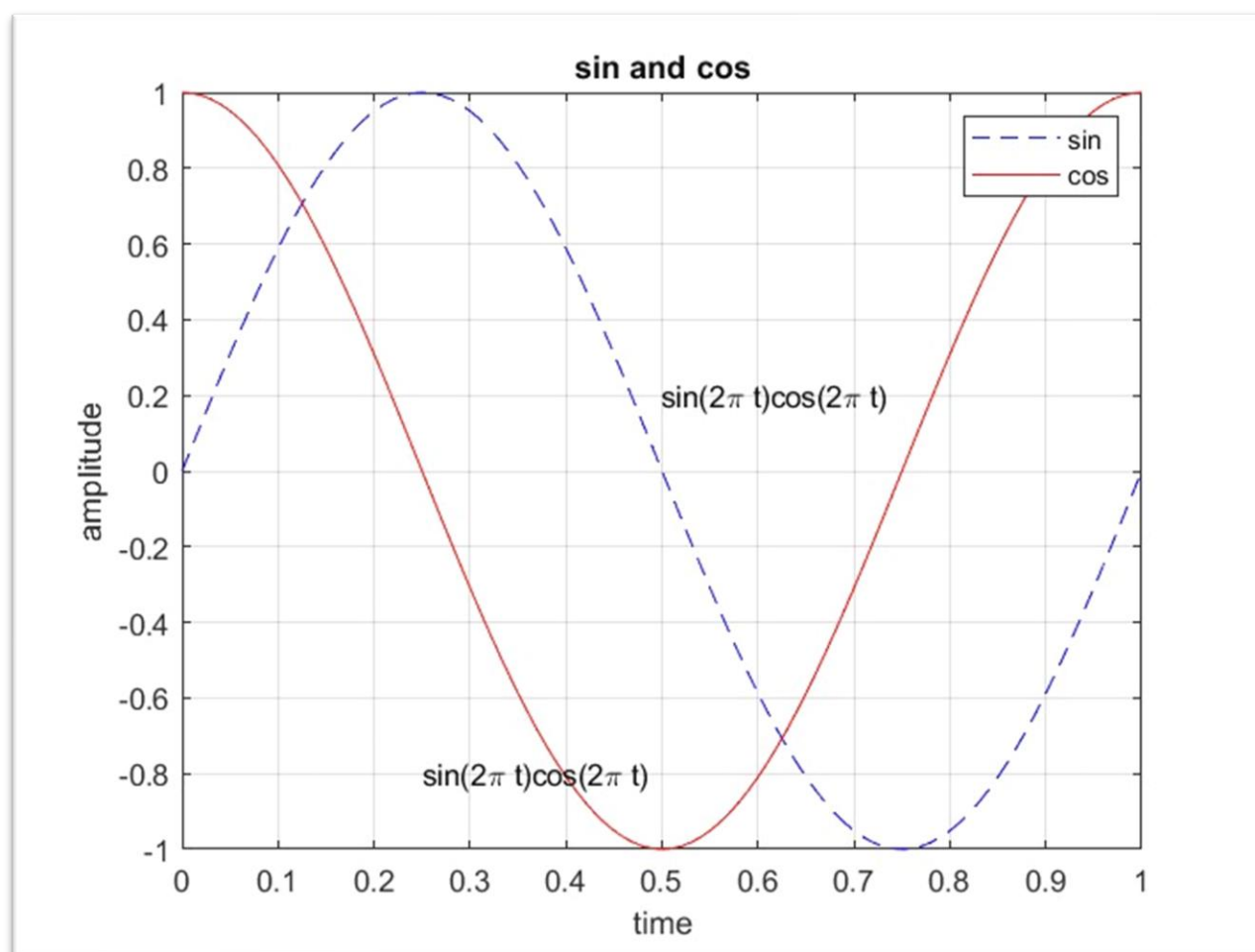


CA 1

امیرعلی شهریاری

۱_۱:

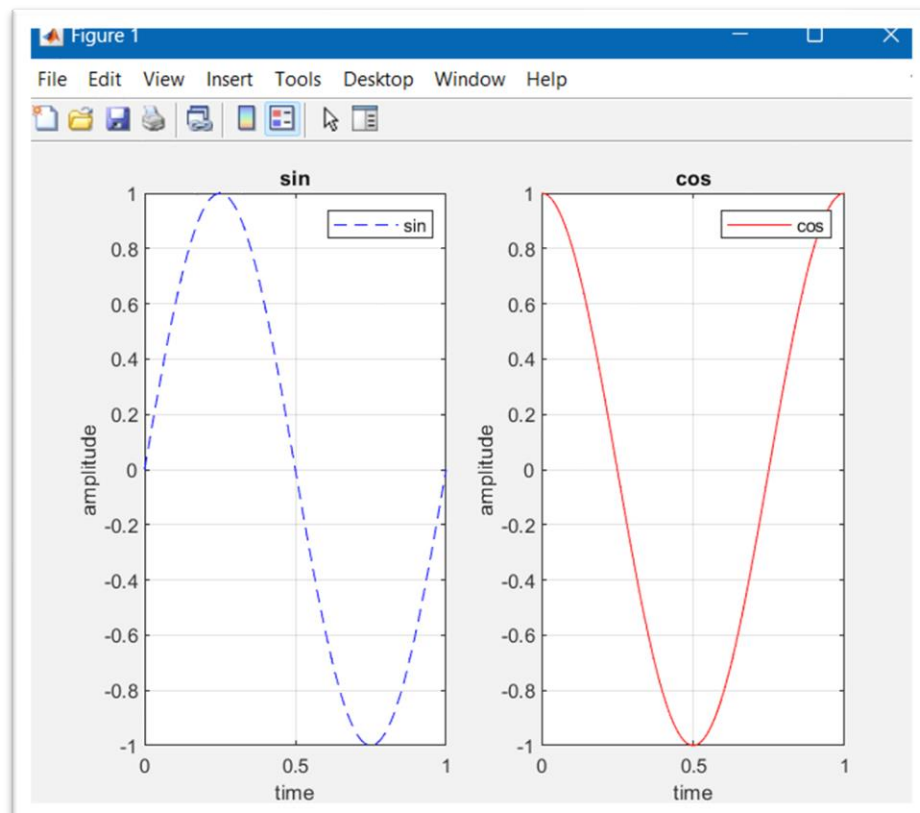
برای خط به خط ران کردن کد پشت آن برک پوینت قرار داده و با گزینه f10 خط به خط به پایین می رویم
شکل نهایی به این صورت می شود.



چنانچه دستور **hold on** را حذف کنیم با نگه داشتن، هر دو نمودار روی یک شکل نمایش داده می شوند و به شما امکان می دهد هر دو موج سینوسی و کسینوس را به طور همزمان مشاهده کنید. اگر **Hold on** را بردارید، فقط موج کسینوس نمایش داده می شود.

۱-۲:

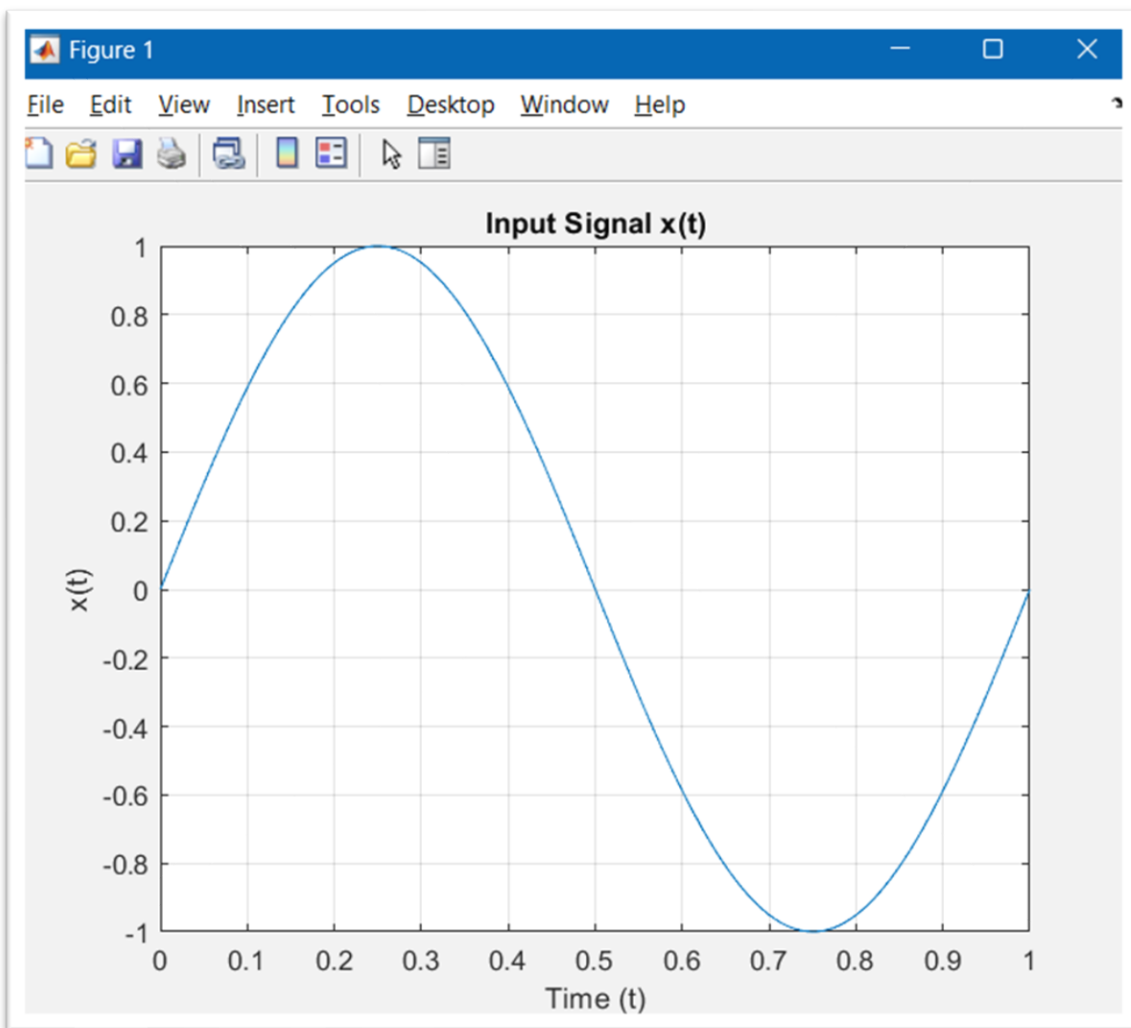
```
1 t=0:0.01:1;
2 z1=sin(2*pi*t);
3 z2=cos(2*pi*t);
4
5 figure;
6
7 subplot(1,2,1);
8 plot(t, z1, '--b');
9 title('sin');
10 xlabel('time');
11 ylabel('amplitude');
12 grid on;
13 legend('sin');
14
15 subplot(1,2,2);
16 plot(t, z2, 'r');
17 title('cos');
18 xlabel('time');
19 ylabel('amplitude');
20 grid on;
21 legend('cos');
```



از دستور subplot با دو ارگومان اول یک جدول ۱ در ۲ میسازیم که در ارگومان سوم مکان هر کدام از نمودار ها در جدول ساخته شده را می نویسیم سپس با استفاده از دستور plot دو نمودار را جداگانه رسم میکنیم و در کنار یکدیگر ترسیم می کنیم.

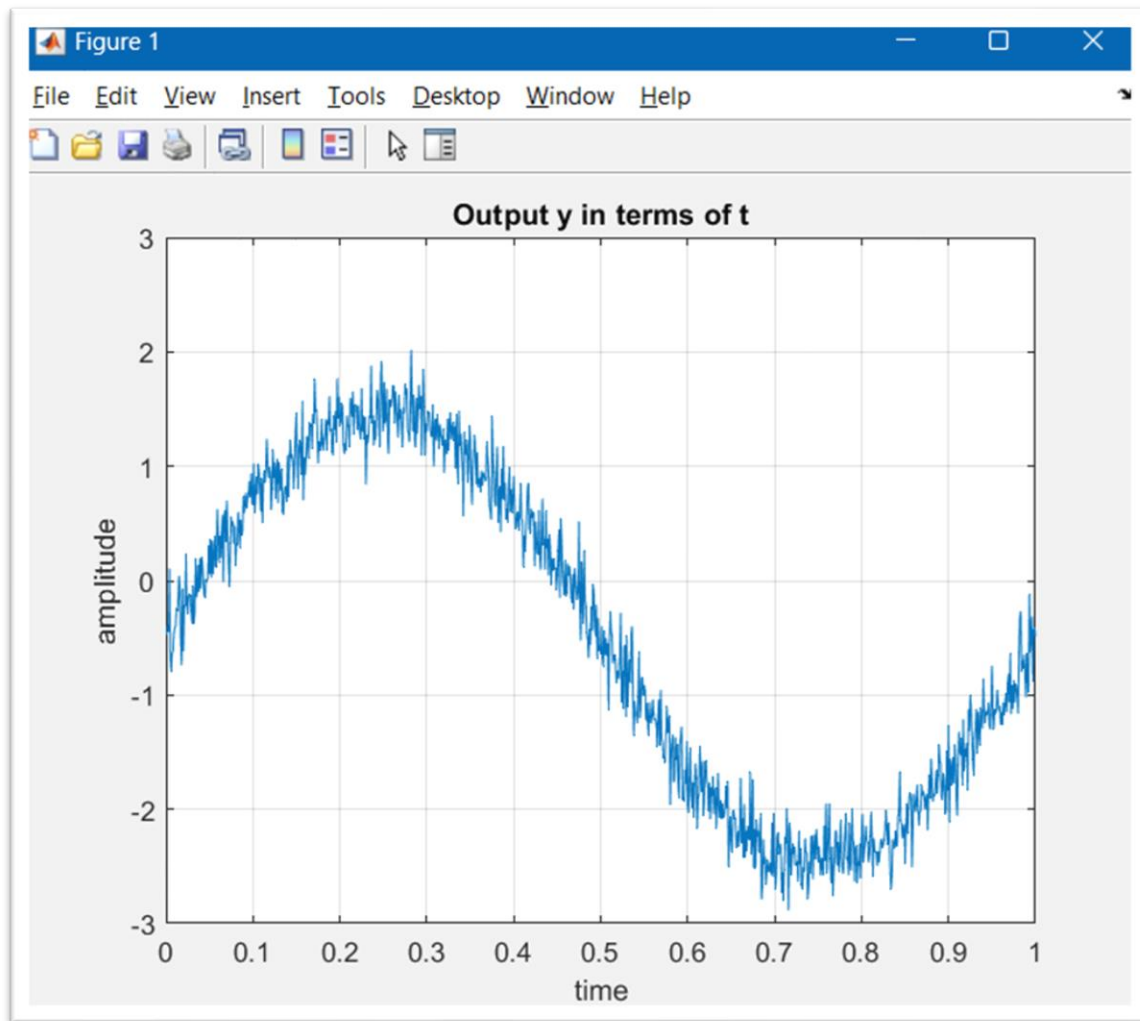
:2_1

```
1  
2     load('p2.mat');  
3     t = 0:0.001:1;  
4  
5     figure;  
6     plot(t, x);  
7     title('Input Signal x(t)');  
8     xlabel('Time (t)');  
9     ylabel('x(t)');  
0     grid on;  
1
```

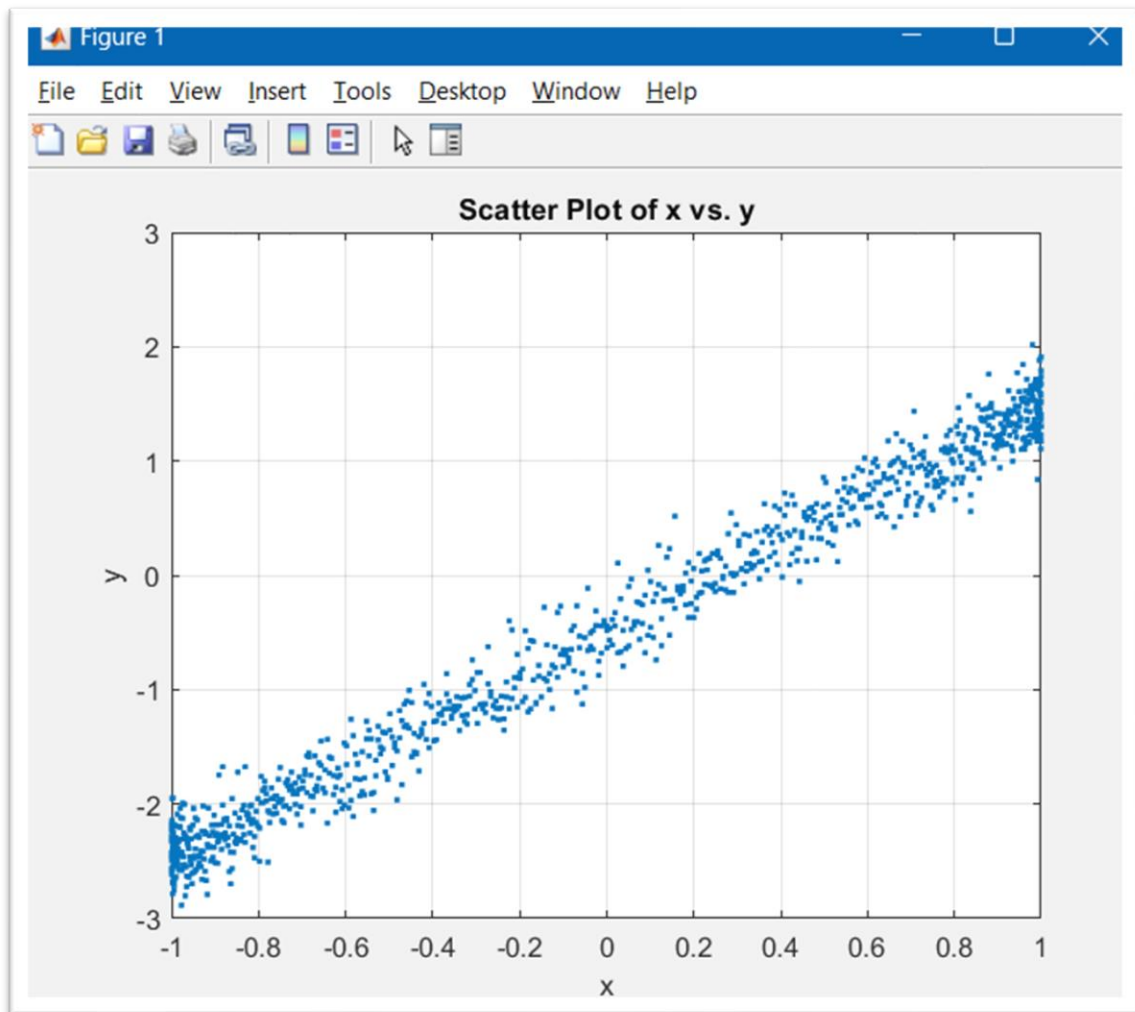


:2_2

```
1  
2     load('p2.mat');  
3  
4     t = 0:0.001:1;  
5     figure;  
6     plot(t, y);  
7     title('Output y in terms of t');  
8     xlabel('time');  
9     ylabel('amplitude');  
.0    grid on;
```



```
1  
2     load('p2.mat');  
3     figure;  
4     plot(x, y, '.');  
5     title('Scatter Plot of x vs. y');  
6     xlabel('x');  
7     ylabel('y');  
8     grid on;  
9
```



شیب این خط بیانگر نرخ تغییر x نسبت به y است که معمولاً به عنوان ضریب معادله در رگرسیون های خطی استفاده می شود شیب طبق رابطه داده شده در صورت سوال همان α است .

عرض از مبدا این خط بیانگر تغییرپذیری یا گسترش مقادیر y را در اطراف خط تشکیل شده منعکس می کند، که نشان دهنده سطح نویز یا عدم قطعیت در رابطه است. طبق رابطه داده شده در صورت سوال عرض از مبدا همان بتا است .

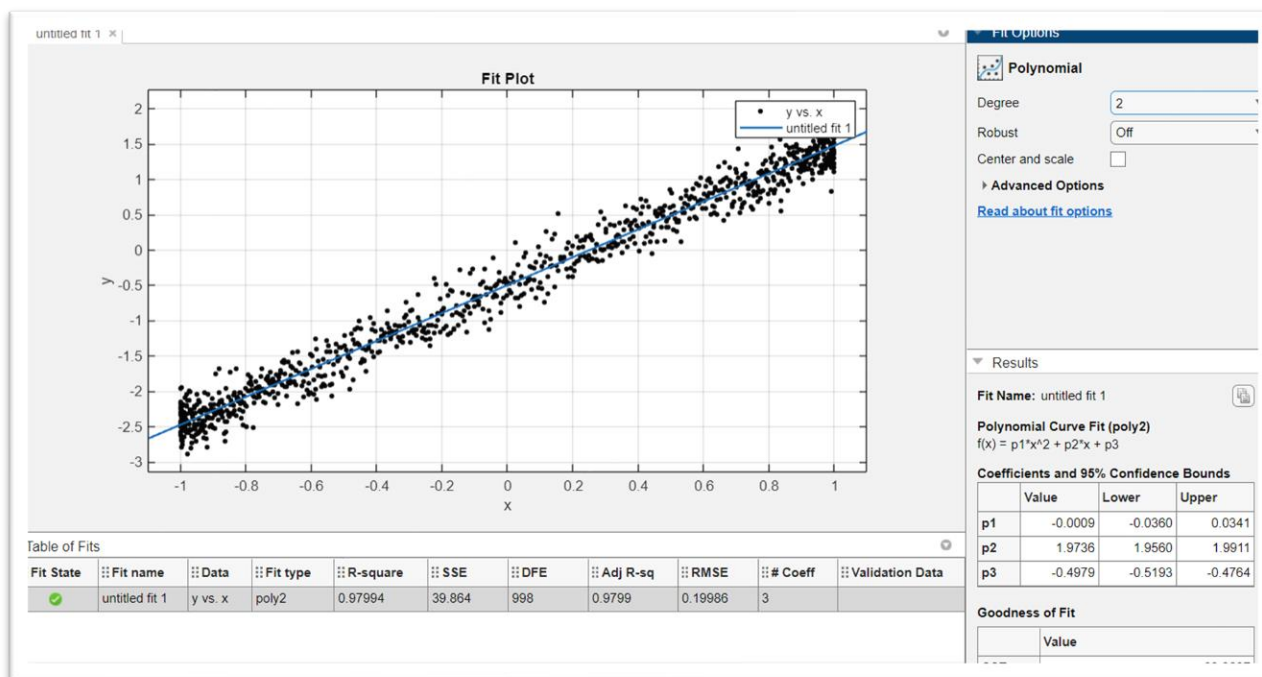
۲_۴:

```
1      % generate data without noise
2      x = linspace(0, 1, 100)';
3      a_true = 2;
4      b_true = 1;
5      y_true = a_true * x + b_true;
6
7      % add noise
8      noise = 0.1 * randn(size(x));
9      y_noisy = y_true + noise;
10
11     % estimate parameters
12     [a_est, b_est] = estimate_ab(x, y_noisy);
13
14     disp(['True a: ', num2str(a_true)]);
15     disp(['Estimated a: ', num2str(a_est)]);
16     disp(['True b: ', num2str(b_true)]);
17     disp(['Estimated b: ', num2str(b_est)]);
18
19
20     function [a, b] = estimate_ab(x, y)
21         n = length(x);
22         A = [x, ones(n, 1)];
23         params = A \ y;
24         a = params(1);
25         b = params(2);
26     end
27
```

```
>> p2_4
True a: 2
Estimated a: 1.9434
True b: 1
Estimated b: 1.0355
>>
```

با توجه به خواسته سوال مبنی بر آزمون مقدار در نظر گرفته شده و مقدار واقعی از مقایسه این دو مشابهت آنها بدست می آید ؛ با سطح نویز کم، تخمین ها باید به طور منطقی نزدیک به مقادیر واقعی باشند. اگر سطح نویز بالا باشد، تخمین ها ممکن است تنوع بیشتری داشته باشند.

2_5:



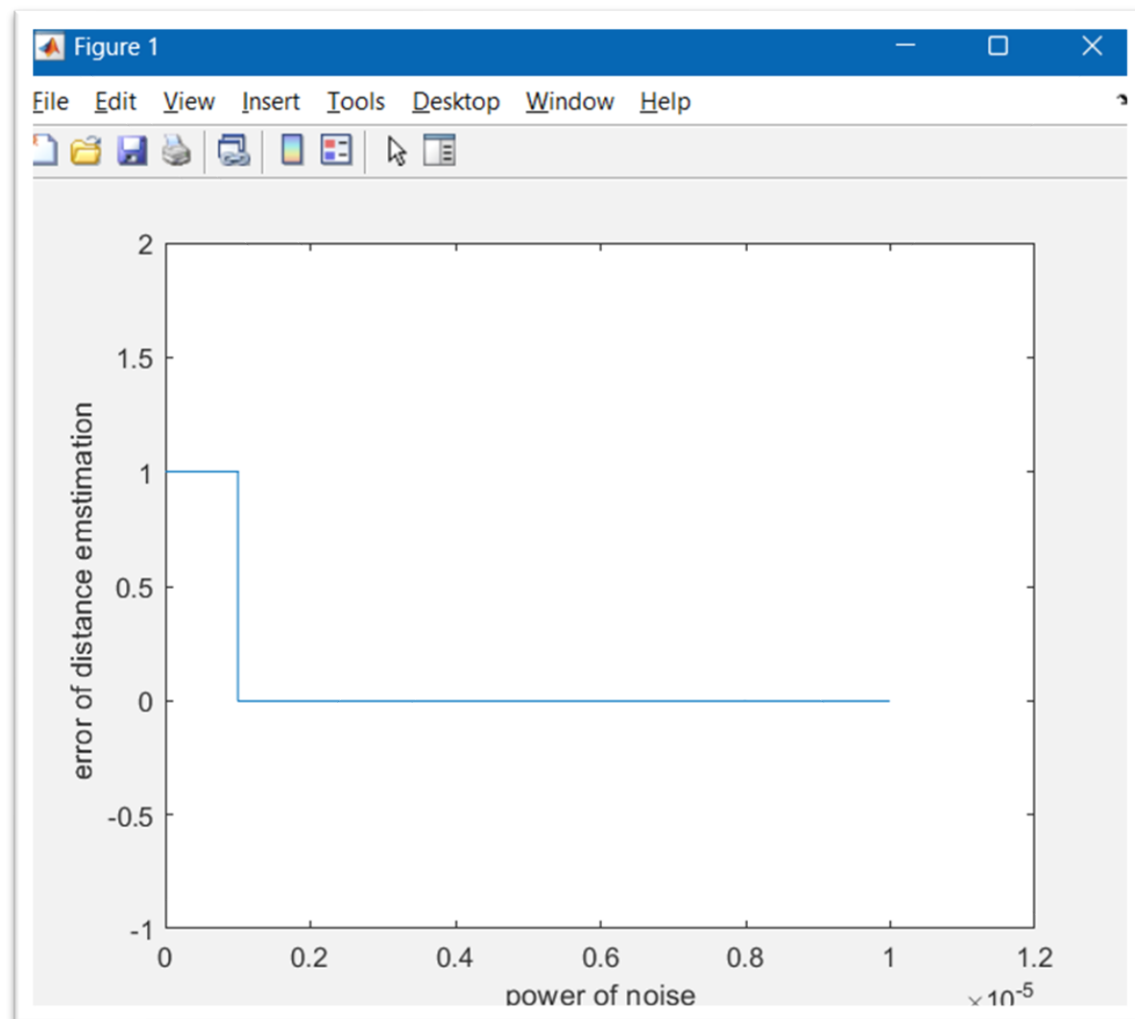
در شکل فوق خط آبی curve به دست آمده برای داده ها است (عملا به مانند مدل سازی رگرسیونی) و جدول سمت راست شیب و عرض از مبدا را مشخص کرده است .

١_٣:

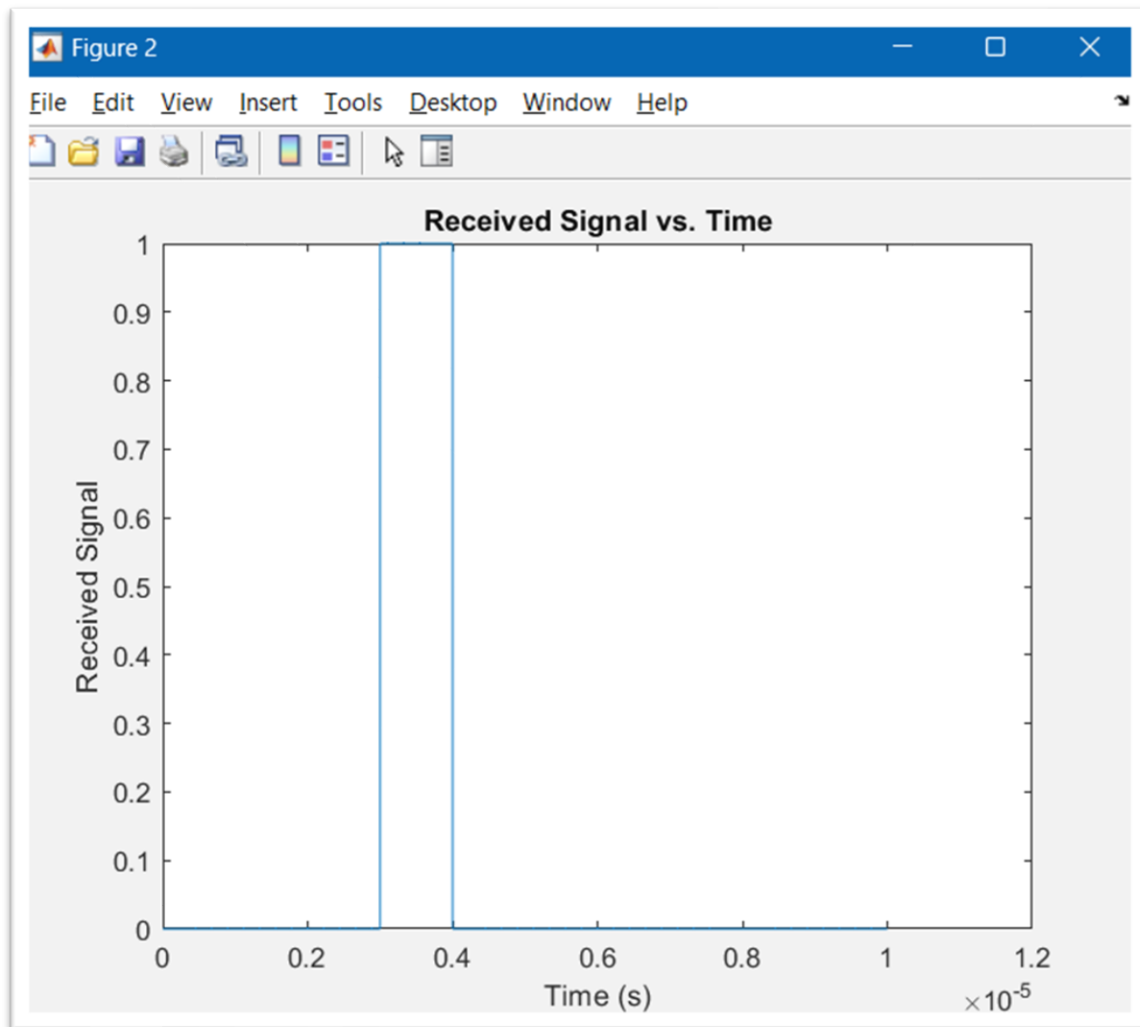
```

1   ts = 1e-9;
2   T = 1e-5;
3   tau = 1e-6;
4
5   t=0:ts:T;
6   tlen=length(t);
7   sent=zeros(1,tlen);
8   sent(1:round(tau/ts))=1;
9
10
11  plot(t,sent);
12  xlabel("power of noise");
13  ylabel("error of distance emstimation ");
14  ylim([-1,2]);
15  hold on;

```



```
1  ts = 1e-9;  
2  T = 1e-5;  
3  tau = 1e-6;  
4  
5  t=0:ts:T;  
6  tlen=length(t);  
7  sent=zeros(1,tlen);  
8  sent(1:round(tau/ts))=1;  
9  
10 R = 450;  
11 a = 0.5;  
12 c = 3e8;  
13  
14 td = 2 * R / c;  
15  
16 received = circshift(sent, round(td/ts));  
17  
18 figure;  
19 plot(t, received);  
20 xlabel('Time (s)');  
21 ylabel('Received Signal');  
22 title('Received Signal vs. Time');
```



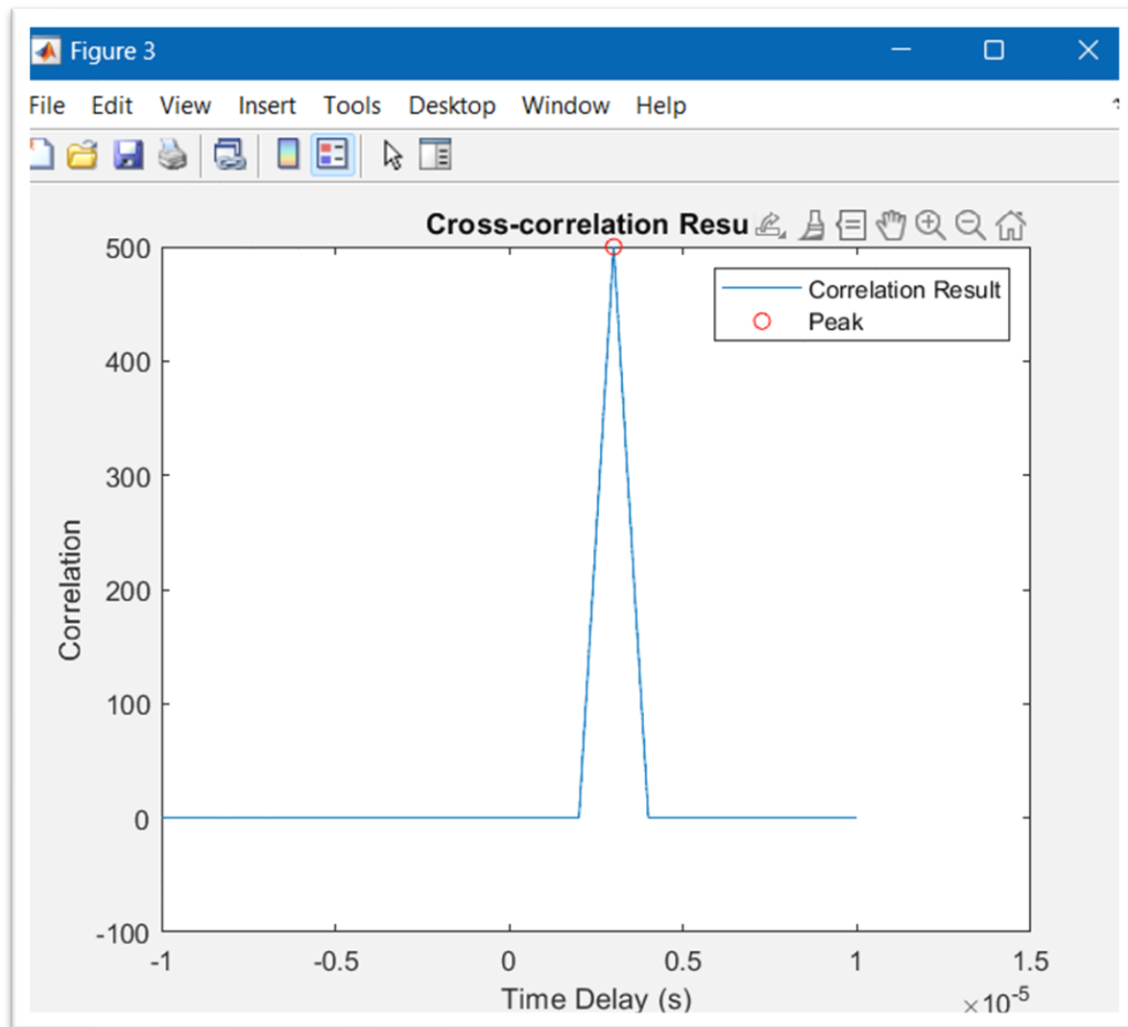
همانطور که در شکل مشخص است در حدود ۰.۳ نمودار بدین شکل است که بیانگر زمان تاخیر است ثانیه $\times 10^{-5}$

۳_۳:

```

1
2     ts = 1e-9;
3     T = 1e-5;
4     tau = 1e-6;
5
6     t = 0:ts:T;
7     tlen = length(t);
8     sent = zeros(1,tlen);
9     sent(1:round(tau/ts)) = 1;
10
11    R = 450;
12    a = 0.5;
13    c = 3e8;
14
15    td = 2 * R / c;
16
17    received = circshift(0.5*sent, round(td/ts));
18
19
20    correlation_result = xcorr(received, sent);
21
22
23    correlation_t = -T:ts:T;
24    |
25    figure;
26
27    plot(correlation_t, correlation_result);
28    xlabel('Time Delay (s)');
29    ylabel('Correlation');
30    title('Cross-correlation Result');
31    hold on;
32
33
34    [max_corr, max_idx] = max(correlation_result);
35    plot(correlation_t(max_idx), max_corr, 'ro');
36    legend('Correlation Result', 'Peak');
37
38
39    time_delay = correlation_t(max_idx);
40    estimated_R = time_delay * c / 2;
41
42    disp(['Estimated Distance (R): ', num2str(estimated_R), ' meters']);
43

```



ضرب داخلی سیگنال ارسالی و دریافتی را محاسبه کرده و زمان روی یکدیگر مرج شدن این دو را پیدا می کنیم که آن زمان دقیقاً همان زمان تاخیر است که به کمک آن می توان فاصله را محاسبه نمود. و با توجه به شکل حدود چهارصد و خورده ای فاصله مد نظر می باشد.

4_1:

```
[x, TS] = audioread( 'sner.wav' );
x
fs
```

```
Command Window

-0.0022    -0.0022
-0.0021    -0.0021
-0.0022    -0.0022
-0.0020    -0.0020
-0.0021    -0.0021
-0.0020    -0.0020
-0.0020    -0.0020
-0.0020    -0.0020
-0.0018    -0.0018
-0.0019    -0.0019

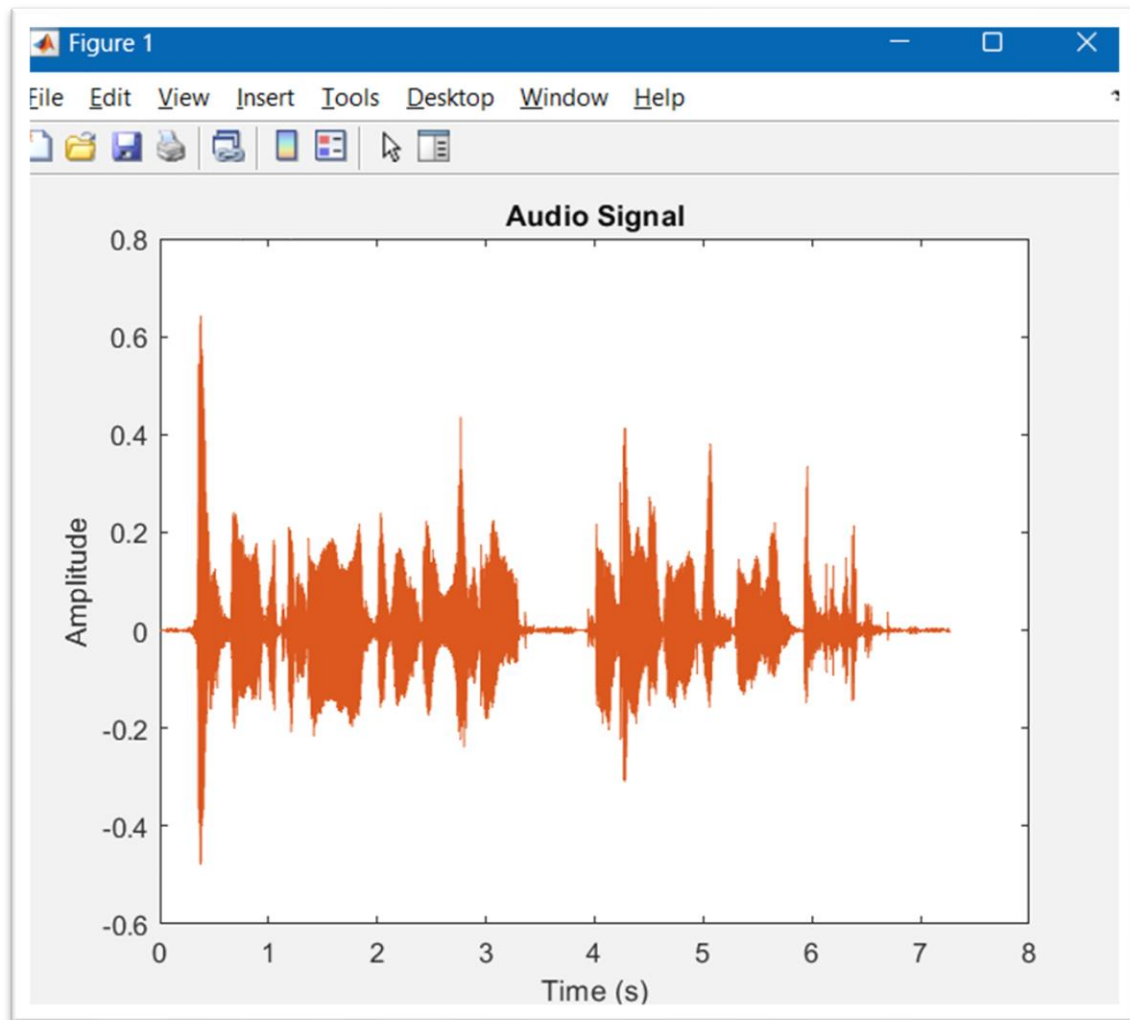
fs =

fx      44100
```

فرکانس عددی ۴۴۱۰۰ هرتز است که با کمک این دستور فرکانس نمونه برداری بدست آمد.

4_2:

```
1
2     [x, fs] = audioread('Sher.wav');
3
4     duration = length(x) / fs;
5
6     t = linspace(0, duration, length(x));
7     figure;
8     plot(t, x);
9     xlabel('Time (s)');
10    ylabel('Amplitude');
11    title('Audio Signal');
12
13    sound(x, fs);|
14    audiowrite('x.wav', x, fs);
15
```



4_3:


```

1
2
3     [x,fs]= audioread('Sher.wav');
4     speed = 2;
5     p4(x, speed,fs);
6
7
8     function y = p4(x, speed,fs)
9         if speed == 2
10            y = x(1:2:end-1, :);
11        elseif speed == 0.5
12            y = zeros(2*length(x)-1, size(x, 2));
13            y(1:2:end, :) = x;
14            y(2:2:end-1, :) = (x(1:end-1, :) + x(2:end, :)) / 2;
15        else
16            error('Invalid speed. Please use 0.5 or 2.');

```

:۴_۴

```

1     [x, fs] = audioread('Sher.wav');
2     speed = 1.9;
3     p4(x, speed, fs);
4
5     function y = p4(x, speed, fs)
6         if speed >= 0.5 && speed <= 2 && mod(speed, 0.1) == 0
7             y = interp1(1:size(x, 1), x, linspace(1, size(x, 1), size(x, 1)*1/speed));
8             sound(y, fs);
9         else
10            error('wrong input use something between 0.5 and 2 & multiple of 0.1.');

```