log10(SQNRb/min(SQNRb));
hfb = figconfig('P1523b','small');
semilogx(eta,SQNRb_db)
xlabel('\eta','fontsize',LFS);
ylabel('SQNR (dB)','fontsize',LFS);
%% Part c
eta = logspace(-1,3,100); N2 = length(eta);
sigx = 1./eta; sigx2 = sigx.^2;
A = sqrt(3*sigx2);
randn('seed',0)
X = bsxfun(@times,rand(N,1),A);
L = 4:2:16; NL = length(L);
SQNRc = zeros(NL,N2);
x1 = -1; xL = 1-0.5.^(L-1);
for jj = 1:NL
    for ii = 1:N2
        ind = (X(:,ii) < xL(jj) & X(:,ii) > x1);
        xng = X(ind,ii);
        [xqg,E,B] = dec2beqR(xng,L(jj));
        sigg2 = mean((xng-xqg).^2);
        xno = X(~ind,ii);
        xqo = -ones(size(xno));
        ind2 = (xno > x1);
        xqo(ind2) = xL(jj);
        sigo2 = mean((xno-xqo).^2);
        if isnan(sigo2)
            sigo2 = 0;
        end
        SQNRc(jj,ii) = sigx2(ii)/(sigg2+sigo2);
    end
end
SQNRc_db = 10*log10(SQNRc/min(SQNRc(:)));
hfc = figconfig('P1523c','small');
semilogx(eta,SQNRc_db)
xlabel('\eta','fontsize',LFS);
ylabel('SQNR (dB)','fontsize',LFS);

```

```

hl = legend('4 bits','6 bits','8 bits','10 bits','12 bits','14 bits',...
            '16 bits','location','northwest');
set(hl,'box','off')

```

24. Proof:

When only granular noise exists and there is no overloading noise, the error is distributed as $e[n] \sim U(-\frac{\Delta}{2}, \frac{\Delta}{2})$, thus, we have

$$\mu_e[n] = 0$$

$$\sigma_e^2 = \int_{-\Delta/2}^{\Delta/2} \frac{1}{\Delta} x^2 dx = \frac{\Delta^2}{12}$$

25. MATLAB script:

```

% P1525: Variance-Gain
clc; close all;
b1 = [1 -8 19 -12]; a1 = [1 -0.4 -0.2375 -0.0188];
b2 = fliplr(b1)/b1(4); a2 = fliplr(a1)/a1(4);
K = b1(4)/a1(4);
[r p k] = residuez(conv(b1,b2),conv(a1,a2));
disp('VG is')
K*(k+sum(r(4:6)))
%% Verification:
hn = filter(b1,a1,[1 zeros(1,999)]);
sum(abs(hn).^2)

```

26. (a) See plot below.

(b) See plot below.

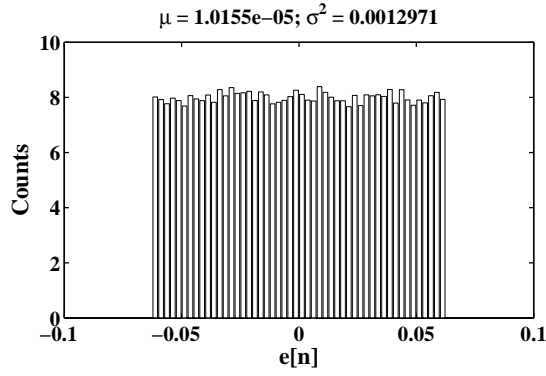
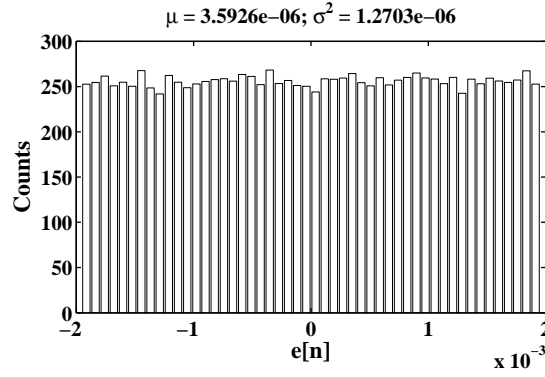
(c) tba.

MATLAB script:

```

% P1526: Quantization noise distribution analysis
clc; close all;
N = 1e5;
n = 1:N;
rand('seed',0)
xn = rand(1,N)*2-1;
B = 3; % Part a

```

FIGURE 15.26: Plot of the quantization histogram for $B = 3$ bits.FIGURE 15.27: Plot of the quantization histogram for $B = 8$ bits.

```
% B = 8; % Part b
[pe, eo, eavg, evar] = QNmodel(xn, B);
hfa = figconfig('P1526a', 'small');
bar(eo, pe, 'facecolor', 'w')
xlabel('e[n]', 'fontsize', LFS);
ylabel('Counts', 'fontsize', LFS);
title(['\mu = ', num2str(eavg), ', \sigma^2 = ', num2str(evar)])
```

27. Proof:

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} S_{x_d}(e^{j\omega}) d\omega = \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{S_x(e^{j\omega})}{D} d\omega = \frac{1}{2\pi} \int_{-\pi/D}^{\pi/D} S_x(e^{j\omega}) d\omega = \sigma_x^2$$

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{\sigma_e^2}{D} d\omega = \frac{1}{2\pi} \int_{-\pi/D}^{\pi/D} \sigma_e^2 d\omega = \frac{\sigma_e^2}{D}$$

28. Proof:

$$v[n] = -y[n-2]r \sin \theta + x[n] + v[n-1]r \cos \theta \quad (\text{P15.28a})$$

$$y[n] = y[n-1]r \cos \theta + v[n]r \sin \theta \quad (\text{P15.28b})$$

Solve (P15.28b) for $v[n]$, we have

$$v[n] = \frac{y[n] - y[n-1]r \cos \theta}{r \sin \theta}$$

Substitute $v[n]$ into (P15.28a) and after a few simple mathematical rearrangements, we have

$$y[n] - y[n-1]2r \cos \theta + y[n-2] = x[n]r \sin \theta$$

Hence, the poles of the above system is

$$p_1 = r \cos \theta, \quad p_2 = r \sin \theta$$

29. (a) Proof:

$$\frac{\frac{\partial C(z)}{\partial b_k} \Big|_{z=z_i}}{\frac{\partial C(z)}{\partial z_i} \Big|_{z=z_i}} = \frac{\partial C(z)}{\partial b_k} \Big|_{z=z_i} \frac{\partial z_i}{\partial C(z)} \Big|_{z=z_i} = \frac{\partial z_i}{\partial b_k}$$

(b) Proof:

$$\frac{\partial C(z)}{\partial b_k} \Big|_{z=z_i} = z^{-k} \Big|_{z=z_i} = z_i^{-k}$$

$$\begin{aligned} \frac{\partial z_i}{\partial C(z)} \Big|_{z=z_i} &= -z^{-1} \prod_{j=1, j \neq i}^M (1 - z_j z^{-1}) \Big|_{z=z_i} = -z_i^{-1} \prod_{j=1, j \neq i}^M (1 - z_j z_i^{-1}) \\ &= -z_i^{-M} \prod_{j=1, j \neq i}^M (z_i - z_j) \end{aligned}$$

$$\frac{\partial z_i}{\partial b_k} = \frac{z_i^{-k}}{-z_i^{-M} \prod_{j=1, j \neq i}^M (z_i - z_j)} = -\frac{z_i^{M-k}}{\prod_{j=1, j \neq i}^M (z_i - z_j)}$$

Hence, we conclude that

$$\Delta z_i = - \sum_{k=1}^M \frac{z_i^{M-k}}{\prod_{j=1, j \neq i}^M (z_i - z_j)} \Delta b_k$$

30. (a) Proof:

The frequency response of the filter is:

$$H(e^{j\omega}) = \frac{1 - \sqrt{3}e^{-j\omega}}{1 - 1/\sqrt{3}e^{-j\omega}}$$

The squared magnitude of the filter is

$$\begin{aligned} |H(e^{j\omega})|^2 &= H(e^{j\omega})H^*(e^{j\omega}) = \frac{1 - \sqrt{3}e^{-j\omega}}{1 - 1/\sqrt{3}e^{-j\omega}} \frac{1 - \sqrt{3}e^{j\omega}}{1 - 1/\sqrt{3}e^{j\omega}} \\ &= \frac{4 - 2\sqrt{3}\cos\omega}{4/3 - 2/\sqrt{3}\cos\omega} = 3 \end{aligned}$$

Hence, the magnitude of the filter is

$$|H(e^{j\omega})| = \sqrt{3}$$

which proves that the filter is an allpass filter.

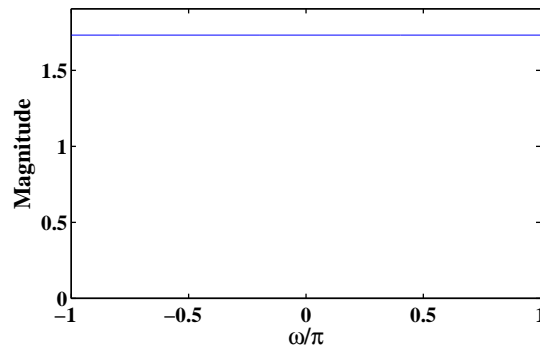


FIGURE 15.28: Plot of the magnitude response of the allpass filter.

(b) See plot below.

(c) See plot below.

MATLAB script:

```
% P1530: All-pass filter
clc; close all;
%% Part a
b = [1 -sqrt(3)]; a = [1 -1/sqrt(3)];
```

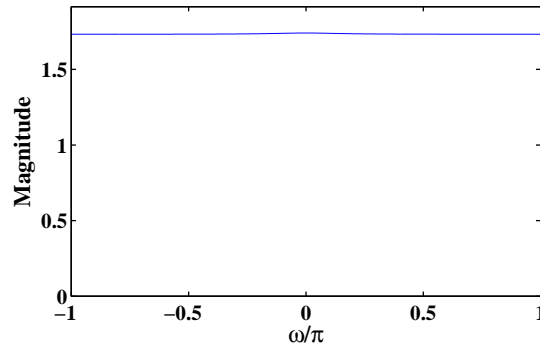


FIGURE 15.29: Plot of the magnitude response of the resulting filter by rounding the filter coefficients to $B = 8$ fraction bits.

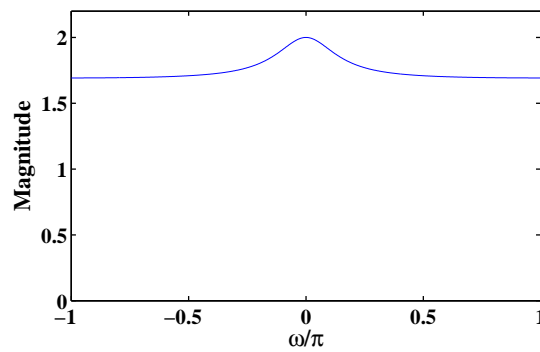


FIGURE 15.30: Plot of the magnitude response of the resulting filter by rounding the filter coefficients to $B = 4$ fraction bits.

```
w = linspace(-1,1,1001)*pi;
H = freqz(b,a,w); Hmag = abs(H);
hfa = figconf('P1530a','small');
plot(w/pi,Hmag)
ylim([0 1.1*max(Hmag)])
xlabel('\omega/\pi','fontsize',LFS);
ylabel('Magnitude','fontsize',LFS);
%% Part b & c
B = 7;
% B = 3;
Ea = max(max(0,fix(log2(abs(a(:))+eps)+1)));
```

```

Eb = max(max(0,fix(log2(abs(b(:))+eps)+1)));
[ar,Ear,Ba] = dec2beqR(a,B+Ea+1);
[br,Ebr,Bb] = dec2beqR(b,B+Eb+1);
Hr = freqz(br,ar,w); Hrmag = abs(Hr);
hfb = figconf('P1530b','small');
plot(w/pi,Hrmag)
ylim([0 1.1*max(Hrmag)])
xlabel('\omega/\pi','fontsize',LFS);
ylabel('Magnitude','fontsize',LFS);

```

31. (a) See plots below.

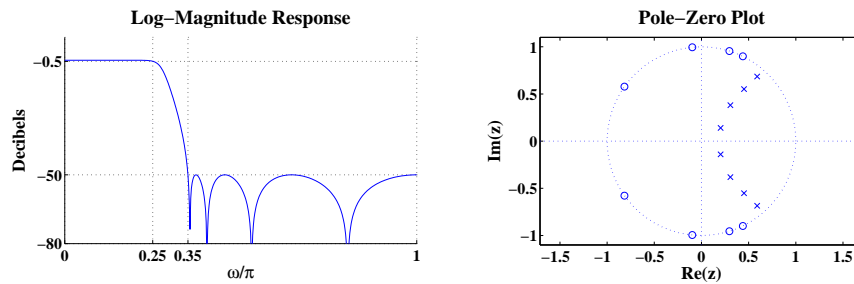


FIGURE 15.31: Plot of the magnitude response and pole-zero diagram of the filter.

(b) See plots below.

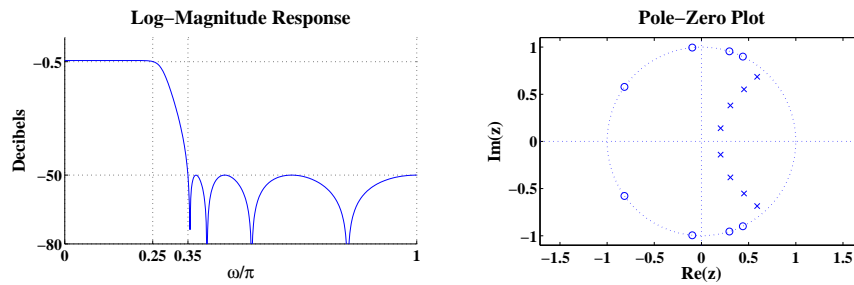


FIGURE 15.32: Plot of the magnitude response and pole-zero diagram of the filter after quantizing the cascade form coefficients to $L = 16$ bits.

(c) See plots below.

(d) See plots below.

(e) tba

MATLAB script:

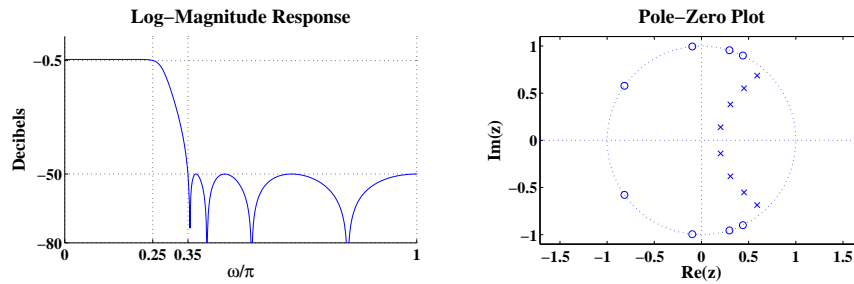


FIGURE 15.33: Plot of the magnitude response and pole-zero diagram of the filter after quantizing the cascade form coefficients to $L = 12$ bits.

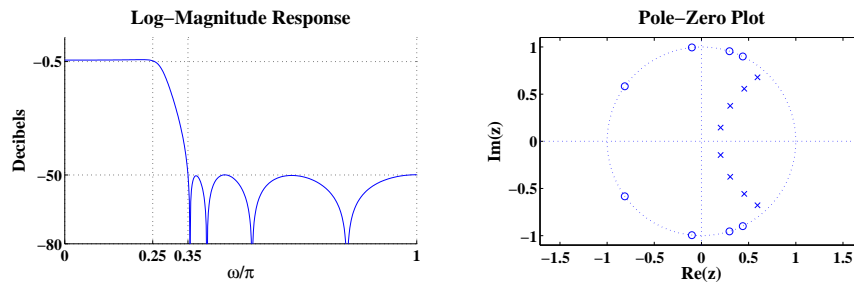


FIGURE 15.34: Plot of the magnitude response and pole-zero diagram of the filter after quantizing the cascade form coefficients to $L = 8$ bits.

```
% P1531: Digital lowpass filter design by Chebyshev II
close all; clc;
%% Filter Design
omegap = 0.25*pi; omegas = 0.35*pi; As = 50; Ap = 0.5; % Specification
[N,omegac] = cheb2ord(omegap/pi,omegas/pi,Ap,As); % Order define
[B,A] = cheby2(N,As,omegac); % coefficients
[sos G] = tf2sos(B,A); % cascade form
%% Filter Coefficient Quantization of Direct Form
L = 16; % Part b
% L = 12; % Part c
% L = 8; % Part d
[soshat,E2,B2] = dec2beqR(sos,L);
[Bhat,Ahat] = sos2tf(soshat,G);
%% Plotting Parameters and Filter Responses
BP = B; AP = A; % Plot infinite precision
% BP = Bhat; AP = Ahat; % Plot finite precision
```

```

om = linspace(0,1,501)*pi;
H = freqz(BP,AP,om); Hmag = abs(H); Hdb = 20*log10(Hmag);
%% Design Plots
hfa = figconf('P1531a','small'); % Log-Magnitude Response in dB
plot(om/pi,Hdb,'b','linewidth',1); axis([0,1,-80,10]);
xlabel('\omega/\pi','fontsize',LFS);
ylabel('Decibels','fontsize',LFS);
title('Log-Magnitude Response','fontsize',TFS);
set(gca,'xtick',[0,omegap,omegas,pi]/pi);
set(gca,'ytick',[-80,-As,-Ap]); grid; box off;

hfb = figconf('P1531b','small'); % Pole-Zero Plot
zplane(BP,AP);
xlabel('Re(z)','fontsize',LFS);
ylabel('Im(z)','fontsize',LFS);
title('Pole-Zero Plot','fontsize',TFS);

```

32. (a) See plots below.

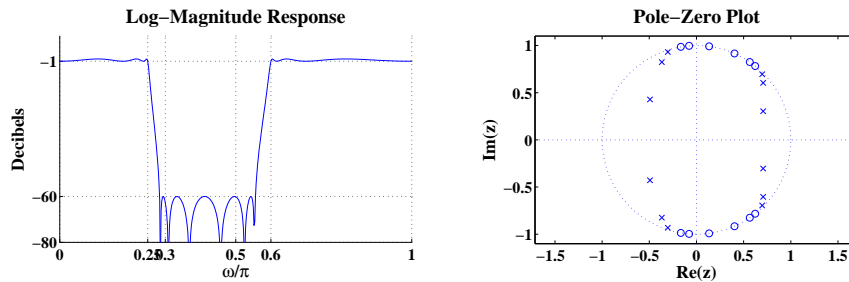


FIGURE 15.35: Plot of the magnitude response and pole-zero diagram of the filter.

(b) See plots below.

(c) See plots below.

(d) See plots below.

(e) tba.

MATLAB script:

```

% P1532: Digital bandstop filter design by elliptic
close all; clc;
%% Filter Design
omegap1 = 0.25*pi; omegas1 = 0.3*pi; omegas2 = 0.5*pi; omegas2 = 0.6*pi;

```

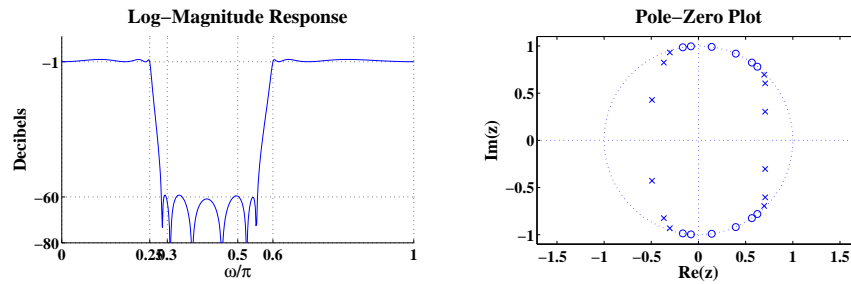


FIGURE 15.36: Plot of the magnitude response and pole-zero diagram of the filter after quantizing the direct form coefficients to $L = 16$ bits.

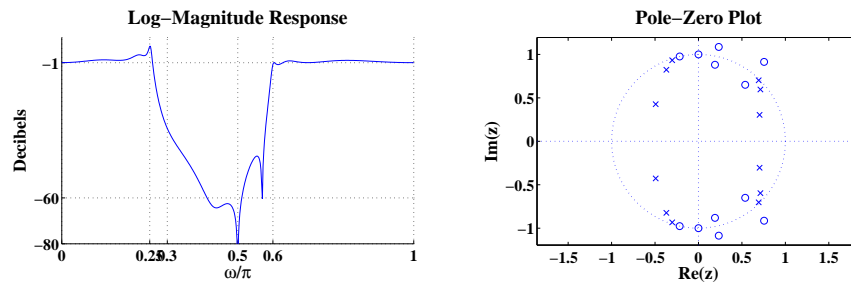


FIGURE 15.37: Plot of the magnitude response and pole-zero diagram of the filter after quantizing the direct form coefficients to $L = 10$ bits.

```

Ap = 1; As = 60; % Specification
omegap = [omegap1 omegap2]; omegas = [omegas1 omegas2];
[N,omegac] = ellipord(omegap/pi,omegas/pi,Ap,As); % Order define
[B,A] = ellip(N,Ap,As,omegac,'stop'); % coefficients
[sos G] = tf2sos(B,A); % cascade form
%% Filter Coefficient Quantization
L = 16; % Part b
% L = 10; % Part c & d
% Direct form Part b & c
% [BAhat,E1,B1] = dec2beqR([B;A],L);
% Bhat = BAhat(1,:); Ahat = BAhat(2,:);
% Cascade form Part d
[soshat,E2,B2] = dec2beqR(sos,L);
[Bhat,Ahat] = sos2tf(soshat,G);
%% Plotting Parameters and Filter Responses
BP = B; AP = A; % Plot infinite precision

```

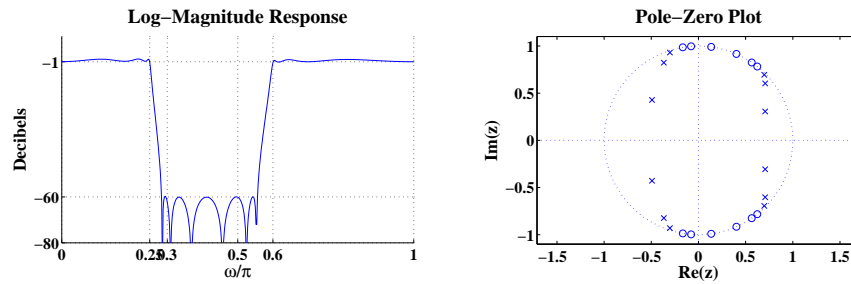


FIGURE 15.38: Plot of the magnitude response and pole-zero diagram of the filter after quantizing the cascade form coefficients to $L = 10$ bits.

```
% BP = Bhat; AP = Ahat; % Plot finite precision
om = linspace(0,1,501)*pi; H = freqz(BP,AP,om);
Hmag = abs(H); Hdb = 20*log10(Hmag);
%% Design Plots
hfa = figconf('P1532a','small'); % Log-Magnitude Response in dB
plot(om/pi,Hdb,'b','linewidth',1); axis([0,1,-80,10]);
xlabel('\omega/\pi','fontsize',LFS);
ylabel('Decibels','fontsize',LFS);
title('Log-Magnitude Response','fontsize',TFS);
set(gca,'xtick',[0,omegap1,omegas1,omegas2,omegap2,pi]/pi);
set(gca,'ytick',[-80,-As,-Ap]); grid; box off;

hfb = figconf('P1532b','small'); % Pole-Zero Plot
zplane(BP,AP);
xlabel('Re(z)','fontsize',LFS);
ylabel('Im(z)','fontsize',LFS);
title('Pole-Zero Plot','fontsize',TFS);
```

33. (a) See plots below.
 (b) See plots below.
 (c) See plots below.
 (d) See plots below.
 (e) tba

MATLAB script:

```
% P1533: Digital highpass filter design by Butterworth
close all; clc;
```

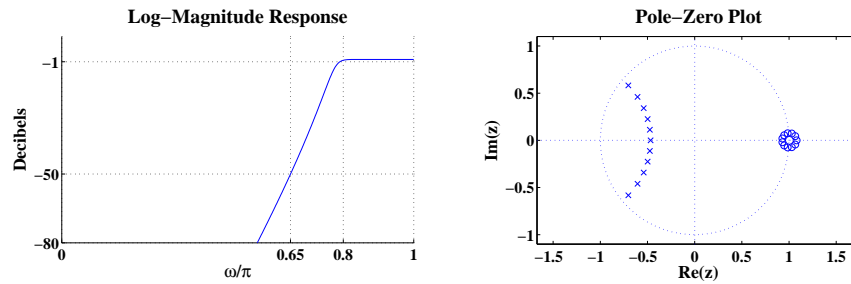
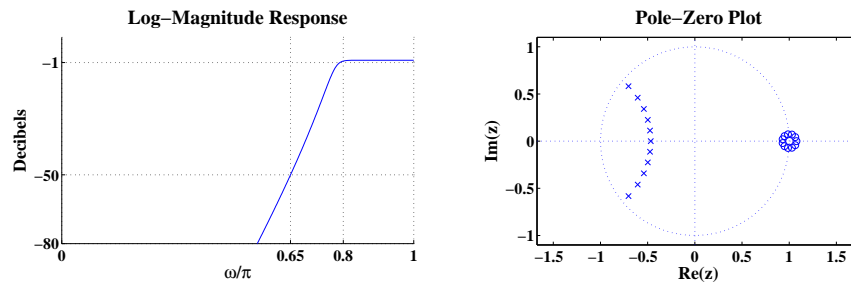



FIGURE 15.39: Plot of the magnitude response and pole-zero diagram of the filter.

FIGURE 15.40: Plot of the magnitude response and pole-zero diagram of the filter after quantizing the cascade form coefficients to $L = 16$ bits.

```

%% Filter Design
omegas = 0.65*pi; omegap = 0.8*pi; As = 50; Ap = 1; % Specification
[N,omegac] = buttord(omegap/pi,omegas/pi,Ap,As); % Order define
[B,A] = butter(N,omegac,'high'); % coefficients
[sos G] = tf2sos(B,A); % cascade form
%% Filter Coefficient Quantization of Direct Form
L = 16; % Part b
% L = 12; % Part c
% L = 8; % Part d
[soshat,E2,B2] = dec2beqR(sos,L);
[Bhat,Ahat] = sos2tf(soshat,G);
%% Plotting Parameters and Filter Responses
BP = B; AP = A; % Plot infinite precision
% BP = Bhat; AP = Ahat; % Plot finite precision
om = linspace(0,1,501)*pi;
H = freqz(BP,AP,om); Hmag = abs(H); Hdb = 20*log10(Hmag);
%% Design Plots

```

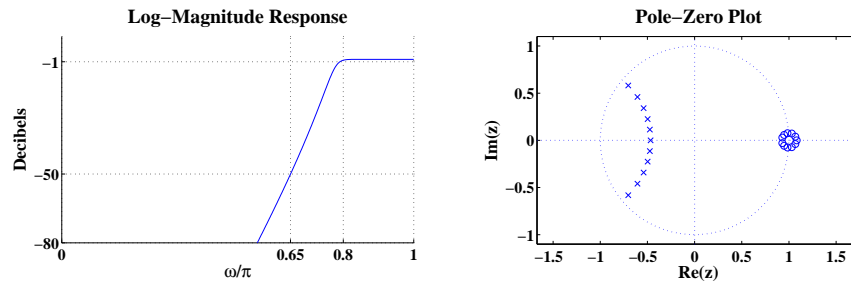


FIGURE 15.41: Plot of the magnitude response and pole-zero diagram of the filter after quantizing the cascade form coefficients to $L = 12$ bits.

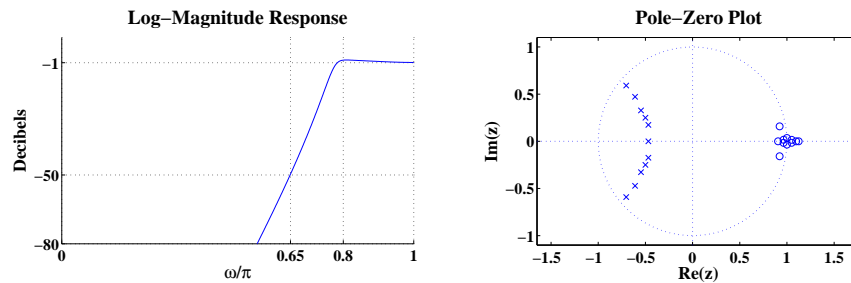


FIGURE 15.42: Plot of the magnitude response and pole-zero diagram of the filter after quantizing the cascade form coefficients to $L = 8$ bits.

```
hfa = figconf('P1533a','small'); % Log-Magnitude Response in dB
plot(om/pi,Hdb,'b','linewidth',1); axis([0,1,-80,10]);
xlabel('\omega/\pi','fontsize',LFS);
ylabel('Decibels','fontsize',LFS);
title('Log-Magnitude Response','fontsize',TFS);
set(gca,'xtick',[0,omegas,omegap,pi]/pi);
set(gca,'ytick',[-80,-As,-Ap]); grid; box off;

hfb = figconf('P1533b','small'); % Pole-Zero Plot
zplane(BP,AP);
xlabel('Re(z)','fontsize',LFS);
ylabel('Im(z)','fontsize',LFS);
title('Pole-Zero Plot','fontsize',TFS);
```

34. (a) See plots below.

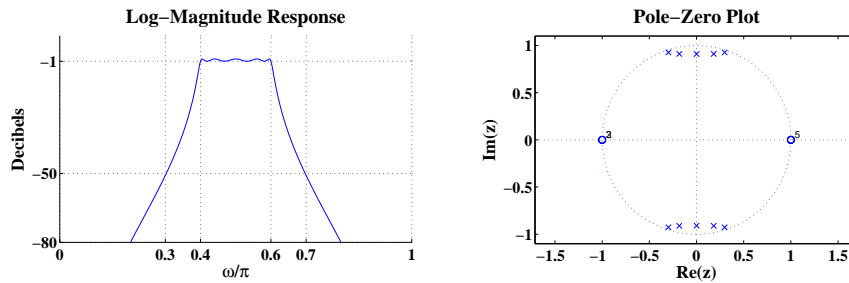


FIGURE 15.43: Plot of the magnitude response and pole-zero diagram of the filter.

(b) See plots below.

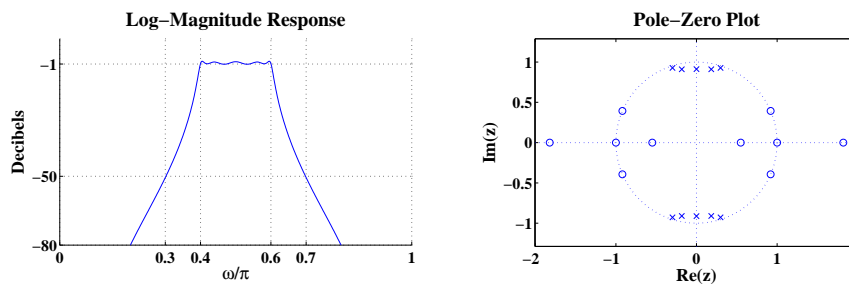


FIGURE 15.44: Plot of the magnitude response and pole-zero diagram of the filter after quantizing the direct form coefficients to $L = 16$ bits.

(c) See plots below.

(d) See plots below.

(e) tba.

MATLAB script:

```
% P1534: Digital bandpass filter design by Chebyshev I
close all; clc;
%% Filter Design
omegas1 = 0.3*pi; omegap1 = 0.4*pi; omegap2 = 0.6*pi; omegas2 = 0.7*pi;
Ap = 1; As = 50; % Specification
omegap = [omegap1 omegap2]; omegas = [omegas1 omegas2];
[N,omegac] = cheb1ord(omegap/pi,omegas/pi,Ap,As); % Order define
[B,A] = cheby1(N,Ap,omegac); % coefficients
```

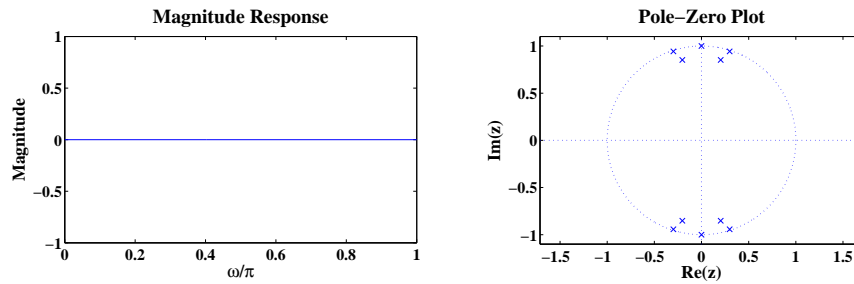


FIGURE 15.45: Plot of the magnitude response and pole-zero diagram of the filter after quantizing the direct form coefficients to $L = 10$ bits.

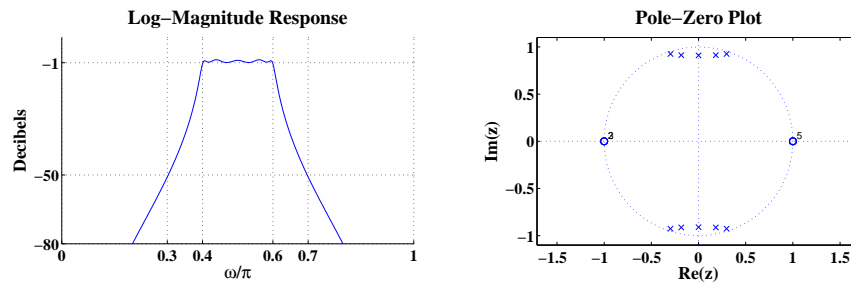


FIGURE 15.46: Plot of the magnitude response and pole-zero diagram of the filter after quantizing the cascade form coefficients to $L = 10$ bits.

```
[sos G] = tf2sos(B,A); % cascade form
%% Filter Coefficient Quantization
L = 16; % Part b
% L = 10; % Part c & d
% Direct form Part b & c
[BAhat,E1,B1] = dec2beqR([B;A],L);
Bhat = BAhat(1,:); Ahat = BAhat(2,:);
% Cascade form Part d
% [soshat,E2,B2] = dec2beqR(sos,L);
% [Bhat,Ahat] = sos2tf(soshat,G);
%% Plotting Parameters and Filter Responses
BP = B; AP = A; % Plot infinite precision
% BP = Bhat; AP = Ahat; % Plot finite precision
om = linspace(0,1,501)*pi; H = freqz(BP,AP,om);
Hmag = abs(H); Hdb = 20*log10(Hmag);
%% Design Plots
```

```

hfa = figconf('P1534a','small'); % Log-Magnitude Response in dB
plot(om/pi,Hdb,'b','linewidth',1); axis([0,1,-80,10]);
xlabel('\omega/\pi','fontsize',LFS);
ylabel('Decibels','fontsize',LFS);
title('Log-Magnitude Response','fontsize',TFS);
set(gca,'xtick',[0,omegas1,omegap1,omegap2,omegas2,pi]/pi);
set(gca,'ytick',[-80,-As,-Ap]); grid; box off;

% hfa = figconf('P1534a','small'); % Magnitude Response
% plot(om/pi,Hmag,'b','linewidth',1); axis([0,1,-1,1]);
% xlabel('\omega/\pi','fontsize',LFS);
% ylabel('Magnitude','fontsize',LFS);
% title('Magnitude Response','fontsize',TFS);

hfb = figconf('P1534b','small'); % Pole-Zero Plot
zplane(BP,AP);
xlabel('Re(z)','fontsize',LFS);
ylabel('Im(z)','fontsize',LFS);
title('Pole-Zero Plot','fontsize',TFS);

```

35. (a) See plot below.

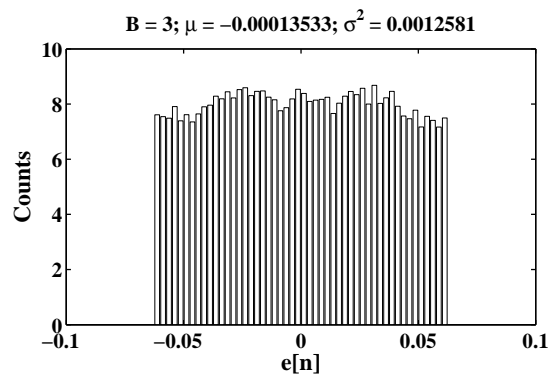


FIGURE 15.47: Plot of the histogram of the resulting error sequence when $ax[n]$ is quantized to $B = 3$.

(b) See plot below.

(c) See plot below.

(d) tba

MATLAB script:

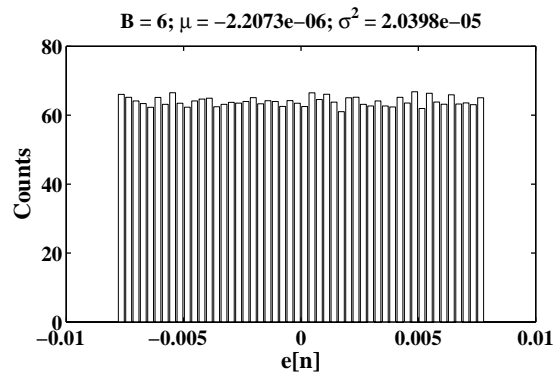


FIGURE 15.48: Plot of the histogram of the resulting error sequence when $ax[n]$ is quantized to $B = 6$.

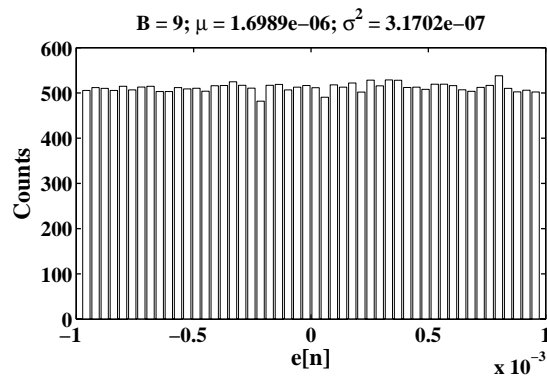


FIGURE 15.49: Plot of the histogram of the resulting error sequence when $ax[n]$ is quantized to $B = 9$.

```
% P1535: Quantization noise distribution analysis
clc; close all;
N = 1e5;
n = 1:N;
rand('seed',0)
xn = rand(1,N)*2-1;
a = 0.6667;
yn = a*xn;
B = 3; % Part a
% B = 6; % Part b
% B = 9; % Part c
```

```

[pe, eo, eavg, evar] = QNmodel(yn, B);
hfa = figconf('P1535a', 'small');
bar(eo, pe, 'facecolor', 'w')
xlabel('e[n]', 'fontsize', LFS);
ylabel('Counts', 'fontsize', LFS);
title(['B = ', num2str(B), ', \mu = ', num2str(eavg), ...
      ', \sigma^2 = ', num2str(evar)])

```

36. (a) Comments:
See script for detail.
- (b) See plot below.

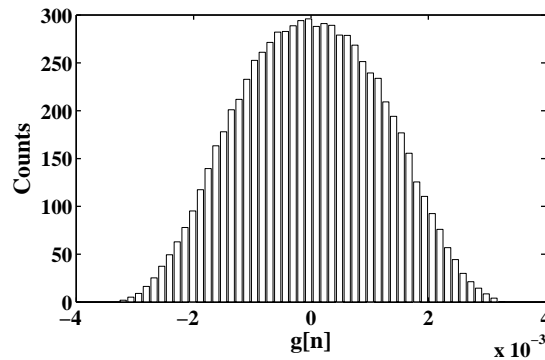


FIGURE 15.50: Plot of the histogram of the resulting error sequence $g[n]$ using all three B bit multipliers in the filter implementation.

- (c) See plot below.

MATLAB script:

```

% P1536:
clc; close all;
B = 8; N = 1e5; n = 1:N;
h = [0.25 0.5 0.25];
S = sqrt(1/sum(h.^2));
xn = cos(n/11);
[xnq, E1, B1] = dec2beqR(S*xn, B+1);
yn = filter(h, 1, xnq); % Part a
delta = sqrt(3)*2^(-B); % Part b
% delta = 2^(-B); % Part c
rand('seed', 0);

```

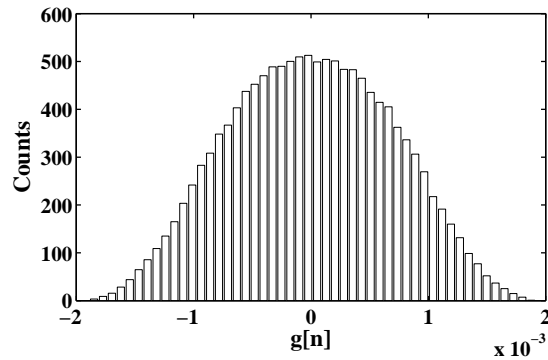


FIGURE 15.51: Plot of the histogram of the resulting error sequence $g[n]$ using only one B bit multiplier in the filter implementation.

```

en = rand(1,N)*delta-delta/2;
ynhat = filter(h,1,xnq+en);
gn = ynhat - yn;
[eo pe] = epdf(gn,50);
SNR = var(yn)/var(gn);
%% Plot
hfa = figconf('P1536a','small');
bar(eo,pe,'facecolor','w')
xlabel('g[n]','fontsize',LFS);
ylabel('Counts','fontsize',LFS);

```

37. (a) Solution:

The output display oscillation and the amplitude is 0.125 and frequency is one sample.

(b) Solution:

The output display oscillation and the amplitude is 0.125 and frequency is one sample.

MATLAB script:

```

% P1537:
clc; close all;
a0 = 0.75; a1 = 0.75; N = 20; B = 4;
yn = zeros(1,N);
yn(1) = a0;
%% Part a:

```

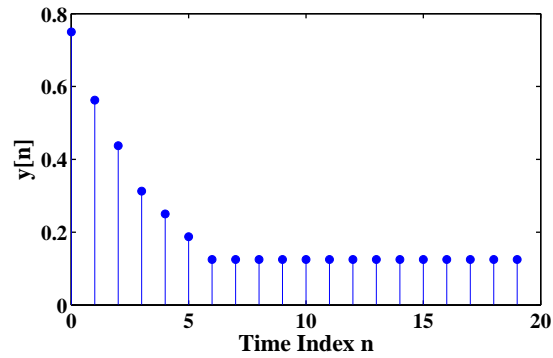



FIGURE 15.52: Plot of the first 20 samples of $y[n]$ when two's-complement overflow is used in the addition.

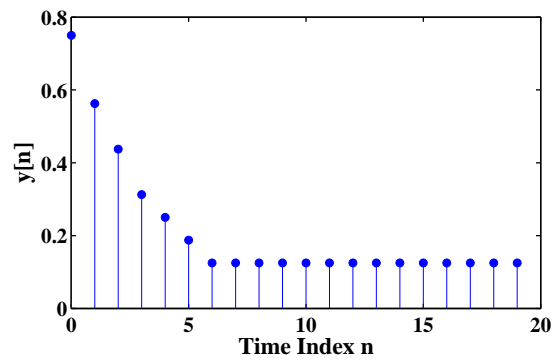


FIGURE 15.53: Plot of the first 20 samples of $y[n]$ when saturation characteristics are used in the addition.

```

for ii = 2:20
    [yn(ii) E1 B1] = dec2beqR(yn(ii-1)*a1,B+1);
    if yn(ii) < -1 || yn(ii) >= 1
        yn(ii) = -sign(yn(ii))*(abs(yn(ii))-1);
    end
end
%% Part b:
% for ii = 2:20
%     [yn(ii) E1 B1] = dec2beqR(yn(ii-1)*a1,B+1);
%     if yn(ii) < -1 || yn(ii) >= 1
%         yn(ii) = sign(yn(ii));
%
```

```
%      end
% end
%% Plot
hfa = figconfig('P1537a','small');
stem(0:N-1,yn,'filled')
xlabel('Time Index n','fontsize',LFS);
ylabel('y[n]','fontsize',LFS);
```

38. tba

Assessment Problems

39. MATLAB script:

```
% P1539: Binary representation conversion
clc; close all;
xd = [0.54321;-0.12345;0.54609;0.862338;-0.497623];
L = 8; N = length(xd);
xd_sign = sign(xd);
%% Part a: Sign-Magnitude Representation
xb_sm = zeros(N,L);
ind = (xd < 0);
xb_sm(ind,1) = 1;
xd_abs = abs(xd);
for ii = 1:L-1
    xb_sm(:,ii+1) = floor(2*xd_abs);
    xd_abs = 2*xd_abs - xb_sm(:,ii+1);
end
disp('Sign-Magnitude Representation is:')
xb_sm
%% Part b: Two's-Complement Representation
xb_tc = zeros(N,L);
ind = (xd < 0);
xd_temp = xd; xd_temp(ind) = 2 + xd_temp(ind);
xb_tc(:,1) = floor(xd_temp);
xd_temp = xd_temp - xb_tc(:,1);
for ii = 1:L-1
    xb_tc(:,ii+1) = floor(xd_temp*2);
    xd_temp = xd_temp*2 - xb_tc(:,ii+1);
end
disp('Two''s-Complement Representation is:')
xb_tc
```

40. (a) Solution:

$$\begin{aligned}
 H(z) &= \frac{(z-j)(z+j)}{(z-r)(z+r)} = \frac{z^2+1}{z^2-r^2} = 1 + \frac{r^2+1}{z^2-r^2} \\
 &= R_0 + \frac{R_1}{z-r} + \frac{R_2}{z+r}
 \end{aligned}$$

where

$$R_0 = 1, \quad R_1 = \frac{r^2+1}{2r}, \quad R_2 = -\frac{r^2+1}{2r}$$

Hence, the variance-gain is

$$\begin{aligned}
 \text{VG} &= R_0^2 + \sum_{i=1}^2 \sum_{j=1}^2 \frac{R_i R_j^*}{1 - p_i p_j^*} \\
 &= 1 + \frac{\left(\frac{r^2+1}{2r}\right)^2}{1-r^2} + \frac{-\left(\frac{r^2+1}{2r}\right)^2}{1+r^2} + \frac{-\left(\frac{r^2+1}{2r}\right)^2}{1+r^2} + \frac{\left(\frac{r^2+1}{2r}\right)^2}{1-r^2} \\
 &= \frac{2}{1-r^2}
 \end{aligned}$$

(b) Solution:

$$\text{VG} = 10.5263$$

(c) Solution:

$$\text{VG} = 100.5025$$

(d) Comments:

See script output for verification.

MATLAB script:

```

% P1540:
close all; clc;
%% Part b
r = 0.9;
%% Part c
% r = 0.99;

%% Computation:
p1 = r; p2 = -r;
VG = 2/(1-r^2);

%% Part d
hn = filter([1 0 1],[1 0 -r^2],[1 zeros(1,1000)]);
VG_ref = sum(abs(hn).^2);

```

41. (a) See plot below.

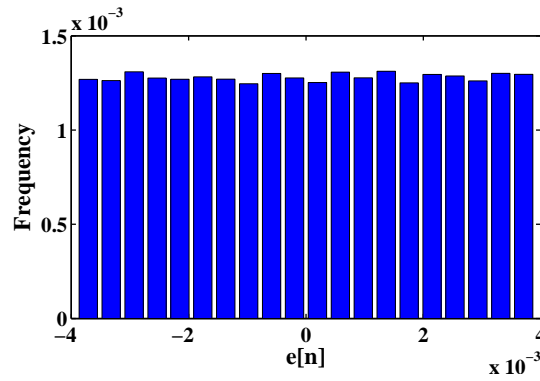


FIGURE 15.54: Plot of the histogram of $e[n]$ using 20 bins.

(b) See plot below.

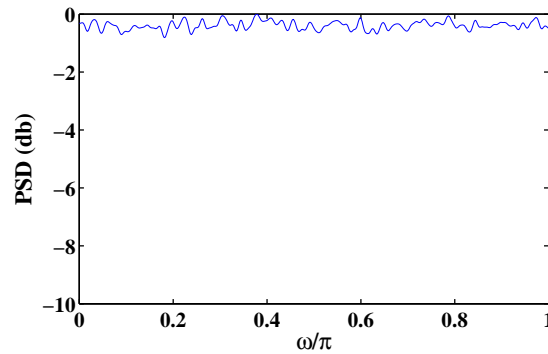


FIGURE 15.55: PSD of $e[n]$ using the `psdwelch` function for $B + 1 = 8$.

(c) See plot below.

MATLAB script:

```
% P1541: Quantization noise distribution analysis
clc; close all;
N = 1e5;
n = 1:N;
randn('seed',0)
xn = randn(1,N+1e3)/4;
xn(abs(xn)>=1) = []; xn = xn(1:N);
```

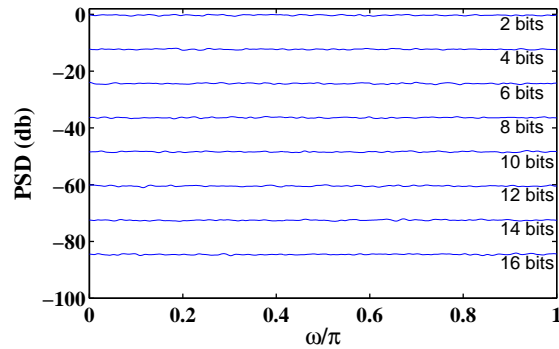


FIGURE 15.56: PSD of $e[n]$ using the `psdwelch` function for $B + 1 = 2, 4, 6, 8, 10, 12, 14$, and 16 .

```
%% Part a
L = 8; % L = 2:2:16;
[beq,E,B] = dec2beqR(xn,L);
en = beq - xn;
[eo pe] = epdf(en,20);
hfa = figconfig('P1541a','small');
bar(eo,pe/numel(en),'facecolor','b')
xlabel('e[n]','fontsize',LFS);
ylabel('Frequency','fontsize',LFS);
%% Part b
Se = psdwelch(en(:),250,1024);
w = linspace(0,1,512)*pi;
Se_mag = abs(Se); Se_db = 10*log10(Se_mag/max(Se_mag));
hfb = figconfig('P1541b','small');
plot(w/pi,Se_db); ylim([-10 0])
xlabel('\omega/\pi','fontsize',LFS);
ylabel('PSD (db)','fontsize',LFS);
%% Part c
L = 2:2:16;
hfc = figconfig('P1541c','small');
w = linspace(0,1,512)*pi;
for ii = 1:length(L)
    [beq,E,B] = dec2beqR(xn,L(ii));
    en = beq - xn;
    Se = psdwelch(en(:),250,1024);
```

```

Se_mag = abs(Se);
if ii == 1
    Se_max = max(Se_mag);
end
Se_db = 10*log10(Se_mag/max(Se_max));
figure(hfc)
plot(w/pi,Se_db); hold on
text(w(450)/pi,Se_db(450)-3,[num2str(L(ii)),' bits'],'fontsize',LFS-4)
xlabel('\omega/\pi','fontsize',LFS);
ylabel('PSD (db)','fontsize',LFS);
end
ylim([-100 2])

```

42. (a) See plot below.

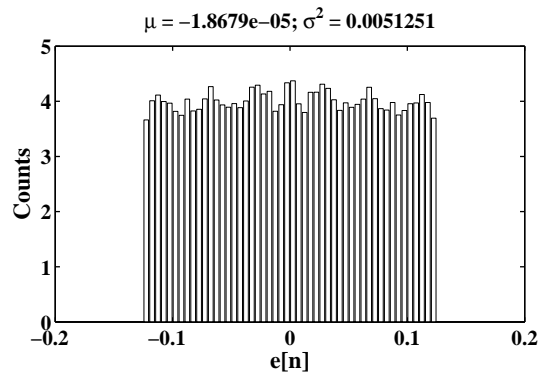


FIGURE 15.57: Plot of the quantization histogram for $B = 2$ bits.

(b) See plot below.

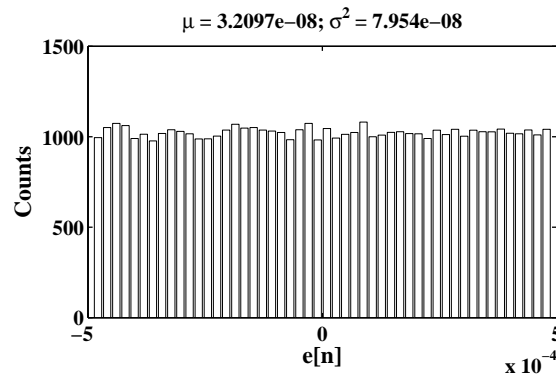
(c) tba.

MATLAB script:

```

% P1542: Quatization noise distribution analysis
clc; close all;
N = 1e5;
n = 1:N;
xn = 0.49*(cos(n/11) + sin(n/31));
B = 2; % Part a
% B = 10; % Part b
[pe, eo, eavg, evar] = QNmodel(xn,B);

```

FIGURE 15.58: Plot of the quantization histogram for $B = 10$ bits.

```
hfa = figconf('P1542a','small');
bar(eo,pe,'facecolor','w')
xlabel('e[n]','fontsize',LFS);
ylabel('Counts','fontsize',LFS);
title(['\mu = ',num2str(eavg),'; \sigma^2 = ',num2str(evar)])
```

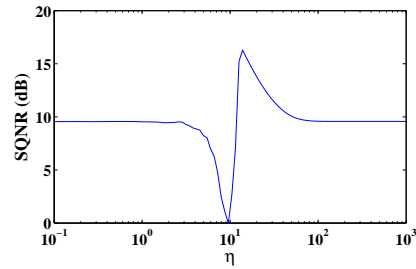
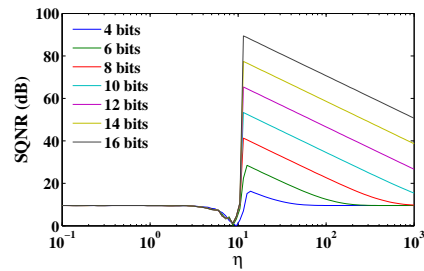
43. MATLAB script:

```
% P1543: Variance-Gain
clc; close all;
b1 = [1 3 -4]; a1 = [1 0.6 0.08];
b2 = fliplr(b1)/b1(3); a2 = fliplr(a1)/a1(3);
K = b1(3)/a1(3);
[r p k] = residuez(conv(b1,b2),conv(a1,a2));
disp('VG is')
K*(k+r(3)+r(4))
%% Verification:
hn = filter(b1,a1,[1 zeros(1,999)]);
sum(abs(hn).^2)
```

44. (a) See script output.
 (b) See plot below.
 (c) See plot below.

MATLAB script:

```
% P1544: Quantizer performance investigation
```


FIGURE 15.59: Plot of the noise variance as a function of η for $B + 1 = 4$ bits.FIGURE 15.60: Plot of the noise variance as a function of η for $B + 1 = 6, 8, 10, 12, 14,$ and 16 bits.

```

clc; close all;
%% Part a
N = 1e5;
sigx2 = 1;
xn = laprnd(N,1,0,1)*sqrt(sigx2);
L = 4;
x1 = -1; xL = 1-0.5^(L-1);
ind = (xn < xL & xn > x1);
xng = xn(ind);
[xqg,E,B] = dec2beqR(xng,L);
sigg2 = mean((xng-xqg).^2);
xno = xn(~ind);
xqo = -ones(size(xno));
ind2 = (xno > x1);
xqo(ind2) = xL;
sigo2 = mean((xno-xqo).^2);
SQNRa = sigx2/(sigg2+sigo2);

```

```

%% Part b
eta = logspace(-1,3,100); N2 = length(eta);
sigx = 1./eta; sigx2 = sigx.^2;
X = bsxfun(@times,laprnd(N,1,0,1),sigx);
SQNRb = zeros(1,N2);
L = 4;
x1 = -1; xL = 1-0.5^(L-1);
for ii = 1:N2
    ind = (X(:,ii) < xL & X(:,ii) > x1);
    xng = X(ind,ii);
    [xqg,E,B] = dec2beqR(xng,L);
    sigg2 = mean((xng-xqg).^2);
    xno = X(~ind,ii);
    xqo = -ones(size(xno));
    ind2 = (xno > x1);
    xqo(ind2) = xL;
    sigo2 = mean((xno-xqo).^2);
    if isnan(sigo2)
        sigo2 = 0;
    end
    SQNRb(ii) = sigx2(ii)/(sigg2+sigo2);
end
SQNRb_db = 10*log10(SQNRb/min(SQNRb));
hfb = figconfig('P1544b','small');
semilogx(eta,SQNRb_db)
xlabel('\eta','fontsize',LFS);
ylabel('SQNR (dB)','fontsize',LFS);
%% Part c
eta = logspace(-1,3,100); N2 = length(eta);
sigx = 1./eta; sigx2 = sigx.^2;
X = bsxfun(@times,laprnd(N,1,0,1),sigx);
L = 4:2:16; NL = length(L);
SQNRc = zeros(NL,N2);
x1 = -1; xL = 1-0.5^(L-1);
for jj = 1:NL
    for ii = 1:N2
        ind = (X(:,ii) < xL(jj) & X(:,ii) > x1);
        xng = X(ind,ii);
        [xqg,E,B] = dec2beqR(xng,L(jj));
        sigg2 = mean((xng-xqg).^2);
    end
end

```

```

xno = X(~ind,ii);
xqo = -ones(size(xno));
ind2 = (xno > x1);
xqo(ind2) = xL(jj);
sigo2 = mean((xno-xqo).^2);
if isnan(sigo2)
    sigo2 = 0;
end
SQNRc(jj,ii) = sigx2(ii)/(sigg2+sigo2);
end
end
SQNRc_db = 10*log10(SQNRc/min(SQNRc(:)));
hfc = figconf('P1544c','small');
semilogx(eta,SQNRc_db)
xlabel('\eta','fontsize',LFS);
ylabel('SQNR (dB)','fontsize',LFS);
hl = legend('4 bits','6 bits','8 bits','10 bits','12 bits','14 bits',...
    '16 bits','location','northwest');
set(hl,'box','off')

```

45. tba

46. (a) See plots below.

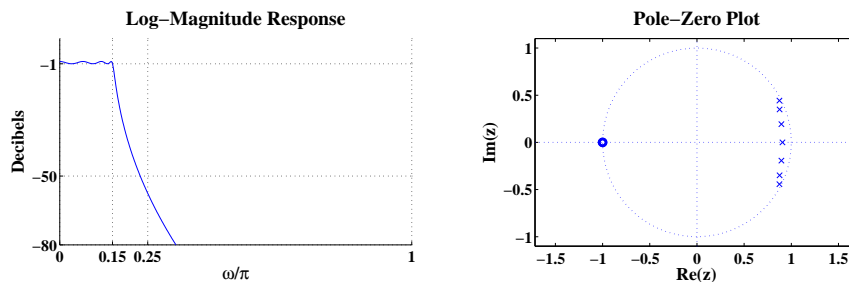


FIGURE 15.61: Plot of the magnitude response and pole-zero diagram of the filter.

(b) See plots below.

(c) See plots below.

(d) See plots below.

(e) tba.

MATLAB script:

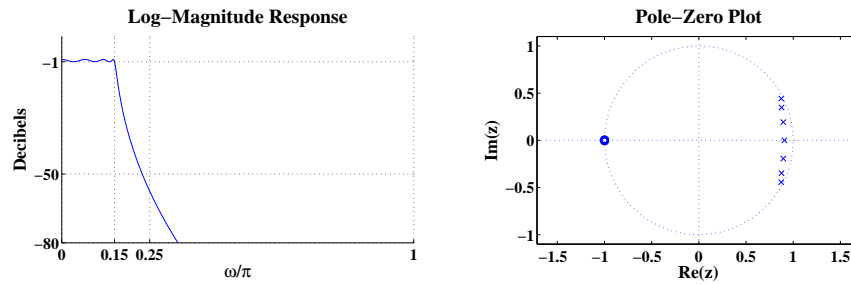


FIGURE 15.62: Plot of the magnitude response and pole-zero diagram of the filter after quantizing the cascade form coefficients to $L = 16$ bits.

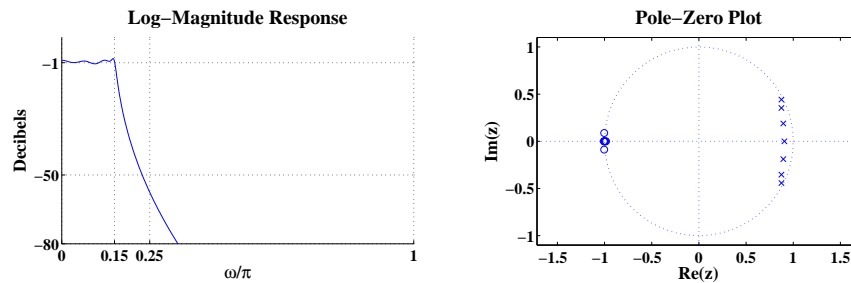


FIGURE 15.63: Plot of the magnitude response and pole-zero diagram of the filter after quantizing the cascade form coefficients to $L = 10$ bits.

```
% P1546: Digital lowpass filter design by Chebyshev I
close all; clc;
%% Filter Design
omegap = 0.15*pi; omegas = 0.25*pi; As = 50; Ap = 1; % Specification
[N,omegac] = cheb1ord(omegap/pi,omegas/pi,Ap,As); % Order define
[B,A] = cheby1(N,Ap,omegac); % coefficients
[sos G] = tf2sos(B,A); % cascade form
%% Filter Coefficient Quantization of Direct Form
L = 16; % Part b
% L = 10; % Part c
% L = 6; % Part d
[soshat,E2,B2] = dec2beqR(sos,L);
[Bhat,Ahat] = sos2tf(soshat,G);
%% Plotting Parameters and Filter Responses
BP = B; AP = A; % Plot infinite precision
% BP = Bhat; AP = Ahat; % Plot finite precision
```

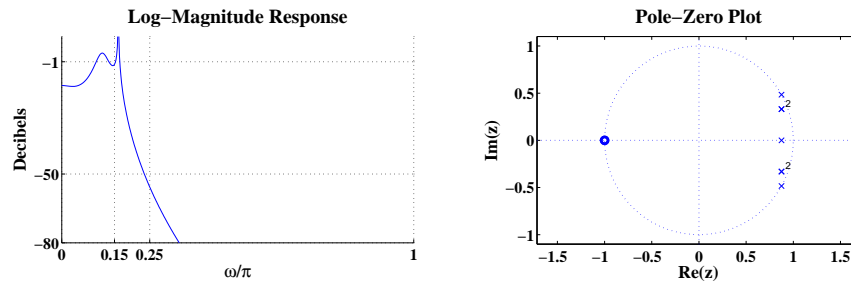


FIGURE 15.64: Plot of the magnitude response and pole-zero diagram of the filter after quantizing the cascade form coefficients to $L = 6$ bits.

```

om = linspace(0,1,501)*pi;
H = freqz(BP,AP,om); Hmag = abs(H); Hdb = 20*log10(Hmag);
%% Design Plots
hfa = figconf('P1546a','small'); % Log-Magnitude Response in dB
plot(om/pi,Hdb,'b','linewidth',1); axis([0,1,-80,10]);
xlabel('\omega/\pi','fontsize',LFS);
ylabel('Decibels','fontsize',LFS);
title('Log-Magnitude Response','fontsize',TFS);
set(gca,'xtick',[0,omegap,omegas,pi]/pi);
set(gca,'ytick',[-80,-As,-Ap]); grid; box off;
hfb = figconf('P1546b','small'); % Pole-Zero Plot
zplane(BP,AP);
xlabel('Re(z)','fontsize',LFS);
ylabel('Im(z)','fontsize',LFS);
title('Pole-Zero Plot','fontsize',TFS);

```

47. (a) See plots below.

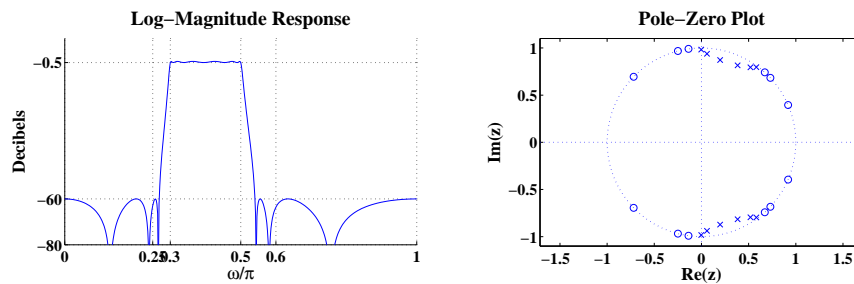


FIGURE 15.65: Plot of the magnitude response and pole-zero diagram of the filter.

(b) See plots below.

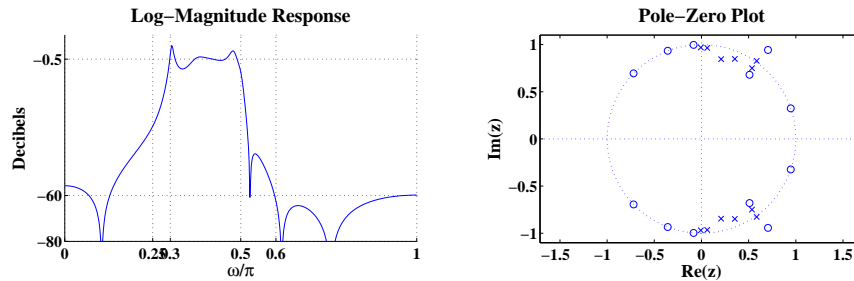


FIGURE 15.66: Plot of the magnitude response and pole-zero diagram of the filter after quantizing the direct form coefficients to $L = 16$ bits.

(c) See plots below.

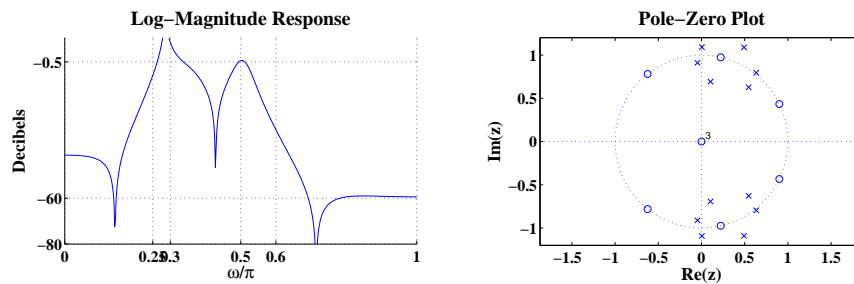


FIGURE 15.67: Plot of the magnitude response and pole-zero diagram of the filter after quantizing the direct form coefficients to $L = 12$ bits.

(d) See plots below.

(e) tba.

MATLAB script:

```
% P1532: Digital bandpass filter design by elliptic
close all; clc;
%% Filter Design
omegas1 = 0.25*pi; omegap1 = 0.3*pi; omegap2 = 0.5*pi; omegas2 = 0.6*pi;
Ap = 0.5; As = 60; % Specification
omegap = [omegap1 omegap2]; omegas = [omegas1 omegas2];
[N,omegac] = ellipord(omegap/pi,omegas/pi,Ap,As); % Order define
[B,A] = ellip(N,Ap,As,omegac); % coefficients
[sos G] = tf2sos(B,A); % cascade form
```

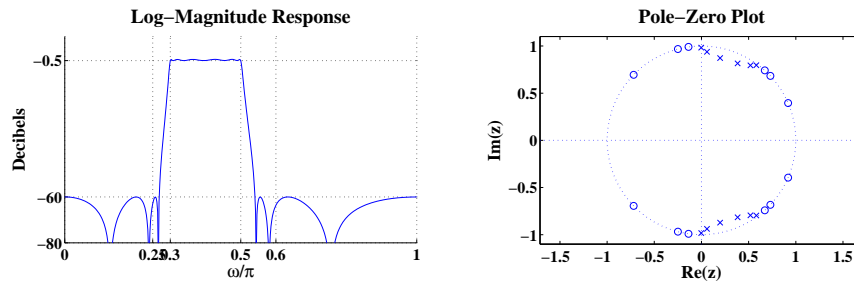


FIGURE 15.68: Plot of the magnitude response and pole-zero diagram of the filter after quantizing the cascade form coefficients to $L = 12$ bits.

```

%% Filter Coefficient Quantization
L = 16; % Part b
% L = 12; % Part c & d
% Direct form Part b & c
[BAhat,E1,B1] = dec2beqR([B;A],L);
Bhat = BAhat(1,:); Ahat = BAhat(2,:);
% Cascade form Part d
% [soshat,E2,B2] = dec2beqR(sos,L);
% [Bhat,Ahat] = sos2tf(soshat,G);
%% Plotting Parameters and Filter Responses
BP = B; AP = A; % Plot infinite precision
% BP = Bhat; AP = Ahat; % Plot finite precision
om = linspace(0,1,501)*pi; H = freqz(BP,AP,om);
Hmag = abs(H); Hdb = 20*log10(Hmag);
%% Design Plots
hfa = figconf('P1547a','small'); % Log-Magnitude Response in dB
plot(om/pi,Hdb,'b','linewidth',1); axis([0,1,-80,10]);
xlabel('\omega/\pi','fontsize',LFS);
ylabel('Decibels','fontsize',LFS);
title('Log-Magnitude Response','fontsize',TFS);
set(gca,'xtick',[0,omegas1,omegap1,omegap2,omegas2,pi]/pi);
set(gca,'ytick',[-80,-As,-Ap]); grid; box off;

hfb = figconf('P1547b','small'); % Pole-Zero Plot
zplane(BP,AP);
xlabel('Re(z)','fontsize',LFS);
ylabel('Im(z)','fontsize',LFS);
title('Pole-Zero Plot','fontsize',TFS);

```

48. (a) See plots below.

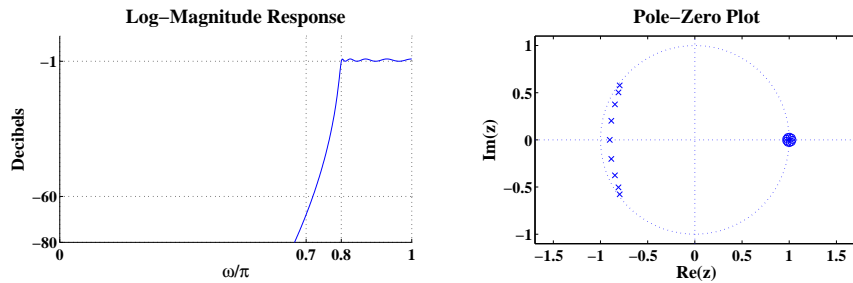


FIGURE 15.69: Plot of the magnitude response and pole-zero diagram of the filter.

(b) See plots below.

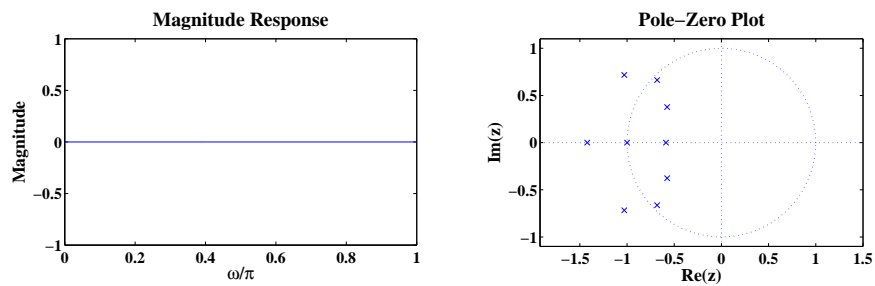


FIGURE 15.70: Plot of the magnitude response and pole-zero diagram of the filter after quantizing the direct form coefficients to $L = 16$ bits.

(c) See plots below.

(d) See plots below.

(e) tba.

MATLAB script:

```
% P1548: Digital highpass filter design by Chebyshev I
close all; clc;
%% Filter Design
omegas = 0.7*pi; omegap = 0.8*pi; As = 60; Ap = 1; % Specification
[N,omegac] = cheblord(omegap/pi,omegas/pi,Ap,As); % Order define
[B,A] = cheby1(N,Ap,omegac,'high'); % coefficients
[sos G] = tf2sos(B,A); % cascade form
%% Filter Coefficient Quantization
```

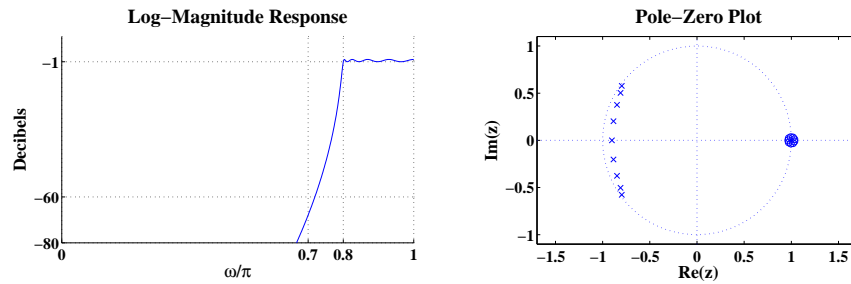



FIGURE 15.71: Plot of the magnitude response and pole-zero diagram of the filter after quantizing the cascade form coefficients to $L = 16$ bits.

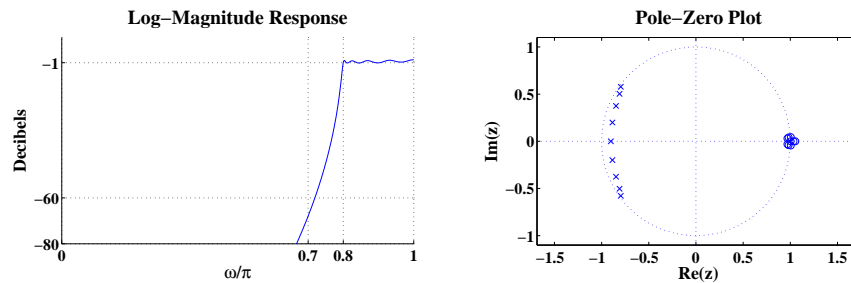


FIGURE 15.72: Plot of the magnitude response and pole-zero diagram of the filter after quantizing the cascade form coefficients to $L = 10$ bits.

```
L = 16; % Part b
% L = 12; % Part c & d
% Direct form Part b & c
[BAhat,E1,B1] = dec2beqR([B;A],L);
Bhat = BAhat(1,:); Ahat = BAhat(2,:);
% Cascade form Part d
% [soshat,E2,B2] = dec2beqR(sos,L);
% [Bhat,Ahat] = sos2tf(soshat,G);
%% Plotting Parameters and Filter Responses
BP = B; AP = A; % Plot infinite precision
% BP = Bhat; AP = Ahat; % Plot finite precision
om = linspace(0,1,501)*pi;
H = freqz(BP,AP,om); Hmag = abs(H); Hdb = 20*log10(Hmag);
%% Design Plots
hfa = figconf('P1548a','small'); % Log-Magnitude Response in dB
plot(om/pi,Hdb,'b','linewidth',1); axis([0,1,-80,10]);
```

```

xlabel('\omega/\pi','fontsize',LFS);
ylabel('Decibels','fontsize',LFS);
title('Log-Magnitude Response','fontsize',TFS);
set(gca,'xtick',[0,omegas,omegap,pi]/pi);
set(gca,'ytick',[-80,-As,-Ap]); grid; box off;

% hfa = figconfig('P1548a','small'); % Magnitude Response
% plot(om/pi,Hmag,'b','linewidth',1); axis([0,1,-1,1]);
% xlabel('\omega/\pi','fontsize',LFS);
% ylabel('Magnitude','fontsize',LFS);
% title('Magnitude Response','fontsize',TFS);

hfb = figconfig('P1548b','small'); % Pole-Zero Plot
zplane(BP,AP);
xlabel('Re(z)','fontsize',LFS);
ylabel('Im(z)','fontsize',LFS);
title('Pole-Zero Plot','fontsize',TFS);

```

49. (a) See plots below.

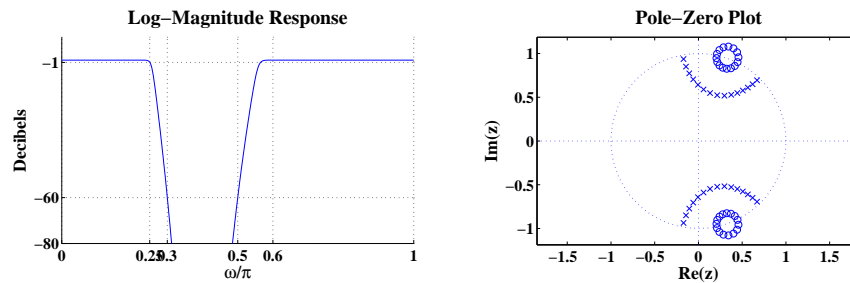


FIGURE 15.73: Plot of the magnitude response and pole-zero diagram of the filter.

(b) See plots below.

(c) See plots below.

(d) See plots below.

(e) tba.

MATLAB script:

```

% P1549: Digital bandstop filter design by butterworth
close all; clc;
%% Filter Design

```

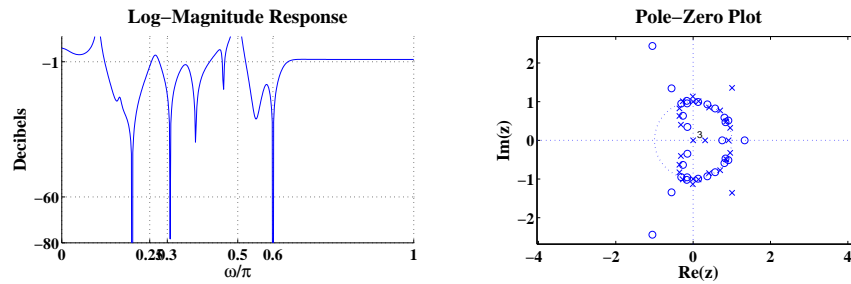


FIGURE 15.74: Plot of the magnitude response and pole-zero diagram of the filter after quantizing the direct form coefficients to $L = 16$ bits.

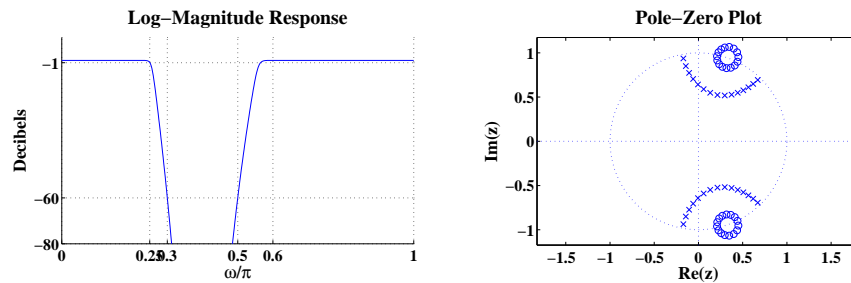


FIGURE 15.75: Plot of the magnitude response and pole-zero diagram of the filter after quantizing the cascade form coefficients to $L = 16$ bits.

```

omegap1 = 0.25*pi; omegas1 = 0.3*pi; omegas2 = 0.5*pi; omegap2 = 0.6*pi;
Ap = 1; As = 60; % Specification
omegap = [omegap1 omegap2]; omegas = [omegas1 omegas2];
[N,omegac] = buttord(omegap/pi,omegas/pi,Ap,As); % Order define
[B,A] = butter(N,omegac,'stop'); % coefficients
[sos G] = tf2sos(B,A); % cascade form
%% Filter Coefficient Quantization
L = 16; % Part b
% L = 10; % Part c & d
% Direct form Part b & c
[BAhat,E1,B1] = dec2beqR([B;A],L);
Bhat = BAhat(1,:); Ahat = BAhat(2,:);
% Cascade form Part d
% [soshat,E2,B2] = dec2beqR(sos,L);
% [Bhat,Ahat] = sos2tf(soshat,G);
%% Plotting Parameters and Filter Responses

```

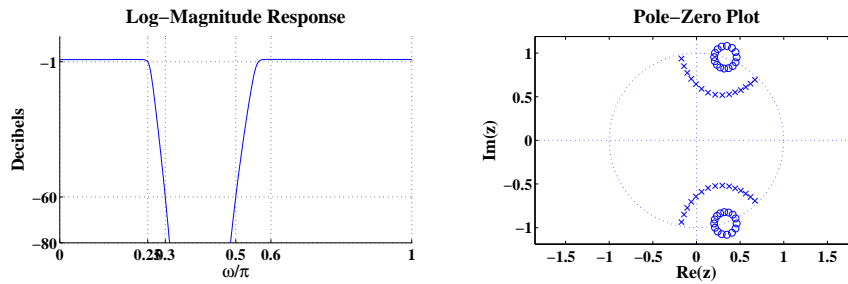


FIGURE 15.76: Plot of the magnitude response and pole-zero diagram of the filter after quantizing the cascade form coefficients to $L = 10$ bits.

```
BP = B; AP = A; % Plot infinite precision
% BP = Bhat; AP = Ahat; % Plot finite precision
om = linspace(0,1,501)*pi; H = freqz(BP,AP,om);
Hmag = abs(H); Hdb = 20*log10(Hmag);
%% Design Plots
hfa = figconfig('P1549a','small'); % Log-Magnitude Response in dB
plot(om/pi,Hdb,'b','linewidth',1); axis([0,1,-80,10]);
xlabel('\omega/\pi','fontsize',LFS);
ylabel('Decibels','fontsize',LFS);
title('Log-Magnitude Response','fontsize',TFS);
set(gca,'xtick',[0,omegap1,omegas1,omegas2,omegap2,pi]/pi);
set(gca,'ytick',[-80,-As,-Ap]); grid; box off;

hfb = figconfig('P1549b','small'); % Pole-Zero Plot
zplane(BP,AP);
xlabel('Re(z)','fontsize',LFS);
ylabel('Im(z)','fontsize',LFS);
title('Pole-Zero Plot','fontsize',TFS);
```

50. (a) See plot below.
 (b) See plot below.
 (c) See plot below.
 (d) tba.

MATLAB script:

```
% P1550: Quantization noise distribution analysis
clc; close all;
```

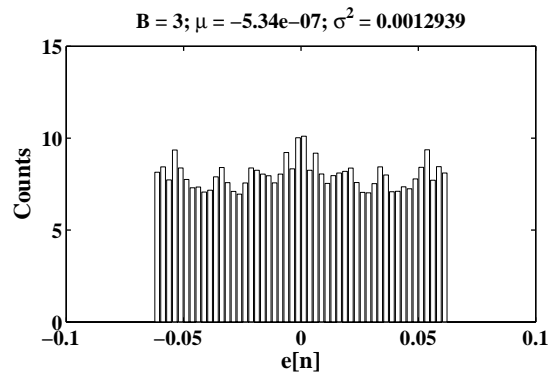


FIGURE 15.77: Plot of the histogram of the resulting error sequence when $ax[n]$ is quantized to $B = 3$.

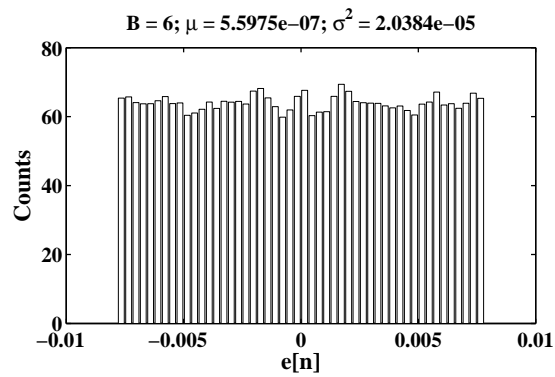


FIGURE 15.78: Plot of the histogram of the resulting error sequence when $ax[n]$ is quantized to $B = 6$.

```

N = 1e5;
n = 1:N;
xn = (cos(n/13) + sin(n/29))*0.49;
a = 0.3333;
yn = a*xn;
B = 3; % Part a
% B = 6; % Part b
% B = 9; % Part c
[pe, eo, eavg, evar] = QNmodel(yn, B);
hfa = figconfg('P1550a', 'small');
bar(eo, pe, 'facecolor', 'w')

```

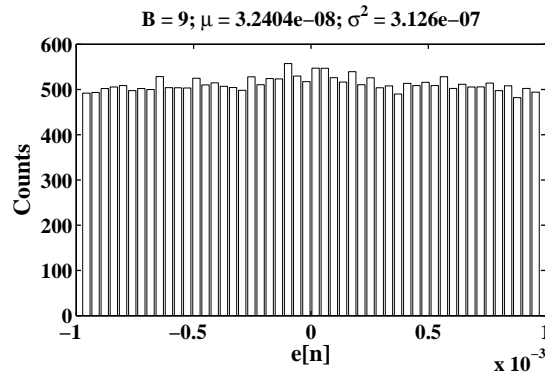


FIGURE 15.79: Plot of the histogram of the resulting error sequence when $ax[n]$ is quantized to $B = 12$.

```
xlabel('e[n]', 'fontsize', LFS);
ylabel('Counts', 'fontsize', LFS);
title(['B = ', num2str(B), '; \mu = ', num2str(eavg), ...
      '; \sigma^2 = ', num2str(evar)])
```

51. (a) Comments:
See script for detail.
- (b) See plot below.

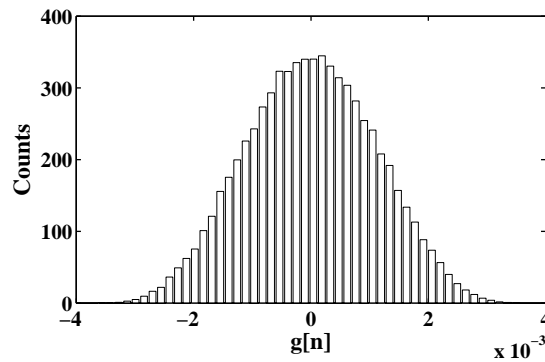


FIGURE 15.80: Plot of the histogram of the resulting error sequence $g[n]$ using all five B bit multipliers in the filter implementation.

- (c) See plot below.

MATLAB script:

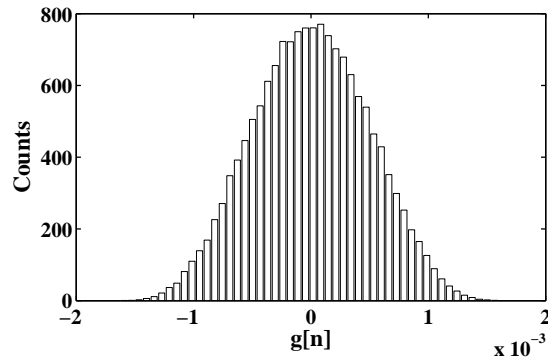


FIGURE 15.81: Plot of the histogram of the resulting error sequence $g[n]$ using only one B bit multiplier in the filter implementation.

```
% P1551:
clc; close all;
B = 8; N = 1e5; n = 1:N;
h = [0.1 0.2 0.3 0.2 0.1];
S = sqrt(1/sum(h.^2));
xn = cos(n/13);
[xnq,E1,B1] = dec2beqR(S*xn,B+1);
yn = filter(h,1,xnq); % Part a
delta = sqrt(5)*2^(-B); % Part b
% delta = 2^(-B); % Part c
rand('seed',0);
en = rand(1,N)*delta-delta/2;
ynhat = filter(h,1,xnq+en);
gn = ynhat - yn;
[eo pe] = epdf(gn,50);
SNR = var(yn)/var(gn);
%% Plot
hfa = figconfg('P1551a','small');
bar(eo,pe,'facecolor','w')
xlabel('g[n]','fontsize',LFS);
ylabel('Counts','fontsize',LFS);
```

52. (a) Comments:
See script for detail.
- (b) See plot below.

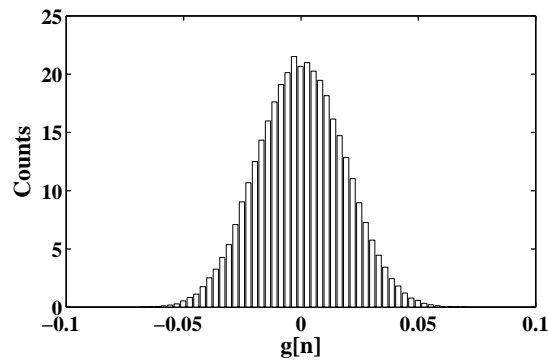


FIGURE 15.82: Plot of the histogram of the resulting error sequence $g[n]$ using $B = 5$ bit multiplication quantization in the filter implementation.

- (c) See plot below.

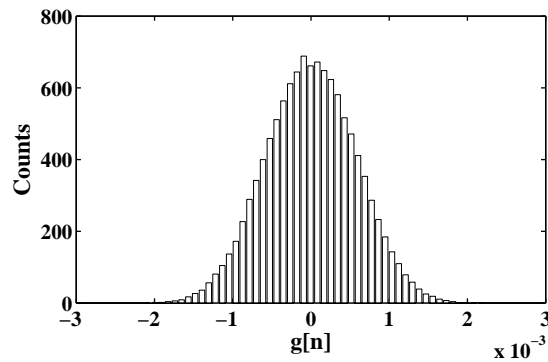


FIGURE 15.83: Plot of the histogram of the resulting error sequence $g[n]$ using $B = 10$ bit multiplication quantization in the filter implementation.

- (d) Comments:
See script output for comparison.

MATLAB script:

```
% P1552:
clc; close all;
```



```

a = 0.875; N = 1e5; n = 1:N; S = 1 - abs(a);
B = 5; % Part a & b
% B = 10; % Part c
rand('seed',0);
xn = rand(1,N)*2-1;
[xnq,E1,B1] = dec2beqR(S*xn,B+1);
yn = filter(1,[1 a],xnq); % Part a
delta = 2^(-B);
rand('seed',0);
en = rand(1,N)*delta-delta/2;
ynhat = filter(1,[1 a],xnq+en);
gn = ynhat - yn;
[eo pe] = epdf(gn,50);
SNR = var(yn)/var(gn);
SNR_ref = 2^(2*B+2)*(1-abs(a))^2; % Part d
%% Plot
hfa = figconf('P1552a','small');
bar(eo,pe,'facecolor','w')
xlabel('g[n]','fontsize',LFS);
ylabel('Counts','fontsize',LFS);

```

53. (a) Solution:

The output display oscillation and the amplitude is 0.125 and frequency is four sample.

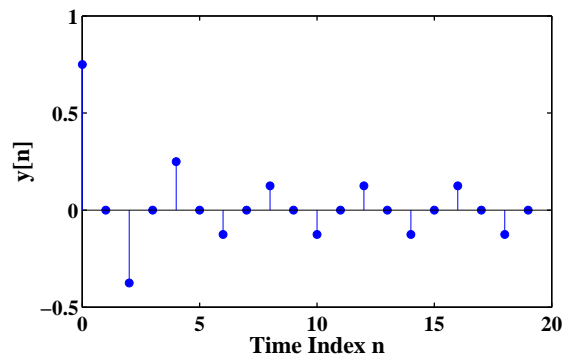


FIGURE 15.84: Plot of the first 20 samples of $y[n]$ when two's-complement overflow is used in the addition.

(b) Solution:

The output display oscillation and the amplitude is 0.125 and frequency is four sample.

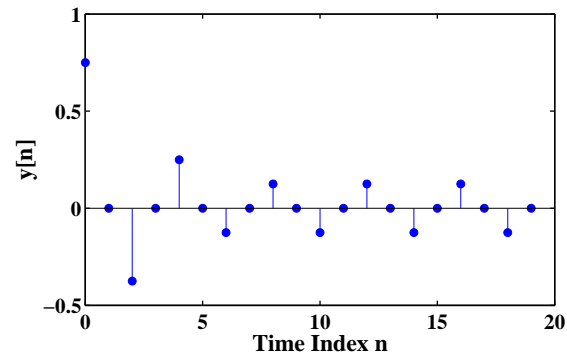


FIGURE 15.85: Plot of the first 20 samples of $y[n]$ when saturation characteristics is used in the addition.

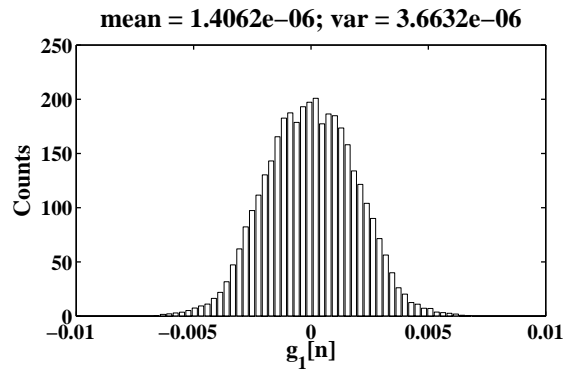
MATLAB script:

```
% P1553:
clc; close all;
a0 = 0.75; a2 = -0.5; N = 20; B = 3;
yn = zeros(1,N);
yn(1) = a0; yn(2) = 0;
%% Part a:
for ii = 3:20
    [yn(ii) E1 B1] = dec2beqR(yn(ii-2)*a2,B+1);
    if yn(ii) < -1 || yn(ii) >= 1
        yn(ii) = -sign(yn(ii))*(abs(yn(ii))-1);
    end
end
%% Part b:
% for ii = 3:20
%     [yn(ii) E1 B1] = dec2beqR(yn(ii-2)*a2,B+1);
%     if yn(ii) < -1 || yn(ii) >= 1
%         yn(ii) = sign(yn(ii));
%     end
% end
%% Plot
hfa = figconfig('P1553a','small');
```

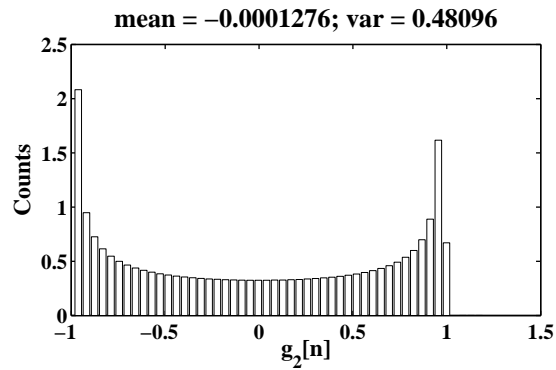
```
stem(0:N-1,yn,'filled')  
xlabel('Time Index n','fontsize',LFS);  
ylabel('y[n]','fontsize',LFS);
```

Review Problems

54. (a) Comments:
See script for detail.
- (b) See plot below.

FIGURE 15.86: Plot of the histogram of the error $g_1[n]$.

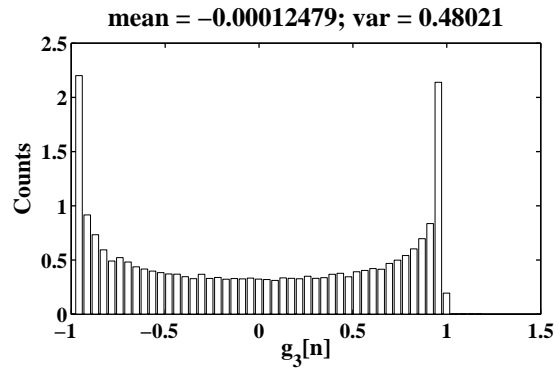
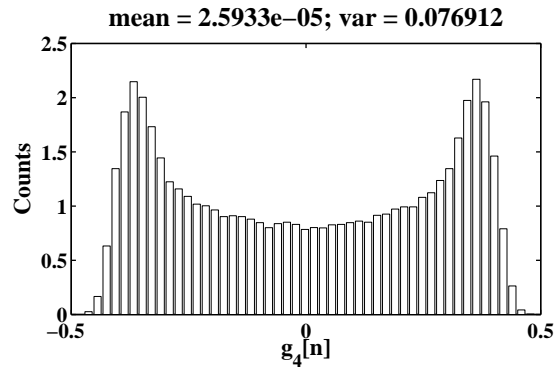
- (c) See plot below.

FIGURE 15.87: Plot of the histogram of the error $g_2[n]$.

- (d) See plot below.
- (e) See plot below.
- (f) tba

MATLAB script:

% P1554:

FIGURE 15.88: Plot of the histogram of the error $g_3[n]$.FIGURE 15.89: Plot of the histogram of the error $g_4[n]$.

```

close all; clc;
%% Filter Design
omegap = 0.2*pi; As = 40; Ap = 1; N = 5; % Specification
[b,a] = ellip(N,Ap,As,omegap/pi); % coefficients
om = linspace(0,1,501)*pi; H = freqz(b,a,om);
Hmag = abs(H); Hdb = 20*log10(Hmag);
hfa = figconf('P1554a','small'); % Log-Magnitude Response in dB
plot(om/pi,Hdb,'b','linewidth',1); axis([0,1,-60,10]);
xlabel('\omega/\pi','fontsize',LFS);
ylabel('Decibels','fontsize',LFS);
title('Log-Magnitude Response','fontsize',TFS);
set(gca,'xtick',[0,omegap,pi]/pi);
set(gca,'ytick',[-60,-As,-Ap]); grid; box off;

```

```

%% Specification:
N = 1e5; n = 1:N; xn = 0.99*cos(n/19); B = 6;
%% Part a
yn = filter(b,a,xn);
%% Part b
[xnq,B1,E1] = dec2beqR(xn,B+1);
y1 = filter(b,a,xnq);
g1 = y1 - yn; g1_avg = mean(g1); g1_var = var(g1);
[e1o pe1] = epdf(g1,50);
hfb = figconf('P1554b','small');
bar(e1o,pe1,'facecolor','w')
xlabel('g_1[n]','fontsize',LFS);
ylabel('Counts','fontsize',LFS);
title(['mean = ',num2str(g1_avg),'; var = ',num2str(g1_var)],...
      'fontsize',TFS)
%% Part c
dm = [b;a];
E = max(max(0,fix(log2(abs(dm(:))+eps)+1)));
[baq,E2,B2] = dec2beqR(dm,E+B+1);
bq = baq(1,:); aq = baq(2,:);
y2 = filter(bq,aq,xn);
g2 = y2 - yn;
% g2(isnan(g2)) = []; g2(abs(g2) == inf) = [];
g2_avg = mean(g2); g2_var = var(g2);
[e2o pe2] = epdf(g2,50);
hfc = figconf('P1554c','small');
bar(e2o,pe2,'facecolor','w')
xlabel('g_2[n]','fontsize',LFS);
ylabel('Counts','fontsize',LFS);
title(['mean = ',num2str(g2_avg),'; var = ',num2str(g2_var)],...
      'fontsize',TFS)
%% Part d
y3 = filter(bq,aq,xnq);
g3 = y3 - yn;
g3_avg = mean(g3); g3_var = var(g3);
[e3o pe3] = epdf(g3,50);
hfd = figconf('P1554d','small');
bar(e3o,pe3,'facecolor','w')
xlabel('g_3[n]','fontsize',LFS);
ylabel('Counts','fontsize',LFS);

```

```

title(['mean = ',num2str(g3_avg),'; var = ',num2str(g3_var)],...
      'fontsize',TFS)
%% Part e
Hq = freqz(bq,aq,om); Hqmag = abs(Hq); S = 1/max(Hqmag);
xsn = S*xn;
[xsq,E3,B3] = dec2beqR(xsn,B+1);
ysn = filter(b,a,xsn);
rand('seed',0)
en = rand(1,2*N);
delta = 2^(-B);
en1 = en(1:N)*delta*2*sqrt(5) - delta*sqrt(5);
en2 = en(N+1:end)*delta*2*sqrt(6) - delta*sqrt(6);
y4 = filter(bq,aq,xsq+en1) + en2;
g4 = y4 - ysn;
g4_avg = mean(g4); g4_var = var(g4);
[e4o pe4] = epdf(g4,50);
hfe = figconfig('P1554e','small');
bar(e4o,pe4,'facecolor','w')
xlabel('g_4[n]','fontsize',LFS);
ylabel('Counts','fontsize',LFS);
title(['mean = ',num2str(g4_avg),'; var = ',num2str(g4_var)],...
      'fontsize',TFS)

```

55. tba