

9.7 Applications - *Optional Material?*

9.7.1 Cloning of Signals

9.7.2 Harmonic Analysis

Even though spectral analysis, which is based upon the discrete Fourier transform, is a readily used technique for analyzing the spectral content of a signal, one might only be interested in the spectral content for particular frequencies that are not included in the discrete set of frequencies provided by the Fourier transform. This can happen when the number of samples is large and the number of frequencies of interest is much smaller. In this section we address the approximation of our signal, or time series, to a sum over a small set of frequency components. The frequencies of interest are known but the associated amplitude is not known. This will be accomplished using the method of least squares.

We begin by reviewing the method of least squares for determining the best fit of a line. The equation of a line only has two unknown parameters, the slope and the intercept. An understanding of the derivation for this simpler approximation should make that for the harmonic analysis more transparent.

We begin with a set of data, presented in the usual pairs (x_i, y_i) for $i = 1, \dots, N$. We are interested in finding the best approximation, in some sense, of this data by some function. In the case of linear regression, we seek a linear relationship of the form $y = ax + b$. Though this line is not expected to agree with the data, it is expected to be as “close as possible”.

What does one mean by “as close as possible”? We could mean that the total distance between the known data points and the line is as small as possible. Though there are many ways we could quantify this, the most natural would be to sum over the standard distance between the points for all data points. Thus, we would look at an expression like

$\sum_{n=1}^N \sqrt{(x_i - x_i)^2 + (y_i - (ax_i + b))^2}$. However, since the first term under the square root vanishes and the square root only returns a positive number, we could instead just consider the expression $\sum_{n=1}^N (y_i - (ax_i + b))^2$. Making

the later as small as possible only gives the same result as for the first expression, but is easier to compute. Thus, this leads to the phrase “least square” regression.

We are interested in minimizing this quantity, which could be thought of as a variance about the straight line mean. We minimize this “error” by varying a and b . Thus, we have a two variable minimization problem. In order to minimize a function of one variable, we differentiate the function with respect to the variable and set it equal to zero to determine that value of the independent variable that yields a minimum. In this case, we need to simultaneously set the derivatives with respect to a and b to zero and find the values of a and b that solve the resulting equations.

Differentiating $\sum_{n=1}^N (y_i - (ax_i + b))^2$ with respect to a and b separately, gives

$$\begin{aligned} 0 &= 2 \sum_{n=1}^N (y_i - (ax_i + b))(-x_i) \\ 0 &= 2 \sum_{n=1}^N (y_i - (ax_i + b))(-1). \end{aligned}$$

Regrouping, we find that these are simultaneous equations for the unknowns:

$$a \sum_{n=1}^N x_i^2 + b \sum_{n=1}^N x_i = \sum_{n=1}^N x_i y_i$$

$$a \sum_{n=1}^N x_i + bN = \sum_{n=1}^N y_i.$$

Solving this system of equations gives expressions for a and b in terms of various sums over expressions the involving the data. This is the basis of the so called “best fit line” that is used in many programs, such as those in calculators and in MS Excel.

However, we are interested in fitting data to a more complicated expression than that of a straight line. In particular, we want to fit our data to a sum over sines and cosines of arguments involving particular frequencies.

We will consider a set of data consisting of N values at equally spaced times, $t_n = n\Delta t$, $n = 1, \dots, N$. We are interested in finding the best approximation to a function consisting of M particular frequencies, f_k , $k = 1, \dots, M$. Namely, we wish to match the data to the function

$$f(t) = A_0 + \sum_{k=1}^M [A_k \cos(2\pi f_k t) + B_k \sin(2\pi f_k t)].$$

The unknown parameters in this case are the A_k 's and B_k 's. We will not determine a function that exactly fits the data, as in the DFT case, but only seek the best fit curve to the data. [Note: for simplicity of the results, we have redefined the constant term A_0 .

In the following we will pick times $t_n = n\Delta t = \frac{nT}{N}$. Then $f_k t = f_k T \frac{n}{N}$. In the text, the authors let $\alpha_k = f_k T$, making $f_k t = \alpha_k \frac{n}{N}$. For now we will hold off from this notation.

To determine the unknowns, we need to minimize the “variance”

$$e^2 = \sum_{n=1}^N [y(t_n) - (A_0 + \sum_{k=1}^M [A_k \cos(2\pi f_k t_n) + B_k \sin(2\pi f_k t_n)])]^2.$$

To do so, we differentiate this expression with respect to all of the parameters. Namely, for a particular q , we have

$$0 = \frac{\partial e^2}{\partial A_q} = 2 \sum_{n=1}^N [y(t_n) - (A_0 + \sum_{k=1}^M [A_k \cos(2\pi f_k \frac{n}{N}) + B_k \sin(2\pi f_k \frac{n}{N})])](- \cos(2\pi f_q \frac{n}{N})), \quad q =$$

$$0 = \frac{\partial e^2}{\partial B_q} = 2 \sum_{n=1}^N [y(t_n) - (A_0 + \sum_{k=1}^M [A_k \cos(2\pi f_k \frac{n}{N}) + B_k \sin(2\pi f_k \frac{n}{N})])](- \sin(2\pi f_q \frac{n}{N})), \quad k =$$

As with the previous example of a best fit line, we have here a system of linear equations for the unknowns. We can rewrite this system in a more apparent form and then make use of matrix theory for solving systems of equations.

$$A_0 \sum_{n=1}^N \cos(2\pi f_q t_n) + \sum_{k=1}^M \left[A_k \sum_{n=1}^N \cos(2\pi f_k t_n) \cos(2\pi f_q t_n) + B_k \sum_{n=1}^N \sin(2\pi f_k t_n) \cos(2\pi f_q t_n) \right] = \sum_{n=1}^N y(t_n) \cos(2\pi f_q t_n)$$

and

$$A_0 \sum_{n=1}^N \sin(2\pi f_q t_n) + \sum_{k=1}^M \left[A_k \sum_{n=1}^N \cos(2\pi f_k t_n) \sin(2\pi f_q t_n) + B_k \sum_{n=1}^N \sin(2\pi f_k t_n) \sin(2\pi f_q t_n) \right] = \sum_{n=1}^N y(t_n) \sin(2\pi f_q t_n)$$

for $q = 1, \dots, M$.

Finally, we need to consider the equations for $q = 0$. In this case we have

$$0 = \frac{\partial e^2}{\partial A_0} = 2 \sum_{n=1}^N [y(t_n) - (A_0 + \sum_{k=1}^M [A_k \cos(2\pi f_k t_n) + B_k \sin(2\pi f_k t_n)])(-1)],$$

which reduces to

$$\sum_{n=1}^N [(A_0 + \sum_{k=1}^M [A_k \cos(2\pi f_k t_n) + B_k \sin(2\pi f_k t_n)])] = \sum_{n=1}^N y(t_n)..$$

We now note that we can write this system of $2M + 1$ equations in matrix form. We first define the $M \times N$ matrices C and S with elements

$$C_{qn} = \cos(2\pi f_k t_n), \quad q = 1, \dots, M, \quad n = 1, \dots, N$$

$$S_{qn} = \sin(2\pi f_k t_n), \quad q = 1, \dots, M, \quad n = 1, \dots, N.$$

The above sums over n can be written as the $M \times M$ matrix products with the entry in the q th row and k th column as

$$(CC^T)_{qk} = \left(\sum_{n=1}^N \cos(2\pi f_k t_n) \cos(2\pi f_q t_n) \right)$$

$$(CS^T)_{qk} = \left(\sum_{n=1}^N \sin(2\pi f_k t_n) \cos(2\pi f_q t_n) \right)$$

$$(SS^T)_{qk} = \left(\sum_{n=1}^N \sin(2\pi f_k t_n) \sin(2\pi f_q t_n) \right) ..$$

Here C^T is the transpose of C . Thus, $C_{ij}^T = C_{ji}$ and $(AB)^T = B^T A^T$.

Inserting these expressions into the system of equations, we have

$$A_0 \sum_{n=1}^N C_{qn} + \sum_{k=1}^M \left[(CC^T)_{qk} A_k + (SC^T)_{qk} B_k \right] = \sum_{n=1}^N C_{qn} y(t_n), \quad q = 1, \dots, M$$

$$A_0 \sum_{n=1}^N S_{qn} + \sum_{k=1}^M \left[(SC^T)_{qk} A_k + (SS^T)_{qk} B_k \right] = \sum_{n=1}^N S_{qn} y(t_n), \quad q = 1, \dots, M$$

and

$$A_0 N + \sum_{k=1}^M A_k c_k + B_k s_k = \sum_{n=1}^N y(t_n).$$

Finally, these equations can be combined by defining $y = \begin{pmatrix} y(t_1) \\ \vdots \\ y(t_N) \end{pmatrix}$,

$$Y = \begin{pmatrix} \bar{y} \\ Cy \\ Sy \end{pmatrix}, \quad Z = \begin{pmatrix} A \\ B \end{pmatrix}, \quad c_q = \sum_{n=1}^N C_{qn}, \quad s_q = \sum_{n=1}^N S_{qn} \text{ and}$$

$$D = \begin{pmatrix} N & c^T & s^T \\ c & CC & CS \\ s & CS & SS \end{pmatrix}.$$

Note that Y and Z are $2M + 1$ dimensional column vectors, c and s are M dimensional column vectors and D is a $(2M + 1) \times (2M + 1)$ matrix.

The system of equations in matrix form looks like $DZ = Y$. Solving for the unknowns in the column vector Z , we have $Z = D^{-1}Y$.

Implementation in Matlab

```
%
% Harmonic Analysis
%
% Enter Data in y
y=[7.6 7.4 8.2 9.2 10.2 11.5 12.4 13.4 13.7 11.8 10.1 ...
9.0 8.9 9.5 10.6 11.4 12.9 12.7 13.9 14.2 13.5 11.4 10.9 8.1];
N=length(y);
% Number of Harmonics Desired and frequency dt
M=2;
f=1/12*(1:M);
T=24;
alpha=f*T;
% Compute the matrices of trigonometric functions
n=1:N;
C=cos(2*pi*alpha'*n/N);
S=sin(2*pi*alpha'*n/N);
c.row=ones(1,N)*C';
s.row=ones(1,N)*S';
```

```

D(1,1)=N;
D(1,2:M+1)=c_row;
D(1,M+2:2*M+1)=s_row;
D(2:M+1,1)=c_row';
D(M+2:2*M+1,1)=s_row';
D(2:M+1,2:M+1)=C*C';
D(M+2:2*M+1,2:M+1)=S*S';
D(2:M+1,M+2:2*M+1)=C*S';
D(M+2:2*M+1,M+2:2*M+1)=S*S';
yy(1,1)=sum(y);
yy(2:M+1)=y*C';
yy(M+2:2*M+1)=y*S';
z=D (-1)*yy';

```

9.7.3 Audio Compression

9.7.4 Image Compression - JPEGs

9.7.5 Filtering and Windowing

We will consider the DFT of a windowed input sequence. We consider the time series x_n , $n = 0, 1, \dots, N - 1$. Though this is done in the time domain, we get similar effects when we consider filters in the frequency domain. We define the window (filter) coefficients w_n , $n = 0, 1, \dots, N - 1$. The windowed time series is then $x_n w_n$. The DFT of this series is

$$\sum_{n=0}^{N-1} w_n x_n e^{-2\pi i n m / N}.$$

There are several types of windowing functions. We will consider the following:

Rectangular Window: $w_n = 1, \quad n = 0, 1, \dots, N - 1.$

Triangular Window: $w_n = \begin{cases} \frac{n}{N/2}, & n = 0, \dots, \frac{N}{2} \\ 2 - \frac{n}{N/2}, & \frac{N}{2} + 1, \dots, N - 1. \end{cases}$

Hanning Window: $w_n = 0.5 - 0.5 \cos\left(\frac{2\pi n}{N-1}\right), \quad n = 0, 1, \dots, N - 1.$

Hamming Window: $w_n = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right), \quad n = 0, 1, \dots, N - 1.$

Let's plot these windowing functions and their DFTs.

```

N=64;
n=1:N;
pi=3.1415926535;
w1=ones(N);
w2(1:N/2)=(1:N/2)/(N/2);
w2(N/2+1:N)=2-(N/2+1:N)/(N/2);
w3=0.5-0.5*cos(2*pi*(1:N)/(N-1));
w4=0.54-0.46*cos(2*pi*(1:N)/(N-1));
subplot(2,2,1)
plot(w1)
title('Rectangular window')
subplot(2,2,2)
plot(w2)
title('Triangular window')
subplot(2,2,3)
plot(w3)
title('Hanning window')

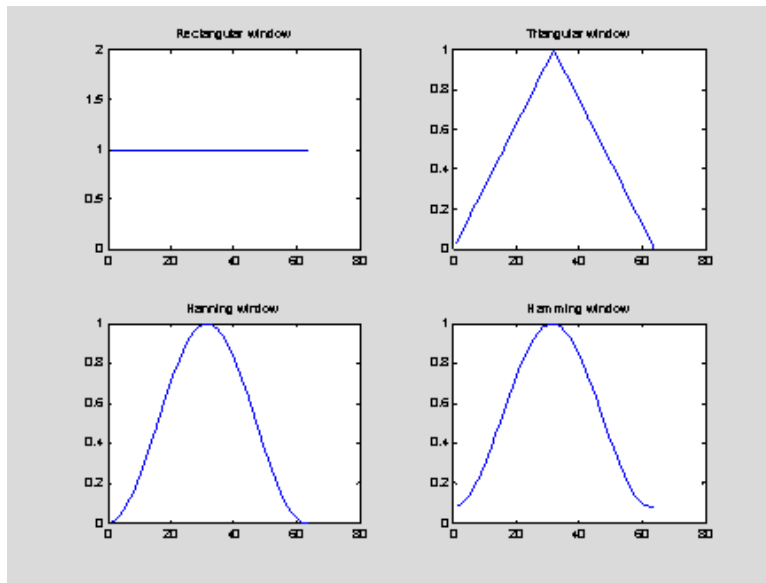
```

226

```
subplot(2,2,4)
```

```
plot(w4)
```

```
title('Hamming window')
```



DFTs for these windows can also be determined. This is important because the convolution of these functions with the DFT of the original signal is the same as the above DFT of the windowed time series.

```
N=1024;
```

```
n=1:N;
```

```
pi=3.1415926535;
```

```
w1=ones(N);
```

```
w2(1:N/2)=(1:N/2)/(N/2);
```

```
w2(N/2+1:N)=2-(N/2+1:N)/(N/2);
```

```
w3=0.5-0.5*cos(2*pi*(1:N)/(N-1));
```

```
w4=0.54-0.46*cos(2*pi*(1:N)/(N-1));
```

```

subplot(2,2,1)
plot(abs(fft(w1)))
title('Rectangular - DFT')
subplot(2,2,2)
plot(abs(fft(w2)))
title('Triangular - DFT')
subplot(2,2,3)
plot(abs(fft(w3)))
title('Hanning - DFT')
subplot(2,2,4)
plot(abs(fft(w4)))
title('Hamming - DFT')

```

We can consider the continuous transforms for these functions.

$$H(\omega) = \int_0^T h(t)e^{-i\omega t} dt.$$

$$H(\omega) = \int_0^T e^{-i\omega t} dt = e^{i\omega T/2} \int_{-\frac{T}{2}}^{\frac{T}{2}} e^{-i\omega\tau} d\tau$$

$$|H(\omega)| = T \frac{\sin(\frac{\omega T}{2})}{\frac{\omega T}{2}} = \frac{N}{2} \frac{\sin(m\pi)}{m\pi}$$

$$\begin{aligned}
 H(\omega) &= \frac{2}{T} \int_0^{\frac{T}{2}} t e^{-i\omega t} dt + \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} (T-t) e^{-i\omega t} dt \\
 &= -\frac{2}{T} e^{-i\omega T/2} \left(\frac{e^{i\omega T/4} - e^{-i\omega T/4}}{\omega} \right)^2 \\
 &= \frac{T}{2} e^{-i\omega T/2} \left(\frac{\sin(\frac{\omega T}{4})}{\frac{\omega T}{4}} \right)^2
 \end{aligned}$$

$$\begin{aligned}
 \int_0^T (a + b \cos(\frac{2\pi t}{T})) e^{-i\omega t} dt &= a \frac{1 - e^{-i\omega T}}{i\omega} + \frac{b}{2} \int_0^T \left[e^{-i(\omega - \frac{2\pi}{T})t} + e^{-i(\omega + \frac{2\pi}{T})t} \right] dt \\
 &= a e^{i\omega T/2} \frac{2 \sin(\frac{\omega T}{2})}{\omega} - \frac{b}{2} e^{i\omega T/2} \sin(\frac{\omega T}{2}) \left[\frac{1}{\omega - \frac{2\pi}{T}} + \frac{1}{\omega + \frac{2\pi}{T}} \right] \\
 &= e^{i\omega T/2} T \frac{\sin(\frac{\omega T}{2})}{\frac{\omega T}{2}} \left[a - \frac{b}{2} \left(\frac{\omega^2}{\omega^2 - (\frac{2\pi}{T})^2} \right) \right]
 \end{aligned}$$

N=64;

k=1.000001;

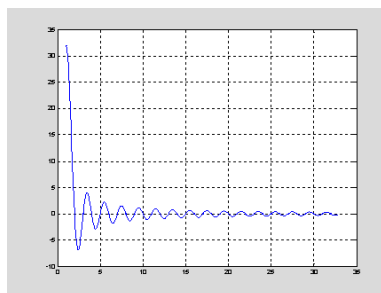
m=1:32;

plot(N/2*sin(pi*(k-m))/pi./(k-m),'.')

x=0.000001:0.1:32;

plot(x+1,N/2*sin(pi*(k-x-1))/pi./(k-x-1))

grid on



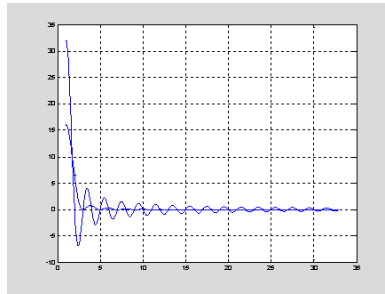
plot(N/4*(sin(pi*(m-k)/2))/pi./(m-k)*2, 2, '.')

hold on

```
plot(x+1,N/4*(sin(pi*(x)/2)/pi./(x)*2). 2)
```

```
grid on
```

```
hold off
```



```
hold off
```

```
A=0.5;
```

```
B=-0.5;
```

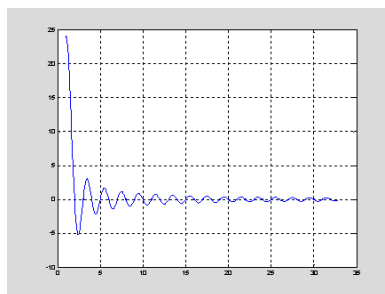
```
plot(N/2*sin(pi*(k-m))/pi./(k-m)*(A-B/2*((k-m). 2/((k-m). 2-1))),'.')
```

```
hold on
```

```
x=0.000001:0.1:32;
```

```
plot(x+1,N/2*sin(pi*x)/pi./x*(A-B/2*(x. 2/(x. 2-1))))
```

```
grid on
```



```
hold off
```

230

A=0.54;

B=-0.46;

plot(N/2*sin(pi*(k-m))/pi./(k-m)*(A-B/2*((k-m). 2/((k-m). 2-1))),'.')

hold on

x=0.000001:0.1:32;

plot(x+1,N/2*sin(pi*x)/pi./x*(A-B/2*(x. 2/(x. 2-1))))

grid on

