# EE654 Adaptive Algorithms – final exam

## Task 1

### A)



% Task A

fs=48000;

f=1000;

N=5000;

% Generate 2000 samples of data

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

clip=1.3;

x\_0=abs(x)/clip;

phi=angle(x);

y1=clip\*(x\_0./(1+x\_0.^6).^(1/6)).\*cos(phi);

% Plot non-linear transfer function

figure

subplot(3,1,1)

clip=1.3;

x\_dat=0:0.02:2;

x\_0=abs(x\_dat)/clip;

y\_dat=clip\*(x\_0./(1+x\_0.^6).^(1/6));

plot(x\_dat,x\_dat,'linewidth',2)

hold on

plot(x\_dat,y\_dat,'r','linewidth',2)

plot([1 1]\*clip,[0.80 1.1]\*clip,'r','linewidth',2)

hold off

grid

title('Nonlinear Transfer Function of Amplifier')

text(1.0,0.8,'1-dB Compression Point')

% Plot 200 samples of input and output data

subplot(3,1,2)

plot(0:199,x(1:200),'r')

hold on

plot(0:199,y1(1:200),'b')

title('First 200 samples of input/output data')

legend('Input','Output')

% Plot spectrum of distorted

subplot(3,1,3)

ww=kaiser(2000)';

ww=ww/sum(ww);

plot(linspace(-0.5,0.5,2000)\*fs,fftshift(20\*log10(abs(fft(y1(1:2000)).\*ww))))

title('Spectrum of distorted signal')

### B)



% Task B

n\_taps=4;

reg=zeros(1,n\_taps);

wts=zeros(1,4);

y2=zeros(1,N);

y3=zeros(1,N);

%wts(1)=1;

mu=0.1;

y3\_sv=zeros(1,N);

% Run LMS algorithm

for n=1:N

reg=[x(n) reg(1:3)];

y2(n)=reg\*wts';

if n>n\_taps

y3(n)=y1(n)-y2(n);

wts=wts+mu\*reg\*conj(y3(n));

end

end

figure

subplot(3,1,1)

plot(0:N-1,20\*log10(abs(y3)))

title('Learning curve of LMS algorithm (logmag of error)')

subplot(3,1,2)

plot(1001:1200,y3(1001:1200));

title('200 samples of error signal after transient')

subplot(3,1,3);

plot(linspace(-0.5,0.5,2000)\*fs,fftshift(20\*log10(abs(fft(y3(1501:3500)).\*ww))))

title('Spectrum of distorted signal')

THD=100\*var(y3)/var(y2);

disp(['Total harmonic distortion: ' num2str(THD) '%'])

Total harmonic distortion: 0.16583%

### C)



% Task C

reg=zeros(1,n\_taps)';

wts=zeros(1,n\_taps)';

wts(1)=1;

delta=0.5;

lambda=0.995;

y3=zeros(1,1000);pp=(1/delta)\*eye(n\_taps);

for n=1:N

y2(n)=reg'\*conj(wts);

y3(n)=y1(n)-y2(n);

C=pp\*reg;

KK\_eq=C/(lambda+reg'\*C);

wts=wts+KK\_eq\*conj(y3(n));

pp=(1/lambda)\*pp -(1/lambda)\*KK\_eq\*reg'\*pp;

reg=[x(n); reg(1:3)];

end

figure

subplot(3,1,1)

plot(0:N-1,20\*log10(abs(y3)))

title('Learning curve of RLS algorithm (logmag of error)')

subplot(3,1,2)

plot(1001:1200,y3(1001:1200));

title('200 samples of error signal after transient')

subplot(3,1,3);

plot(linspace(-0.5,0.5,2000)\*fs,fftshift(20\*log10(abs(fft(y3(1001:3000)).\*ww))))

title('Spectrum of distorted signal')

THD=100\*var(y3)/var(y2);

disp(['Total harmonic distortion: ' num2str(THD) '%'])

Total harmonic distortion: 0.06355%

## Task 2

### A)



% Task a)

n\_dat=1024;

n1=10;

bw=0.20;

x1=(-n1\*bw:bw:(2\*n1-1)\*bw/2); % time sample locations

yy=sinc(x1); % Correlation Sequence

x2=(-n1:n1-1);

rr=zeros(n1,n1); rd=zeros(1,n1); % Form Correlation Matrix rr and cross Correlation Vector rd

for n=1:n1

rr(n1+1-n,:)=yy(n+1:n+n1);

rd(n1+1-n)=yy(n);

end

add=10^(-3)\*eye(n1,n1); % add small term to Diagonal to raise matrix condition number

rrp=rr+add;

wts=inv(rrp)\*conj(rd'); % form filter weights

aa=[1 -wts'];

fwts=fftshift(20\*log10(abs(fft([1 -wts'],1024))));

% Plot impulse response of predictive filter

figure

subplot(2,1,1);

stem(0:length(wts)-1,wts)

title('Impulse response of predictive filter')

subplot(2,1,2);

plot(linspace(-0.5,0.5,1024),fwts);

title('Frequency response of predictive filter')

### B)





% Task b

x=0.8\*sin(2\*pi\*(0:(n\_dat-1))\*0.06);

reg=zeros(1,n1);

err=zeros(1,n\_dat);

qq=4; % number of ADC bits

scl=2^(qq-2);

for nt=1:n\_dat;

sm1=x(nt)+reg\*wts;

q\_out=round(scl\*sm1)/scl;

n(nt)=q\_out;

err(nt)=sm1-q\_out;

reg=[err(nt) reg(1:n1-1)];

end;

% Plot input and output time series

figure

subplot(2,1,1)

stem(924:n\_dat-1,x(925:end),'b')

title('Input time series')

subplot(2,1,2)

stem(924:n\_dat-1,n(925:end),'r')

title('Output time series')

% Plot input and output spectrum

figure

ww=kaiser(n\_dat)';

ww=ww/sum(ww);

subplot(2,1,1)

plot(linspace(-0.5,0.5,n\_dat),fftshift(20\*log10(abs(fft(x).\*ww))))

title('Input spectrum')

subplot(2,1,2)

plot(linspace(-0.5,0.5,n\_dat),fftshift(20\*log10(abs(fft(n).\*ww))))

title('Output spectrum')

### C)



% Task C

f\_n = 200; % Filter order

f = [0 0.1 0.2 1]; % Frequency band edges

a = [1 1 0 0]; % Desired amplitudes

h = firpm(f\_n,f,a); % Generate filter using remez algorithm

y2=filter(h,1,n);

figure

subplot(2,1,1)

stem(924:n\_dat-1,y2(925:end),'r')

title('Filtered time series')

subplot(2,1,2)

plot(linspace(-0.5,0.5,n\_dat),fftshift(20\*log10(abs(fft(y2).\*ww))))

title('Output spectrum of low pass filter')