## EE654 Adaptive Algorithms – Midterm

## 1

### A)



% Task 1 a)

N = 1000;

f = 1000;

fs = 10000;

ws = fs\*2\*pi;

x0 = cos(2\*pi\*[0:999]\*f/fs);

y0 = sin(2\*pi\*[0:999]\*f/fs);

ee = 0.2;

aa = 0.2;

dc\_x = 0.2;

dc\_y =0.1;

x1 = x0 + dc\_x;

y1 = (1+ee).\*y0+aa.\*x0+dc\_y;

% Plot spectrum of signal

figure

ww = kaiser(1000,10)';

ww = ww/sum(ww);

subplot(2,1,1)

plot(linspace(-0.5,0.5,1000)\*fs,fftshift(20\*log10(abs(fft(x0+j\*y0)).\*ww)))

xlim([-5000 5000])

ylim([-100 0])

title('Signal spectrum')

xlabel('f[hz]')

ylabel('logMagnitude[dB]')

%Plot spectrum of signal with gain/phase imbalance and DC offset

subplot(2,1,2)

plot(linspace(-0.5,0.5,1000)\*fs,fftshift(20\*log10(abs(fft(x1+j\*y1)).\*ww)))

title('Signal spectrum with gain/phase imbalance and DC offset')

xlabel('f[hz]')

ylabel('logMagnitude[dB]')

xlim([-5000 5000])

ylim([-100 0])

% Plot Lissajous pattern of signal

figure

subplot(1,2,1)

plot(x0,imag(j\*y0))

axis('square')

title('Lissajous pattern of signal')

xlabel('I')

ylabel('Q')

ylim([-1.5 1.5])

xlim([-1.5 1.5])

% Plot lissajous pattern of signal with DC offset and gain/phase imbalance

subplot(1,2,2)

plot(x1,imag(j\*y1))

axis('square')

title('Lissajous pattern of signal with gain/phase imbalance and DC offset')

xlabel('I')

ylabel('Q')

ylim([-1.5 1.5])

xlim([-1.5 1.5])

### B)



The following function was used for the DC canceller:

function [y, dc\_hat\_sv] = DC\_canceller(x,N, mu)

% Adaptive DC\_canceller using LMS algorithm

% x = input

% N = number of samples

% mu = self explanatory

w = 0;

w\_new = 0;

w\_sv = zeros(1,N);

y = zeros(1,N);

dc\_hat\_sv = zeros(1,N);

for nn = 1:N

w = w\_new;

w\_sv(nn) = w;

dc\_hat = conj(w);

dc\_hat\_sv(nn) = dc\_hat;

err = x(nn) - dc\_hat;

y(nn) = err;

w\_new = conj(err)\*mu+w;

end

end

And the script to solve the problem follows:

% Task b)

% Run DC canceller for I

[x2, dc\_hat\_x\_sv] = DC\_canceller(x1, N, 0.01);

% Run DC canceller for Q

[y2, dc\_hat\_y\_sv] = DC\_canceller(y1, N, 0.01);

%Plot DC estimates for I and Q

figure

plot(0:N-1,dc\_hat\_x\_sv,'r')

hold on

plot(0:N-1,dc\_hat\_y\_sv,'b')

legend('I','Q')

title('DC Estimate for I and Q')

### C)



The following function was used for the phase balancer:

function [y, aa\_hat\_sv] = phase\_balancer(x1,x2,N,mu)

% Adaptive phase balancer using LMS algorithm

% x1= Input signal

% x2= Signal which causes the phase imbalance

% N= Number of samples

% mu = self explanatory

w = 0;

w\_new = 0;

w\_sv = zeros(1,N);

aa\_hat\_sv = zeros(1,N);

y = zeros(1,N);

for nn = 1:N

w = w\_new;

w\_sv(nn) = w;

aa\_hat = conj(w);

aa\_hat\_sv(nn) = aa\_hat;

err = x1(nn) - aa\_hat\*x2(nn);

y(nn) = err;

w\_new = conj(err)\*x2(nn)\*mu+w;

end

end

And the following script was used to solve the problem:

% Task d

% Run gain balancer

[y4, ee\_hat\_sv] = gain\_balancer(y3,x2,N,0.01);

% Plot epsilon estimates and lissajous pattern of complex signal

figure

subplot(2,1,1)

plot(0:N-1,ee\_hat\_sv)

title('Estimates of epsilon')

subplot(2,1,2)

plot(x2(800:end),imag(j\*y4(800:end)))

axis('square')

title('Lissajous pattern of signal with corrected gain')

xlabel('I')

ylabel('Q')

ylim([-1.5 1.5])

xlim([-1.5 1.5])

### E)



The following function was used for the gain balancer:

function [y, ee\_hat\_sv] = gain\_balancer(x1,x2,N,mu)

% Adaptive gain balancer using LMS algorithm

% x1= Input signal

% x2= Signal which causes the gain imbalance

% N= Number of samples

% mu = self explanatory

w\_new = 1;

w\_sv = zeros(1,N);

ee\_hat\_sv = zeros(1,N);

y = zeros(1,N);

for nn = 1:N

w = w\_new;

ee\_hat\_sv(nn) = w;

y(nn) = x1(nn)\*w;

w\_new = (abs(x2(nn))-abs(y(nn)))\*mu+w;

end

end

And the following script was used to solve the problem:

% Task d

% Run gain balancer

[y4, ee\_hat\_sv] = gain\_balancer(y3,x2,N,0.01);

% Plot epsilon estimates and lissajous pattern of complex signal

figure

subplot(2,1,1)

plot(0:N-1,ee\_hat\_sv)

title('Estimates of epsilon')

subplot(2,1,2)

plot(x2(800:end),imag(j\*y4(800:end)))

axis('square')

title('Lissajous pattern of signal with corrected gain')

xlabel('I')

ylabel('Q')

ylim([-1.5 1.5])

xlim([-1.5 1.5])

### E)

% Task e)

N = 20000;

mu = 0.001

% Generate random QPSK data

x0=((floor(2\*rand(1,N))-0.5)/0.5);

y0=((floor(2\*rand(1,N))-0.5)/0.5);

x1 = x0 + dc\_x;

y1 = (1+ee)\*y0+aa\*x0+dc\_y;

% Run DC canceller for I

[x2, ~] = DC\_canceller(x1, N, mu);

% Run DC canceller for Q

[y2, ~] = DC\_canceller(y1, N, mu);

% Run phase balancer

[y3, ~] = phase\_balancer(y2,x2,N,mu);

% Run gain balancer

[y4, ~] = gain\_balancer(y3,x2,N,mu);

% Plot constellation diagrams at each point in the signal processing chain

figure

subplot(2,3,1)

plot\_constellation(x0+j\*y0,N,1,'Signal')

subplot(2,3,2)

plot\_constellation(x1+j\*y1,N,1,'Signal with DC offset and gain/phase imbalance')

subplot(2,3,3)

plot\_constellation(x2+j\*y2,N,N-1000,'DC-cancelled signal')

subplot(2,3,4)

plot\_constellation(x2+j\*y3,N,N-1000,'Phase-balanced signal')

subplot(2,3,5)

plot\_constellation(x2+j\*y4,N,N-1000,'Gain-balanced signal')

ha = axes('Position',[0 0 1 1],'Xlim',[0 1],'Ylim',[0

1],'Box','off','Visible','off','Units','normalized', 'clipping' , 'off');

text(0.5, 1,'\bf Constellation Diagrams','HorizontalAlignment' ,'center','VerticalAlignment', 'top');

## 2)

### A)





% Task a)

f = 107000;

fs = 1000000;

N = 10000;

% Form even indexed samples

y0=zeros(1,N\*2);

y0(1:2:end) = cos(2\*pi\*[0:N-1]\*f/fs);

% Form odd indexed samples

y1=zeros(1,N\*2);

y1(2:2:end)=cos(2\*pi\*[0.5:N-0.5]\*f/fs);

% Form interleaved samples

y01=y0+y1;

% Plot time series of ADC samples

figure

subplot(3,1,1)

stem(y0(1:100))

title('Time series of even indexed samples')

ylim([-1 1])

subplot(3,1,2)

stem(y1(1:100))

title('Time series of odd indexed samples')

ylim([-1 1])

subplot(3,1,3)

stem(y01(1:100))

title('Time series of interleaved samples')

ylim([-1 1])

% % Plot spectrum of ADC samples

figure

subplot(3,1,1)

ww = kaiser(20000)';

ww = ww/sum(ww);

plot(linspace(-0.5,0.5,N\*2)\*fs\*2,fftshift(20\*log10(abs(fft(y0).\*ww))))

title('Spectrum of even indexed samples')

ylim([1-100 0])

subplot(3,1,2)

plot(linspace(-0.5,0.5,N\*2)\*fs\*2,fftshift(20\*log10(abs(fft(y1).\*ww))))

title('Spectrum of odd indexed samples')

ylim([-100 0])

subplot(3,1,3)

plot(linspace(-0.5,0.5,N\*2)\*fs\*2,fftshift(20\*log10(abs(fft(y01).\*ww))))

title('Spectrum of interleaved samples')

ylim([-100 0])

xlim([-1e6 1e6])

### B)



% Task b)

g0=1;

g1=1.05;

dc0=-0.06;

dc1=0.05;

% Apply gain imbalance and DC offset to even samples

y0g=zeros(1,20000);

y0g(1:2:end) = g0\*y0(1:2:end)+dc0;

% Apply gain imbalance and DC offset to odd samples

y1g=zeros(1,20000);

y1g(2:2:end) = g1\*y1(2:2:end)+dc1;

% Interleave odd and even samples

y01g=y0g+y1g;

% Plot time series of distorted ADC samples

figure

subplot(3,1,1)

stem(y0g(1:100))

title('Time series of even indexed samples with DC offset and gain imbalance')

ylim([-1.5 1.5])

subplot(3,1,2)

stem(y1g(1:100))

title('Time series of odd indexed samples with DC offset and gain imbalance')

ylim([-1.5 1.5])

subplot(3,1,3)

stem(y01g(1:100))

title('Time series of interleaved samples with DC offset and gain imbalance')

ylim([-1.5 1.5])

figure

subplot(3,1,1)

plot(linspace(-0.5,0.5,N\*2)\*fs\*2,fftshift(20\*log10(abs(fft(y0g).\*ww))))

title('Spectrum of even indexed samples with DC offset and gain imbalance')

ylim([-100 0])

subplot(3,1,2)

plot(linspace(-0.5,0.5,N\*2)\*fs\*2,fftshift(20\*log10(abs(fft(y1g).\*ww))))

title('Spectrum of odd indexed samples with DC offset and gain imbalance')

ylim([-100 0])

subplot(3,1,3)

plot(linspace(-0.5,0.5,N\*2)\*fs\*2,fftshift(20\*log10(abs(fft(y01g).\*ww))))

title('Spectrum of interleaved samples with DC offset and gain imbalance')

ylim([-100 0])

### C)



Same function as in task 1 is used for the dc cancelling.

% Task c)

% Cancel DC of even samples

y0c=zeros(1,N\*2);

[y0c(1:2:N\*2), dc\_hat\_0]=DC\_canceller(y0g(1:2:N\*2),N,0.01);

% Cancel DC of odd samples

y1c=zeros(1,N\*2);

[y1c(2:2:N\*2), dc\_hat\_1]=DC\_canceller(y1g(2:2:N\*2),N,0.01);

% Interleave DC-cancelled samples

x0=y0c+y1c;

% Plot DC estimates

figure

subplot(2,1,1)

plot(0:999,dc\_hat\_0(1:1000))

title('DC estimates for even samples')

subplot(2,1,2)

plot(0:999,dc\_hat\_1(1:1000))

title('DC estimates for odd samples')

% Plot spectrum of DC cancelled signal

ww2=kaiser(N\*2-2000)';

ww2=ww2/sum(ww2);

figure

subplot(3,1,1)

plot(linspace(-0.5,0.5,N\*2-2000)\*fs\*2,fftshift(20\*log10(abs(fft(y0c(2001:end)).\*ww2))))

title('Spectrum of even indexed samples with DC cancelled')

ylim([-100 0])

subplot(3,1,2)

plot(linspace(-0.5,0.5,N\*2-2000)\*fs\*2,fftshift(20\*log10(abs(fft(y1c(2001:end)).\*ww2))))

title('Spectrum of odd indexed samples with DC cancelled')

ylim([-100 0])

subplot(3,1,3)

plot(linspace(-0.5,0.5,N\*2-2000)\*fs\*2,fftshift(20\*log10(abs(fft(x0(2001:end)).\*ww2))))

title('Spectrum of interleaved samples with DC cancelled')

ylim([-100 0])

### D)



k

%Task d)

Fpass = 800000;

Fstop = 750000;

Ap = 1;

Ast = 100;

fs = 2000000;

% Generate filter

d = designfilt('highpassfir','PassbandFrequency',Fpass,...

'StopbandFrequency',Fstop,'PassbandRipple',Ap,'StopbandAttenuation',...

Ast, 'SampleRate', fs);

h = tf(d);

cg\_new=1;

% Balance gain

reg0=zeros(1,135);

reg1=zeros(1,135);

y4\_sv=zeros(1,N\*2);

y3\_sv=zeros(1,N\*2);

y2\_sv=zeros(1,N\*2);

x2\_sv=zeros(1,N\*2);

x3=zeros(1,N\*2);

cg=zeros(1,N\*2);

mu=0.01;

% Run gain balancing algorithm

for nn = 1:N\*2

cg(nn)=cg\_new;

x1=x0(nn)\*cos(pi\*nn);

x2=cg(nn)\*x1;

x2\_sv(nn)=x2;

x3(nn)=x0(nn)-x2;

reg0=[x0(nn) reg0(1:134)];

reg1=[x1 reg1(1:134)];

y1=reg1\*h';

y2=y1\*cg(nn);

y2\_sv(nn)=y2;

y3=reg0\*h';

y3\_sv(nn)=y3;

y4=abs(y3)-abs(y2);

y4\_sv(nn)=y4;

cg\_new=y4\*mu+cg(nn);

end

% Plot spectrum of signal presented to the LMS canceller

figure

subplot(3,1,1)

plot(linspace(-0.5,0.5,N\*2-2000)\*fs,fftshift(20\*log10(abs(fft(y2\_sv(2001:end)).\*ww2))))

title('Spectrum of y2')

subplot(3,1,2)

plot(linspace(-0.5,0.5,N\*2-2000)\*fs,fftshift(20\*log10(abs(fft(y3\_sv(2001:end)).\*ww2))))

title('Spectrum of y3')

subplot(3,1,3)

plot(linspace(-0.5,0.5,N\*2-2000)\*fs,fftshift(20\*log10(abs(fft(y4\_sv(2001:end)).\*ww2))))

title('Spectrum of error signal')

figure

plot(0:1999,cg(1:2000))

title('Canceller learning curve')

## E)



% task e)

figure

subplot(3,1,1)

plot(linspace(-0.5,0.5,N\*2-2000)\*fs,fftshift(20\*log10(abs(fft(x0(2001:end)).\*ww2))))

ylim([-100 0])

title('Spectrum of x0')

subplot(3,1,2)

plot(linspace(-0.5,0.5,N\*2-2000)\*fs,fftshift(20\*log10(abs(fft(x2\_sv(2001:end)).\*ww2))))

ylim([-100 0])

title('Spectrum of x2')

subplot(3,1,3)

plot(linspace(-0.5,0.5,N\*2-2000)\*fs,fftshift(20\*log10(abs(fft(x3(2001:end)).\*ww2))))

title('Spectrum of gain cancelled signal')

ylim([-100 0])

xlim([-1e6 1e6])