Spectral Processing of Signals
Homework 2:
Rational Parametric Methods

Philip Ahl

I have chosen to investigate the least squares approach to the AR and ARMA models. First, looking at the least square AR (LSAR), the model order choice was aided by plotting the white noise variance estimate against the order of the AR polynomial. The plots can be seen for all data sets in Fig.1. The white noise variance estimate gives a picture of how much of the spectrum is given by the signal and how much is given by the noise. When the noise variance estimate decreases with increased order, less of the full signal variance is represented by the error. Here a trade-off is made between model order and variance (not to mix up with the variance of the whole signal). Using this methodology of analysis, the loglynx data set order seems good at 12, the lynx data set at 8 and the sunspot at 9.

Four different information criterion were used separately on each data set. Akaike's Information Criterion (AIC), Akaike's Information Criterion with correction (AICc), Bayesian Information Criterion (BIC), Generalized Information criterion (GIC) were used. Four different were used to get a nuanced picture of where the criterion suggests the order, but mainly the AIC was considered. They were usually pretty coherent, as seen in Fig.2 for the sunspot data set where 9 seemed as a good minimum. This fenomena was seen in all three data sets, where all criterion minimum matched with the variance analysis, hence the above mentioned orders were chosen for each data set.

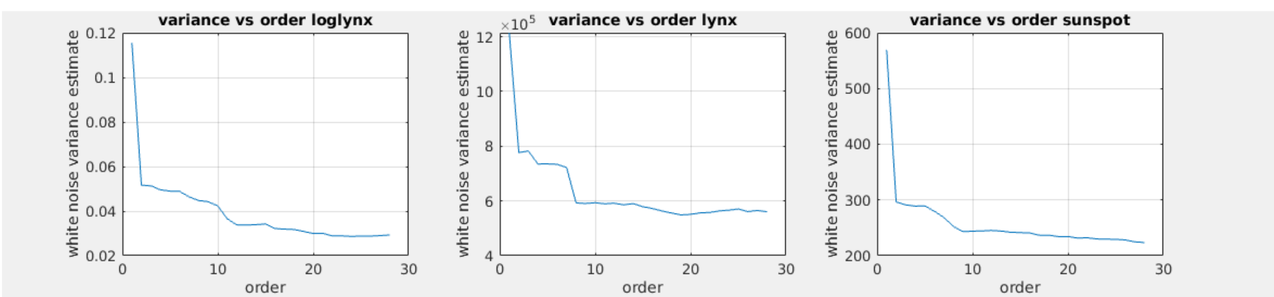

Figure 1

For the ARMA model, similar analyses were made, but slightly more complex since the ARMA model consists of three different orders to choose, AR, MA and the truncated model size K. The methodology of analysis was with the following scheme. A fixed K was chosen, while all permutations of M and N up to the value of K was investigated. The same information criterion was used. The best (lowest) for each criterion and data set was saved for each tried M, together with the corresponding N that yielded that value, were plotted against each other. In Fig.3 the criterion for the lynx data set is shown with K fixed at 20. It is not as clear as with the AR model, but the M (MA) order was set to 11 and the corresponding N (AR) order was found as 8. Similar

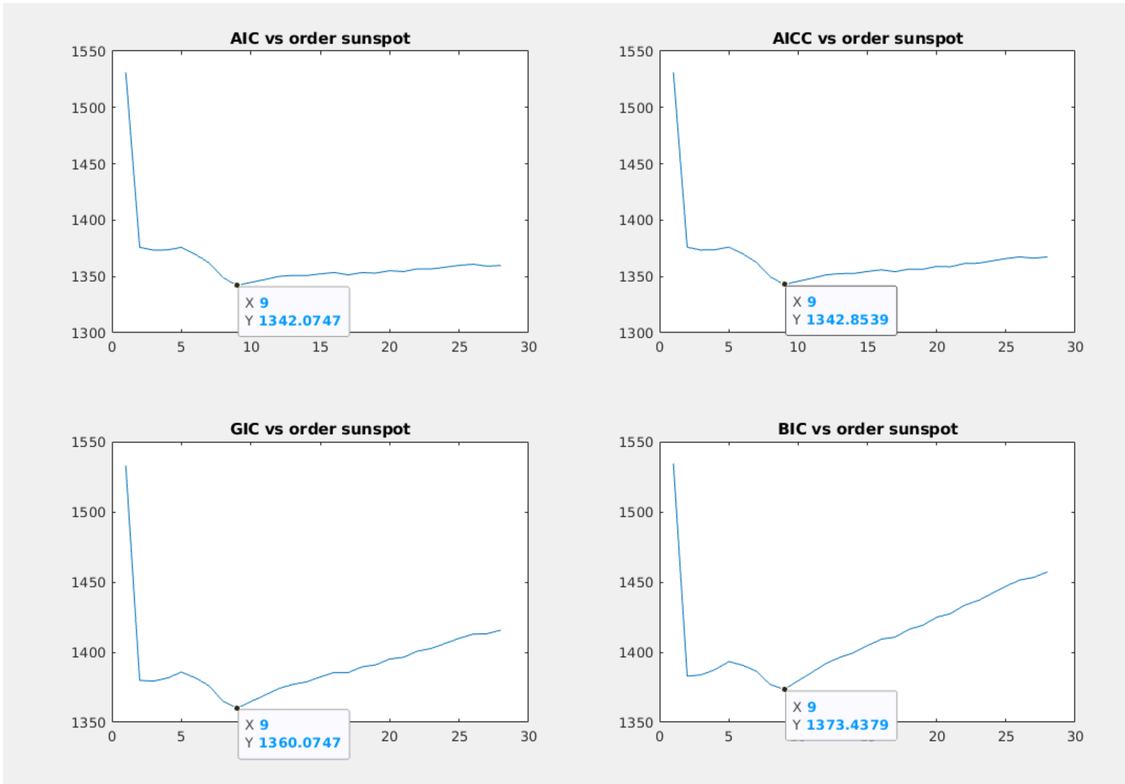

Figure 2

approach was used for the other data sets as well and got M = 5 and N = 2 for the loglynx and M = 9, N = 3 for the sunspot data set. At some point different local minima were tried out, steering towards lower orders due to unstable poles. The steering towards less complex models with lower orders is due to the mind set of Occam's razor.

The spectrum and pole-zero plots of the final models are shown in Fig.4 and Fig.5. Where all poles on the periodogram from zero towards positive pi corresponds on rotating anti clockwise on the PZ-plot starting from "3 o'clock".

The cons of parametric methods such as the ones looked at in this paper is that parameters = freedom of choice. Since parameters can be correlated, all feasible permutations should be considered. For larger data sets, this can be a computationally heavy process. The analysis can grow immensely and the common feel for the goodness of the methods may disappear. The non-parametric method works with less tweaks, henceforth also with less accuracy. A trade-off depending on the signal needs to be done if a full parametric analysis is worth the time and computational burden. A non-parametric method could here be used as a first step as guidance in a rough estimation on how many peaks the spectrum should have.

Having too large order size with respect to the length of the signal can lead to overfitting. This is when we are fitting the model on the training data, giving it a high variance. Having too small order size will lead to underfitting, which will decrease the models variance while increase its bias. Here, I'd say the plotted LSARMA on the loglynx data set was underfitted, biased to a few amount of parameters, loosing information about frequency peaks compared to the others.

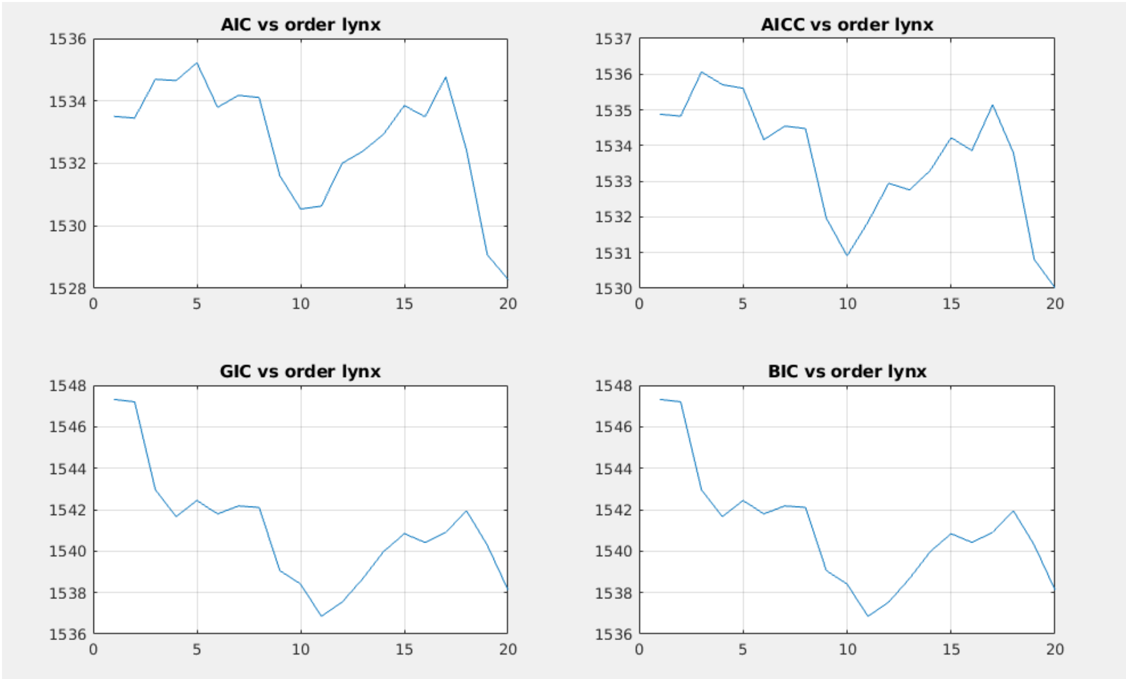

Figure 3

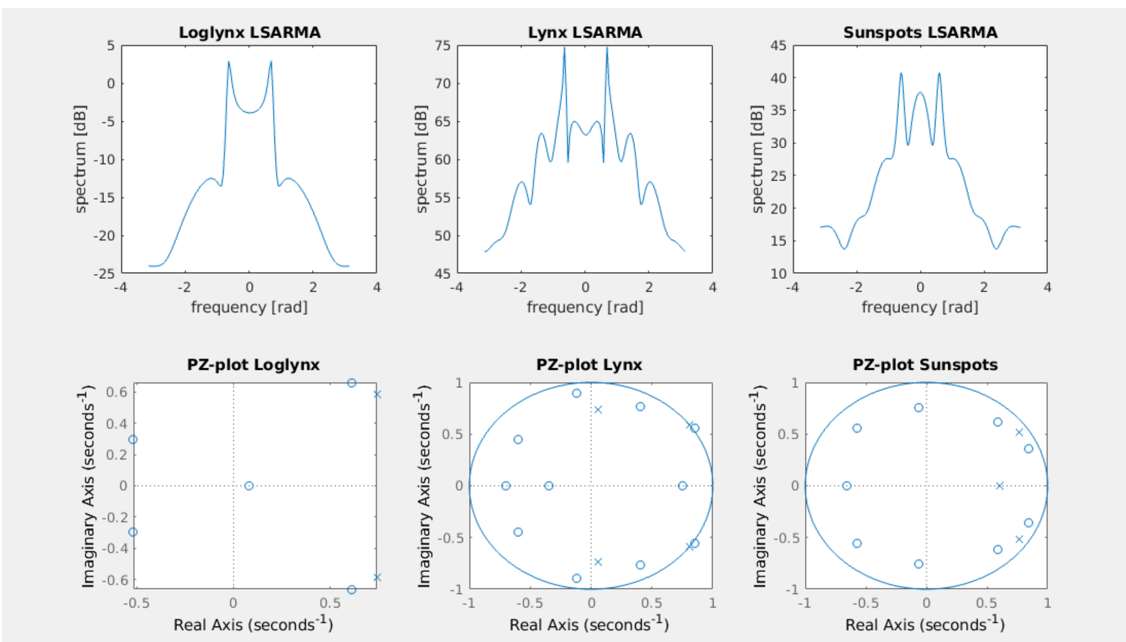

Figure 4

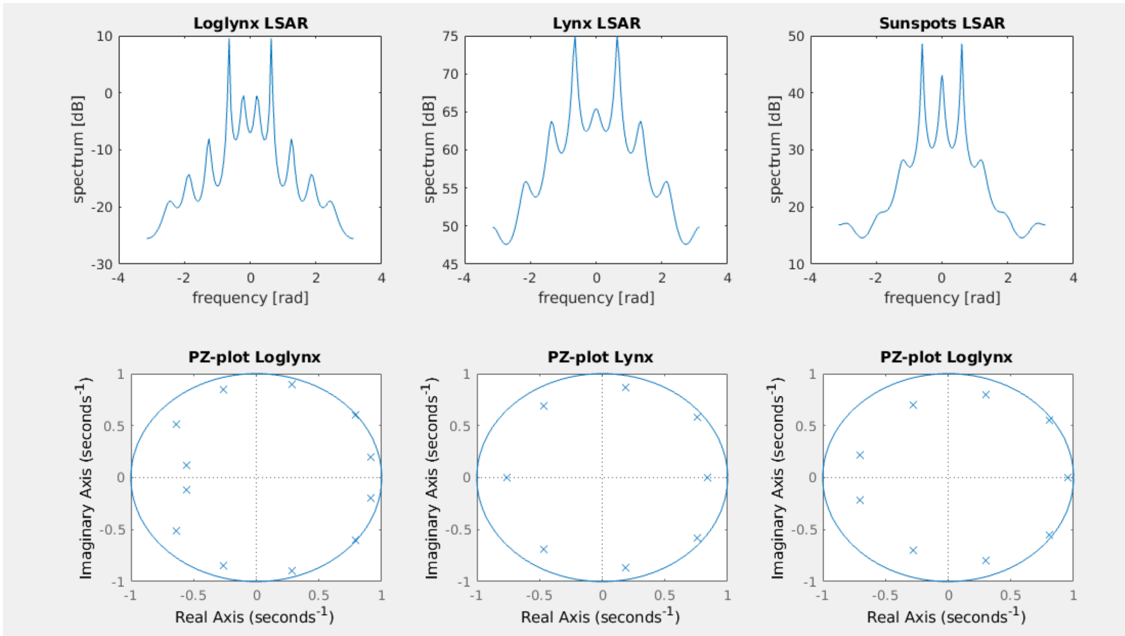

Figure 5

I. Appendix

I. ARMA Matlab script

```
clear all
close all

% import data from current folder
load('lynxdata.mat')
load('sunspotdata.mat')

% removing mean
loglynx = loglynx - mean(loglynx);
lynx = lynx - mean(lynx);
sunspot = sunspot - mean(sunspot);

len_logl = length(loglynx);
len_lynx = length(lynx);
len_sun = length(sunspot);

orders_n = 20;
orders_m = 20;

% vector for saving variances for different orders_n
vars_logl = zeros(orders_n);
vars_lynx = zeros(orders_n);
```

```
vars_sun = zeros(orders_n);

% vectors for saving aic, aicc, gic and bic values for each order
aic_logl = zeros(1,orders_n);
aic_lynx = zeros(1,orders_n);
aic_sun = zeros(1,orders_n);
aicc_logl = zeros(1,orders_n);
aicc_lynx = zeros(1,orders_n);
aicc_sun = zeros(1,orders_n);
gic_logl = zeros(1,orders_n);
gic_lynx = zeros(1,orders_n);
gic_sun = zeros(1,orders_n);
bic_logl = zeros(1,orders_n);
bic_lynx = zeros(1,orders_n);
bic_sun = zeros(1,orders_n);

aic_logl_ = zeros(1,orders_m);
aic_lynx_ = zeros(1,orders_m);
aic_sun_ = zeros(1,orders_m);
aicc_logl_ = zeros(1,orders_m);
aicc_lynx_ = zeros(1,orders_m);
aicc_sun_ = zeros(1,orders_m);
gic_logl_ = zeros(1,orders_m);
gic_lynx_ = zeros(1,orders_m);
gic_sun_ = zeros(1,orders_m);
bic_logl_ = zeros(1,orders_m);
bic_lynx_ = zeros(1,orders_m);
bic_sun_ = zeros(1,orders_m);

% match the index of the lowest criterion with this array
% i1 corresp to aic for logl
% i12 corresp to bic to sunspot
% etc. .. ... . .
i1 = zeros(1,orders_m);
i2 = zeros(1,orders_m);
i3 = zeros(1,orders_m);
i4 = zeros(1,orders_m);
i5 = zeros(1,orders_m);
i6 = zeros(1,orders_m);
i7 = zeros(1,orders_m);
i8 = zeros(1,orders_m);
i9 = zeros(1,orders_m);
i10 = zeros(1,orders_m);
i11 = zeros(1,orders_m);
i12 = zeros(1,orders_m);

for m=1:orders_m
    for n=1:orders_n
```

```
        k = 20;
        m_lynx = 11;
        n_lynx = 4;
        m_sun = 9;
        n_sun = 3;
        m_logl = 5;
        n_logl = 2;

%         m_lynx = m;
%         n_lynx = n;
%         m_sun = m;
%         n_sun = n;
%         m_logl = m;
%         n_logl = n;

        % Least Square ARMA coeff est
        [a_params_logl_lsarma,b_params_logl_lsarma,var_logl] = lsarma(
            loglynx,n_logl,m_logl,k);
        [a_params_lynx_lsarma,b_params_lynx_lsarma,var_lynx] = lsarma(
            lynx,n_lynx,m_lynx,k);
        [a_params_sun_lsarma,b_params_sun_lsarma,var_sun] = lsarma(
            sunspot,n_sun,m_sun,k);

        phi_logl_arma = armase(b_params_logl_lsarma,
            a_params_logl_lsarma,var_logl,len_logl);
        phi_lynx_arma = armase(b_params_lynx_lsarma,
            a_params_lynx_lsarma,var_lynx,len_lynx);
        phi_sun_arma = armase(b_params_sun_lsarma,a_params_sun_lsarma,
            var_sun,len_sun);

        vars_logl(n) = var_logl;
        vars_lynx(n) = var_lynx;
        vars_sun(n) = var_sun;

        aic_logl(n) = aic(var_logl,n,len_logl);
        aic_lynx(n) = aic(var_lynx,n,len_lynx);
        aic_sun(n) = aic(var_sun,n,len_sun);
        aicc_logl(n) = aicc(var_logl,n,len_logl);
        aicc_lynx(n) = aicc(var_lynx,n,len_lynx);
        aicc_sun(n) = aicc(var_sun,n,len_sun);
        gic_logl(n) = gic(var_logl,n,len_logl,4);
        gic_lynx(n) = gic(var_lynx,n,len_lynx,4);
        gic_sun(n) = gic(var_sun,n,len_sun,4);
        bic_logl(n) = bic(var_logl,n,len_logl);
        bic_lynx(n) = bic(var_lynx,n,len_lynx);
        bic_sun(n) = bic(var_sun,n,len_sun);
    end
    % i corresp to n = AR order = poles
```

```
    [aic_logl_(m),i1(m)] = min(aic_logl);
    [aic_lynx_(m),i2(m)] = min(aic_lynx);
    [aic_sun_(m),i3(m)] = min(aic_sun);
    [aicc_logl_(m),i4(m)] = min(aicc_logl);
    [aicc_lynx_(m),i5(m)] = min(aicc_lynx);
    [aicc_sun_(m),i6(m)] = min(aicc_sun);
    [gic_logl_(m),i7(m)] = min(gic_logl);
    [gic_lynx_(m),i8(m)] = min(gic_lynx);
    [gic_sun_(m),i9(m)] = min(gic_sun);
    [bic_logl_(m),i10(m)] = min(gic_logl);
    [bic_lynx_(m),i11(m)] = min(gic_lynx);
    [bic_sun_(m),i12(m)] = min(gic_sun);
end

omega_logl = -pi:(2*pi/(len_logl-1)):pi;
omega_lynx = -pi:(2*pi/(len_lynx-1)):pi;
omega_sun = -pi:(2*pi/(len_sun-1)):pi;
w = 0:1/100:2*pi;
unitcircle = cos(w)+1j*sin(w);

figure(1)
subplot(2,3,1), plot(omega_logl,fftshift(10*log10(phi_logl_arma))),
    title('Loglynx LSARMA'),xlabel('frequency [rad]'), ylabel('spectrum
    [dB]');
subplot(2,3,2), plot(omega_lynx,fftshift(10*log10(phi_lynx_arma))),
    title('Lynx LSARMA'),xlabel('frequency [rad]'), ylabel('spectrum [dB
    ]');
subplot(2,3,3), plot(omega_sun,fftshift(10*log10(phi_sun_arma))), title
    ('Sunspots LSARMA'),xlabel('frequency [rad]'), ylabel('spectrum [dB]
    ');
sys_logl_lsar = tf(b_params_logl_lsarma',a_params_logl_lsarma');
sys_lynx_lsar = tf(b_params_lynx_lsarma',a_params_lynx_lsarma');
sys_sun_lsar = tf(b_params_sun_lsarma',a_params_sun_lsarma');
subplot(2,3,4), pzplot(sys_logl_lsar), hold on, plot(unitcircle), title
    ('PZ-plot Loglynx');
subplot(2,3,5), pzplot(sys_lynx_lsar), hold on, plot(unitcircle), title
    ('PZ-plot Lynx');
subplot(2,3,6), pzplot(sys_sun_lsar), hold on, plot(unitcircle), title(
    'PZ-plot Sunspots');

figure(3)
subplot(1,3,1), plot(1:orders_n,vars_logl), title('variance vs order
    loglynx');
subplot(1,3,2), plot(1:orders_n,vars_lynx), title('variance vs order
    lynx');
subplot(1,3,3), plot(1:orders_n,vars_sun), title('variance vs order
    sunspot');
```

```
figure(4)
subplot(2,2,1), plot(1:orders_n,aic_logl_), title('AIC vs order loglynx
    ');
subplot(2,2,2), plot(1:orders_n,aicc_logl_), title('AICC vs order
    loglynx');
subplot(2,2,3), plot(1:orders_n,gic_logl_), title('GIC vs order loglynx
    ');
subplot(2,2,4), plot(1:orders_n,bic_logl_), title('BIC vs order loglynx
    ');

figure(5)
subplot(2,2,1), plot(1:orders_n,aic_lynx_),grid on, title('AIC vs order
     lynx');
subplot(2,2,2), plot(1:orders_n,aicc_lynx_),grid on, title('AICC vs
    order lynx');
subplot(2,2,3), plot(1:orders_n,gic_lynx_), grid on,title('GIC vs order
     lynx');
subplot(2,2,4), plot(1:orders_n,bic_lynx_), grid on,title('BIC vs order
     lynx');

figure(6)
subplot(2,2,1), plot(1:orders_n,aic_sun_), title('AIC vs order sunspot'
    );
subplot(2,2,2), plot(1:orders_n,aicc_sun_), title('AICC vs order
    sunspot');
subplot(2,2,3), plot(1:orders_n,gic_sun_), title('GIC vs order sunspot'
    );
subplot(2,2,4), plot(1:orders_n,bic_sun_), title('BIC vs order sunspot'
    );


function b = aic(sig2,order,N)
    b = N*log(sig2) + 2*order;
end

function b = aicc(sig2,order,N)
    b = N*log(sig2) + (2*order * N)./(N-order-1);
end

function b = gic(sig2,order,N,gic_param)
% gic param usually between [2 6], hence set to 4 :)
    b = N*log(sig2) +  order * gic_param;
end

function b = bic(sig2,order,N)
    b = N*log(sig2) +  order * log(N);
end
```

II. AR Matlab script

```
clear all
close all

% import data from current folder
load('lynxdata.mat')
load('sunspotdata.mat')

% removing mean
loglynx = loglynx - mean(loglynx);
lynx = lynx - mean(lynx);
sunspot = sunspot - mean(sunspot);

len_logl = length(loglynx);
len_lynx = length(lynx);
len_sun = length(sunspot);

orders = floor(len_logl/4);

% vector for saving variances for different orders
vars_logl = zeros(1,orders);
vars_lynx = zeros(1,orders);
vars_sun = zeros(1,orders);

% vectors for saving aic, aicc, gic and bic values for each order
aic_logl = zeros(1,orders);
aic_lynx = zeros(1,orders);
aic_sun = zeros(1,orders);
aicc_logl = zeros(1,orders);
aicc_lynx = zeros(1,orders);
aicc_sun = zeros(1,orders);
gic_logl = zeros(1,orders);
gic_lynx = zeros(1,orders);
gic_sun = zeros(1,orders);
bic_logl = zeros(1,orders);
bic_lynx = zeros(1,orders);
bic_sun = zeros(1,orders);

for n=1:orders

        % model orders

        N_logl_lsar = 12;
        N_lynx_lsar = 8;
        N_sun_lsar = 9;
%          N_logl_lsar = n;
%          N_lynx_lsar = n;
```

```
%           N_sun_lsar = n;
%
        % K should not be large wrt N
        K_logl_lsarma = floor(len_logl/20);
        K_lynx_lsarma = floor(len_lynx/20);
        K_sun_lsarma = floor(len_sun/20);


        % Least Square AR coeff est
        [params_logl_lsar,var_logl_lsar] = lsar(loglynx,N_logl_lsar);
        [params_lynx_lsar,var_lynx_lsar] = lsar(lynx,N_lynx_lsar);
        [params_sun_lsar,var_sun_lsar] = lsar(sunspot,N_sun_lsar);

        omega_logl = -pi:(2*pi/(len_logl-1)):pi;
        omega_lynx = -pi:(2*pi/(len_lynx-1)):pi;
        omega_sun = -pi:(2*pi/(len_sun-1)):pi;

        est_logl_lsar = zeros(length(omega_logl),1);
        est_lynx_lsar = zeros(length(omega_lynx),1);
        est_sun_lsar = zeros(length(omega_sun),1);

        % Calculating spectrum manually

        % Loglynx
        k1 = 1:N_logl_lsar+1;
        for i=k1
            est_logl_lsar(:) = est_logl_lsar(:) + lsmerge(
                params_logl_lsar(i),omega_logl,i-1)';
        end

        % Lynx
        k2 = 1:N_lynx_lsar+1;
        for i=k2
            est_lynx_lsar(:) = est_lynx_lsar(:) + lsmerge(
                params_lynx_lsar(i),omega_lynx,i-1)';
        end

        % Sunspot
        k3 = 1:N_sun_lsar+1;
        for i=k3
            est_sun_lsar(:) = est_sun_lsar(:) + lsmerge(params_sun_lsar
                (i),omega_sun,i-1)';
        end

        phi_logl_lsar = var_logl_lsar./abs(est_logl_lsar).^2;
        phi_logl_lsar = 10*log10(phi_logl_lsar);

        phi_lynx_lsar = var_lynx_lsar./abs(est_lynx_lsar).^2;
```

```
        phi_lynx_lsar = 10*log10(phi_lynx_lsar);

        phi_sun_lsar = var_sun_lsar./abs(est_sun_lsar).^2;
        phi_sun_lsar = 10*log10(phi_sun_lsar);

        vars_logl(n) = var_logl_lsar;
        vars_lynx(n) = var_lynx_lsar;
        vars_sun(n) = var_sun_lsar;

        aic_logl(n) = aic(var_logl_lsar,n,len_logl);
        aic_lynx(n) = aic(var_lynx_lsar,n,len_lynx);
        aic_sun(n) = aic(var_sun_lsar,n,len_sun);
        aicc_logl(n) = aicc(var_logl_lsar,n,len_logl);
        aicc_lynx(n) = aicc(var_lynx_lsar,n,len_lynx);
        aicc_sun(n) = aicc(var_sun_lsar,n,len_sun);
        gic_logl(n) = gic(var_logl_lsar,n,len_logl,4);
        gic_lynx(n) = gic(var_lynx_lsar,n,len_lynx,4);
        gic_sun(n) = gic(var_sun_lsar,n,len_sun,4);
        bic_logl(n) = bic(var_logl_lsar,n,len_logl);
        bic_lynx(n) = bic(var_lynx_lsar,n,len_lynx);
        bic_sun(n) = bic(var_sun_lsar,n,len_sun);

end

figure(1)
subplot(2,3,1), plot(omega_logl,phi_logl_lsar), title('Loglynx LSAR'),
    ylabel('spectrum [dB]'),xlabel('frequency [rad]');
subplot(2,3,2), plot(omega_lynx,phi_lynx_lsar), title('Lynx LSAR'),
    ylabel('spectrum [dB]'),xlabel('frequency [rad]');
subplot(2,3,3), plot(omega_sun,phi_sun_lsar), title('Sunspots LSAR'),
    ylabel('spectrum [dB]'),xlabel('frequency [rad]');


sys_logl_lsar = tf(1,params_logl_lsar');
sys_lynx_lsar = tf(1,params_lynx_lsar');
sys_sun_lsar = tf(1,params_sun_lsar');

w = 0:1/100:2*pi;
unitcircle = cos(w)+1j*sin(w);


subplot(2,3,4), pzplot(sys_logl_lsar), hold on, plot(unitcircle), title
    ('PZ-plot Loglynx');
subplot(2,3,5), pzplot(sys_lynx_lsar), hold on, plot(unitcircle), title
    ('PZ-plot Lynx');
subplot(2,3,6), pzplot(sys_sun_lsar), hold on, plot(unitcircle), title(
    'PZ-plot Loglynx');

```

```
figure(3)
subplot(1,3,1), plot(1:orders,vars_logl), grid on, title('variance vs order loglynx'), xlabel('order'),ylabel('white noise variance estimate');
subplot(1,3,2), plot(1:orders,vars_lynx), grid on, title('variance vs order lynx'),xlabel('order'),ylabel('white noise variance estimate');
subplot(1,3,3), plot(1:orders,vars_sun), grid on, title('variance vs order sunspot'),xlabel('order'),ylabel('white noise variance estimate');

figure(4)
subplot(2,2,1), plot(1:orders,aic_logl), title('AIC vs order loglynx');
subplot(2,2,2), plot(1:orders,aicc_logl), title('AICC vs order loglynx');
subplot(2,2,3), plot(1:orders,gic_logl), title('GIC vs order loglynx');
subplot(2,2,4), plot(1:orders,bic_logl), title('BIC vs order loglynx');

figure(5)
subplot(2,2,1), plot(1:orders,aic_lynx), title('AIC vs order lynx');
subplot(2,2,2), plot(1:orders,aicc_lynx), title('AICC vs order lynx');
subplot(2,2,3), plot(1:orders,gic_lynx), title('GIC vs order lynx');
subplot(2,2,4), plot(1:orders,bic_lynx), title('BIC vs order lynx');

figure(6)
subplot(2,2,1), plot(1:orders,aic_sun), title('AIC vs order sunspot');
subplot(2,2,2), plot(1:orders,aicc_sun), title('AICC vs order sunspot');
subplot(2,2,3), plot(1:orders,gic_sun), title('GIC vs order sunspot');
subplot(2,2,4), plot(1:orders,bic_sun), title('BIC vs order sunspot');



function a = lsmerge(coeff,omega,index)
    a = coeff*exp(-1j*omega*index);
end

function b = aic(sig2,order,N)
    b = N*log(sig2) + 2*order;
end

function b = aicc(sig2,order,N)
    b = N*log(sig2) + (2*order * N)./(N-order-1);
end

function b = gic(sig2,order,N,gic_param)
% gic param usually between [2 6], hence set to 4 :)
    b = N*log(sig2) +  order * gic_param;
```

```
end

function b = bic(sig2,order,N)
    b = N*log(sig2) +  order * log(N);
end
```