Signal Processing I

Laboratory

Investigating the Fourier Series & Filtering.

Part I

The following Matlab function (see below) plots the Fourier synthesis of the function $f(t) = t$ for $-\pi < t < \pi$. That is

x(t), in here it's f(t)

a) Calculate mathematically the Fourier coefficients and verify that the expression between the *for loop* is correct.
b) Why are there only Sine terms? Why there is no DC value?
c) Calculate the Fourier series for $f(t) = A + t$ for $-\pi < t < \pi$, where A is a constant. What are the differences in the coefficients compared to part a)? Plot these coefficients for both cases so that you can see the differences and similarities.
d) Modify the program and plot this new series that you obtained in part c)
e) Plot and Calculate the Fourier coefficients for $f(t) = t\,\mathrm{Cos}(t/2)$ for $-\pi < t < \pi$. Plot $f(t)$.
f) Do the coefficients in e) have higher amplitude than in a)? Write an explanation of these differences. Hint, remember that the signal $f(t)$ is not periodic but you are assuming periodicity and your series approximation is only valid for $-\pi < t < \pi$. Therefore, regardless of $f(t)$, this signal repeats itself every 2π.

```
function fourser(N);

% N = number of coefficients to be used (input)
% Plot the Fourier series for f(t) = t for -pi<t<pi

t = -pi:pi/200:pi;

fsm = zeros(N-1,length(t));

for i = 1:(N-1)      cn =
2*((-1)^(i+1))/i;
  fsm(i,:) = cn*sin(i*t);
end

fsm2 = sum(fsm);

plot(t,t,t,fsm2)
title(['Fourier Series', ' N = ', num2str(N)])
xlabel('Time'), ylabel('Amplitude')
legend('Function f(t)', 'Fourier series approx')
```

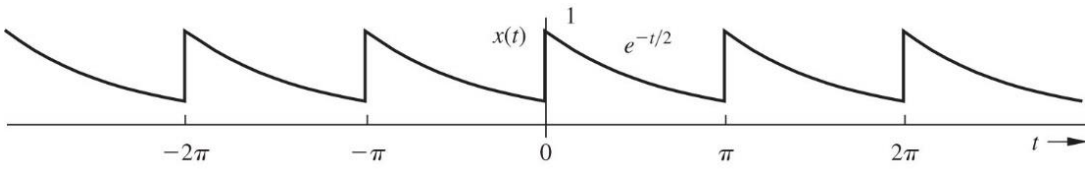
Part II

As you see in Table 6.1, to calculate the different versions of the Fourier series of a periodic signal requires a number of integrals.

TABLE 6.1 Fourier Series Representation of a Periodic Signal of Period T_0 ($\omega_0 = 2\pi/T_0$)

Series Form	Coefficient Computation	Conversion Formulas
Trigonometric	$a_0 = \frac{1}{T_0}\int_{T_0} f(t)\,dt$	$a_0 = C_0 = D_0$
$f(t) = a_0 + \sum_{n=1}^{\infty} a_n \cos n\omega_0 t + b_n \sin n\omega_0 t$	$a_n = \frac{2}{T_0}\int_{T_0} f(t)\cos n\omega_0 t\,dt$	$a_n - jb_n = C_n e^{j\theta_n} = 2D_n$
	$b_n = \frac{2}{T_0}\int_{T_0} f(t)\sin n\omega_0 t\,dt$	$a_n + jb_n = C_n e^{-j\theta_n} = 2D_{-n}$
Compact trigonometric	$C_0 = a_0$	$C_0 = D_0$
$f(t) = C_0 + \sum_{n=1}^{\infty} C_n \cos(n\omega_0 t + \theta_n)$	$C_n = \sqrt{a_n^2 + b_n^2}$	$C_n = 2\lvert D_n\rvert \quad n \geq 1$
	$\theta_n = \tan^{-1}\left(\frac{-b_n}{a_n}\right)$	$\theta_n = \angle D_n$
Exponential		
$f(t) = \sum_{n=-\infty}^{\infty} D_n e^{jn\omega_0 t}$	$D_n = \frac{1}{T_0}\int_{T_0} f(t)e^{-jn\omega_0 t}\,dt$	

Matlab provides functions for solving, plotting, and manipulating symbolic math equations. You can analytically perform differentiation, integration, simplification, transforms, and equation solving.
In order to illustrate this, we will solve Example 6.1 shown in the figure below.



First, create the symbolic variables *t*,

```
syms t
```

Then assign the expression to *x*:

```
x = exp(-0.5*t)
```

Need the expressions now for the Fourier series coeffcients.

First, for the value π notice the difference in Matlab. Type and execute the following in the Comand window

```
>> sin(sym(pi))
>> sin(pi)
```

What values do you have for both operations?

Continuing with the coefficients and calculating the Trigonometric ones

```
syms n
```

Start with a_o and do the integral

```
>> ao = int((1/pi)*x, 0 , pi)
```

Evaluate your result

```
>> ao_coeff = eval(ao)
```

Continue with a_n

```
>> an = int(((2/pi)*x*sym(cos(2*n*t))),t, 0,pi)
```

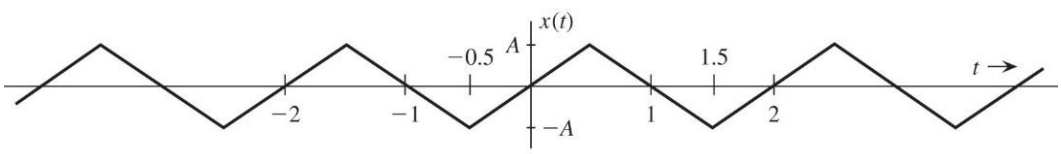
If you want to see the first 10 (a_n and b_n coefficients) for n = 1: 10 do the following

```
>> n=1:10;
>> an_coeffs= eval(an)
>> bn = int(((2/pi)*x*sym(sin(2*n*t))),t, 0,pi)
>> bn_coeffs = eval(bn)
```

Those are for the Trigonometric. The compact trigonometric as calculated in your textbook are:

```
>> cn = sqrt(an_coeffs.^2 + bn_coeffs.^2)
>> subplot(211)
>> inde = 0:10;
>> stem(inde,[ao_coeff cn])
>> title('Example 6.1')
>> xlabel('Index (multiple of fundamental frequency)')
>> ylabel('cn')
>> subplot(212)
>> angles = atan(-bn_coeffs./an_coeffs)
>> stem(inde,[0 angles])
>> title('Phase')
>> xlabel('Index (multiple of fundamental frequency)')
>> ylabel('degrees, radian')
```

Please do the same for the signal in Example 6.2 shown in the next figure.



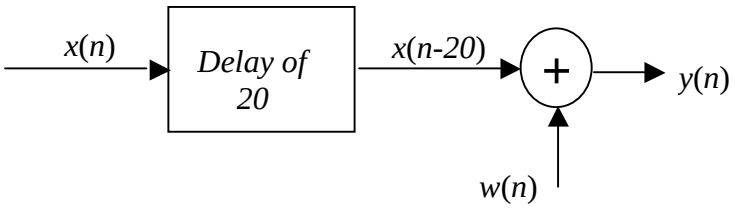
Repeat Example 6.2 when $x(t)$ is $x(t\text{-}0.5)$ and $x(t\text{+}0.5)$. Comment on the symmetry of the original 6.2 problem and the new time shifted versions and the coefficients you obtained.

Part III

Create a signal sequence of length 200 that its first 13 samples are:

```
x(1:13)=[1 1 1 1 -1 -1 1 1 -1 1 -1 1];
x(14:200)=0;
```

Then, create a Gaussian random sequence (use the command "randn" in MATLAB) with zero mean and a variance that is being determined by the Signal to Noise Ratio (SNR). Now, write a simple script that generates the output of the following system.



Furthermore, calculate the crosscorrelation of the output of the above system and the original signal and plot it.

Calculate the variance of the random signals for the SNR values of 10, 5, 1 and 0.1 and observe the results. What can you say about correlations and detection of signals buried in noise?

Handout:

1. Plot of the original signal and the output signal (use subplot to save paper!).
2. Plot of the crosscorrelation of the y(n) and x(n) for only 50 samples chosen from the middle of $r_{xy}(n)$ at different SNR values and the estimated delay of the system from the crosscorrlation function.

.

Part IV

Ask the TA for the file seashell.wav. (you can pick any other wave files that are in the windows directory of your lab computer.) You can load this file to MATLAB by the command: `[x, fs]=wavread('seashell');`

And then you can play it by command: `soundsc(x,fs)`

To see what filtering convolution with a simple impulse response looks like try the following:

soundsc(conv(x,[1 -1]),fs); and `soundsc(conv(x,[1 1 1 1 1 1 1]),fs);`

What difference do you hear? Are they making the sound lower in pitch or higher? You may try different length for $h(t)$ and see its effect.