2 Linear stochastic modelling

Contents

2.1 ACF of uncorrelated sequences

2.1.1 Matlab xcorr function

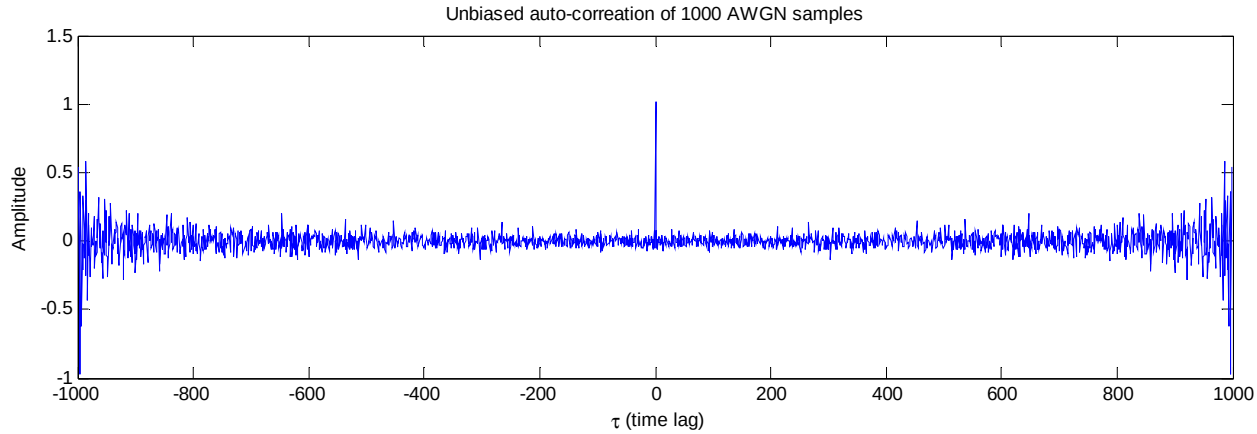
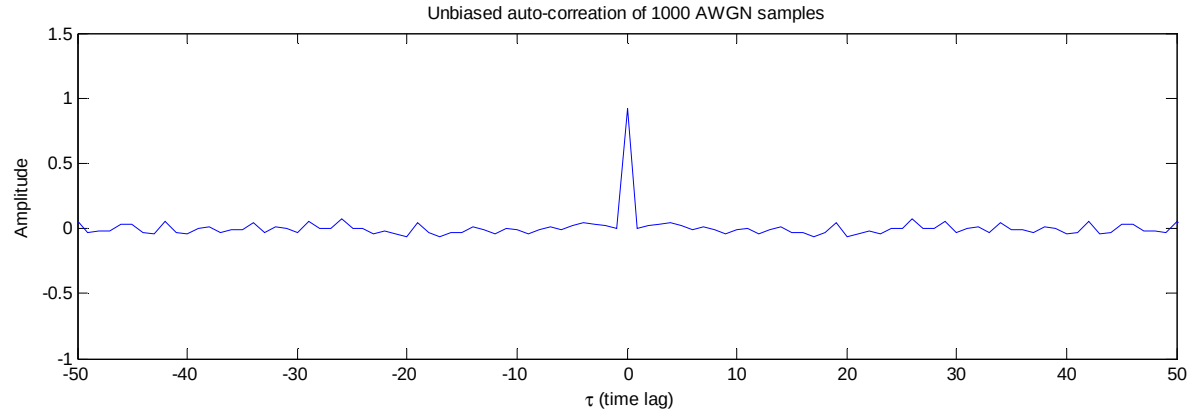


Figure 1 – Unbiased autocorrelation of white Gaussian noise. The equation for unbiased autocorrelation is $\hat{R}_x[l] = \frac{1}{N-|l|} \sum_{n=1}^{N-|l|} x[n]x[n+l]$

This is the autocorrelation function of 1000 AWGN samples. We notice the discrete delta at $\tau = 0$ (the peak of 1) but more importantly the cross-correlation estimate increases as $\tau = \pm 999$ is reached starting from around $\tau = \pm 500$. This is because past this limit less than half the values of x (the vector storing the instance of the 1000 AWGN samples) overlap due to being of length 1000. Thus once ± 999 is reached it is understandable that an amplitude increase may increase due to only one sample being available for comparison. We also notice that the function is symmetric around $\tau = 0$. This is due to the autocorrelation being the correlation of a signal with itself thus at equal distances from $\tau = 0$ the same values of x will be compared.

2.1.2 Zoomed in instance



In the above figured it can be noticed that the amplitude is reasonably constant for values other than $\tau = 0$. This is, as explained above, due to the samples always overlapping and "cancelling" each other out giving out a more reasonable estimate than for larger τ's.

The more discrete appearance of this graph is simply due to less samples being present.

2.1.3 Empirical bound on τ

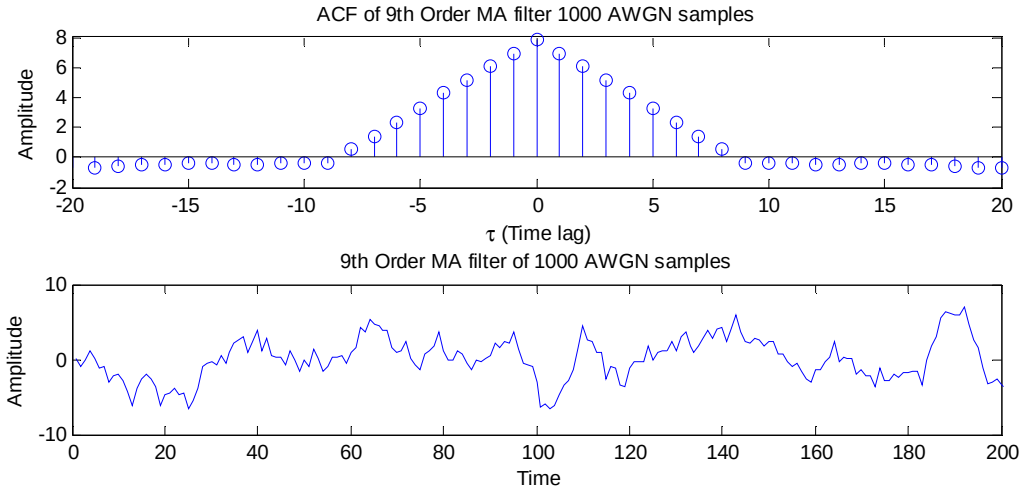
We notice that for $\hat{R}_X(\tau) = \frac{1}{N-|\tau|}\sum_{n=1}^{N-|\tau|} x[n] * x[n+\tau]$ values past a certain limit start to diverge and suggest that for large lags the signal is in fact similar to itself τ samples before. As the samples taken are AWGN this is not expected (while it is possible the probability is extremely low).

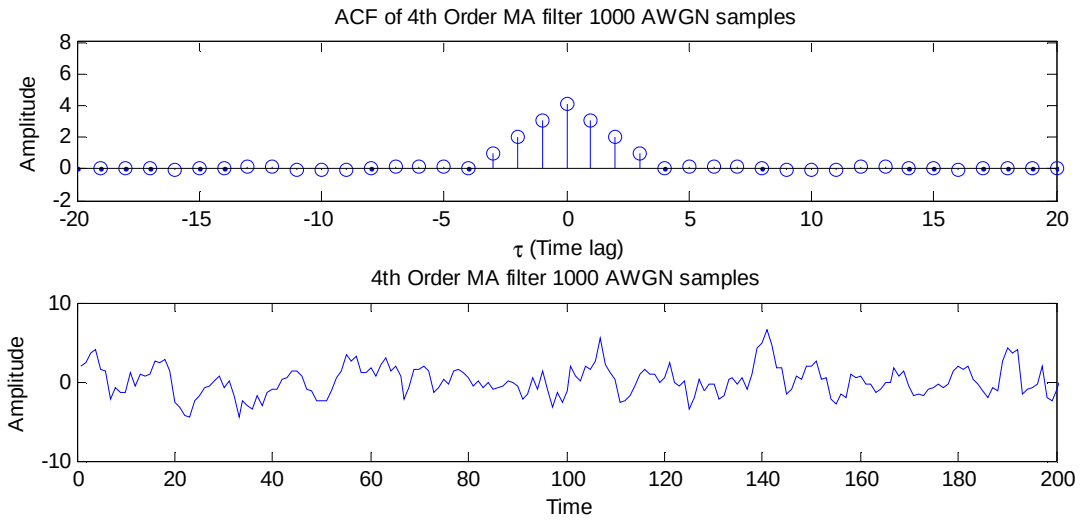
In the previous case for 1000 AWGN samples a value of $\tau = \pm 500$ was a good estimate i.e. $\pm\frac{N_{samples}}{2}$. This is because past this value more than half of the samples will be available for comparison and the probability of the series resembling itself increases. Looking at $\hat{R}_X$ we notice that the factor $\frac{1}{N-|\tau|}$ also suggests that as τ gets larger the accuracy decreases due to less samples being available, confirming that estimates do get worse as the lag increases.

This due to the probability of values being the same for a randomly generate process increasing when the number of samples decrease. For example if only 1 sample pair is used there is a 50% chance of both of those to overlap, for 2 sample pairs there is

2.2 ACF of filtered sequences

2.2.1 Moving Average Filter





The graphs above show use the effects of passing AWGN samples through an n[th] order MA filter. In this case orders of 4 and 9 were chosen. As we can see the ideal ACF function for AWGN passed through a MA filter of order N is $ACF_{ideal}(\tau) = N \cdot \Lambda\left(\frac{\tau}{N}\right)$, where $\Lambda(t)$ the triangle function is. This is due to the way the moving average filter works –which is by taking the coefficients it has and recursively adding up past elements (the equation for FIR filter of order N is $y[n] = \sum_{i=0}^{N} b_i x[n-i]$). Thus by taking the average of elements at relative index 1 to N will be correlated to each other, with a higher correlation for being closer to the original element, explaining the triangle shape of the ACF.

The effect of the signal on its output is thus to smooth it out, we can see this from the fact that the 4[th] order MA signal looks more jagged than the 9[th] order one.

If we have an MA of order equal to the number of samples (N) then the value of average could be calculated by $y[N] = \sum_{i=0}^{N-1} b_i x[N-i]$ if $b_i = \frac{1}{N}$ as this would essentially become the equation of average $y[N] = mean(y) = \frac{1}{N}\sum_{i=0}^{N-1} x[N-i]$.

2.2.2 Stochastic process correlation

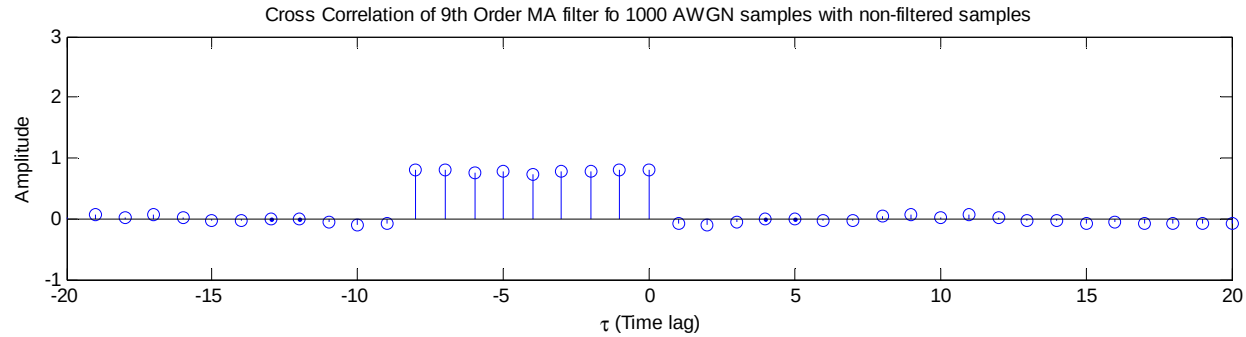
If Y_n is the result of a stochastic process, X_n is obtained from an uncorrelated process and we have

$$R_Y(\tau) = R_X(\tau) * R_h(\tau)$$

Then we have that $R_Y(\tau) = R_h(\tau)$ due to $R_X(\tau) = \delta(\tau)$, which is the result of the autocorrelation of an uncorrelated process (and any function convoluted by $\delta(\tau)$ is equal to itself). Thus the autocorrelation of Y represents the autocorrelation of the impulse response. In the case of processes taken from 2.2.1 the result is, as expected, the function $N \cdot \Lambda\left(\frac{\tau}{N}\right)$.

2.3 Cross-correlation function

2.3.1 Cross correlation of X and Y from 2.2



Above is the plot of the cross correlation of X and Y. We notice that the result is the impulse response (here it is reversed in time but this depends on if xcorr(y,x, 'unbiased') or xcorr(x,y, 'unbiased')is used).

This is the expected response as the cross correlation X and Y is equal to $R_{XY}(\tau) = h(\tau) * R_X(\tau)$ which, by using the fact that X is AWGN which is uncorrelated, is equal to $R_{XY}(\tau) = h(\tau)$.

In this case $h(\tau) = \sum_{i=0}^{N} \delta(\tau - i)$, which is what we have for the observed instance.

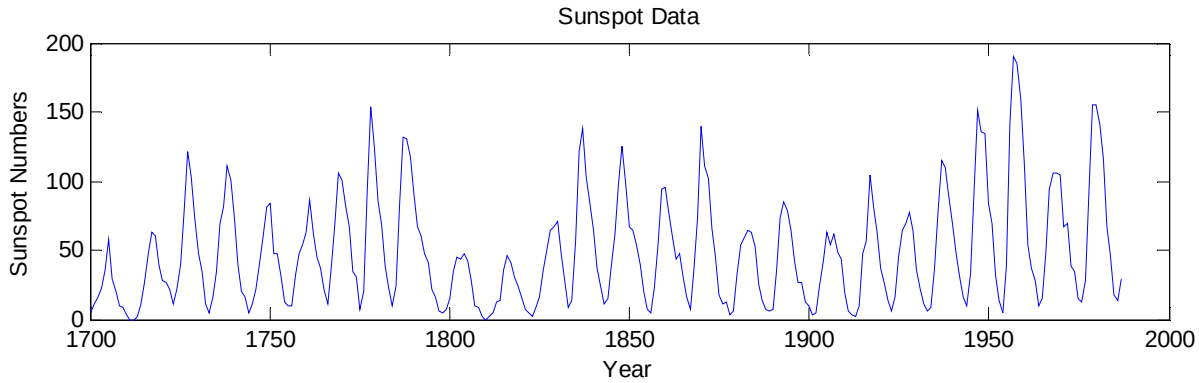
2.3.2 System estimation from cross correlation function

By applying the CCF on two functions, one which we know is the result of a FIR (IIR or any other LTI system would also work) filter and the other function being the original function , the impulse response of the system can be obtained –though this is only possible if the original process is uncorrelated (i.e. we must have that $R_X(\tau) = \delta(\tau)$).

From the impulse response we can deduce the order by the number of delta function and if the system is an FIR the coefficients can be estimated.

2.4 Autoregressive modelling

2.4.1 Sunspot Data



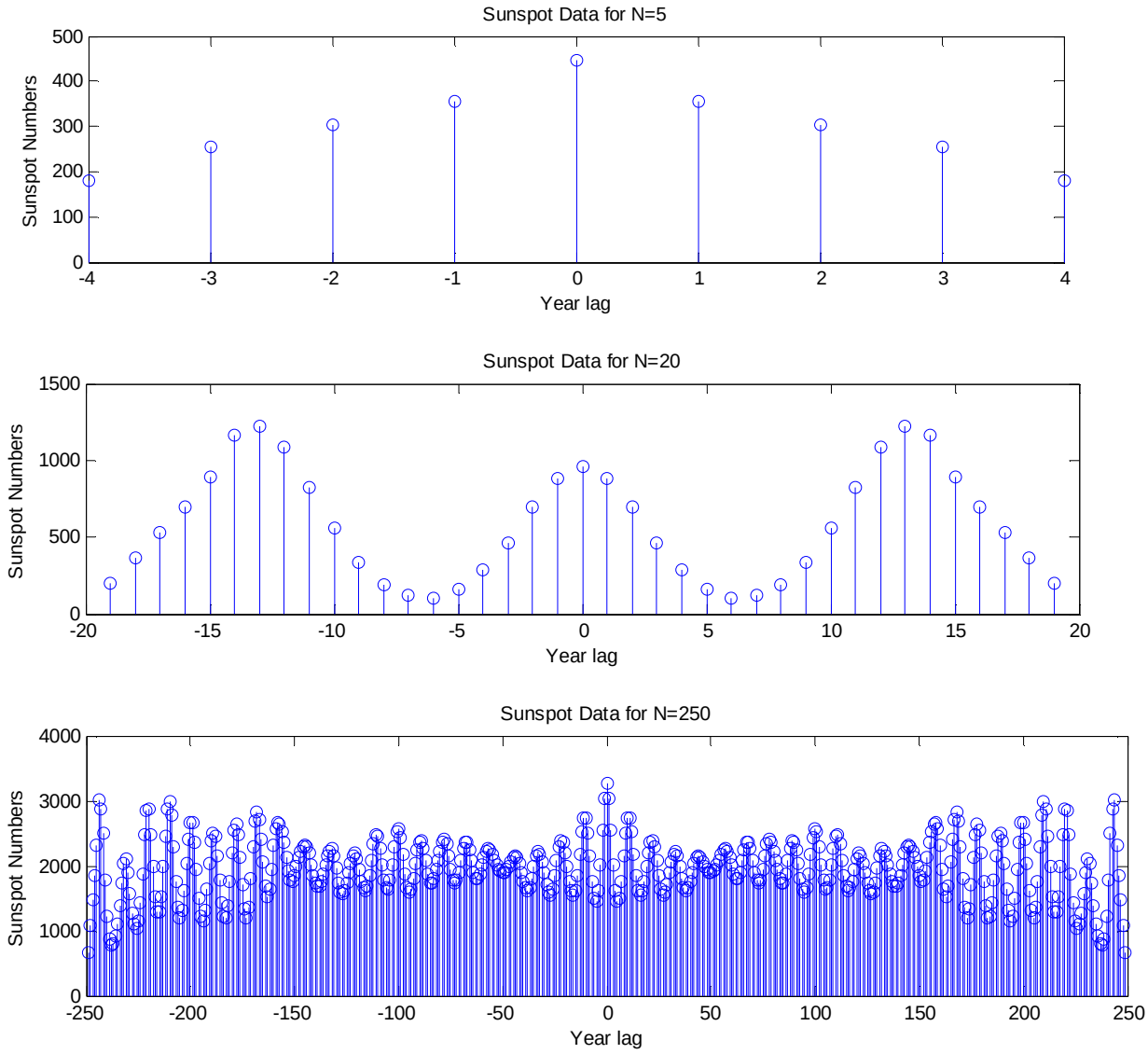
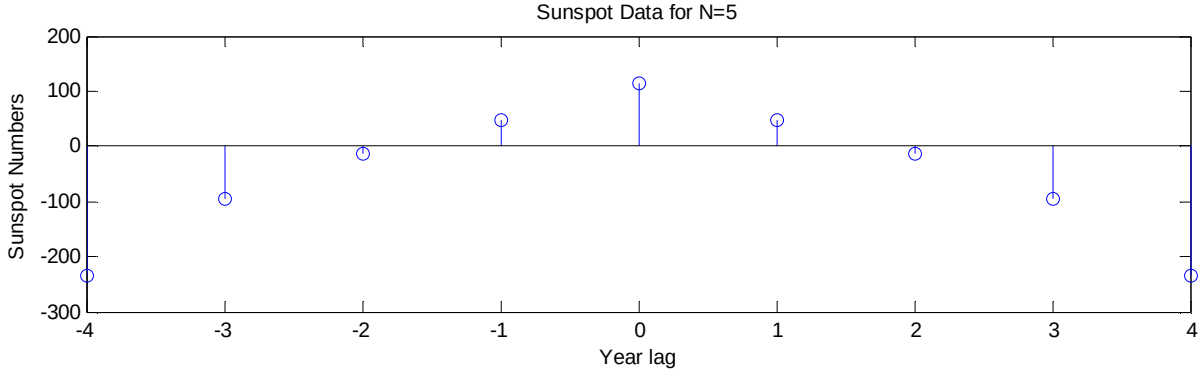


Figure 2 – Autocorrelation of sunspot data for various N values.

This is the sunspot data plotted in time as well as the ACF of the data for various N. We can notice that for very small N (N=5) it is difficult to see that the data is recurring every few years, whereas the shape for N=20 shows how for every 13 years or so sunspot data repeats. When contrasting the ACF for N=20 to N=250 it can be noticed that the peaks of the data are higher originally but once a longer N is applied these apparent peaks do not exist. This is due to the issues with observing/interpreting data past $\pm\frac{N}{2}$ as discussed in section 2.1.

2.4.2 Zero mean data ACF



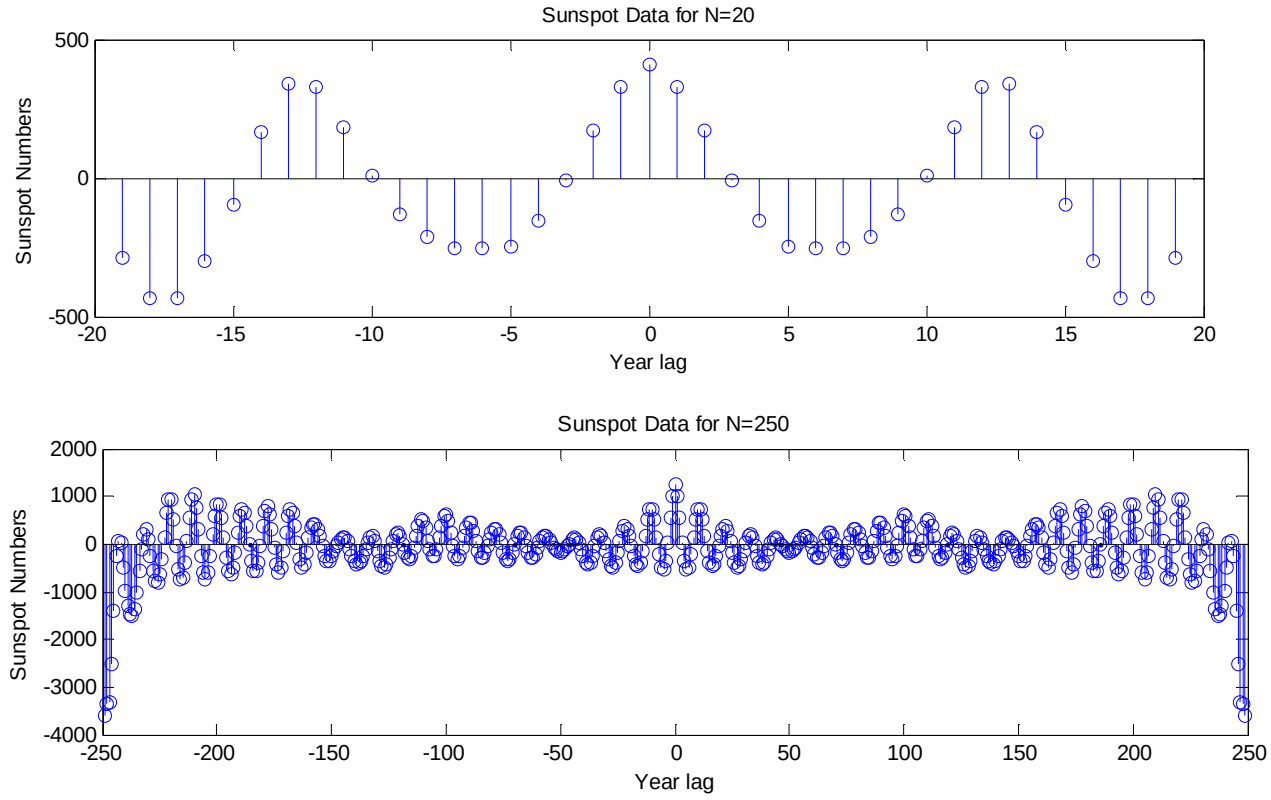
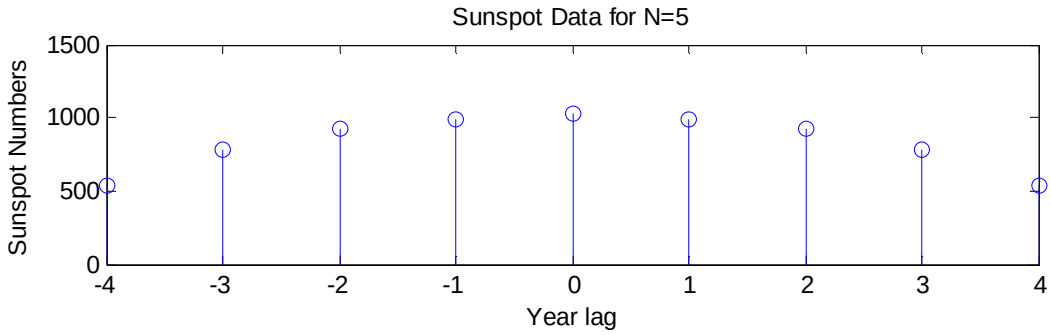
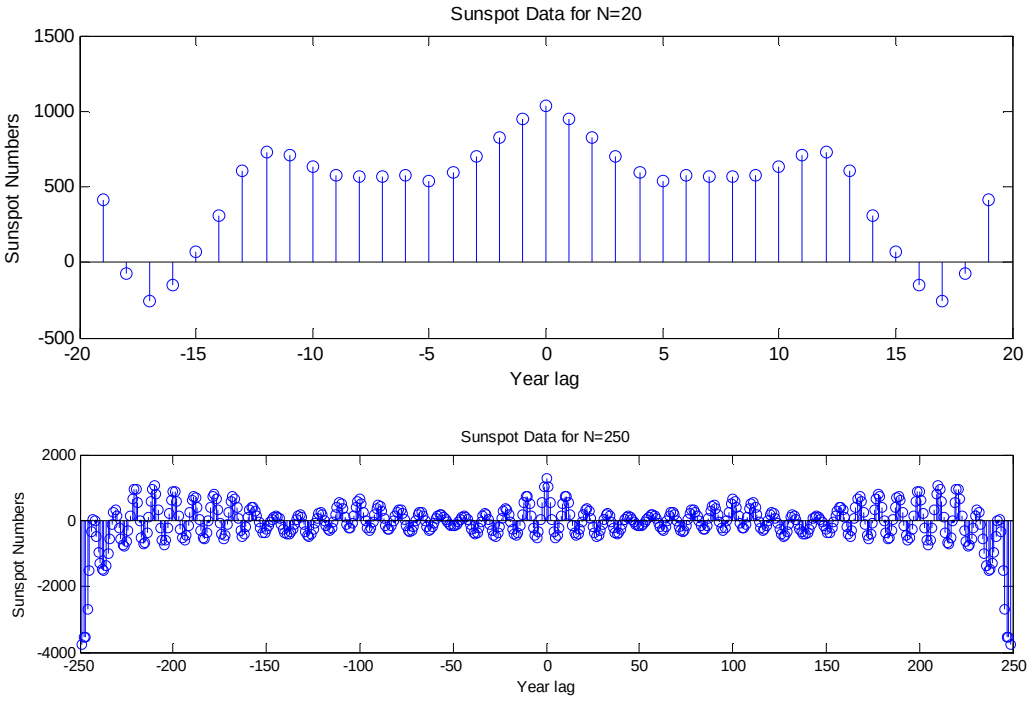


Figure 3 – Autocorrelation done for various N values with a zero mean

This time "zero meaning" the data has fixed a few problems observed earlier. For example the problem with the peaks being higher for N=20 (which implied the signal was more similar to itself when shifted than when it wasn't) is no longer the case. Finally for N=250 the data is somewhat more clear to observe. This is due to the way the ACF works, multiplying each shifted element by another, thus by placing a zero mean we ensure than there is an equal representation in positive as well as negative magnitudes meaning data does not simply accumulate throughout the ACF. This allows the periodicity of the data to be better seen and understood, with the peaks representing periods of repetition.

In the images above we are operating on data which sample set has been zero-meaned in the sense that the mean has been removed after collecting the data sets. It is also possible to remove the mean of the sunspot data before taking the samples. In this this case the following graphs would be obtained:





It can be noticed that the start amplitude is always the same. However it is still difficult to draw conclusions about the data's periodicity thus the previous method is preferred.

2.4.3 AR2 Stability

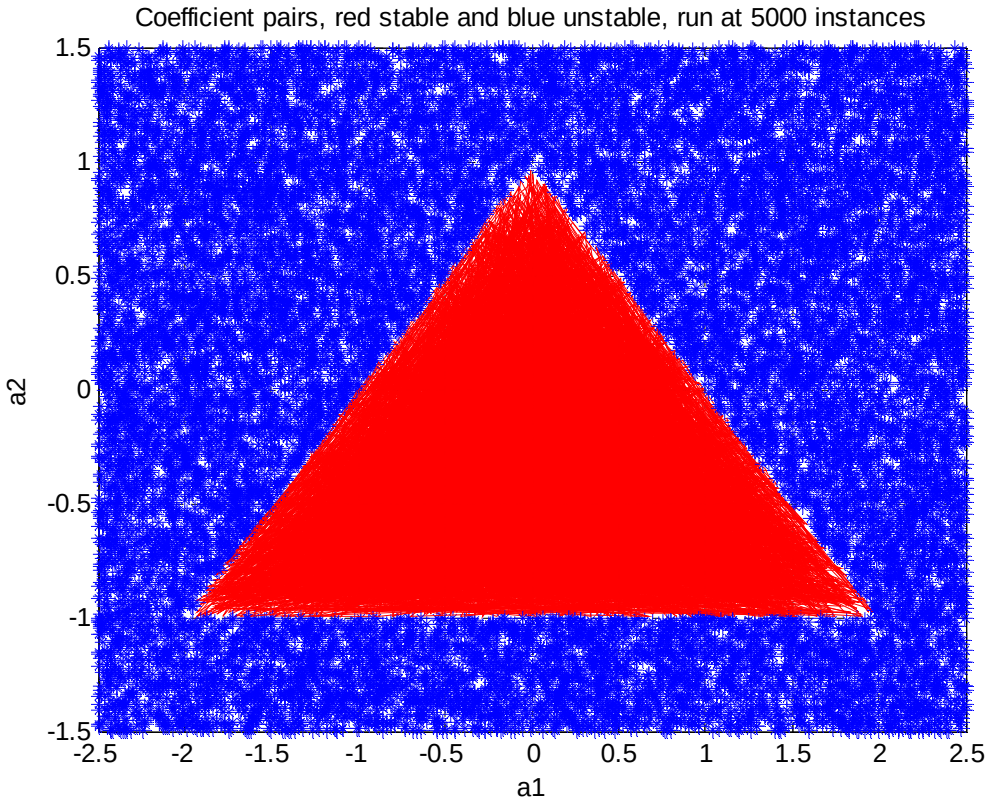


Figure 4 – Stability triangle for an AR2 process

The graph above has blue stars and red dots to represent stable and unstable processes. In red are represented all the coefficients for stable process and in blue we can see the

coefficient pairs for unstable processes. The stability of a process was detected by checking whether or not the final value of x (x[1000]) overflowed or not in matlab.

This graph basically replicated the stability triangle as mentioned in the notes and to be stable a process must satisfy, all at the same time:

$$a1 + a2 < 1$$

$$a2 - a1 < 1$$

$$-1 < a2 < 1$$

These equations arise from having roots inside the unit circle which implies a stable process.

2.4.4 Yule-Walker equations

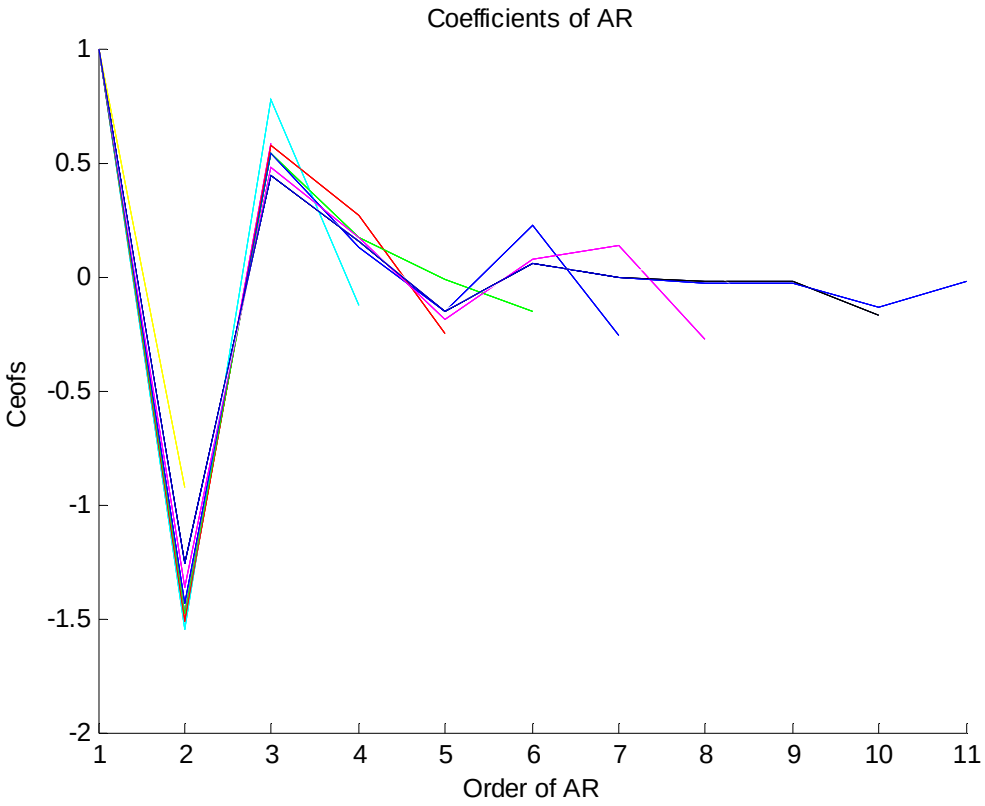


Figure 5 – Plot of coefficients across various AR orders. AS can be seen past 2 most coefficients are around 0 and thus from this graph the optimal order would be 2.

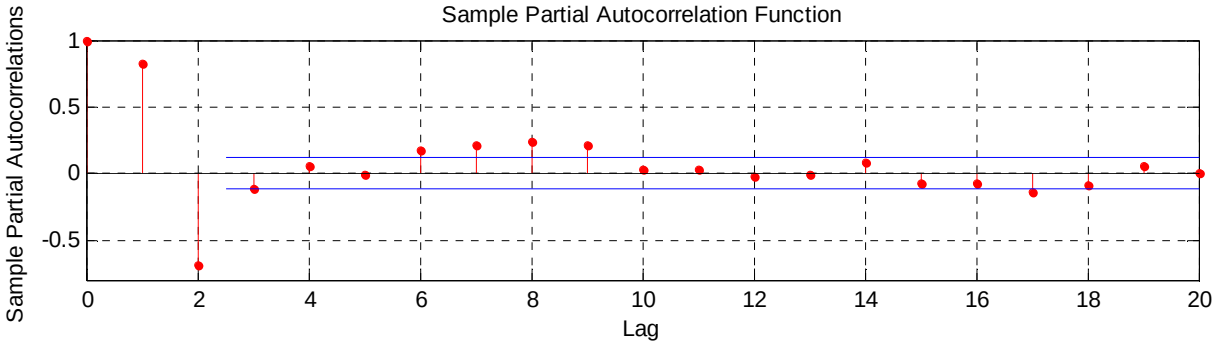


Figure 6 - These lines were obtained from the partial autocorrelation function which calculates the autocorrelation of a series up to a certain time difference ignoring the other lags (thus its name partial autocorrelation) - meaning linear dependence of samples on each other is removed. This

allows the determination of whether or not a sample is worth relative to another one and can help in the identification of the order of an AR process.

From the above sample partial autocorrelation function we can estimate the most likely order of the series which seems to be 2 as all subsequent coefficients are much smaller than the previous ones. Additionally the guidance lines the partial autocorrelation provides in Figure 6 also point towards the sunspot process being best approximated by an AR(2) process.

2.4.5 Determining the correct model order

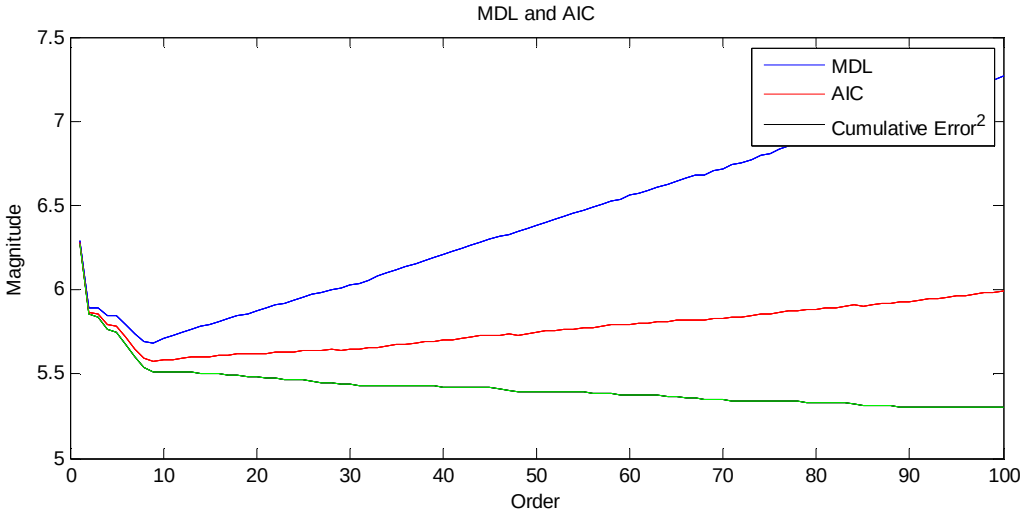


Figure 7 – MDL, AIC and cumulative error squared between the orders of 0 to 100. The "first" minimum is found to be at 2 but both global minima occur at 9 for both AIC and MDL.

Here we use the Minimum Description Length (MDL) and Akaike Information Criteria (AIC) to help us determine the most optimal order. They work by calculating the expected error which leaves us to choose the minimum value which will guide us to choosing the order. The equations are:

$$MDL(p) = \log(E_p) + \frac{p \cdot log(N)}{N}$$

$$AIC(p) = \log(E_p) + \frac{2 \cdot p}{N}$$

P is the order of the model , N being the number of samples and E is the cumulative squared error.

Figure 7 shows how order 9 is the most optimal model for the sunspot data. This means that for an order of 9 the amount of information lost is minimized however this does not predict whether or not the data will be predicted correctly. This is due to MDL and AIC only being relative comparators – the best estimated order could still return a faulty model.

2.4.6 AR modelling

Using the coefficients found from the Yule-Walker equations we can construct models to predict future sunspot data. From the Y-W coefficients we can use the following equation

$x[n] = \sum_{k=1}^{p} a[k]x[n-k]$ to predict future values by first filtering the p previous values (p is the order of the AR process).

From Figure 8 and Figure 9 we can see that the AR2 and AR6 process can predict data to a certain extent –with the AR6 having a better success due to being of higher order. AR2 badly performing is not too surprising due to the data having a repeating pattern in the order of 10 years.

Figure 10 looks at how these predictions do over a very long period of time. The result is not surprising – all the AR models tend to 0 meaning that more input values are needed to "re-energize" them.

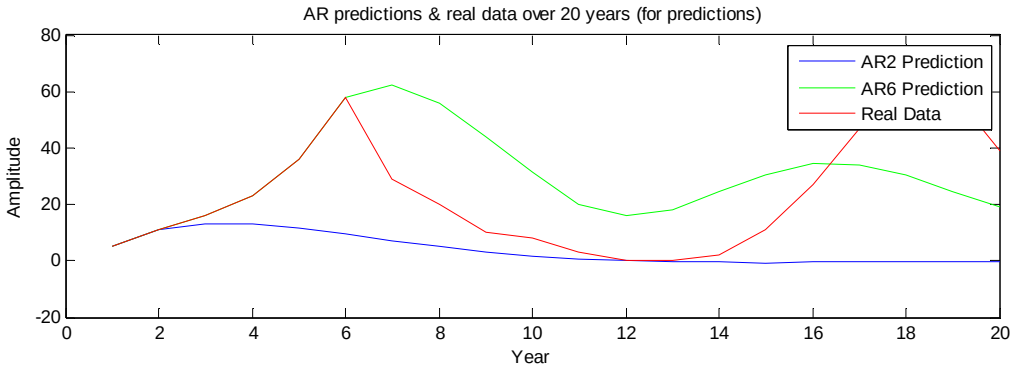


Figure 8 – Prediction for an AR2 and AR6 process of future sunspot data over a period of 20 years. This means that 18 years are predicted by the AR2 process and 14 years predicted by the AR6 process.

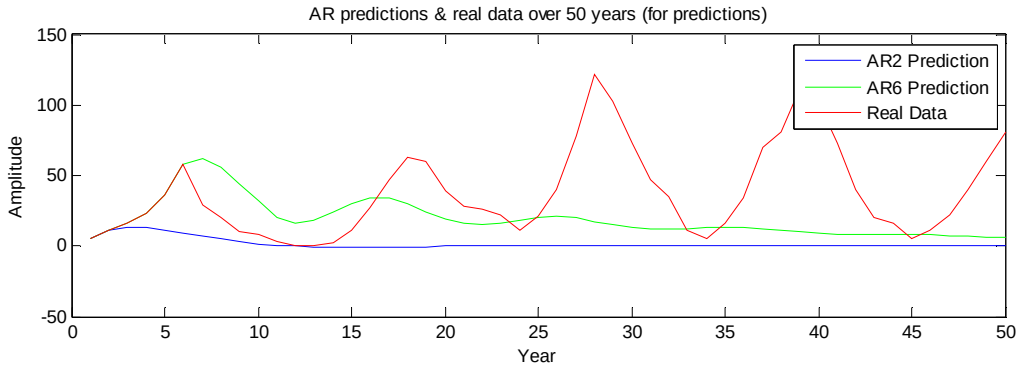


Figure 9 - Prediction for an AR2 and AR6 process of future sunspot data over a period of 50 years. As can be seen the AR predicted data decays reasonably fast and would need more data input points to continue predicting.

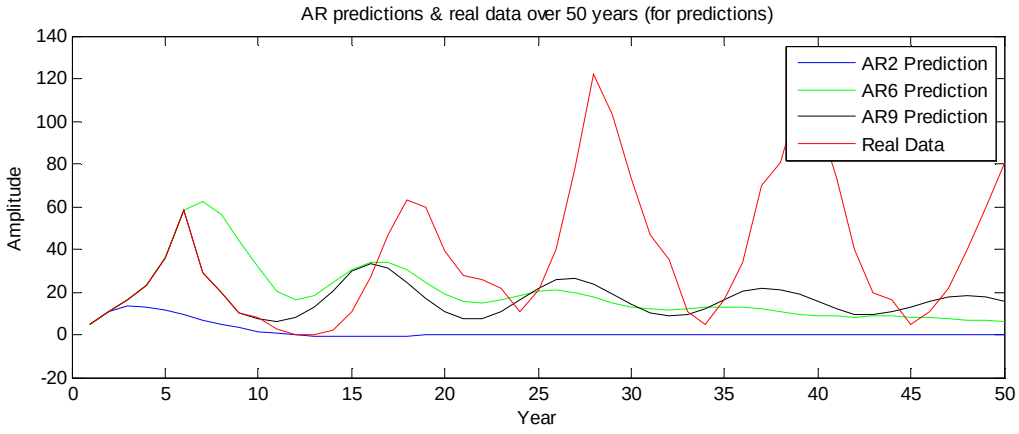


Figure 10 - Prediction for an AR2, AR6 and AR9 process of future sunspot data over a period of 50 years. The AR9 process can be seen to oscillate much longer without need of further input – however the values start to get out of phase and have other issues such as having a very low magnitude.

Appendix

Table of figures

Matlab Code

Part 2.1

```
clc;
x = randn(1,1000);
[a,b] = xcorr(x, 'unbiased');

stem([-999:1:999],a)
xlabel(' \tau (time lag)')
ylabel('Amplitude')
title('Unbiased auto-correation of 1000 AWGN samples')
axis([-50 50 -1 1.5])
```

Part 2.2

```
clc;
Or = 1001;
x = randn(1,1000);
y=filter(ones(Or,1),[Or],x);
[a,b] = xcorr(y, 'unbiased');
subplot(2,1,1)
stem(b,a)
xlabel(' \tau (Time lag)')
ylabel('Amplitude')
str = sprintf('ACF of %dth Order MA filter 1000 AWGN samples',Or)
title(str)
axis ([-20 20 -2 8])
subplot(2,1,2)
plot(y)
xlabel(' Time')
ylabel('Amplitude')
str = sprintf('%dth Order MA filter 1000 AWGN samples',Or)
title(str)
y(1000)
mean(y)
%axis([0 200 -10 10])
```

Part 2.3

```
clc;
Or = 9;
x = randn(1,1000);
y=filter(ones(Or,1),[1],x);
[a,b] = xcorr(y,x, 'unbiased');
stem(b,a)
xlabel(' \tau (Time lag)')
ylabel('Amplitude')
str = sprintf('Cross Correlation of %dth Order MA filter fo 1000 AWGN samples with non-filtered
samples',Or)
title(str)
axis ([-20 20 -1 3])
```

Part 2.4

```
clc;
close all;
clear;
N = 250;
```

```
load sunspot.dat
year=sunspot(1:N,1);
x =sunspot(:,2);
x=x-mean(x);
x = x(1:N);
[a,b] = xcorr(x, 'unbiased');
stem(b,a)
xlabel('Year lag')
ylabel('Sunspot Numbers')
str = sprintf('Sunspot Data for N=%d',N);
title(str)
```

Part 2.4.3

```
clc;
clear;
%plots the stability triangle if called mutliple times
N = 1000;
a1=5.*rand(100,1)-2.5;
a2=3.*rand(100,1)-1.5;
w=randn(N,1);
% a1_2 = zeros(100,1)
% for j=1:100
%     a1_2(i) = a1(i)^i;
%     a2_2(i) = a2(i)^i;
% end

x = zeros(N,N);
x(:,1) = w(1);
x(:,2) = w(2)+a1(1)*x(:,1);
for j=1:100
    for i=3:N
        x(j,i) = w(i) + a1(j)*x(j,i-1) + a2(j)*x(j,i-2);
    end
end
a = zeros(100,1);
b = zeros(100,1);
for j=1:100
    %cehck for overshoot
    if (abs(x(j,N))>10)||(isnan(x(j,N)))
        disp(j)
        a(j) = a1(j);
        b(j) = a2(j);
        a1(j)=0;
        a2(j)=0;
    else
        str = sprintf('_____________\nfor
j=%d\na1:%f\na2:%f\nx_end:%f\na1+a2:%f\n',j,a1(j),a2(j),x(j,N),a1(j)+a2(j));
        disp(str);
    end
end
%clear coef list
a1(a1==0)=[];
a2(a2==0)=[];
a(a==0)=[];
b(b==0)=[];
plot(a1,a2,'r-');hold on
grid on
plot(a,b,'*');
xlabel('a1')
ylabel('a2')
```

```
str = sprintf('Coefficient pairs, red stable and blue unstable');
title(str)
```

Part 2.4.4

```
clc;
clear;
N = 288;
load sunspot.dat
year=sunspot(1:N,1);
x =sunspot(1:N,2);
% for i=1:10
% str = sprintf('ar_%d = aryule(x,%d);',i,i);
% disp(str)
% end
%get coefs
ar_1 = aryule(x,1);
ar_2 = aryule(x,2);
ar_3 = aryule(x,3);
ar_4 = aryule(x,4);
ar_5 = aryule(x,5);
ar_6 = aryule(x,6);
ar_7 = aryule(x,7);
ar_8 = aryule(x,8);
ar_9 = aryule(x,9);
ar_10 = aryule(x,10);
% for i=1:10
% str = sprintf('plot(ar_%d);',i);
% disp(str)
% end
%plot them
figure(1)
hold on
plot(ar_1,'y');
plot(ar_2,'m');
plot(ar_3,'c');
plot(ar_4,'r');
plot(ar_5,'g');
plot(ar_6,'b');
plot(ar_7,'m');
plot(ar_8,'w');
plot(ar_9,'k');
plot(ar_10);
hold off
xlabel('Order of AR')
ylabel('Ceofs')
str = sprintf('Coefficients of AR');
title(str)
figure(2)
% parcorr(x,[],2)
pyulear(x,2)
```

Part 2.4.5 & 2.4.6

```
clc;
close all;
clear all;
% PART 2.4.5 & 2.5.6
load sunspot.dat;
```

```
year = sunspot(:,1);
data = sunspot(:,2);
N = length(data);
years = 50;
%Process the MDL and AIC for 100 orders
for i = 1:100
    [coefficients,E] = aryule(data, i);
    ord = length(coefficients);
    y(:,i) = filter(-1*coefficients(2:end),1,data(:));%cut off first value and negate the rest due to the way
the equation works
    MDL(i,1) = log(sum(E)) + (i*log(N))/N;%Calculate MDL
    AIC(i,1) = log(sum(E)) +(2*i)/N;%Calculate AIC
    Er(i) = log(sum(E));
end
 %Generate coefficients for AR processes
coeffs_order2 = -1*aryule(data, 2);
coeffs_order6 = -1*aryule(data, 6);
coeffs_order9 = -1*aryule(data, 9);
ar2(1) = data(1);
ar2(2) = data(2);

%Initialise all
for i = 1:20
 ar2(i) = data(i);
 ar6(i) = data(i);
 ar9(i) = data(i);
end

%AR2 prediction
for i = 1:years-2,
    ar2(2+i) = coeffs_order2(2)*ar2(1+i) + coeffs_order2(3)*ar2(i);
end
%AR6 prediction
for i = 1:years-6,
   ar6(6+i) = coeffs_order6(2)*ar6(5+i) + coeffs_order6(3)*ar6(4+i) + coeffs_order6(4)*ar6(3+i) +
coeffs_order6(5)*ar6(2+i) + coeffs_order6(6)*ar6(1+i) + coeffs_order6(7)*ar6(i);
end

%AR9 future
for i = 1:years-9,
    ar9(9+i) = 0;
    for j = 1:9
        ar9(9+i) = ar9(9+i) + coeffs_order9(1+j)*ar9(9+i-j);
    end
end

%Plot all
figure(1)
plot(MDL), title('MDL and AIC'), xlabel('Order'), ylabel('Magnitude');
hold on;
plot(AIC,'r');
plot(Er,'g');
legend('MDL','AIC','Cumulative Error^2');

figure(2)
str = sprintf('AR predictions & real data over %d years (for predictions)', years);
plot(ar2), title(str), xlabel('Year'), ylabel('Amplitude');
hold on;
plot(ar6, 'g');
plot(ar9, 'k');

plot(data(1:years),'r');
```

```
legend('AR2 Prediction','AR6 Prediction','AR9 Prediction','Real Data')
hold off;
```