UNIVERSITY OF OULU

FACULTY OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING - ITEE

STATISTICAL SIGNAL PROCESSING 1

MATLAB TASK #3

Group: 14

Students:
- AZZAZ Aissa (Number: 2207335)
- BOULFRAD Mourad (Number: 2207592)

Due date: October 11th 2022

1. ML estimator

The parameters that need to estimated are the unknown characteristics $\theta = [A \ f_0 \ \phi]^T$ of a sinewave using the observed data that reassembles the following model

$$x[n] = A\cos(2\pi f_0 n + \phi) + w[n]$$

with $n = 0, 1, 2 \ldots N-1$ and $A > 0$, and $0 < f_0 < \frac{1}{2}$.

Examples 3.14 and 7.16 present the estimators of the parameters as:

- For the Amplitude A:

$$\hat{A} = \frac{2}{N}\left|\sum_{n=0}^{N-1} x[n]e^{-j2\pi\hat{f}_0 n}\right| \text{ with } var(\hat{A}) \geq \frac{2\sigma^2}{N}$$

- For the phase ϕ:

$$\hat{\phi} = \arctan\left(\frac{-\sum_{n=0}^{N-1} x[n]\sin\left(2\pi\hat{f}_0 n\right)}{\sum_{n=0}^{N-1} x[n]\cos\left(2\pi\hat{f}_0 n\right)}\right) \text{ with } var(\hat{\phi}) \geq \frac{4\sigma^2(2N-1)}{A^2N(N+1)}$$

- For the frequency f_0: it's the value that minimizes the periodogram

$$I(f) = \frac{1}{N}\left|\sum_{n=0}^{N-1} x[n]e^{-j2\pi fn}\right|^2 \text{ with } var(\hat{f}_0) \geq \frac{24\sigma^2}{(2\pi)^2A^2N(N+1)}$$

The code below implements these estimators and their theoretical variances. It starts by setting the different Monte-Carlo simulation parameters and generating the noisy samples. Then it calculates the estimators. Here, the code calculates the periodogram for different values in the frequency range and finds the index of the maximum values in each MC loop, which then is used to determine the frequency values of the estimator. The mean value of the frequency estimator is then used in the amplitude and phase estimators. After that, the code calculates the theoretical and simulated variances and means of the estimators which are presented in Tables 1 and 2.

```
close all ; clear ; clc ;
rng(0) ; % reset the random number generator (for reproducibility)

A = 1 ;
f0 = 1/4 ; f_min = 0 ; f_max = 1/2 ;
phi = pi/3 ;
sigma_squared = 0.001 ;
MC = 100000 ; % number of Monte Carlo loops
N = 5 ; % Number of samples in the data
n = (0:N-1)' ; % col vector to represent the index of each sample

%%% Generating of the samples
noise = randn(MC,N)*sqrt(sigma_squared) ; % W ==> N columns of WGN repeated MC rows
signal = A*cos(2*pi*f0.*n + phi) ; % H*A ==> linear model of the signal of interest
X = signal + noise' ; % captured samples with MC times in columns and has N samples in rows

%%% estimators
f = linspace(f_min,f_max,MC/100) ; % frequency samples (only MC/100 for simulation speed)
f = f(2:end-1) ; % truncate the limits
I = nan*ones(MC,length(f)) ;
for ii = 1:length(f)
    I(:,ii) = (1/N) * abs( sum( X.*exp(-1j*2*pi*f(ii).*n) ) ).^2 ;
end
[~,index] = max(I,[],2) ; % index of maximum of I(f) along each MC loop
f_estimate_values = f(index) ; % the value of f that resulted in the maximum in each MC loop
f_estimate = mean(f_estimate_values)
```

```
  f_estimate =
      249.993473473474e-003
```

```
A_estimate = (2/N)* abs( sum(X.*exp(-1j*2*pi*f_estimate.*n)) ) ;

phi_estimate = atan(-sum(X.*sin(2*pi*f_estimate.*n))./sum(X.*cos(2*pi*f_estimate.*n))) ;

%%% Theoretical variances
var_f_estimate_theory = (24*sigma_squared)/((2*pi)^2*A^2*N*(N-1))
var_A_estimate_theory = 2*sigma_squared/N
var_phi_estimate_theory = (4*sigma_squared*(2*N-1))/(A^2*N*(N+1))

results_var = [var_A_estimate_theory var_f_estimate_theory var_phi_estimate_theory ; ...
                  var(A_estimate) var(f_estimate_values) var(phi_estimate)]
```

```
  results_var = 2x3
      400.000000000000e-006    30.3963550927013e-006    1.20000000000000e-003
      389.709034711724e-006    8.99251827906217e-006    486.734453869912e-006
```

```
results_mean = [A f0 phi ; mean(A_estimate) mean(f_estimate_values) mean(phi_estimate)]
```

```
    results_mean = 2x3
        1.00000000000000e+000    250.000000000000e-003    1.04719755119660e+000
        916.724555046527e-003    249.993473473474e-003    857.231348328749e-003
```

Table 1: Simulated and theoretical variances of the estimators

	$\sigma^2_{\hat{A}}$	$\sigma^2_{\hat{f}_0}$	$\sigma^2_{\hat{\phi}}$
Simulated	0.000400	0.00003039	0.001200
Theoretical	0.000389	0.00000899	0.000486

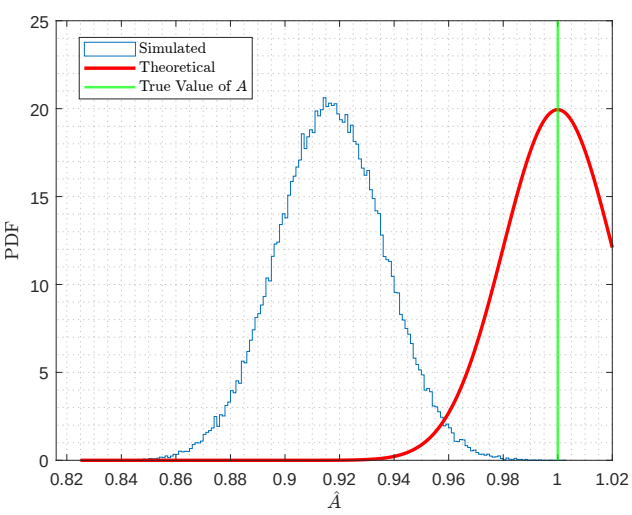
Table 2: Simulated and theoretical means of the estimators

	$E\{\hat{A}\}$	$E\{\hat{f}_0\}$	$E\{\hat{\phi}\}$
Simulated	1.0000	0.2500	1.0472
Theoretical	0.9167	0.2500	0.8572

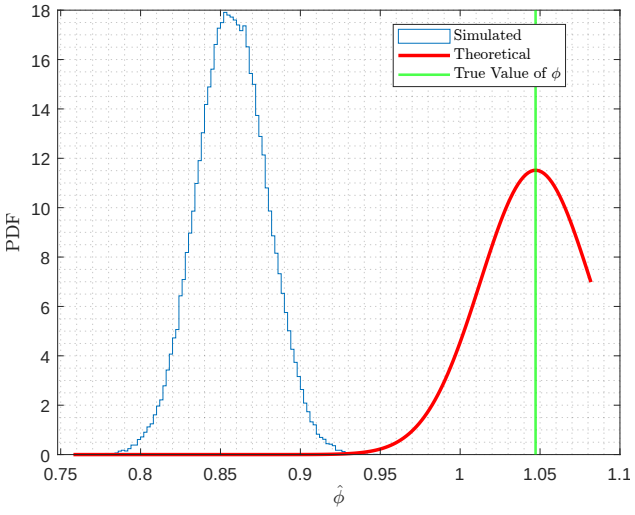
From the results on Tables 1 and 2, we notice that the frequency estimator results in a very accurate estimate that reaches the theoretical characteristics where the mean is almost identical to specified simulation parameter and the variance reaches the CRLB. However, the amplitude and phase estimators have a small bias and their variances (especially the phase estimator) don't approach the CRLB well enough. To have better estimations, one way is to increase the number of samples. To further visualize the estimation and the previously mentioned results, the code below plot the theoretical and simulated pdfs of each estimator, as shown in Figure 1.

```
fig1 = figure ;
A1 = histogram(A_estimate,'Normalization','pdf','DisplayStyle',"stairs") ;
hold on
A2 = linspace(min(A1.BinLimits(1),(A-sqrt(var_A_estimate_theory))), ...
      max(A1.BinLimits(2),A+sqrt(var_A_estimate_theory)),200) ;
A_theor_PDF = normpdf(A2,A,sqrt(var_A_estimate_theory)) ;
plot(A2, A_theor_PDF,'r-','Linewidth',2) ;
xline(A,'-g','Linewidth',1.5) ; % true value
hold off
grid minor
xlabel('$\hat{A}$','interpreter','latex') ;
ylabel('PDF','interpreter','latex') ;
legend('Simulated','Theoretical','True Value of $A$','interpreter','latex','Location','best') ;

fig2 = figure ;
```

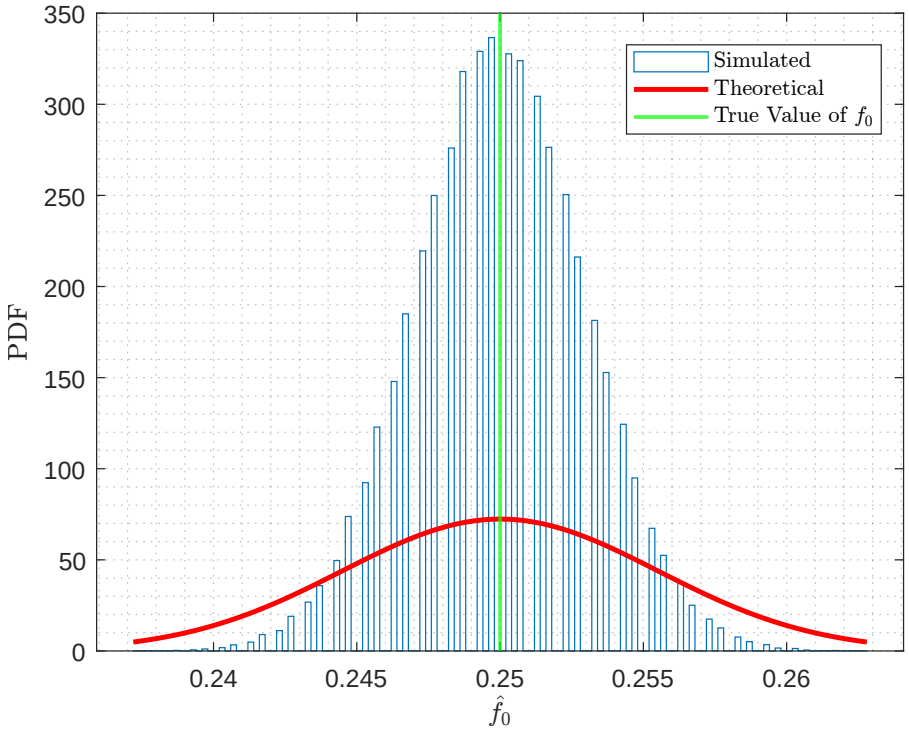
```
f1 = histogram(f_estimate_values,'Normalization','pdf','DisplayStyle',"stairs") ;
hold on
f2 = linspace(min(f1.BinLimits(1),(f0-sqrt(var_f_estimate_theory))), ...
      max(f1.BinLimits(2),f0+sqrt(var_f_estimate_theory)),200) ;
f_theor_PDF = normpdf(f2,f0,sqrt(var_f_estimate_theory)) ;
plot(f2, f_theor_PDF,'r-','Linewidth',2) ;
xline(f0,'-g','Linewidth',1.5) ; % true value
hold off
grid minor
xlabel('$\hat{f}_0$','interpreter','latex') ;
ylabel('PDF','interpreter','latex') ;
legend('Simulated','Theoretical','True Value of $f_0$','interpreter','latex','Location','best') ;

fig3 = figure ;
phi1 = histogram(phi_estimate,'Normalization','pdf','DisplayStyle',"stairs") ;
hold on
phi2 = linspace(min(phi1.BinLimits(1),(phi-sqrt(var_phi_estimate_theory))), ...
       max(phi1.BinLimits(2),phi+sqrt(var_phi_estimate_theory)),200) ;
phi_theor_PDF = normpdf(phi2,phi,sqrt(var_phi_estimate_theory)) ;
plot(phi2, phi_theor_PDF,'r-','Linewidth',2) ;
xline(phi,'-g','Linewidth',1.5) ; % true value
hold off
grid minor
xlabel('$\hat{\phi}$','interpreter','latex') ;
ylabel('PDF','interpreter','latex') ;
legend('Simulated','Theoretical','True Value of $\phi$','interpreter','latex','Location','best') ;
```



(a) Amplitude



(b) Phase



(c) Frequency

Figure 1: The theoretical and simulated pdfs of each estimator

2. LSE Estimator

The parameters that need to estimated are the unknown characteristics $\theta = [f_0 \ \phi]^T$ of a sinewave using the observed data that reassembles the following model

$$x[n] = A\cos(2\pi f_0 n + \phi) + w[n]$$

with $n = -M, \ldots, 0, \ldots, M$.

For $M \gg 1$, Example 8.14 presented the LSE estimators of the parameters as:

- For the frequency f_0:

$$f_{0k+1} = f_{0k} - \frac{3}{2\pi M^2} \sum_{n=-M}^{M} nx[n] \sin\left(2\pi f_{0k} + \phi_k n\right)$$

- For the phase ϕ:

$$\phi_{k+1} = \phi_k - \frac{1}{M} \sum_{n=-M}^{M} nx[n] \sin\left(2\pi f_{0k} + \phi_k n\right)$$

The code below implements these estimators in an iterative approach.

```
close all ; clear ; clc ;
rng(0) ; % reset the random number generator (for reproducibility)

A = 1 ;
M = 2 ;
sigma_squared = 0.001 ;
f0 = 1/4 ;
phi0 = pi/3 ;
iter = 10 ;
n = (-M:M)' ; % col vector to represent the index of each sample

%%% Generating of the samples
noise = randn(length(n),1)*sqrt(sigma_squared) ; % W ==> N columns of WGN repeated MC rows
% var(noise)
signal = A*cos(2*pi*f0.*n + phi0) ; % H*A ==> linear model of the signal of interset
X = signal + noise ; % captured samples with MC times in columns and has N samples in rows

%%% estimators
f = nan*ones(1,iter+1) ;
phi = nan*ones(1,iter+1) ;

f(1) = 0.001 ;  % initial staring point for the iterations
phi(1) = 0.001 ; % initial staring point for the iterations

for ii =1:iter
    f(ii+1) = f(ii) - (3/(4*pi*M^3))*sum( n.*X.*sin(2*pi*f(ii).*n+phi(ii)) ) ;
    phi(ii+1) = phi(ii) - (1/M)*sum( X.*sin(2*pi*f(ii).*n+phi(ii)) ) ;
end

f0
```

```
  f0 = 0.2500
```

```
f_value = f(end)
```

```
  f_value = 0.2384
```

```
phi0
```

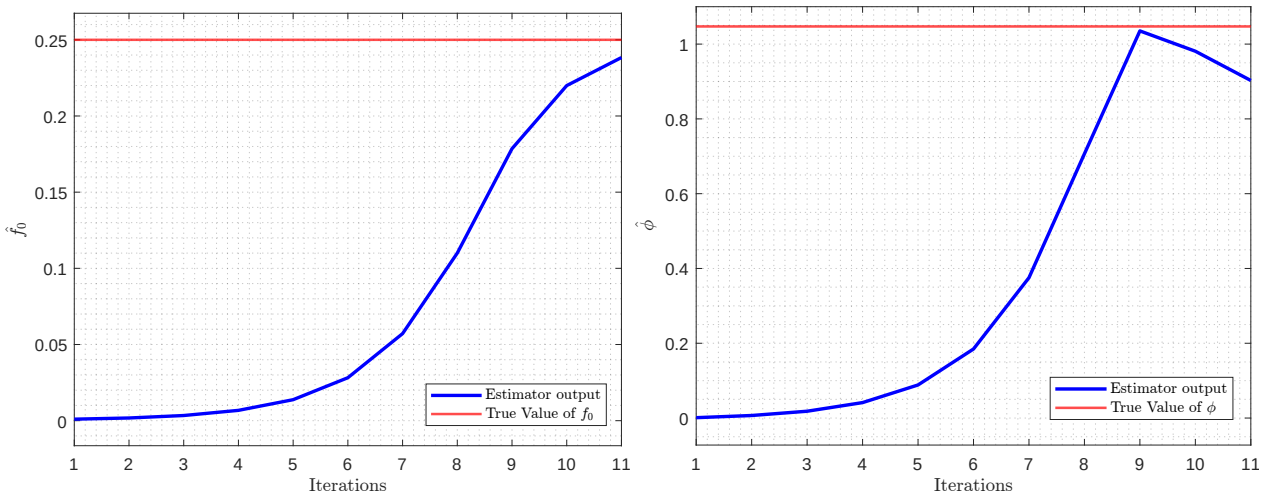
```
phi0 = 1.0472
```

```
phi_value = phi(end)
```

```
phi_value = 0.9027
```

Both estimator convergence approximately to the theoretical values (especially the frequency estimator) after only few iterations. To bttter present these results, the code below plots the convergence of the estimators outputs in each iteration, as shown in Figure 2. Because the phase and frequency estimation are coupled in the PLL approach, sometimes a better estimation of the frequency would result in divergence of the phase estimator from the actual value. To have a better convergence, we can the number of samples must be increased.

```
fig1 = figure ;
plot(1:iter+1,f,'b-','Linewidth',2) ;
yline(f0,'-r','Linewidth',1.5) ; % true value
hold off
grid minor
ylim padded
xlabel('Iteration','interpreter','latex') ;
ylabel('$\hat{f}_0$','interpreter','latex') ;
legend('Estimator output','True Value of $f_0$','interpreter','latex','Location','best') ;

fig2 = figure ;
plot(1:iter+1,phi,'b-','Linewidth',2) ;
yline(phi0,'-r','Linewidth',1.5) ; % true value
hold off
grid minor
ylim padded
xlabel('Iteration','interpreter','latex') ;
ylabel('$\hat{phi}$','interpreter','latex') ;
legend('Estimator output','True Value of $\phi_0$','interpreter','latex','Location','best') ;
```



(a) Frequency **(b)** Phase

Figure 2: The convergence of the LSE estimators