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: Quatization noise distribution analysis
clc; close all;
```

```
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.

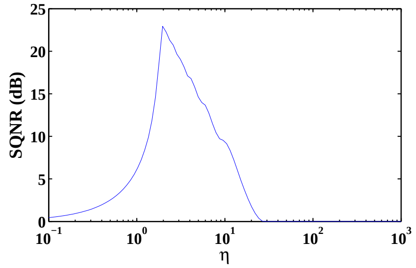

FIGURE 15.24: Plot of the noise variance as a function of η for $B+1=4$ bits.

(c) See plot below.

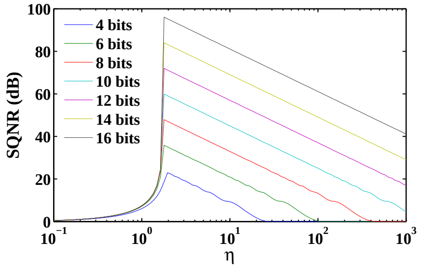

FIGURE 15.25: Plot of the noise variance as a function of η for $B+1=$ 6, 8, 10, 12, 14, and 16 bits.

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