Bangladesh University of Engineering and Technology

Numerical Technique Laboratory

EEE 212

Experiment No.: 04

Name of the Experiment: Curve Fitting

Department: EEE

Section: C1

Group: 01

Student No.: 1406131

1406132

Date of Performance: September 24, 2016

Date of Submission: September 30, 2016

Curve Fitting

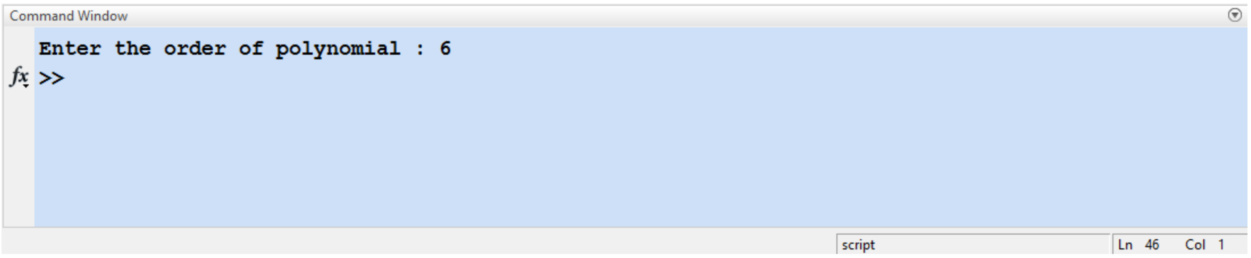
Polynomial Regression for nth Order

Code:

```
clearall ;
clc ,clf , close all ;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Curve Fitting
x = [0 1 2 3 4 5 ] ;
y = [ 2.1 7.7 13.6 27.2 40.9 61.1] ;

% plotting x and y as points
plot(x,y,'O','MarkerSize',6,'MarkerFaceColor','w','MarkerEdgeColor',...
'b','LineWidth',2) ;
holdon ;

% taking order of polynomial as input
input_value= input('Enter the order of polynomial : ') ;
order =  1 ;

% plotting curve from order 1 to input value
while order <= input_value + 1
power = 0 ;
for row = 1 : order
for col = 1 : order
A(row,col) = sum(x .^ power) ;
power = power + 1 ;
end
power = row ;    % as every time power of x start from the previous
% row value
end

for row = 1 : order
B(row,1) = sum( (x.^(row-1)) .* y) ;
end

coefficient = A\B ;

total = 0 ;
for i = 1 : length(coefficient) - 1
total = total + coefficient(i+1) * (x.^i) ;
end

new_y = total + coefficient(1) ;
plot(x,new_y) ;
% changing order
order = order + 1 ;
end
```

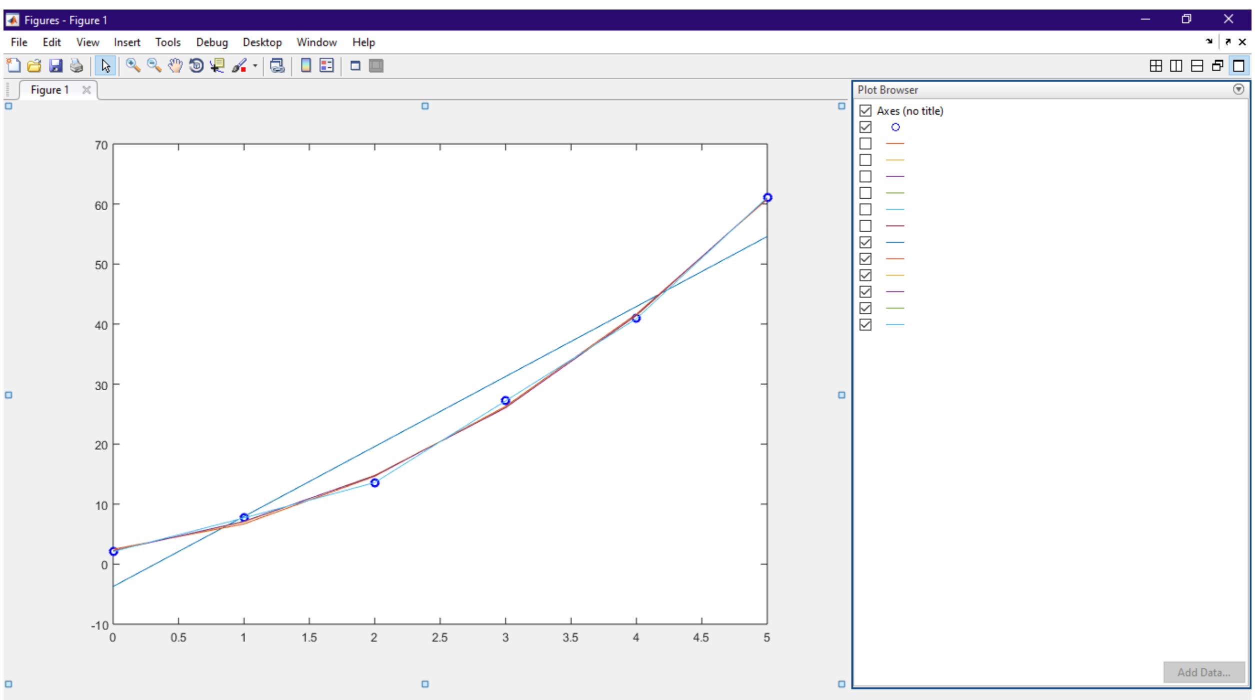
Output:

Command window output:



Output:

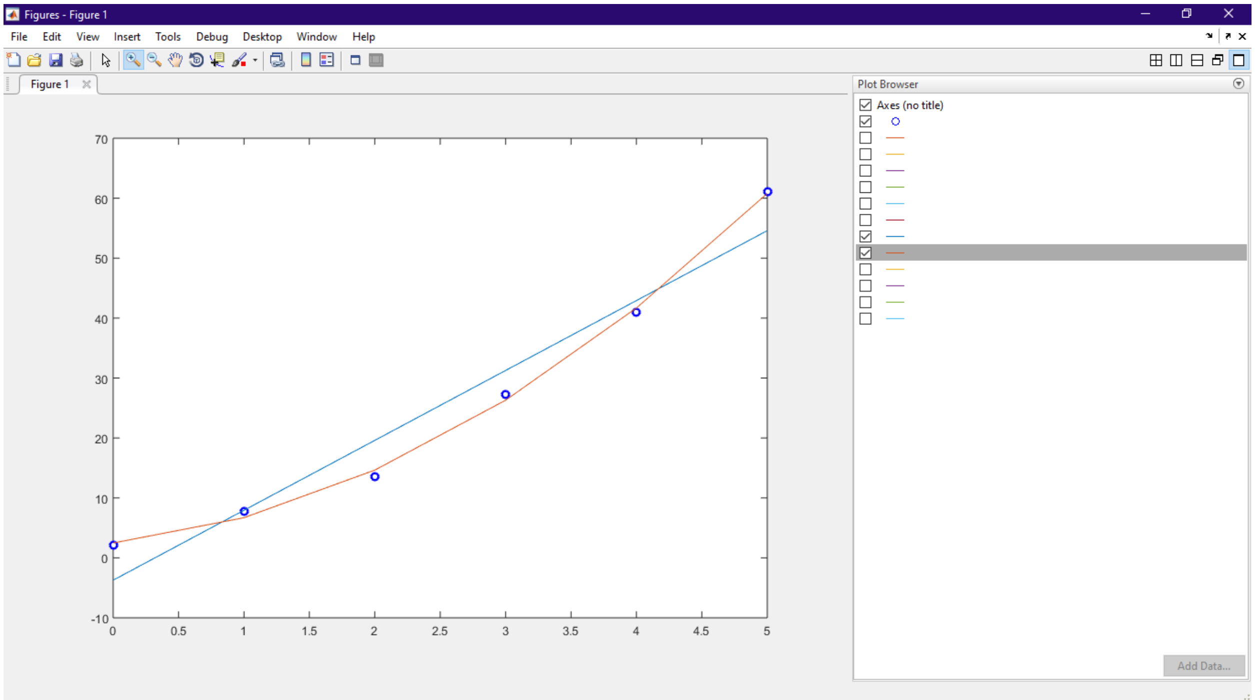
Output of polynomial regression for order 1 to 6 in a single plot



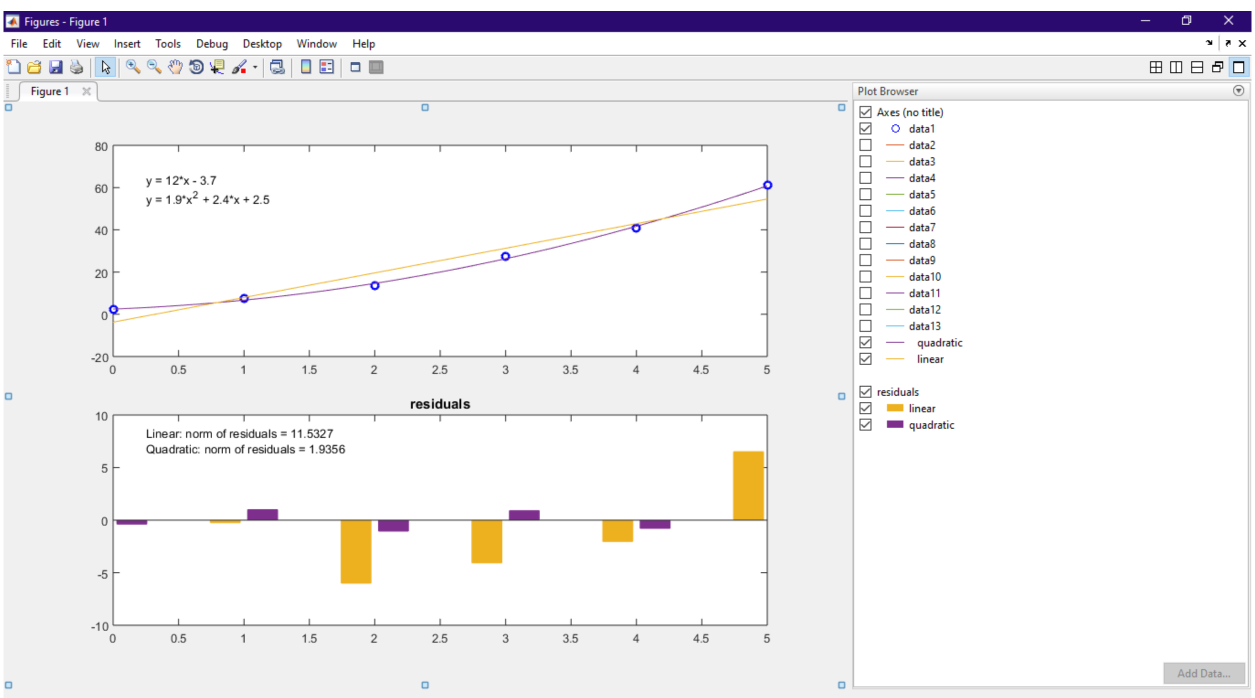
Comments:

<u>Output plot for order 1 & 2</u>

Here, two plots are highly separated.



Finding residuals and norm of the residuals from plot window:

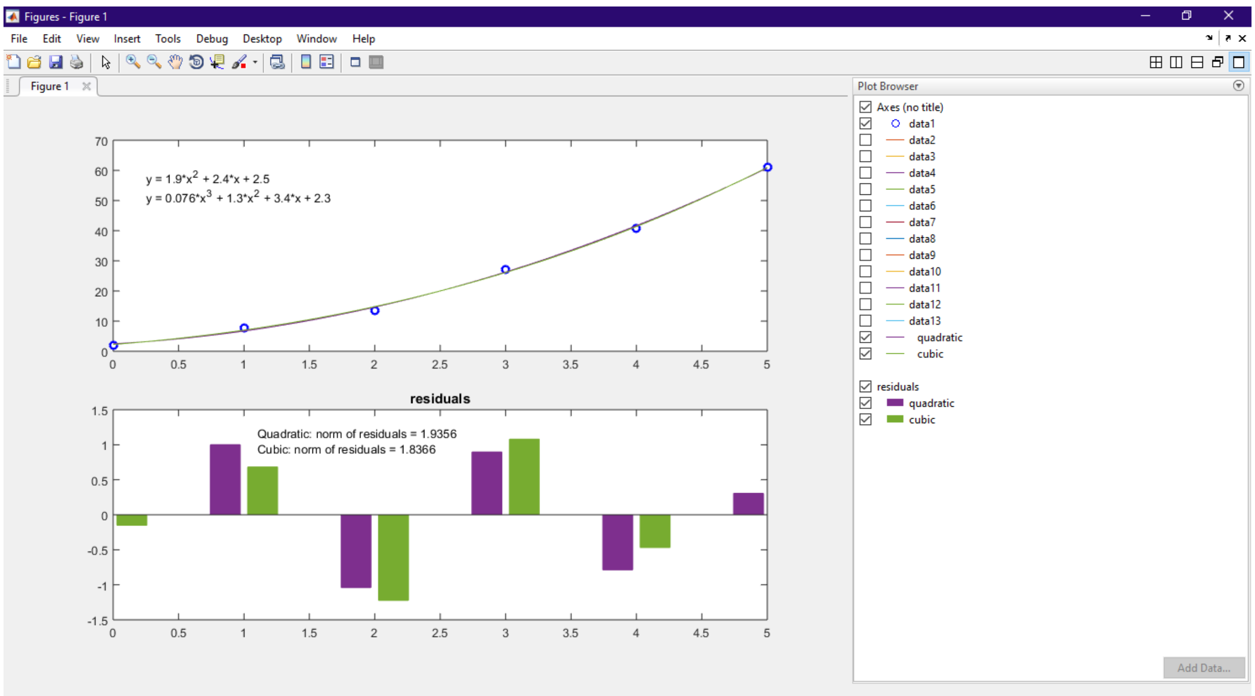


Residual is the difference between the sample and the estimated function value. If the points in a residual plot are randomly dispersed around the horizontal axis, a linear regression model is appropriate for the data otherwise a non-linear model is more appropriate. Here, residuals value is not randomly dispersed and non-random residuals indicate a non-linear better fitted curve.

So, a non-linear curve would be better fitted for the given points.

<u>Output plot for order 2 &3:</u>

Here, two plots are closely located. [Higher order plots are also closely located.]



Norm of the residuals is a measure of the deviation between the correlation and data. A lower norm of the residuals means a better fit. Here, norm of the residuals are:

Order 1: norm of residuals = 11.5327

Order 2: norm of residuals = 1.9356

Order 3: norm of residuals = 1.8366

The difference between the norm of residuals for order 1 & 2 is: 9.5971

The difference between the norm of residuals for order 2 & 3 is: .099

That is why plot for order 1 & 2 are highly separated and plot for order 2 & 3 are closely located.

Curve Fitting Of a Noisy Sin Wave

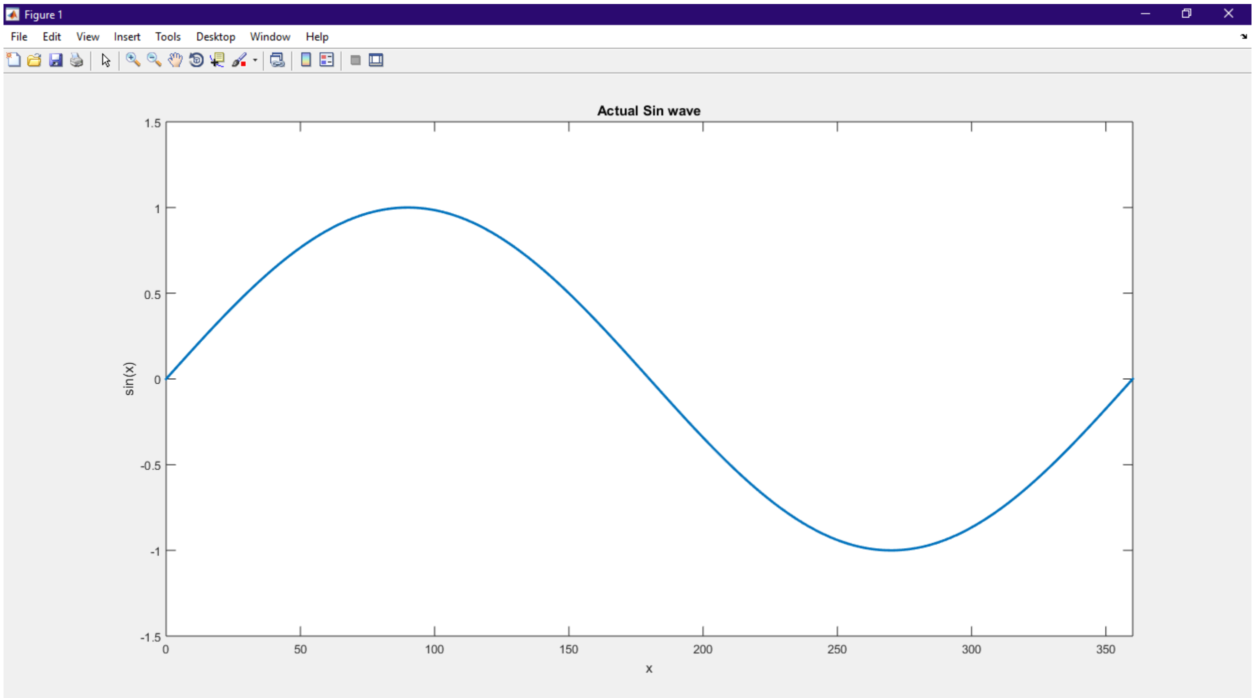
Code:

```
clearall ;
clc ,clf , close all ;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
x = 0:360 ;
y = sind(x) ;

plot(x,y,'LineWidth',2) ;
axis([0 360 -1.5 1.5]) ;
title('Actual Sin wave','LineWidth',2) ;
xlabel('x','LineWidth',2);
ylabel('sin(x)','LineWidth',2) ;

figure ;

min = -.1 ;
max = .1 ;
n = length(x) ;

% Random number generator within limit
noise = min + rand(1,n) * (max-min) ;
y = y + noise ;

plot(x,y,'LineWidth',1.5) ;
axis([0 360 -1.5 1.5]) ;
title('Sin wave with noise','LineWidth',2) ;
xlabel('x','LineWidth',2);
ylabel('sin(x) + Noise','LineWidth',2) ;

figure ;

new_x = [] ;
new_y = [] ;

for i = 1: 15 : 361
new_x = [new_xx(i)] ;
new_y = [new_yy(i)] ;
end

plot(new_x,new_y,'O','Linewidth',1.5) ;
axis([0 360 -1.5 1.5]) ;

holdon ;

% Curve Fitting
n = length(new_x) ;
order = 8 ;
power = 0 ;
```

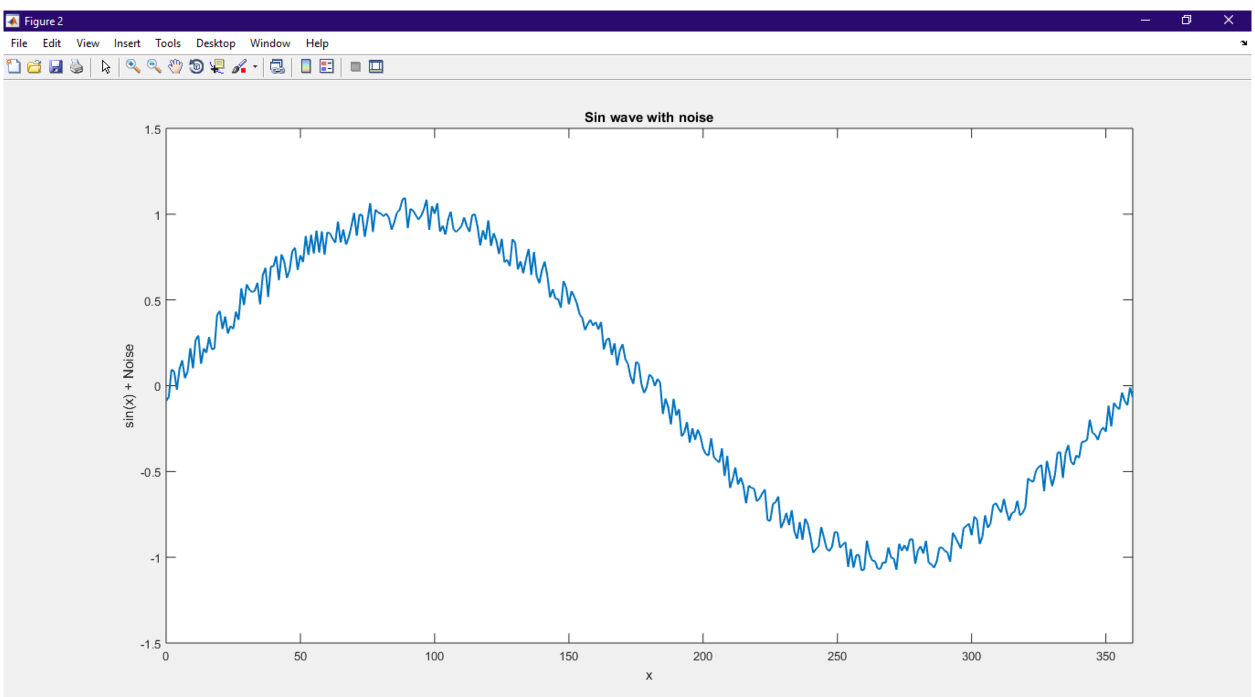
```
for row = 1 : order
for col = 1 : order
A(row,col) = sum(new_x .^ power) ;
power = power + 1 ;
end
power = row ;
end

for row = 1 : order
B(row,1) = sum( (new_x.^(row-1)) .* new_y) ;
end

coefficient = A\B ;

total = 0 ;
for i = 1 : length(coefficient) - 1
total = total + coefficient(i+1) * (new_x.^i) ;
end

estimated_y = total + coefficient(1) ;
plot(new_x,estimated_y,'Linewidth',2) ;
title('Estimated Sin wave','LineWidth',2) ;
axis([0 360 -1.5 1.5]) ;

x = 0:360 ;
y = sind(x) ;
actual_y = [] ;
for i = 1: 15 : 361
actual_y = [actual_yy(i)] ;
end

error = (actual_y - estimated_y) ./ actual_y ;
error = abs(error .* 100) ;

% Removing Inf and NaN from error matrix
error(~isfinite(error)) = 0;

figure ;

% plotting error
plot(error,'LineWidth',1.5) ;
title('Percentage of error between origianl sin wave and estimated sin
wave',...
'LineWidth',2) ;
ylabel('Error','LineWidth',2) ;
```

Output:

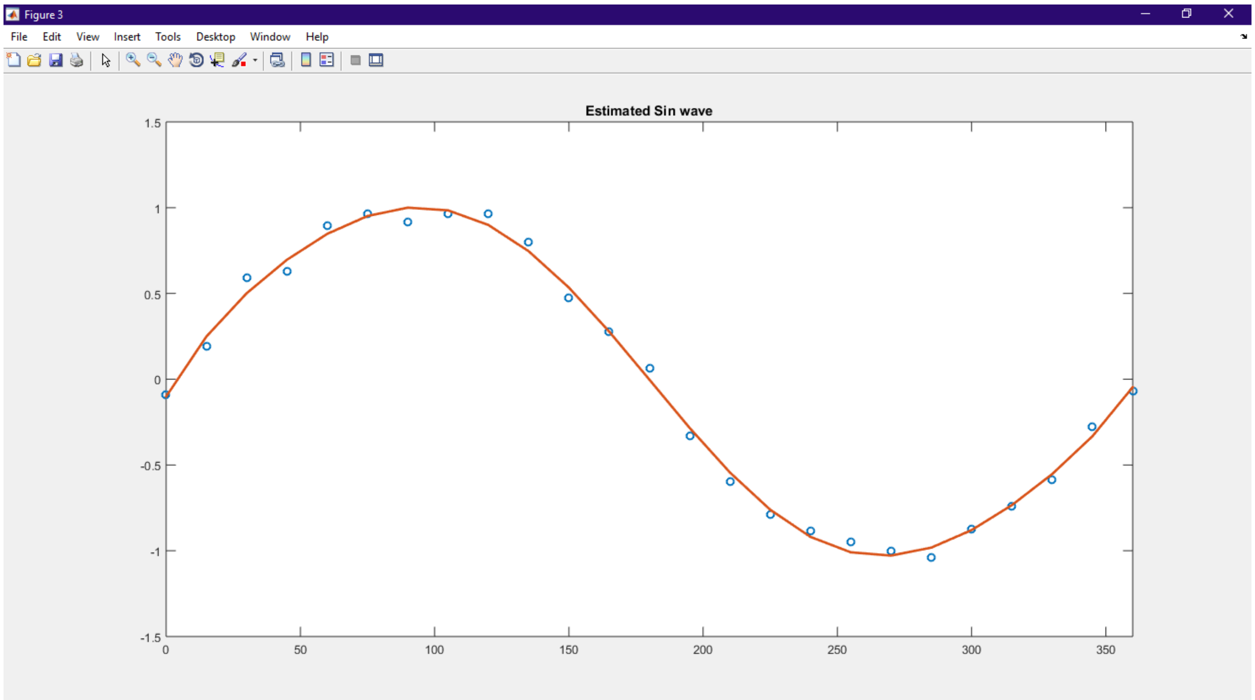
Actual Sin Wave:



Adding noise to the sin wave:



Estimated sin wave:



Percentage of error:

$$\text{Calculated as, percentage of error} = \frac{\text{Actual value} - \text{Estimated value}}{\text{Actual value}} \times 100\%$$