Worksheet for Fourier Series, Applied to a Photo Finish Problem

Andreas Stahel

Version of March 25, 2015

Getting Started

1. This mathematics class is building on the lecture notes on Fourier series, in particular the introductionary example of image processing. Make sure you have a printed copy of this section of the lecture notes on your desk. ☐
2. Find code on the web page `staff.ti.bfh.ch/sha1` using the link `schedule` and then the section *Files for the Lab Sessions*. Find the information for this lab in the directory `F2/Fourier`. Create a subdirectory for this lab and copy the files to your directory. ☐
3. You should start your own script file `goFourier.m` to run your codes. Then use your new command `goFourier` to run your code. ☐
4. At the end and for verification you may consult the file `PhotoFinishScan.m` which contains a complete solution for this lab. ☐

Starting up `MATLAB`

1. Start up `MATLAB`. You should work with your script file `goFourier.m` and not on the command line of `MATLAB`. ☐
2. Throughout this section you might have to use the built-in help by `MATLAB`. To read the documentation on the command `imread()`, just type `help imread` and read the result. ☐

Starting up *Octave*

1. Start up *Octave* and your favorite text editor. You should work with your script file `goFourier.m` and not on the command line of *Octave*. ☐
2. Throughout this section you might have to use the built-in help by *Octave*. To read the documentation on the command `imread()`, just type `help imread` and read the result. ☐

Analyzing a Photo Finish Scan

1 Read the image in `Image.bmp` with the help of `a = imread('Image.bmp')`. The result will be the RGB values of a bitmap image, generated by a photo finish scan. The result `a` is a 3 dimensional matrix, use `size(a)`. The result of `a(3,5,:)` should be a vector of integers between 0 and 256, representing the Red, Green and Blue components of the pixel in row 3 and column 5.

2 Using `imagesc(a)` a colored picture should be shown on the screen.

3 Transform the color image to a gray-scale image by adding up the RGB values for each pixel in the image. Use `ad = sum(a,3)` and then `colormap(gray)` and `imagesc(ad)` to display the gray scale image.

4 Choose **one line** in the image to be analyzed with the help of a Fourier series e.g. line number $nn = 300$. Read out only the information of this line by `inline = ad(nn,:);`.

5 With `np=length(inline)`, *Octave*/`MATLAB` will count the number of data points in one line. Show a plot of the result by `plot(inline)` and make sure you recognize the periodic pattern.

6 The sampling interval was 0.5 ms . Construct a corresponding time vector t and plot the amplitude as function of time. Hint: `t = dt*[0:np-1]`. Label the axis with the help of `xlabel()` and `ylabel()`.

Filtering with Fourier

1 Apply the command `fft()` and plot the absolute value of the Fourier coefficients, e.g. `c=fft(inline);`, followed by `amp=abs(c)/np;` and `plot(amp)`. With the division by `np` you will obtain the actual amplitude of the sin/cos-function with the corresponding frequency. Make sure you recognize the symmetry in the spectrum and the peaks caused by the surrounding lights in the stadium.

2 Determine the time T of measurement and construct a vector of frequencies. Then plot the amplitude as function of the frequency, e.g. by `plot(freq,amp)`. The scaling in vertical direction is not very useful since it is dominated by the DC component. Use the command `axis()` to choose a better scaling.

3 Generate a new plot with only a restricted domain of frequencies shown, e.g. only frequencies smaller than 300 Hz. You may determine the index of the limiting frequency graphically (`plot(amp)`) or use the command `find()`.

4 Identify the frequencies of the undesired perturbation graphically or with `find()`.

5 Set the corresponding coefficients in the Fourier coefficients to 0. Then use `ifft()` to reconstruct a new, improved signal. Compare the original and your modification graphically.