Chapter 6 Appendix.

TOC: 1. Maximally Decimated 6-Path Analysis Channelizer

2. Non-Maximally Decimated 6-Path Analysis Channelizer

3. Maximally Decimated 6-Path Synthesis Channelizer

4. Non-Maximally Decimated 6-Path Synthesis Channelizer

5. Non-Maximally Decimated 6-Path Cascade Analysis and Synthesis Channelizers  
 Impulse Response, Full Bandwidth Cascade

6. Non-Maximally Decimated 6-Path Cascade Analysis and Synthesis Channelizers  
 Impulse Response, Synthesized Three Channel Super-Channel

% 1 % %%%%%%%%%%%%% Maximally Decimated 6-path Analysis Channelizer %%%%%%%%

hh=sinc(-6.0+1/6:1/6:6.0-1/6).\*kaiser(71,9.2)'; %Prototype Filter

hh=hh/6;

hh2=reshape(6\*[hh 0],6,12); % Polyphase partition of Filter Weights

reg=zeros(6,12); % Input Sample data Array for 6-path filter

v1=zeros(1,6)'; % 6 Sample Input Vector

v2=zeros(1,6)'; % 6 Path Filter Output Vector

v3=zeros(1,6)'; % Channelizer Output Vector, IFFT of Filter Vector

v4=zeros(6,100); % Time Series Output Buffer

x=[0 1 zeros(1,600)]; % Input Sequence to Filter

mm=1; % Output Clock Index

for n=1:6:length(x)-6 % For Loop, 6-Input Samples

v1(1:6)=fliplr(x(n:n+5)).'; % Re-order, start Bottom, go to top

reg=[v1 reg(:,1:11)]; % Insert Vector in Filter Delay Line

for k=1:6 % Compute 6 Path Outputs

v2(k)=reg(k,:)\*hh2(k,:)';

end

v3=6\*ifft(v2); % Phase Coherent Sum, Channelizer Output

v4(:,mm)=fftshift(v3); % Store Channel Output Samples

mm=mm+1; % Increment Output Clock

end

figure(1)

subplot(3,1,1)

plot((-0.5:1/500:0.5-1/500)\*6,fftshift(20\*log10(abs(fft(hh,500)))),'linewidth',2)

hold on

plot([-0.5 -0.5 +0.5 +0.5],[-100 0 0 -100],'r','linewidth',2)

hold off

grid on

axis([-3 3 -100 10])

title('Spectrum, Prototype Low Pass Filter, 6-Path Maximally Decimated Filter Bank')

xlabel('Frequency')

ylabel('Log Mag (dB)')

for k=1:6

subplot(3,3,k+3)

plot(real(v4(k,:)))

plot((-0.5:1/500:0.5-1/500)\*1,fftshift(20\*log10(abs(fft(v4(k,:),500)))),'linewidth',2)

grid on

axis([-0.5 0.5 -2 1])

title(['Spectrum, Channel(',num2str(k-4),')'])

xlabel('Frequency')

ylabel('Log Mag (dB)')

end

% 2 % %%%%%% Non-Maximally Decimated 6-path Analysis Channelizer %%%%%%%%

hh=sinc(-6.0+1/6:1/6:6.0-1/6).\*kaiser(71,9.2)'; %Prototype Filter

hh=hh/6;

hh2=reshape(6\*[hh 0],6,12); % Polyphase partition of Filter Weights

reg=zeros(6,12); % Input Sample data Array for 6-path filter

v1=zeros(1,3)'; % 3 Sample Input Vector

v2=zeros(1,6)'; % 6 Path Filter Output Vector

v3=zeros(1,6)'; % Channelizer Output Vector, IFFT of Filter Vector

v4=zeros(6,200); % Time Series Output Buffer

flg=0; % Circular Buffer Flag

x=[0 1 zeros(1,100)]; % Input Sequence to Filter

mm=1; % Output Clock Index

for n=1:3:length(x)-6 % For Loop, 3-Input Samples

v1(1:3)=fliplr(x(n:n+2)).'; % Re-order, start center, go to top

reg=[reg(4:6,:);reg(1:3,:)]; % Roll Buffer for Serpentine Shift

reg(1:3,:)=[v1 reg(1:3,1:11)]; % Insert Vector in Filter Delay Line

for k=1:6 % Compute 6 Path Outputs

v2(k)=reg(k,:)\*hh2(k,:)';

end

if flg==0; % Test flag for end around shift

flg=1; % no shift, set flag for next test

else

flg=0; % set flag for next test

v2=[v2(4:6);v2(1:3)]; % perform end around shift

end

v3=3\*ifft(v2); % Phase Coherent Sum, Channelizer Output

v4(:,mm)=fftshift(v3); % Store Channel Output Samples

mm=mm+1; % Increment Output Clock

end

figure(2)

subplot(3,1,1)

plot((-0.5:1/500:0.5-1/500)\*6,fftshift(20\*log10(abs(fft(hh,500)))),'linewidth',2)

hold on

plot([-0.5 -0.5 +0.5 +0.5],[-100 0 0 -100],'r','linewidth',2)

plot([-1.0 -1.0],[-100 10],'--r','linewidth',2)

plot([+1.0 +1.0],[-100 10],'--r','linewidth',2)

hold off

grid on

axis([-3 3 -100 10])

title('Spectrum, Prototype Low Pass Filter, 6-Path Non-Maximally Decimated Filter Bank')

xlabel('Frequency')

ylabel('Log Mag (dB)')

for k=1:6

subplot(3,3,k+3)

plot(real(v4(k,:)))

plot((-0.5:1/500:0.5-1/500)\*2,fftshift(20\*log10(abs(fft(v4(k,:),500)))),'linewidth',2)

grid on

axis([-1 1 -100 10])

title(['Spectrum, Channel(',num2str(k-4),')'])

xlabel('Frequency')

ylabel('Log Mag (dB)')

end

% 3 % %%%%%%%%%%%%% Maximally Decimated 6-path Synthesis Channelizer %%%%%%%%

hh=remez(94,[0 8 12 60]/60,{'myfrf',[1 1 0 0]},[1 200]); %Prototype Filter

hh2=reshape(6\*[0 hh],6,16); % Polyphase partition of Filter Weights

v4=zeros(6,100); % Input array of 6-pnt vectors

u3=zeros(1,6)'; % Input vector to IFFT

u2=zeros(1,6)'; % Output vector from IFFT

reg=zeros(6,16); % Input Sample data Array for 6-path filter

u1=zeros(1,6)'; % 6 output samples from channelizer

y=zeros(1,500); % Output Sequence from Channelizer

w2=kaiser(100,14)'/100; w1=kaiser(39,14)'/4;

v4(5,20)=1; % impulse bin -2

v4(1,20)=1; % impulse bin 0

v4(3,1:39)=sinc(-5+1/4:1/4:5-1/4).\*w1; % sinc pulse bin +2

v4(6,1:39)=sinc(-6-1/3:1/3:6+1/3).\*w1; % sinc pulse bin -2

v4(2,1:100)=exp(j\*2\*pi\*(0:99)\*0.1).\*w2; % cmplx tone bin +1

mm=1; % Output Clock Index

for n=1:length(v4) % For Loop, 1-Input 6-sample vector

u3=v4(:,n); % Input vector to IFFT

u2=ifft(u3); % Interpolated output from IFFT

reg=[u2 reg(:,1:15)]; % Insert Vector in Filter Delay Line

for k=1:6 % Compute 6 Path Outputs

u1(k)=reg(k,:)\*hh2(k,:)';

end

y(mm:mm+5)=u1.'; % Store 6 Channel Output Samples

mm=mm+6; % Increment Output Clock

end

figure(3)

subplot(3,1,1)

plot((-0.5:1/600:0.5-1/600)\*6,fftshift(20\*log10(abs(fft(y,600)))),'linewidth',2)

hold on

plot([-0.5 -0.5 +0.5 +0.5],[-100 0 0 -100],'r','linewidth',2)

hold off

grid on

axis([-3 3 -100 10])

title('Spectrum, Prototype Low Pass Filter, 6-Path Non-Maximally Decimated Filter Bank')

xlabel('Frequency')

ylabel('Log Mag (dB)')

qq=[4 5 6 1 2 3];

aa1=[-20 -20 -20 -20 -50 -20];

aa3=[-0.5 -0.1 -0.05 -0.1 -0.012 -0.05];

aa4=[0.5 1.1 0.3 1.1 0.012 0.3];

for k=1:6

subplot(3,3,k+3)

if k==5

plot(-50:49,real(v4(qq(k),1:100)),'linewidth',2)

hold on

plot(-50:49,imag(v4(qq(k),1:100)),'r','linewidth',2)

hold off

else

plot(-19:19,real(v4(qq(k),1:39)),'linewidth',2)

end

grid on

axis([aa1(k) -aa1(k) aa3(k) aa4(k)]);

title(['Input Time Series, Channel(',num2str(k-4),')'])

xlabel('Time Index')

ylabel('Amplitude')

end

% % 4 % %%%%%% Non-Maximally Decimated 6-path Synthesis Channelizer %%%%%%

hh=remez(94,[0 8 12 60]/60,{'myfrf',[1 1 0 0]},[1 200]); %Prototype Filter

hh2=reshape(6\*[0 hh],6,16); % Polyphase partition of Filter Weights

v4=zeros(6,200); % Input array of 6-pnt vectors

u3=zeros(1,6)'; % Input vector to IFFT

u2=zeros(1,6)'; % Output vector from IFFT

reg=zeros(6,32); % Input Sample data Array for 6-path filter

u1=zeros(1,3)'; % 3 output samples from channelizer

y=zeros(1,600); % Output Sequence from Channelizer

w0=kaiser(63,10)'/12;

w1=kaiser(39,10)'/6;

w2=kaiser(100,10)'/100;

v4(5,20)=1; % impulse bin -2

v4(1,20)=1; % impulse bin 0

v4(3,1:63)=sinc(-2.5-1/12:1/12:2.5+1/12).\*w0; % sinc pulse bin +2

v4(6,1:39)=sinc(-3-1/6:1/6:3+1/6).\*w1; % sinc pulse bin -2

v4(2,1:100)=exp(j\*2\*pi\*(0:99)\*0.05).\*w2; % cmplx tone bin +1

mm=1; % Output Clock Index

flg=0;

for n=1:length(v4) % For Loop, 6-Input Samples

u3=v4(:,n); % Input vector to IFFT

u2=ifft(u3); % Interpolated output from IFFT

if flg==0; % test for circular shift

flg=1; % set flag for next test

else

flg=0; % set flag for next test

u2=[u2(4:6);u2(1:3)]; % perform circular shift

end

reg=[u2 reg(:,1:31)]; % Insert Vector in Filter Delay Line

for k=1:3 % Compute 3 Path Outputs

p1=reg(k,1:2:32)\*hh2(k,:)';

p2=reg(k+3,2:2:32)\*hh2(k+3,:)';

u1(k)=(p1+p2);

end

y(mm:mm+2)=u1.'; % Store 3 Channel Output Samples

mm=mm+3; % Increment Output Clock

end

figure(4)

subplot(3,1,1)

plot((-0.5:1/600:0.5-1/600)\*6,fftshift(20\*log10(abs(fft(y,600)))),'linewidth',2)

hold on

plot([-0.5 -0.5 +0.5 +0.5],[-100 0 0 -100],'r','linewidth',2)

hold off

grid on

axis([-3 3 -100 10])

title('Spectrum, Prototype Low Pass Filter, 6-Path Non-Maximally Decimated Filter Bank')

xlabel('Frequency')

ylabel('Log Mag (dB)')

qq=[4 5 6 1 2 3];

aa1=[-20 -20 -20 -20 -50 -34];

aa3=[-0.5 -0.1 -0.05 -0.1 -0.012 -0.02];

aa4=[0.5 1.1 0.2 1.1 0.012 0.1];

for k=1:6

subplot(3,3,k+3)

if k==5

plot(-50:49,real(v4(qq(k),1:100)),'linewidth',2)

hold on

plot(-50:49,imag(v4(qq(k),1:100)),'r','linewidth',2)

hold off

elseif k==6

plot(-31:31,real(v4(qq(k),1:63)),'linewidth',2)

else

plot(-19:19,real(v4(qq(k),1:39)),'linewidth',2)

end

grid on

axis([aa1(k) -aa1(k) aa3(k) aa4(k)]);

title(['Input Time Series, Channel(',num2str(k-4),')'])

xlabel('Time Index')

ylabel('Amplitude')

end

% 5 % %%%%%% Cascade 3-to-1, and 1-to-3 Non-Maximally Decimated %%%%%%

%%%%%% 6-path Analysis and synthesis Channelizer %%%%%%%%

%%%%% Impulse Response, Full Channel Bandwidth %%%%%%%%%

hh=sinc(-6+1/6:1/6:6-1/6).\*kaiser(71,9.2)';

hh=hh/6;

gg=remez(70,[0 15 25 60]/60,{'myfrf',[1 1 0 0]},[3 1]);

hh2=reshape(6\*[0 hh],6,12);

gg2=reshape([0 gg],6,12);

reg1=zeros(6,12);

reg2=zeros(6,24);

v1=zeros(1,3)';

v2=zeros(1,6)';

v3=zeros(1,6)';

v4=zeros(6,33);

u3=zeros(1,6)';

u2=zeros(1,6)';

flg1=0;

flg2=0;

x=zeros(1,160);

y=zeros(1,160);

x(1)=1;

m1=1;

m2=0;

for n=1:3:160-3

v1(1:3)=fliplr(x(n:n+2)).';

reg1=[reg1(4:6,:);reg1(1:3,:)];

reg1(1:3,:)=[v1 reg1(1:3,1:11)];

for k=1:6

v2(k)=reg1(k,:)\*hh2(k,:)';

end

if flg1==0;

flg1=1;

else

flg1=0;

v2=[v2(4:6);v2(1:3)];

end

v3=3\*ifft(v2);

v4(:,m1)=fftshift(v3);

m1=m1+1;

v3=v3.\*[1 1 0 0 0 1]'; % binary mask

u3=v3;

u2=6\*ifft(u3);

%

if flg2==0;

flg2=1;

else

flg2=0;

u2=[u2(4:6);u2(1:3)];

end

reg2=[u2 reg2(:,1:23)];

for k=1:3

p1=reg2(k,1:2:24)\*gg2(k,:)';

p2=reg2(k+3,2:2:24)\*gg2(k+3,:)';

y(m2+k)=p1+p2;

end

m2=m2+3;

end

figure(5)

subplot(3,1,1)

plot(0:144,y(1:145),'linewidth',2)

grid on

axis([-5 150 -0.1 1.1])

title('Impulse Response, Cascade 3-to-1 Downsample 6-Path Analysis Filter Bank and 1-to-3 6-Path Synthesis Filter Bank')

xlabel('Time Index')

ylabel('Amplitude')

subplot(3,1,2)

plot(0:144,y(1:145),'linewidth',2)

grid on

axis([-5 150 -0.00002 .00002])

title('Zoom to Reconstruction Artifacts Levels of Cascade Filter Bank Impulse Response')

xlabel('Time Index')

ylabel('Amplitude')

text(49,-0.00001,'Artifacts 5 Orders of Magnitude Below Desired Signal Level','fontsize',14 )

subplot(3,1,3)

plot((-0.5:1/600:0.5-1/600)\*6,fftshift(20\*log10(abs(fft(y,600)))),'r','linewidth',3)

hold on

plot((-0.5:1/600:0.5-1/600)\*6,fftshift(20\*log10(abs(fft(hh,600)))),'linewidth',2)

plot((-0.5:1/600:0.5-1/600)\*6+1,fftshift(20\*log10(abs(fft(hh,600)))),'linewidth',2)

plot((-0.5:1/600:0.5-1/600)\*6-1,fftshift(20\*log10(abs(fft(hh,600)))),'linewidth',2)

plot((-0.5:1/600:0.5-1/600)\*6+2,fftshift(20\*log10(abs(fft(hh,600)))),'linewidth',2)

plot((-0.5:1/600:0.5-1/600)\*6-2,fftshift(20\*log10(abs(fft(hh,600)))),'linewidth',2)

hold off

grid on

axis([-1.5 1.5 -0.0005 0.0005])

title('Spectrum, Zoom to Spectrum Ripple Levels of Impulse Response and of Prototype Low Pass Filter Embedded in Analysis Filter Bank')

xlabel('Frequency')

ylabel('Log Mag (dB)')

text(0.28, 0.00025,' Peak Ripple = 0.00019 dB','fontsize',14)

% 6 % %%%%%% Cascade 3-to-1, and 1-to-3 Non-Maximally Decimated

%%%%%% 6-path Analysis and synthesis Channelizer %%%%%%%%

hh=sinc(-6+1/6:1/6:6-1/6).\*kaiser(71,9.2)';

hh=hh/6;

gg=remez(70,[0 15 25 60]/60,{'myfrf',[1 1 0 0]},[3 1]);

hh2=reshape(6\*[0 hh],6,12);

gg2=reshape([0 gg],6,12);

reg1=zeros(6,12);

reg2=zeros(6,24);

v1=zeros(1,3)';

v2=zeros(1,6)';

v3=zeros(1,6)';

v4=zeros(6,33);

u3=zeros(1,6)';

u2=zeros(1,6)';

flg1=0;

flg2=0;

x=zeros(1,160);

y=zeros(1,160);

x(1)=1;

m1=1;

m2=0;

for n=1:3:160-3

v1(1:3)=fliplr(x(n:n+2)).';

reg1=[reg1(4:6,:);reg1(1:3,:)];

reg1(1:3,:)=[v1 reg1(1:3,1:11)];

for k=1:6

v2(k)=reg1(k,:)\*hh2(k,:)';

end

if flg1==0;

flg1=1;

else

flg1=0;

v2=[v2(4:6);v2(1:3)];

end

v3=3\*ifft(v2);

v4(:,m1)=fftshift(v3);

m1=m1+1;

%v3=v3.\*[1 1 0 0 0 1]'; % binary mask

u3=v3;

u2=6\*ifft(u3);

%

if flg2==0;

flg2=1;

else

flg2=0;

u2=[u2(4:6);u2(1:3)];

end

reg2=[u2 reg2(:,1:23)];

for k=1:3

p1=reg2(k,1:2:24)\*gg2(k,:)';

p2=reg2(k+3,2:2:24)\*gg2(k+3,:)';

y(m2+k)=p1+p2;

end

m2=m2+3;

end

figure(6)

subplot(4,1,1)

plot((-0.5:1/600:0.5-1/600)\*6,fftshift(20\*log10(abs(fft(hh,600)))),'linewidth',2)

hold on

plot([-0.5 -0.5 +0.5 +0.5],[-100 0 0 -100],'r','linewidth',2)

hold off

grid on

axis([-3 3 -100 10])

title('Spectrum, Prototype Low Pass Filter, 6-Path Non-Maximally Decimated Filter Bank')

xlabel('Frequency')

ylabel('Log Mag (dB)')

for k=1:6

subplot(4,3,k+3)

plot(-0.5:1/600:0.5-1/600,fftshift(20\*log10(abs(fft(v4(k,:),600)))),'linewidth',2)

grid on

axis([-1 1 -100 10]);

title(['Frequency Response, Channel(',num2str(k-4),')'])

xlabel('Frequency')

ylabel('Log mag (dB)')

end

subplot(4,1,4)

plot([-0.5 -0.5 +0.5 +0.5],[-100 0 0 -100],'r','linewidth',2)

hold on

plot((-0.5:1/600:0.5-1/600)\*2,fftshift(20\*log10(abs(fft(v4(1,:),600)))),'r','linewidth',2)

plot((-0.5:1/600:0.5-1/600)\*2+1,fftshift(20\*log10(abs(fft(v4(1,:),600)))),'r','linewidth',2)

plot((-0.5:1/600:0.5-1/600)\*2-1,fftshift(20\*log10(abs(fft(v4(1,:),600)))),'r','linewidth',2)

plot((-0.5:1/600:0.5-1/600)\*6,fftshift(20\*log10(abs(fft(y,600)))),'linewidth',2)

hold off

grid on

axis([-3 3 -100 10])

title('Spectrum, Super Channel Formed by Merging 3 Channels, of 6-Path Non-Maximally Decimated Filter Bank')

xlabel('Frequency')

ylabel('Log Mag (dB)')