UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN
Department of Electrical and Computer Engineering

ECE 498MH Signal and Image Analysis

Homework 7

Fall 2013

Assigned: Friday, November 1, 2013 — Due: Friday, November 8, 2013

Reading: SPF Chapter 12-3 covers almost this material, but for continous↔discrete rather than discrete↔discrete

Problem 7.1

Consider the signal $x[n] = \cos(0.9\pi n)$.

(a) The signal $x[n]$ is downsampled by a factor of two, resulting in $y[n]$. What is $y[n]$? Express your answer in the form $y[n] = \cos(\omega_a n)$, where $-\pi \leq \omega_a \leq \pi$.

(b) The signal $y[n]$ is upsampled by a factor of two, resulting in $z[n]$. What is $z[n]$? Express your answer in two different ways:

(1) Express your answer in the following form, where you need to tell me what is the value of ω_b:

$$z[n] = \begin{cases} \cos(\omega_b n) & n \text{ even} \\ 0 & n \text{ odd} \end{cases}$$

(2) Express your answer in the following form, where you need to tell me what are the values of ω_c and ω_d:

$$z[n] = \frac{1}{2}\cos(\omega_c n) + \frac{1}{2}\cos(\omega_d n) \quad \text{for all } n$$

(c) Let $f[n] = h[n] * z[n]$. What is the value of $h[n]$ that will produce $f[n] = \cos(0.1\pi n)$?

Matlab Exercises

Problem 7.2

Choose any image from the web – anything you like.

Read your image into matlab, e.g., using `A=double(imread(filename));` (the "double" command should make it possible to allow filtering later). Check its size, e.g., using `size(A)`. Downsample the image to roughly 50×50×3. Thus for example, if the original image was 480×640×3, the command `B = A(1:10:480,1:13:640,:)` would result in an image that was 48×50×3.

Upsample it by a factor of ten, with no interpolation. Thus for example, if your image is 48×50×3, you can use `C=zeros(480,500,:); C(1:10:480,1:10:500,:)=B;`.

Show your original, downsampled, and upsampled images, e.g., using

```
subplot(2,2,1);image(A);subplot(2,2,2);image(B);subplot(2,2,3);image(C);
```

You should find that the upsampled image has tiny dots of color surrounded by big swatches of black (you might even find it hard to see the tiny spots of color).

Perform piece-wise constant interpolation of your upsampled image. The command `h0=ones(1,10);` creates an impulse response, $h_0[n]$, that is equal to one for the first ten samples, and zero everywhere else. The command `D=conv2(conv2(C,h0,'same'),h0','same');` convolves image C with $h_0[n]$ along each row, then (because h0' is the transpose of h0) along each column. The keyword 'same' forces the output image to be exactly as many rows and columns as the input image. Show this image—you should be able to see the 10×10-pixel blocks of constant color.

Perform piece-wise linear interpolation of your upsampled image. Create the impulse response `h1=[0.1:0.1:0.9, 1, 0.9:-0.1:0.1]`, then plot it (`plot(h1);`) so that you see what it looks like. Linearly interpolate C in both rows and columns using something like `E=conv2(conv2(C,h1,'same'),h1','same');`. Show this image. You may be able to see linear interpolation of color over large blocks.

Perform sinc interpolation of your upsampled image. I recommend that you sample the sinc function at half-sample values. Create the half-sample indices using something like `n=[-49.5:0.5:49.5]`. Then create the sinc function using `hs=sin(pi*n/10)./(pi*n/10)`. Sinc-interpolate C in both rows and columns using something like `F=conv2(conv2(C,hs,'same'),hs','same');`. Show this image. The image should look "ideally smooth;" unfortunately, in practice, ideally smooth pictures may look a bit blurry.

Plot the impulse responses of your three filters in three sub-plots. Hand in a copy of this plot. You should see a rectangle in the first sub-plot, a triangle in the second sub-plot, and a sinc function in the third.

Create another figure with four sub-plots. In the first, plot one of the colors (red=1, green=2, blue=3) from one of the rows of image C that has some non-zero samples (e.g., `plot(C(501,:,2));` may have nonzero samples). You should see a non-zero sample once every ten samples, and the rest should be zeros. In the second sub-plot, plot the same row of D (e.g., `plot(D(501,:,2)))`. You should see piece-wise constant interpolation of the values in image C. In the third figure, plot the same row of image E; you should see piece-wise linear interpolation. In the fourth row, plot the same row of image F.