UNIVERSITY COLLEGE DUBLIN

SCHOOL OF ELECTRICAL & ELECTRONIC ENG.

EEN30110 SIGNALS AND SYSTEMS

Experiment 3SS2

TRIGONOMETRIC FOURIER SERIES

1. Objective:

To investigate the trigonometric Fourier series.

2. Background Information:

See information in handout for laboratory 3SS1 and module notes.

The Matlab command which will approximately determine the Fourier coefficients of a periodic signal is **fft**.

Consider the periodic continuous-time signal with period of 1 sec defined by

$$f(t) = \begin{cases} 1 & 0 \le t < 0.5 \\ 0 & 0.5 \le t < 1 \end{cases}$$

over one period. To generate the approximate Fourier coefficients of this periodic square-wave one must first generate a vector of samples uniformly spaced over one period of the signal. To achieve good speed of execution of the command it is better to ensure that the number of samples is a power of 2. The following code generates 256 samples:

```
>> N = 256;
>> f = [ones(1,N/2) zeros(1,N/2)];
```

where / is the Matlab command for division. Now the command

```
>> FF = fft(f);
```

generates the sequence of numbers called F_n in the module notes. From the notes the Fourier coefficients are approximately F_n/N, at least for small *n*.

```
>> Fcoeff = FF/N;
```

The vector Fcoeff is now approximately the vector of Fourier coefficients. The first term is the zeroth coefficient, i.e. the DC or average value. One may determine this directly as follows:

```
>> Fcoeff(1)

ans =

   0.5000
```

where the command Fcoeff(1) just picks out the first element of vector Fcoeff. Of course it makes sense that the average value of the square-wave should be a half. The Fourier coefficients are complex numbers in general. In this case for example:

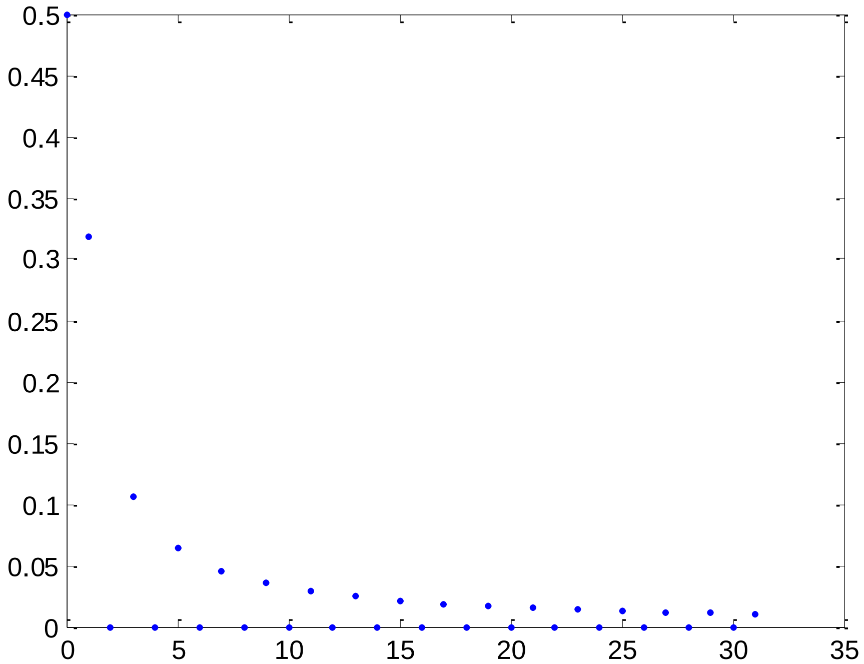
```
>> Fcoeff(2)

ans =

   0.0039 - 0.3183i
```

Note that by default Matlab uses i to denote the square root of -1. Command **help i** gives more information. It is not possible to plot complex data directly using the plot command in Matlab.

Consider the command

```
>> plot([0:31], abs(Fcoeff(1:32)), '. ')
```

which produces the following figure



There are a number of observations to be made concerning this plot command. The command [0:31] generates the vector of numbers 0, 1, 2, 3, ... , 31. If the increment in the definition of a vector is omitted it is assigned the default value of 1, so the command [0:31] and the command [0:1:31] generate the same vector. As noted in the table of basic functions the command **abs** determines the moduli of the elements of the vector, i.e. produces real data which the **plot** command can plot. The command Fcoeff(1:32) generates a sub-vector of the vector Fcoeff comprising the first 32 elements of Fcoeff, i.e. those elements with indices 1 through to 32. The reason for throwing away the rest of vector Fcoeff is that the approximation involved in determining the Fourier coefficients is only accurate for the first eighth or so of the coefficients. The additional term '.' in the plot command results in the plotting of the data using dots only, i.e. prevents the rather dubious default practice of joining up of the data points by lines. From the plot we see that the zeroth coefficient has modulus 0.5 and that the second, fourth, sixth, *etc* coefficients are zero which is in agreement with the formula obtained in module notes.

Consideration of the module notes reveals that, given samples of the signal, the **fft** command will approximately determine the Fourier coefficients (or at least the first few coefficients). However, the command will assume that the sample times are uniformly distributed over one period of the signal *starting at* $t = 0$. What if the sample times are uniformly distributed over one period starting at some other time? To be specific suppose they are uniformly distributed over one period starting at $t = -\frac{T}{2}$. In this case the first sample, which **fft** assumes is the value of the signal at $t = 0$ is actually $f\left(-\frac{T}{2}\right)$, that is to say is the value of the related signal $f\left(t-\frac{T}{2}\right)$ at time $t = 0$. Similar comments apply to the second sample, the third sample, *etc*. Accordingly the Fourier coefficients approximately calculated by **fft** will be the coefficients, not of the signal $f(t)$, but of the related signal $f\left(t-\frac{T}{2}\right)$. But if $f(t) = \sum_{n=-\infty}^{\infty} c_n \exp(jn\omega_0 t)$, where the coefficients c_n are the Fourier coefficients of $f(t)$ which we seek, then:

$$f\left(t-\frac{T}{2}\right) = \sum_{n=-\infty}^{\infty} c_n \exp\left(jn\omega_0\left(t-\frac{T}{2}\right)\right) = \sum_{n=-\infty}^{\infty} c_n \exp(-jn\pi)\exp(jn\omega_0 t) = \sum_{n=-\infty}^{\infty} (-1)^n c_n \exp(jn\omega_0 t).$$

So the n^{th} Fourier coefficient of the signal $f\left(t-\frac{T}{2}\right)$, which we calculate, equals $(-1)^n c_n$ where c_n is the n^{th} Fourier coefficient of the signal $f(t)$ which we seek. In this case the Fourier coefficients calculated by **fft** are correct for n even, but must be negated for n odd to yield the actual coefficients required. Similar comments and a similar corrective analysis may be provided for the situation where the first sample is at some other time, i.e. at a time other than $t = 0$ or $t = -T/2$.

3. Problems

1. The signal $q(t)$ is periodic of period 3π. Over one period it is given by:

$$q(t) = \frac{t^2}{\pi^2} \qquad -2\pi \le t < \pi$$

   Plot five cycles of the signal running over the time interval -8π to 7π.

2. Analytically determine the trigonometric Fourier series of this signal $q(t)$.

3. Numerically determine the first seven Fourier coefficients of this signal $q(t)$.

4. Using this result write out the first thirteen terms of the trigonometric Fourier series of this signal q(t).

5. The voltage signal $f(t)$ is periodic and has frequency 500Hz. Over one period it is given by:

$$f(t) = \begin{cases} 0.8(2000)^2 t(0.001-t) & 0 \le t < 0.001 \\ 0.8(2000)^2 t(0.001+t) & -0.001 \le t < 0 \end{cases}$$

Plot three cycles of the signal for t in the range -0.003 to 0.003. Confirm that the signal *looks like* a sinusoid of amplitude 0.8 V and frequency 500 Hz by plotting three cycles of a 0.8V amplitude, 500 Hz sinusoid in red or green *on the same axis*. You will need to use the **hold on** command to ensure that the sinusoid is directed to the same figure. You will also need to include the character string **'r'** or **'g'** to ensure that the data plots in red or green. As usual the command **help** will give further information.

6. Generate a vector of samples of the signal $f(t)$ comprising 512 consecutive cycles. What is the time duration of this signal? If you have headphones (and if the version of Matlab which you are running supports it) you may listen to this signal using the **sound** command. The **help** command tells you how to do this. You will find that you need to give the **sound** command both the samples of the signal to be sounded and the frequency at which those samples were taken. This frequency is the reciprocal of the time-gap between your samples. Likewise generate the samples of 512 consecutive cycles of the 0.8 V amplitude 500 Hz sinusoid and listen to this, again using the sound command. You should find that not only does signal $f(t)$ look like a 0.8 V amplitude, 500 Hz sinusoid, *it also sounds like one*.

7. Numerically determine the first ten Fourier coefficients of the signal $f(t)$ and confirm that, notwithstanding the facts that it looks and sounds like a sinusoid, *it is not a sinusoid*.

8. By looking at the relative values of the first few Fourier coefficients offer an explanation as to why the signal looks and sounds sinusoidal.

9. The audio file Audio1.wav comprises samples of three notes played on a Hohner harpsichord. Once you have downloaded the file Audio1.wav from Blackboard you should be able to use the **wavread** command to import the downloaded file into MATLAB from wherever you placed it. Subsequently you can play this file using the **sound** command (provided your version of MATLAB supports this command). Plot the samples of the downloaded signal and confirm that there are clearly three distinct audio signals in succession. The audio signal for each of the notes is very clearly divided into three phases, an early phase (the attack phase), a middle phase (the sustain phase) and an end phase (the decay phase). By plotting a suitable number of the samples from the sustain phase of the first note confirm that the signal is approximately *periodic* for the duration of this phase. Find the first five terms of the trigonometric Fourier series of this sustain-phase periodic signal. Find the fundamental frequency in hertz of this signal and hence determine what note was played (if you do not have a knowledge of music you may need to consult websites to find the mapping from notes to frequencies and *vice versa*).