SGN 2016 – Multirate Signal Processing

Exercise 1
(2010)

1. Design a one stage decimator by factor $N = 12$ in Case A, B, and C with the following specifications:

$\delta_p = 0.01$
$\delta_s = 0.001$
$\alpha = 0.6$

Plot in each case the linear-scale magnitude response, the magnitude response in dB, impulse response, group delay, phase delay, impulse response, and pole-zero plot.

2. What are the requirements for the filter if we have sampling rate conversion with the factor $M / N = 4/5$? How about if the factor is 5/4? Design filters in both cases with the following specifications:

$\delta_p = 0.01$
$\delta_s = 0.001$
$\alpha = 0.7$

Plot in each case the linear-scale magnitude response, the magnitude response in dB, impulse response, group delay, phase delay, impulse response, and pole-zero plot.

SGN 2016 – Multirate Signal Processing

Exercise 2
(2010)

1. Design a two-stage decimator in Case A, B, and C with the following specifications:

$\delta_p = 0.01$
$\delta_s = 0.001$
$\alpha = 0.8$
$N = 8$, $(N_1 = 4$, $N_2 = 2)$

N_1 and N_2 are the decimator factors for stage 1 and stage 2, respectively. Plot the linear-scale amplitude response, impulse response, group delay and pole-zero plots. Plot also the amplitude response of the single-stage equivalent.

2. Design a one-stage decimator in Case A with the specifications given in task 1. Plot the linear-scale amplitude response, impulse response, group delay, and pole-zero plots. Compare the one-stage design with the two-stage design.

From /home/ts/matlab/multirate the files decfir.m, subdecfir.m and zeroam.m are available for this exercise.

SGN 2016 – Multirate Signal Processing

Exercise 3
(2010)

1. Design a half-band FIR filter (Lecture notes: Part II.F, pp. 224-233) with the following specifications:

$\omega_p = 0.35\pi$
$\delta_p = \delta_s = 0.001$

Plot the frequency response, impulse response, group delay and the pole-zero plot of the filter. Write a Matlab function $h = \text{halfband}(\omega_p, \delta)$.

2. Implement a lowpass-highpass filter pair (the half-band FIR filter designed in the previous task). Test your implementation with some signals.

3. Use the half-band filter as a decimator. Implement it with the polyphase structure (Lecture notes: Part II, pp. 19-27). Test your implementation with some signals.

SGN 2016 – Multirate Signal Processing

Exercise 4
(2010)

1. Copy files from folder `/home/ts/matlab/nyquist` into one of your folders. Running matlab program `nykki.m` (all other files in this folder are used by the main program) enables one to design separable and non-separable Lth band filters. Following two facts should be pointed out: First, in all cases use passband weight 0.00001 and stopband weight 1. Second, the number of iterations for single stage design is 1. For a multi-stage design, few iterations should be used.

After copying the files proceed with Task 2.

2. Design a single stage non-separable Lth-band FIR filter (Lecture notes: Part II, pp. 151-225) with the following specifications:

$L = 8$
$\rho = 0.2$
number of grid points = 20
$N = 74$
number of iterations = 1

in the above, N stands for the filter order. Repeat the design for minimax and least square design criteria. In addition to the figures generated by the program, generate a figure showing the zero-plot.

3. Design a two-stage non-separable Lth-band FIR filter by using the minimax design criteria having the same properties as the minimax filter designed in Task 2. What are the required orders for filters in Stage 1 and Stage 2 of the design? What is the overall order of the equivalent single-stage filter?

4. Design Lth-band FIR filters with different specifications of your choice. Optimize your filters for both the minimax and least-square sense. Create Lth-band filters for separable and non-separable case. Try to study the given code and use it or write your own code.

SGN 2016 – Multirate Signal Processing

Exercise 5
(2010)

1. Copy files from folder `/ts/matlab/ex5` into one of your folders.

2. Write a program for designing a cosine-modulated filterbank by using the windowing technique as described in the attached paper. The program should be written as a Matlab function that takes filter order (N), number of channels (M), and desired stopband attenuation (A_s) as input parameters and returns the prototype filter coefficients (h_p). ## KAISER WINDOW ONLY##

Design and compare filterbanks having various numbers of channels, stopband attenuations, and filter lengths. How do the reconstruction error and the aliasing error depend on these parameters?

For evaluating the designed filterbanks, you can use Matlab function `cosfb.m`.

3. Filterbank demo. Seven different two-channel filterbanks have been designed in advance:

- Quadrature mirror filter (QMF) bank with linear-phase subfilters (`qmf_lin.m`)
- Low-delay QMF bank with nonlinear-phase subfilters (`qmf_low.m`)
- Orthogonal filterbank, minimax design (`orth_mima.m`)
- Orthogonal filterbank, least-squared design (`orth_lea.m`)
- Biorthogonal filterbank with linear-phase subfilters (`biorth_lin.m`)
- Low-delay biorthogonal filterbank with nonlinear-phase subfilters (`biorth_low.m`)
- NPR biorthogonal filterbank with linear-phase subfilters (`biorthnpr_lin.m`)

Run all m-files.