Statistical Signal Processing
Laboratory Work 2

Iran Ramezanipour
2365743

Parisa Nouri
2417581

November 13, 2017

1. Task 1

One celestial body a travels along an elliptical orbit that lies on the equatorial plane of another larger body B. The position of the larger celestial body can be considered fixed with respect to a. We set origin $(0;0)$ of a Cartesian coordinate system in correspondence with the center of B. The orbit of the body a is known to have a period of 6 years (72 months) but its trajectory is unknown. During the last 12 months the position of a was observed and recorded at regular time intervals, once during each month, resulting in twelve coordinate pairs $(X1;Y1);...;(X12;Y12)$. However, such recorded values are known to be imprecise due to imperfections in the measuring device. The orbit of a can be modeled in the following way:

$$\begin{bmatrix} X_k \\ Y_k \end{bmatrix} = \begin{bmatrix} R_1 cos(\frac{2\pi}{72}k) + C_1 \\ R_2 sin(\frac{2\pi}{72}k) + C_2 \end{bmatrix} + \begin{bmatrix} \upsilon_k \\ \nu_k \end{bmatrix} \tag{1.1}$$

where $k = 1,...12$ and R_1, R_2, C_1, C_2 are the unknown parameters of an ellipse, representing respectively the lengths of the two main axes, and the xy-coordinates of its center. The quantities υ_k, ν_k are statistically independent noise that afECT the measurements.

The task is to re-express Equation (1.1) according to the generic linear model form and estimate numerically the orbit of a in Matlab using a LS-estimator. The numerical values of the measurements X_k, Y_k are contained in the Matlab script. Can you predict whether a and B are going to collide?

A linear least-squares problem is one where the parameter observation model is linear as $S = H\theta$, and $X = H\theta + W$ where $\hat{\theta} = (H^T H)^{-1} H^T X$. In this task, we have to find the estimated values of R_1, R_2, C_1, C_2 according to LS estimator and observation matrix H. The orbit of a on x-axis and y-axis is

$$\begin{bmatrix} X_1 \\ \vdots \\ X_{12} \end{bmatrix} = \begin{bmatrix} cos(\frac{2\pi}{72} \times 1) & 1 \\ \vdots & \vdots \\ cos(\frac{2\pi}{72} \times 12) & 1 \end{bmatrix} \begin{bmatrix} R_1 \\ C_1 \end{bmatrix} + \begin{bmatrix} \upsilon_1 \\ \vdots \\ \upsilon_{12} \end{bmatrix} \tag{1.2}$$

$$\begin{bmatrix} Y_1 \\ \vdots \\ Y_{12} \end{bmatrix} = \begin{bmatrix} sin(\frac{2\pi}{72} \times 1) & 1 \\ \vdots & \vdots \\ sin(\frac{2\pi}{72} \times 12) & 1 \end{bmatrix} \begin{bmatrix} R_2 \\ C_2 \end{bmatrix} + \begin{bmatrix} \nu_1 \\ \vdots \\ \nu_{12} \end{bmatrix} \tag{1.3}$$

Hence, $\hat{\theta_X} = \begin{bmatrix} \hat{R_1} \\ \hat{C_1} \end{bmatrix}$ and $\hat{\theta_Y} = \begin{bmatrix} \hat{R_2} \\ \hat{C_2} \end{bmatrix}$ and we store the estimated values in a 4-element vector variable named P as $P = [\hat{R_1} \;\; \hat{R_2} \;\; \hat{C_1} \;\; \hat{C_2}]$.

We aim to estimate the trajectory $X(t)$ and $Y(t)$ which is solved according to the recorded positions of a during the 12 months as mentioned in the Matlab script. In order

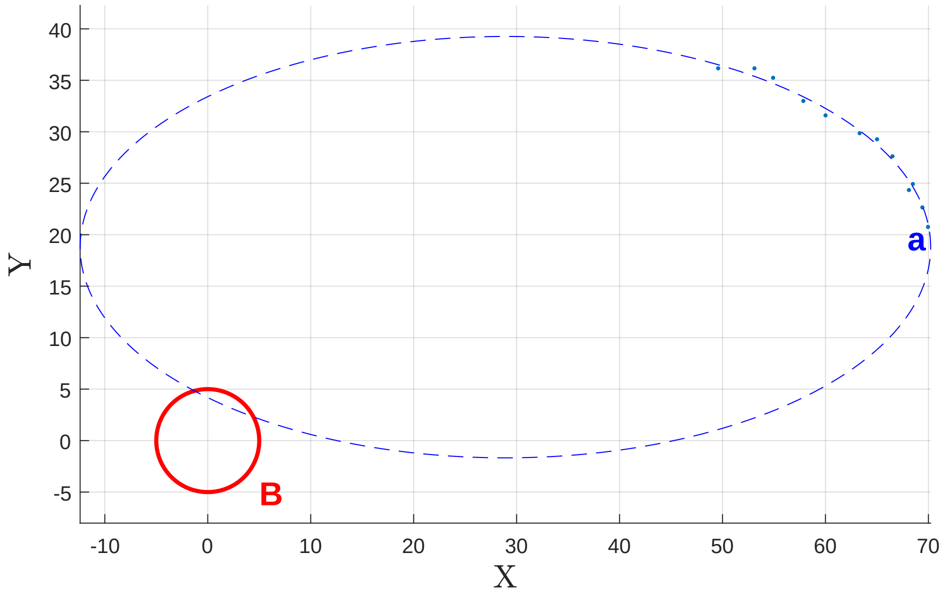

Figure 1: Numerical estimation of the orbit of the celestial body a

to show if two bodies will collide, we have to mathematically solve the following equations to find points which satisfy both equations as follow

$$X^2 + Y^2 = R^2 \text{ for body B} \tag{1.4}$$

$$\frac{(X - C_1)^2}{R_1^2} + \frac{(X - C_2)^2}{R_2^2} = 1 \text{ for body a - ellipsis} \tag{1.5}$$

2. Task 2

1. Write a program to automatically set detection threshold for power samples [dBm]. This threshold should separate the power samples into H0: noise only and H1: signal + noise. When setting the threshold, how did you solve the problem of having signals at unknown locations (no known noise-only samples)? Can you estimate the probability of false alarm that your detection threshold will give? Is this PFA reasonable? Please notice that using solution that will only work for this specific dataset will give less points.
2. Using this threshold (if you could not solve the previous proble, then you may use here manually selected threshold like $-xx$ dBm), plot a figure showing which samples are H_1 and which samples are H_0. Does the result look reasonable?
3. Our task in signal detection is to especially detect weak signals, how should you take this into account when setting the threshold?
4. Using a detection threshold, automatically by MATLAB estimate the signal

parameters: pulse period (after what time is the pulse again repeated?) and pulse duration (how long is each individual pulse?). Do the estimation by taking into account all the received data and not only for example the first pulse. How accurate would you say the signal generator which generated these signals is (does pulse period and duration stay constant)?
5. How would you handle the case of very weak signals (sometimes under noise), sketch an algorithm. Innovative solutions will give more points!

According to the Matlab code provided in Fig.5, we calculate the threshold γ and the probability of false alarm (PFA). First we define the hypotheses as $H_0 : x = n$ and $H_1 : x = r + n$. We know that a false alarm occurs whenever the noise voltage exceeds a defined threshold voltage. Noise level which determines the threshold consists of noise floor $NF = \text{mean}(10log_{10}(|\text{data}|^2)$ and variance of noise voltage which is the noise power in white spaces is written as $\sigma^2_{v_n} = NF - \min(10log_{10}(|\text{data}|^2))$ and threshold is $\gamma = 2 \times \text{mean}(10log_{10}(|\text{data}|^2)) - \min(10log_{10}(|\text{data}|^2))$. According to (2.6), we find the probability of false alarm as follow

$$PFA = Prob(v_\gamma < R < \infty) = \int_{v_\gamma}^{\infty} \frac{R}{\sigma^2_{v_n}} \exp\big(\frac{-R^2}{2\sigma^2_{v_n}}\big) dR = \exp\big(\frac{-v_\gamma^2}{2\sigma^2_{v_n}}\big) \tag{2.6}$$

where v_γ is the voltage of threshold and R is the amplitude of the envelope of the filter output which has the Rayleigh fading distribution. We can simplify (2.6) by updating v_γ^2 and $\sigma^2_{v_n}$ by $10^{0.1\gamma}$ and $10^{0.1NF}$; thus, $PFA = \exp\big(\frac{-(10^{0.1\gamma}-10^{0.1NF})}{2}\big)$. Hence, according to the PFA and threshold formulas, we get $\gamma = -78.1213$ dBm and PFA equal to 6.1742×10^{-5}.

Now, we can determine the region of each hypothesis in such a way that signals over the threshold are in H_1 region and signals lower than the threshold are in H_0 region. In Fig.2, we show the threshold with the solid red line, and the purple shadowed region indicates H_1 region.

*** question 3 ***

According the threshold calculated according to the Matlab code provided in Fig.5b, we find the pulse period and pulse duration parameters. First, we convert the signal to a square signal and then we show the signal over the threshold with 1 and signal lower than that with 0. We use $findpeaks$ function to find the rising and falling edges of the signal as indicated in Fig.3. Afterward, we find the pulse period by finding the time difference between two 0 signals or two 1 signals. Pulse duration is equal to the time difference of a 0 signal and 1 signal or vice versa. We get pulse period equal to 0.001 (sec) and pulse duration equal to 6.4844×10^{-6} (sec).

*** question 5 ***

Numerical Values	
Threshold (γ),	-78.1213 dBm
PFA	6.1742×10^{-5}
Pulse period	0.001 (sec)
Pulse duraiotn	6.4844×10^{-6}(sec)

Table 1: Table of numerical values

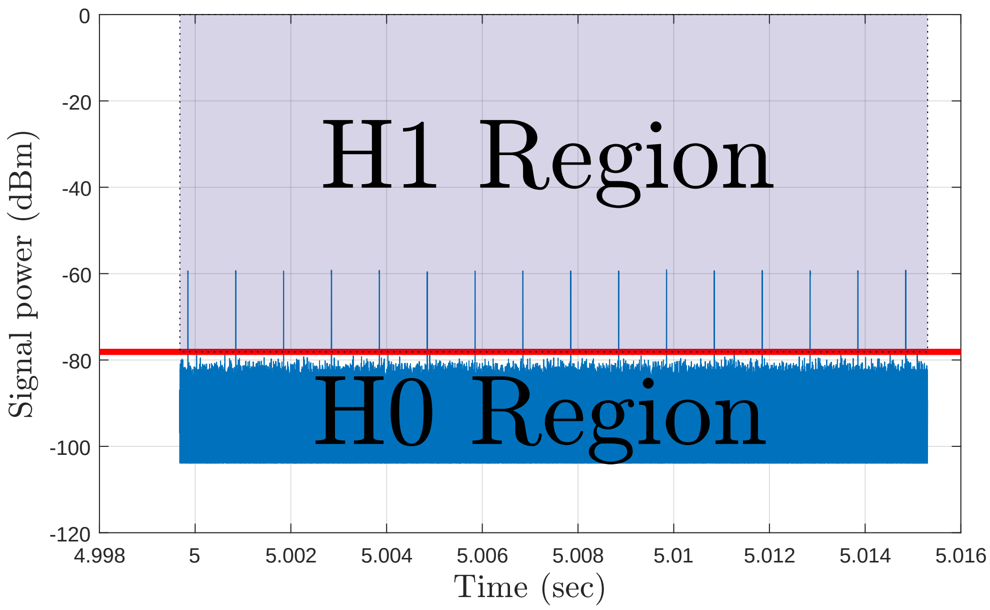

Figure 2: Signal power versus time. H_1 region is the purple shaded region and red line is the threshold $\gamma = -78.1213$ dBm.

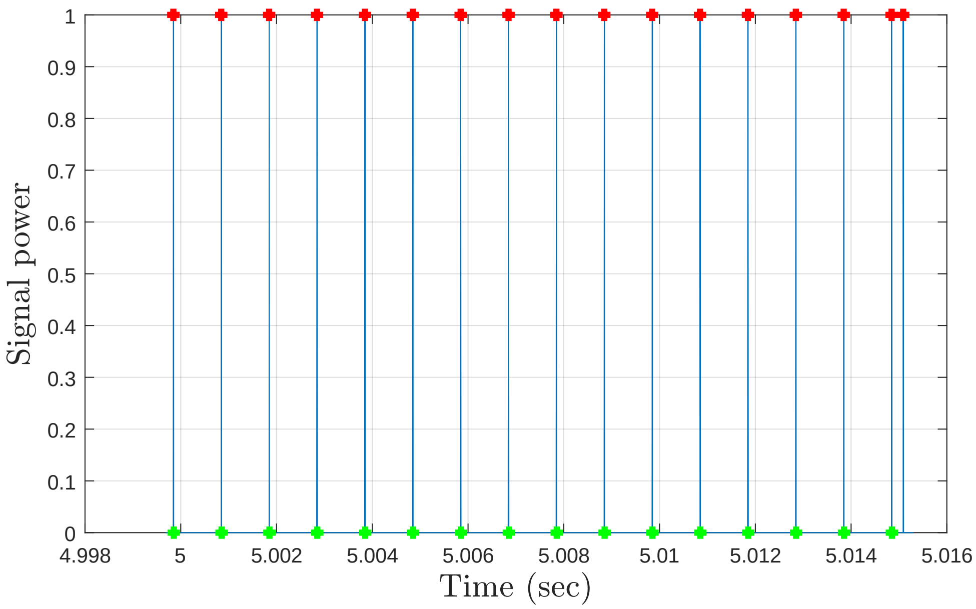

Figure 3: Square signal of our data.

```
%% ----------------------------------------
clc;close all;clear; % reset all
% ----------------------------------------


% (X,Y) Measurements of the position of the celestial body 'a' for
each
% month:
X = [69.9610    69.4111    68.1078    68.4906    66.5073    65.0106
63.3201  60.0063  57.8493  54.9152  53.1017  49.5782];
Y = [20.7559    22.6554    24.3398    24.9237    27.6180    29.2726
29.8596  31.5923  32.9917  35.2425  36.1634  36.1643];

% Orbital period of 'a':
T = 12*6;

% Plot the diagram:
f1 = figure(1)
set(gcf, 'Units', 'centimeters'); % set units
axesFontSize = 16;
afFigurePosition = [2 7 19 11]; % [pos_x pos_y width_x width_y]
set(gcf,            'Position',            afFigurePosition,'PaperSize',[19
11],'PaperPositionMode','auto');  %  [left  bottom  width  height],
setting printing properties
% plotting
scatter(X, Y, '.', 'LineWidth',1);
hold on;
h = ellipse(5, 5, 0, 0,0, 'r');
set(h, 'LineWidth', 2,  'LineStyle','-');
hold off;
axis equal;
grid on
xlabel('X','Interpreter', 'latex','fontsize',axesFontSize);
ylabel('Y','Interpreter', 'latex','fontsize',axesFontSize);
```

(a)

```
text(5,-5, 'B', 'Color', 'Red', 'FontWeight', 'Bold', 'FontSize', 16);
text(X(1)-2,Y(1)-1,    'a',    'Color',    'Blue',    'FontWeight',    'Bold',
'FontSize', 16);

%
************************************************************
**************
%
************************************************************
**************
%
%  ... complete this part by writing a piece of code that estimates the
%  four parameters of the ellipse using LS-estimation ...
%
%  NOTE: The estimated parameters R1,R2,C1,C2 must be stored
in a 4-element
%  vector variable named P.
%% X = HA+W linear LS estimator --> A_hat = (H'H)^-1 H'X
k = 1:12;
%estimator for X_k
Hx(k,1) = cos((2*pi.*k)/72);%observation matrix
Hx(k,2) = 1;%observation matrix
A_hat_x =(Hx'*Hx)^-1* Hx'*X'; %estimator

%estimator for Y_k
Hy(k,1) = sin((2*pi.*k)/72);%observation matrix
Hy(k,2) = 1; %observation matrix
A_hat_y =(Hy'*Hy)^-1* Hy'*Y'; %estimator

P = [A_hat_x(1) A_hat_y(1) A_hat_x(2) A_hat_y(2)];

h = ellipse(P(1), P(2), 0, P(3), P(4), 'b');
set(h, 'LineStyle','--');
filename = ['EX2.pdf'];
print(f1,filename,'-dpdf'); % print pdf file
```

(b)

Figure 4: Corresponding Matlab code for the orbit of the celestial body a

```
%% ----------------------------------------
clc;close all;clear; % reset all
% ----------------------------------------
%% settings (Task 2)
% ----------------------------------------
load data_high_snr
SP = 10*log10(abs(data).^2); % signal power in dBm
SL = abs(data).^2; % linear value of signal power
S = SP(find(SP~=-Inf)); % detecting suitable signals
%%
% Plot the diagram
f1 = figure(1);
set(gcf, 'Units', 'centimeters'); % set units
axesFontSize = 16;
afFigurePosition = [2 7 19 11]; % [pos_x pos_y width_x width_y]]
set(gcf, 'Position', afFigurePosition,'PaperSize',[19
11],'PaperPositionMode','auto'); % [left bottom width height], setting
printing properties
plot(timet,10.*log10(abs(data).^2))
hold on
grid on
xlabel('Time (sec)','Interpreter', 'latex','fontsize',axesFontSize)
```

(a)

```
ylabel ('Signal power (dBm)','Interpreter',
'latex','fontsize',axesFontSize)
filename = ['T2_signal' '.pdf'];
print(f1,filename,'-dpdf'); % print pdf file
%**** PART I ********
% Finding the detection threshold where we have H0: x = n and H1: x =
r+n
% A false alarm occurs whenever the noise voltage exceeds a defined
threshold voltage
% The threshold (gama) is determined according to the noise level which
consists
% of noise floor and variance of noise volatage
% probability of the false alarm is equal to exp(-1/2 10^(noise floor - min
(s)))
gama = 2*mean (S) - min (S); % Threshold in terms of noises
FA = exp(-0.5*10^(0.1*(mean(S)-min(S)))); % Probability of false alarm
%Prininting the value of threshold(gama) and PFA
name1 ='Threshold';
name2 = 'PFA';
X1 = [name1,' is ',num2str(gama)];
X2 = [name2,' is ',num2str(FA)];
disp(X1)
disp(X2)
```

(b)

Figure 5: Corresponding Matlab code for threshold and false alarm calculation.

```
%**** PART II ********

% settings

% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

% Plot the diagram:

f2 = figure(2);

set(gcf, 'Units', 'centimeters'); % set units

axesFontSize = 16;

afFigurePosition = [2 7 19 11]; % [pos_x pos_y width_x width_y]

set(gcf, 'Position', afFigurePosition,'PaperSize',[19 11],'PaperPositionMode','auto'); % [left bottom
width height], setting printing properties

%ploting signal... if signal is larger than the threshold, we select H1 is the decision region, otherwise
H0 is the region. So, we plot the signal for each region as follow

plot(timet,10.*log10(abs(data).^2))

hold on

hline = refline([0 gama]);%Threshold gama = -78.1213 dBm

hline.Color = 'r';

set(hline,'LineWidth',3);

pa = area([min(timet) min(timet) max(timet) max(timet)], [0 gama gama 0],'LineStyle',':');% H1
region

pa.FaceAlpha = 0.2;

hold off

xlabel('Time (sec)','Interpreter', 'latex','fontsize',axesFontSize)

ylabel ('Signal power (dBm)','Interpreter', 'latex','fontsize',axesFontSize)

grid on

filename = ['T2_P2.pdf'];

print(f2,filename,'-dpdf'); % print pdf file

%Signals over the threshold indicated with the red line, are in H1 region(Shadowed region)

%Signals lower than the threshold are in region H0
```

Figure 6: Corresponding Matlab code to show the decision regions.

```matlab
%%
%**** PART IV ********
% settings
% ----------------------------------------
data_db = 10.*log10(abs(data).^2); %signal poswer
data_proc = zeros(size(data_db));
data_proc(data_db - gama>0) =1; % converting to square signal
[pk,lc] = findpeaks(data_proc); % rising edge of square signal
[pk1,lc1] = findpeaks(-data_proc); % falling edge of square signal
f3 = figure(3); %Ploting the square signal with rising and falling edges
set(gcf,'Units','Centimeters');%set units
axesFontSize = 16;
afFigurePosition = [2 7 19 11];
set(gcf,'Position',afFigurePosition,'PaperSize',[19
11],'PaperPositionMode','auto');
plot(timet,data_proc) ;% ploting the signal power
hold on
scatter(timet(lc),pk,'r+', 'LineWidth',3);% ploting the rising edge of signal
scatter(timet(lc1),pk1,'g+','LineWidth',3) ;% ploting the falling edge of signak
grid on
hold off
xlabel('Time (sec)','Interpreter','latex','fontsize',axesFontSize);
ylabel('Signal power','Interpreter','latex','fontsize',axesFontSize);
```

(a)

```matlab
periods = zeros(size(lc,1) - 1,1);
puls_size = zeros(size(lc,1) - 1,1);
for i=1:(size(lc,1)-1)
    periods (i)= timet(lc(i+1)) - timet(lc(i)); % pulse period
    puls_size (i) = timet(lc1(i)) - timet(lc(i)); % pulse duration
end
% all values of periods(i) have the same value so periods(1)=periods(2)=...
% it also holds for puls_size (i) where puls_size(1) = puls_size(2) = ...
% so we can print the value of pulse period and pulse duration by
% considering the the first duration
%Printing the value of pulse period and pulse duration
name1 = 'Pulse period';
name2 = 'Pulse duration';
X1 = [name1, ' is ',num2str(periods(1))];
X2 = [name2, ' is ',num2str(puls_size(1))];
disp(X1)
disp(X2)
```

(b)

Figure 7: Corresponding Matlab code to calculate the pulse period and pulse duration.