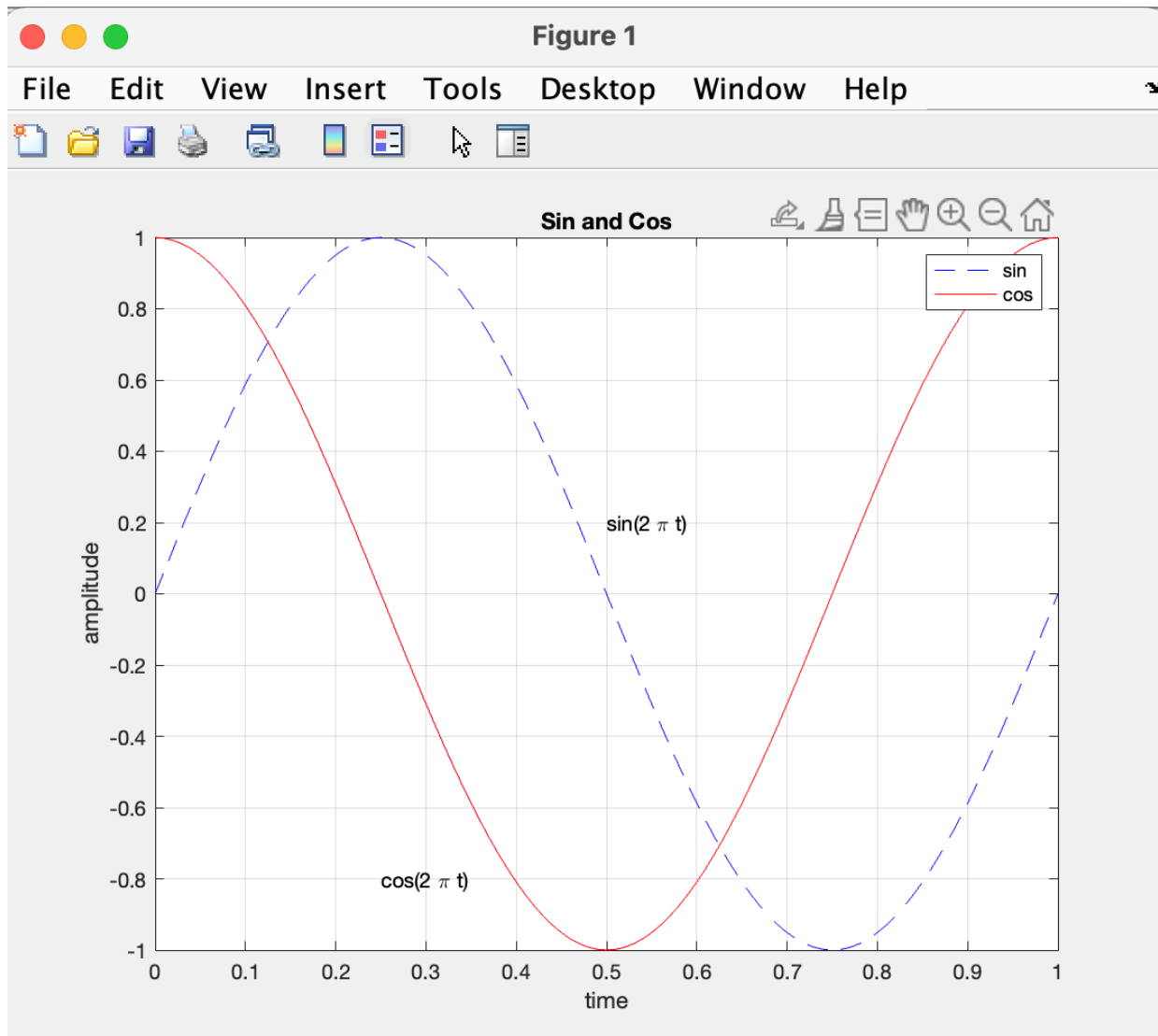


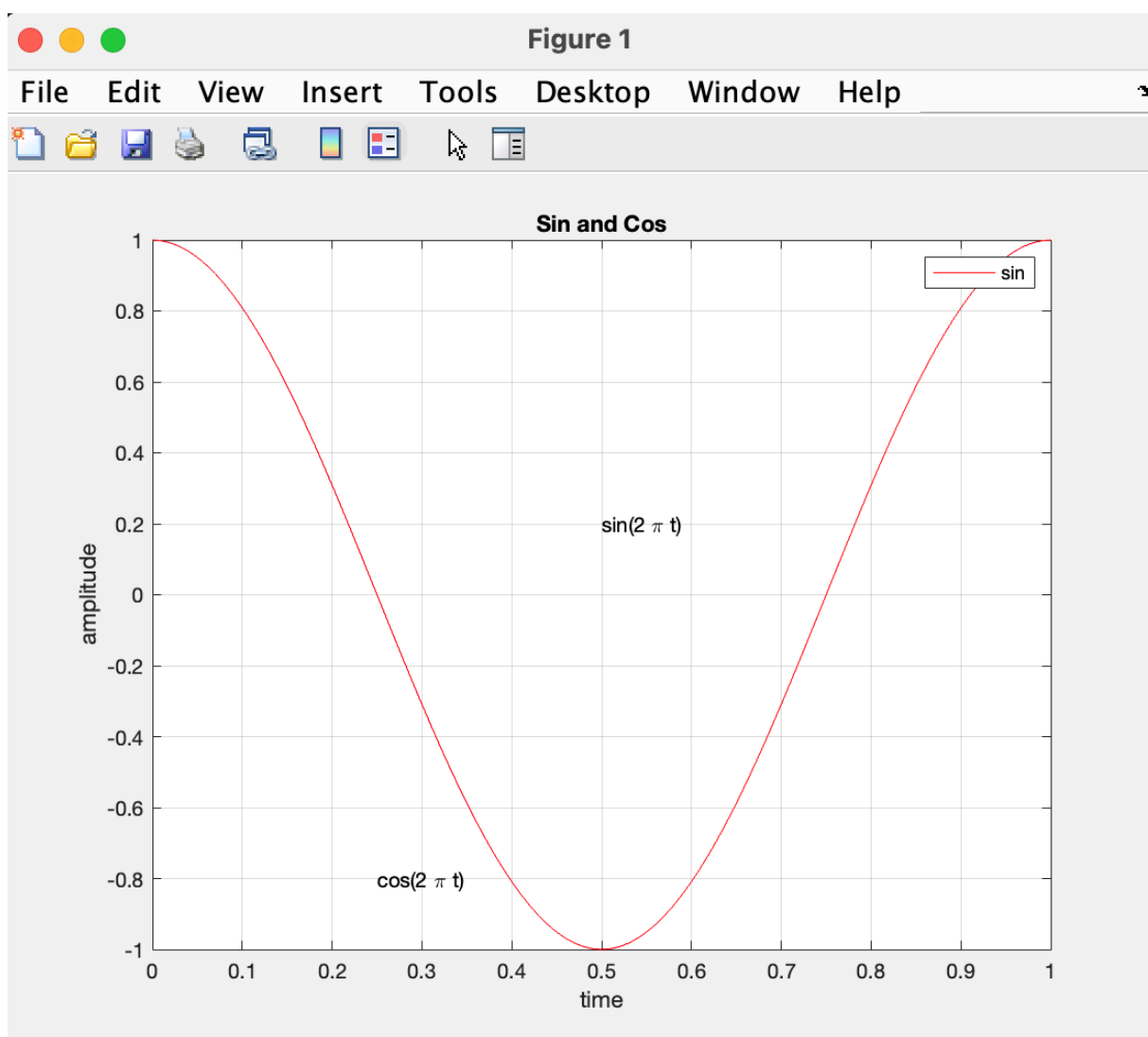
تمرین ۱-۱

شکل نهایی کد به این صورت است:



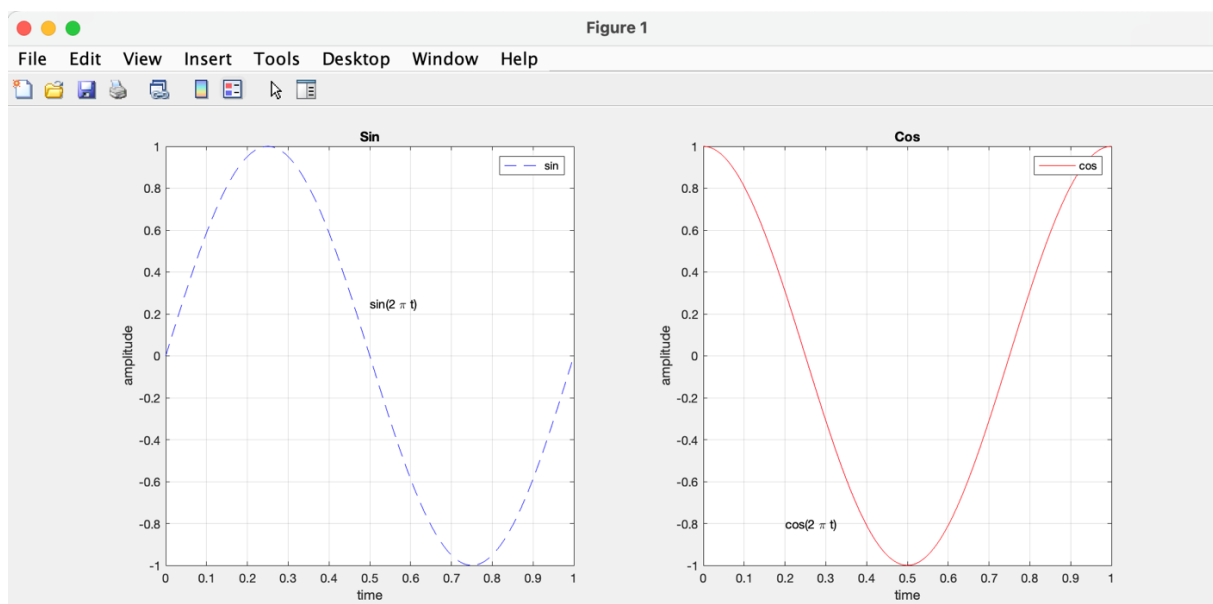
همانطور که دیده می‌شود نمودارهای سینوس و کسینوس به ترتیب به رنگ آبی و قرمز کشیده شده‌اند. دستور **legend** باعث می‌شود که در بالا سمت راست شکل نام هر نمودار همراه با شکل خطوط آن نمایش داده شود. با استفاده از دستور **text** نام هر نمودار در مختصات مناسب در کنار آن نوشته شده است. همچنین نام نمودار و نام هر محور نوشته شده است. دستور **grid on** شکل را بخش بندی می‌کند تا خواننده هر نمودار ساده‌تر شود. دستور **hold on** پس از یکی از نمودارها باعث می‌شود که نمودار بعدی هم روی همین نمودار کشیده شود. اگر از این دستور استفاده نکنیم، نمودار سینوس پاک شده و فقط نمودار کسینوس روی شکل نشان داده می‌شود. همچنین هنگام استفاده از دستور **legend** خطوط قرمز کسینوسی به اشتباه سینوسی نامیده می‌شوند چون تنها نمودار موجود در شکل کسینوسی است. همچنین متلب به ما **warning** در استفاده از **legend** می‌دهد.

شکل آذ به این صورت است:



تمرین ۱-۲)

شکل نهایی به این شکل است:



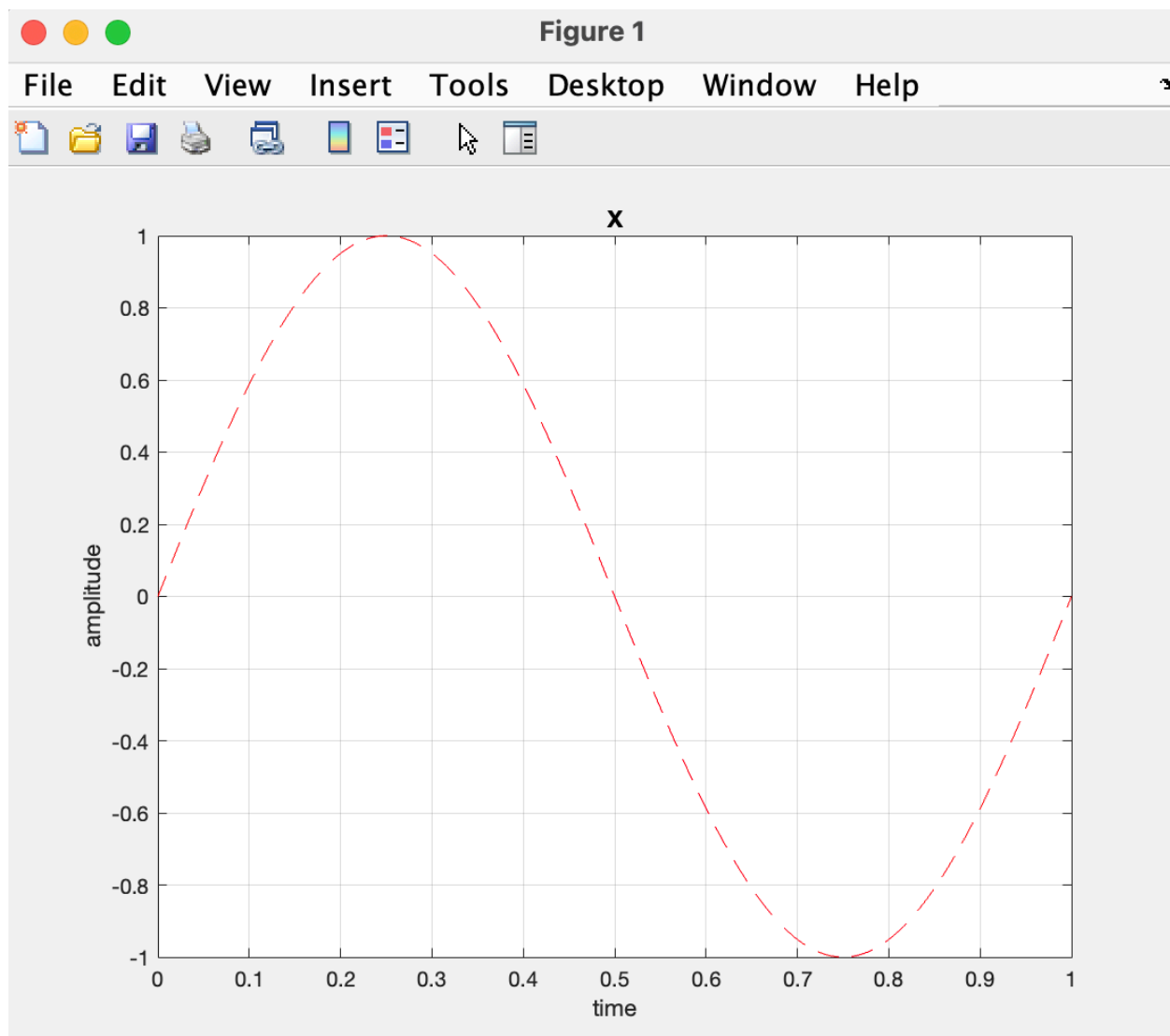
دستور subplot کمک می‌کند که دو نمودار را در دو شکل جدا کنار هم رسم کنیم. پس از استفاده از subplot و مشخص کردن position نموداری که می‌خواهیم رسم کنیم، می‌توانیم ویژگی‌های هر نمودار را به آن اضافه کنیم.

اسکرپتی که منجر به این شکل می‌شود:

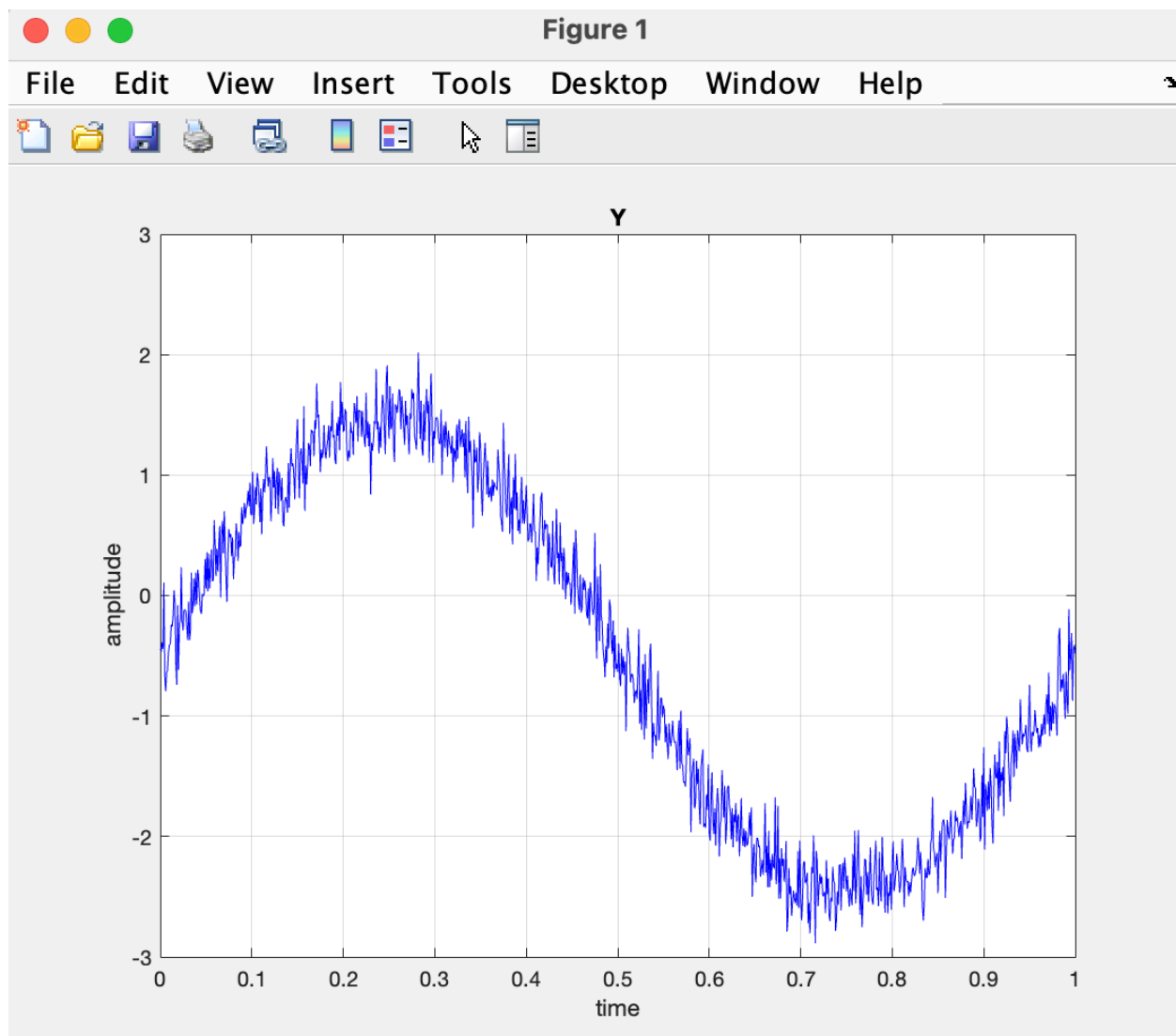
```
1 t = 0:0.01:1;
2 z1 = sin(2*pi*t);
3 z2 = cos(2*pi*t);
4
5 figure;
6 subplot(1, 2, 1)
7 plot(t, z1, '--b')
8 s1 = 'sin(2 \pi t)';
9 text(0.5, 0.25, s1);
10 title('Sin')
11 legend('sin')
12 xlabel('time')
13 ylabel('amplitude')
14 grid on
15
16
17 subplot(1, 2, 2)
18 plot(t, z2, 'r')
19 y0 = [];
20 s2 = 'cos(2 \pi t)';
21 text(0.2, -0.8, s2);
22 title('Cos')
23 legend('cos')
24 xlabel('time')
25 ylabel('amplitude')
26 grid on
27
```

تمرین ۱-۲)

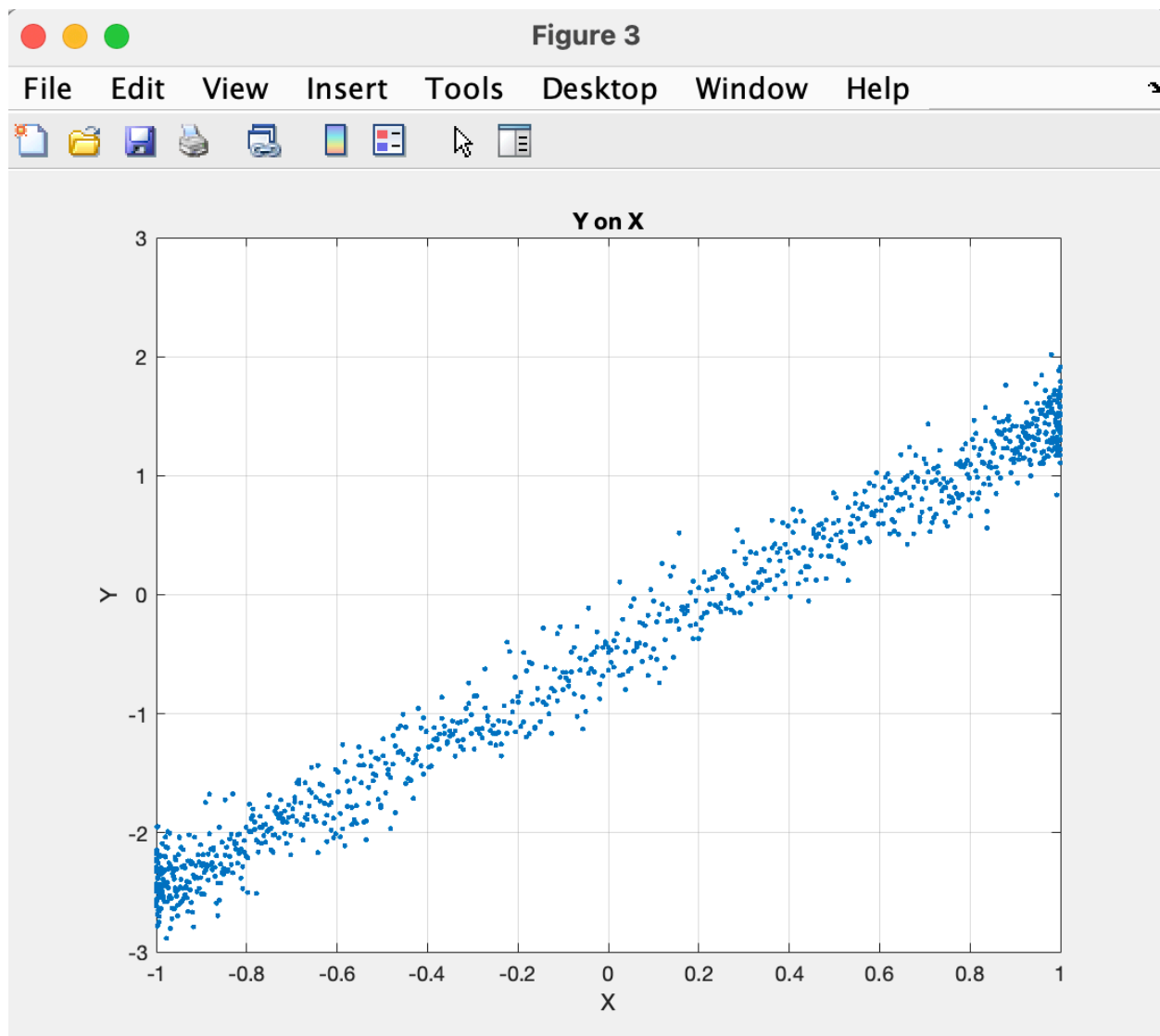
شکل نمودار X بر حسب t به این صورت است:



تمرین ۲-۲) شکل نمودار y بر حسب t به این صورت است که شامل نویز است:



تمرین ۲-۳) شکل نمودار Y بر حسب X به این صورت است:



رابطه Y بر حسب X تقریباً خطی است و شیب این خط نشان‌دهنده میزان تغییرات Y نسبت به X است. به بیاز دیگر در فرمول $y(t) = \alpha x(t) + \beta$ ، شیب خط نشان‌دهنده پارامتر α است. همچنین عرض از مبدا این خط معاداً پارامتر β در این فرمول است.

تمرین ۲-۴)

برای مینیمم کردن تابع هزینه که برابر با $f(\alpha, \beta) = \sum (y(t) - \alpha x(t) - \beta)^2$ است باید مشتق جزئی این تابع را نسبت به α و β به دست آورده و آن دو را برابر با ۰ بگذاریم. با این کار دو معادله به دست می‌آید که دو مجهول α و β را از آنها به دست می‌آوریم.

فانکشن `p2_4.m` با استفاده از `syms` و تابع `solve` دو مجهول را محاسبه می‌کند.

برای چک کردن این که تابع به درستی کار می‌کند ابتدا تابعی به این صورت تولید کردیم:

```
v = 0:0.001:1;
w = 2.572 * v + 6.2419;
```

سپس V و W را به تابع داده تا ضرایب را محاسبه کند. ضرایب به این شکل محاسبه می‌شوند:

alpha: 2.5720
beta: 6.2419

ضرایب محاسبه شده دقیقاً برابر با ضرایب واقعی هستند.

این بار به W نویزی با ضریب 0.3 اضافه می‌کنیم و دوباره V و W را به تابع می‌دهیم. ضرایب به این شکل محاسبه می‌شوند:

alpha: 2.5625
beta: 6.2576

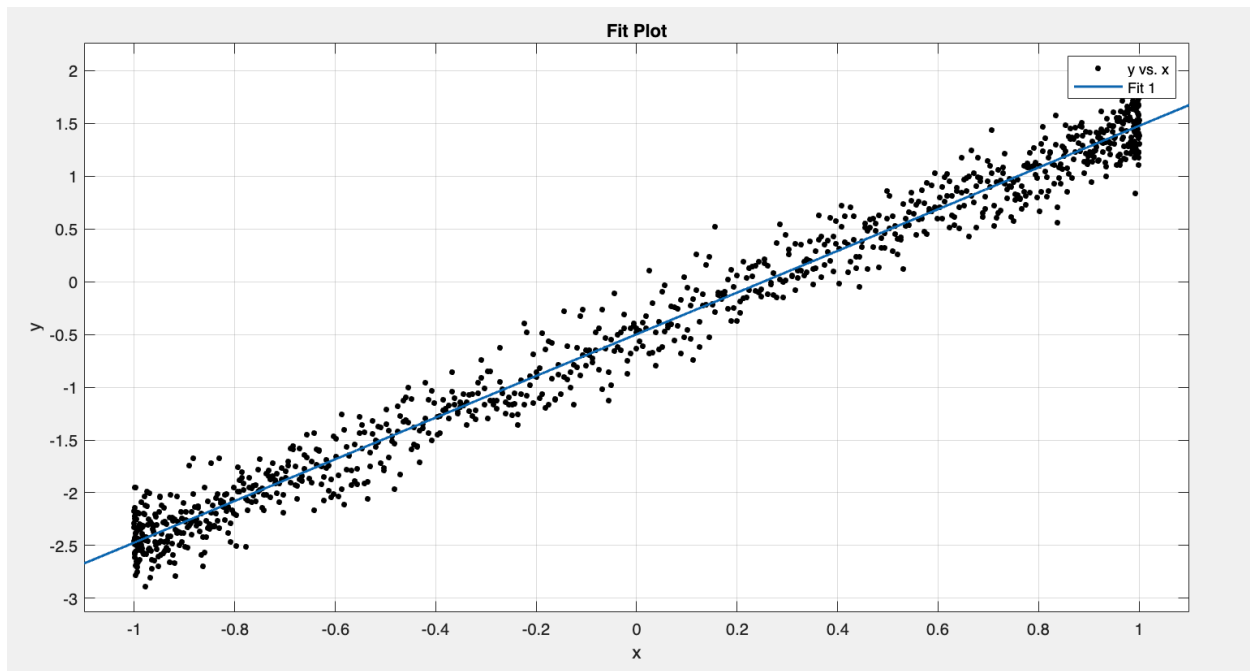
این ضرایب بسیار نزدیک به ضرایب واقعی هستند. در نتیجه تابع به درستی کار می‌کند.

حال برای به دست آوردن ضرایب مربوط به توابع X و Y این دو را به فانکشن می‌دهیم تا ضرایب را محاسبه کند. نتیجه به این صورت است:

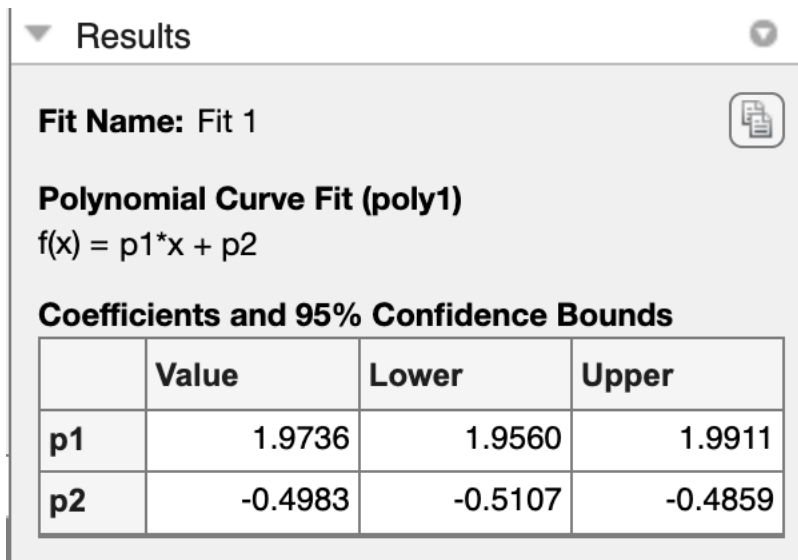
alpha: 1.9736
beta: -0.4983

تمرین ۲-۵)

شکل حاصل از تخمین اپلیکیشن به این صورت است:

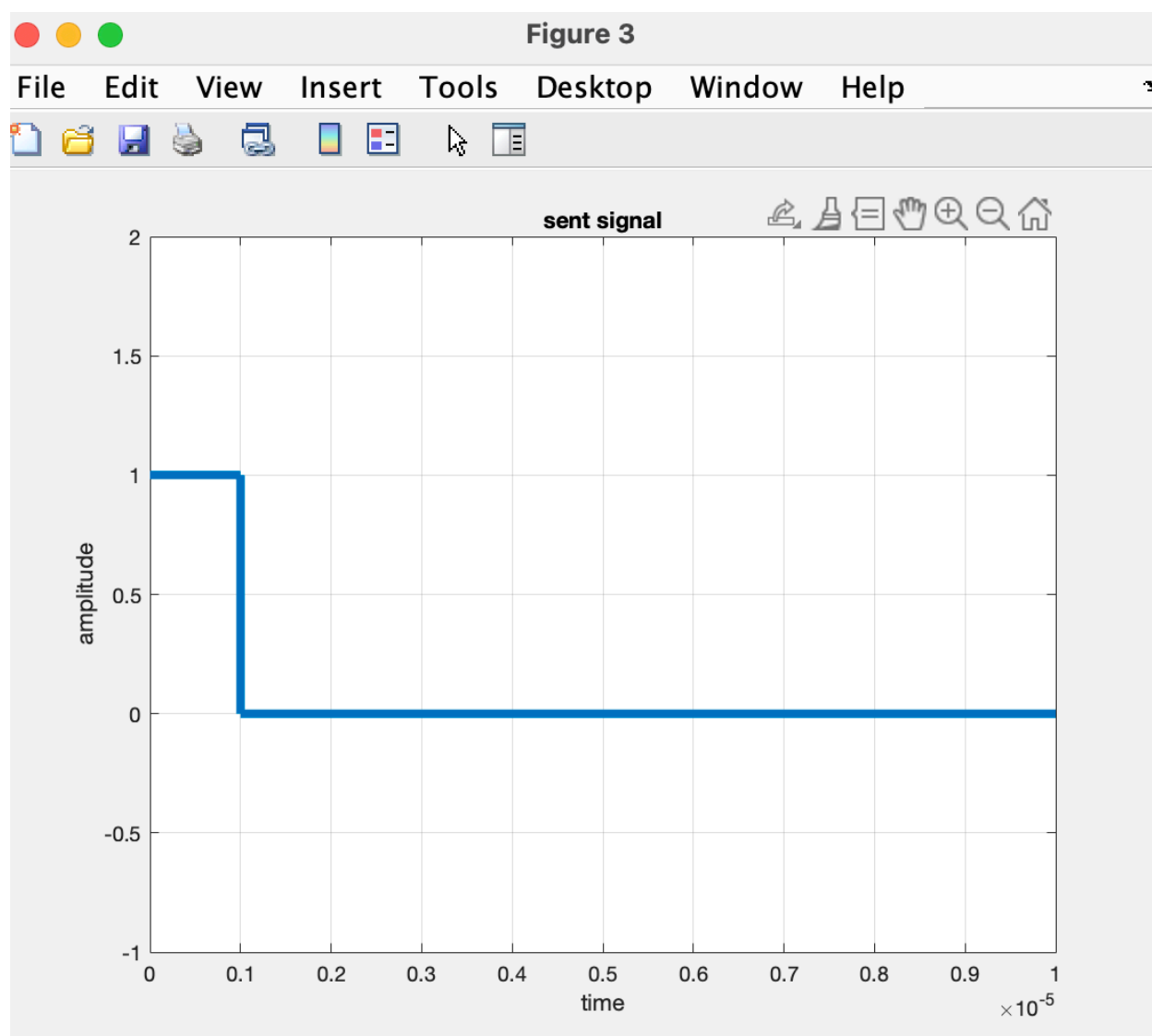


نتایجی که در تخمین به دست آمده در زیر آمده است که برابر با نتایج تخمین زده شده توسط فانکشن $p2_4.m$ است:



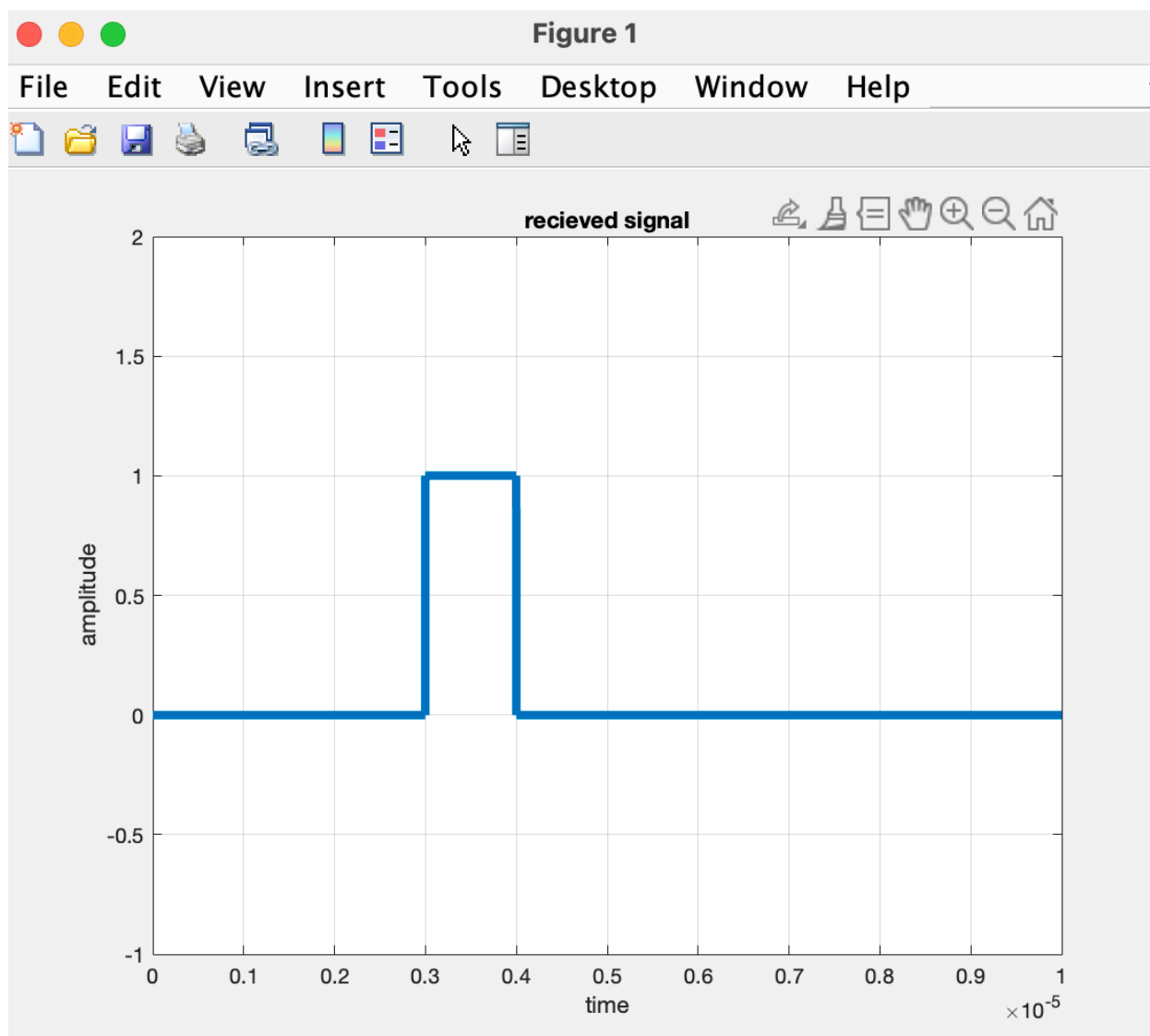
تمرین ۳-۱)

سیگنال ارسالی با فرضیات گفته شده به این شکل است:



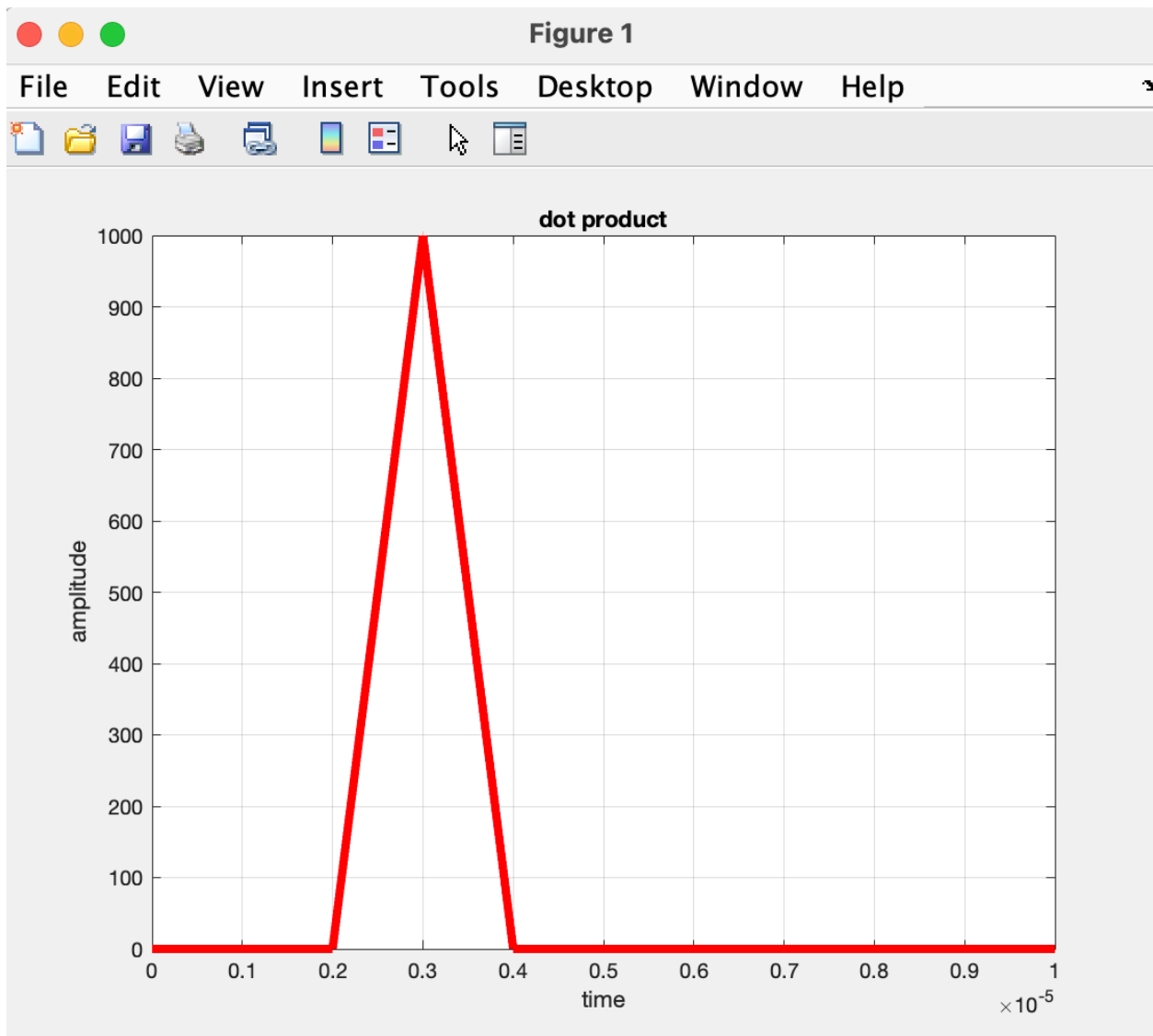
تمرین ۳-۲)

سیگنال دریافتی با فرض $R = 450$ معادله شکل زیر است:



تمرین ۳-۳

ابتدا برای به دست آوردن زمان بازگشت سیگنال اصلی، سیگنال تست متحرک در زمانی در نظر می گیریم و در هر لحظه و هر حالت سیگنال تستی، ضرب داخلی آن را با سیگنال بازگشتی محاسبه می کنیم. زمانی که این ضرب داخلی بیشینه شود یعنی سیگنال تست ما با سیگنال بازگشتی منطبق شده است. با این کار زمان بازگشت سیگنال اصلی به دست می آید. نمودار ضرب داخلی دو سیگنال بر حسب زمان به این صورت است:

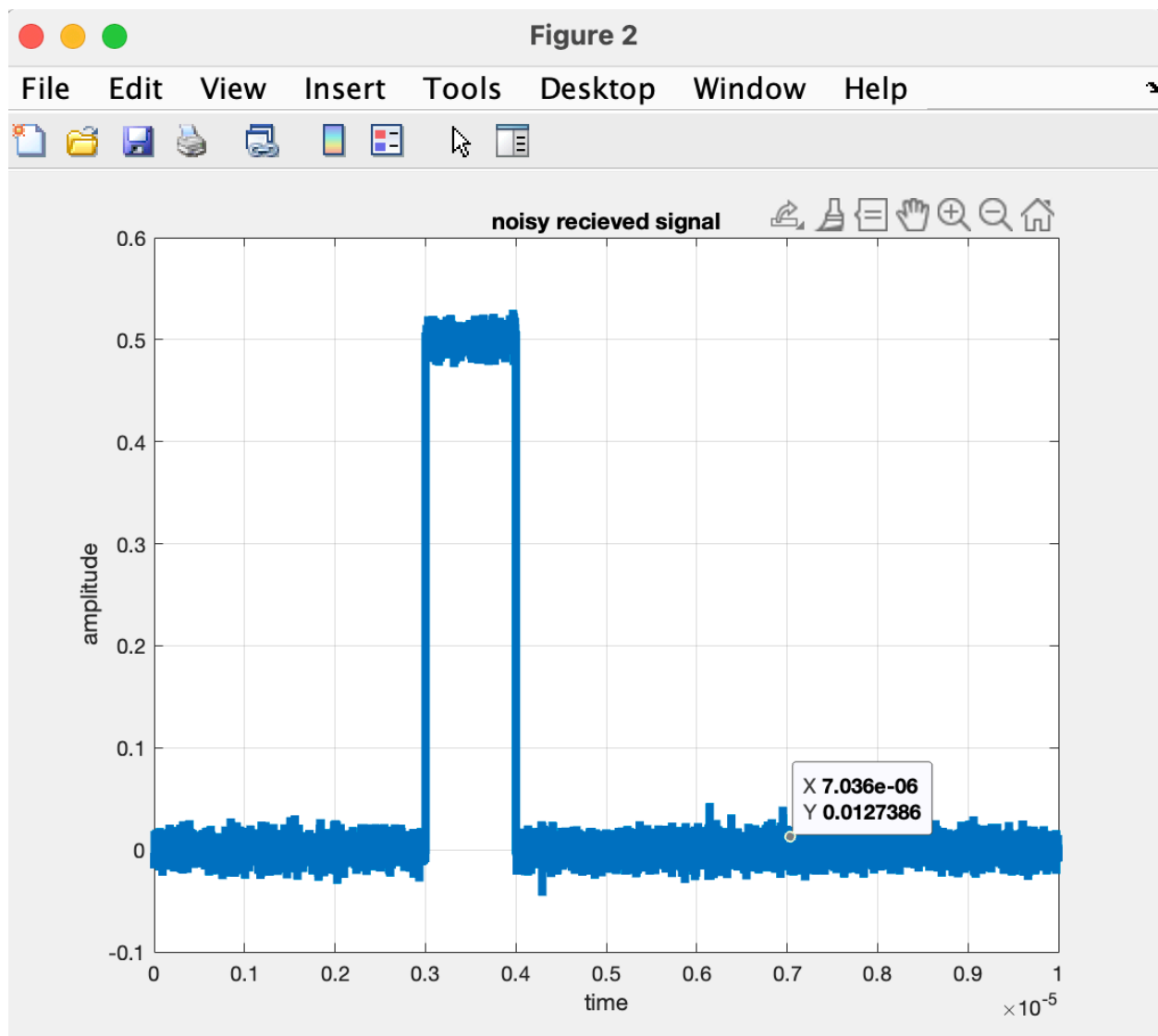


این نمودار نشان می‌دهد دو سیگنال در زمان $td = 3e-6$ با هم منطبق شده‌اند پس سیگنال اصلی هم در همین زمان بازگشته است.

با توجه به فرمول $R = td * c / 2$ می‌توانیم فاصله شی مورد نظر را محاسبه کنیم. این فاصله برابر با ۴۴۹.۸۳۸۶ محاسبه می‌شود تقریب بسیار خوبی از فاصله واقعی (۴۵۰) است.

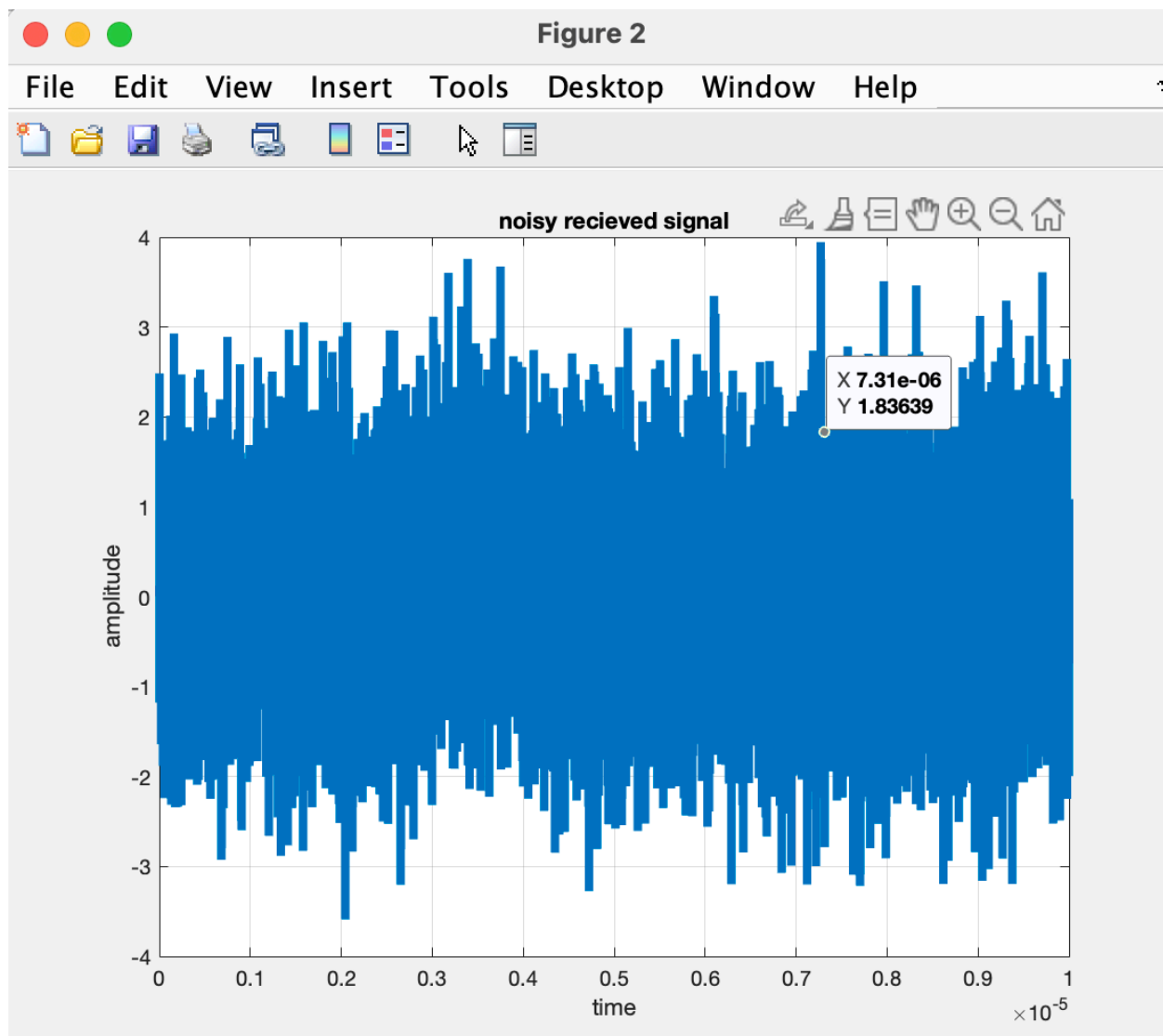
تمرین ۳-۴)

با اضافه کردن نویز به سیگنال دریافتی تشخیص زمان دریافت سیگنال بازگشتی سخت‌تر می‌شود. شکل زیر سیگنال دریافتی با نویز با قدرت ۰.۰۱ را نمایش می‌دهد:

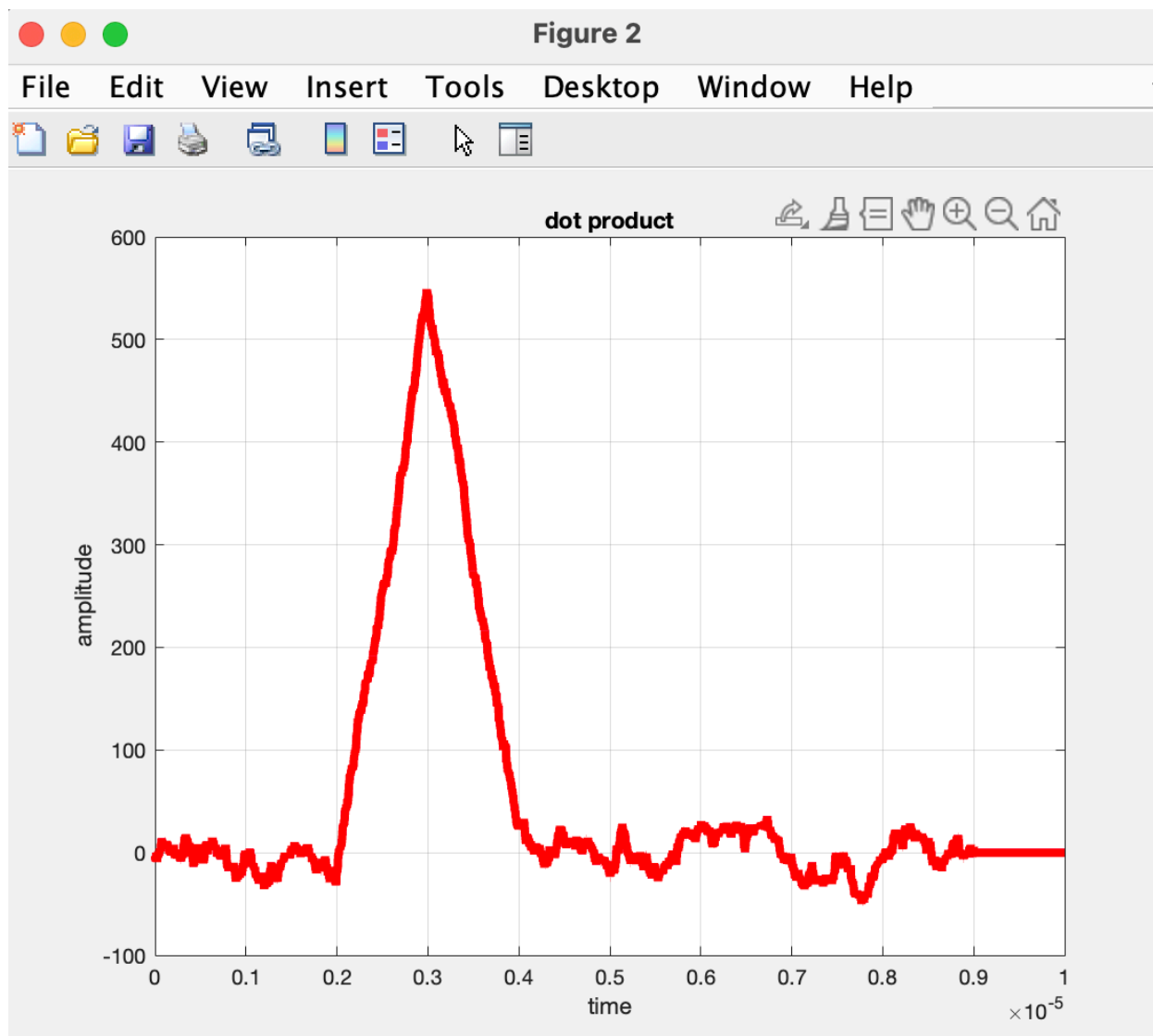


همانطور که معلوم است سیگنال دریافتی هنوز کاملاً مشخص است چون قدرت نویز بسیار کم است. تشخیص زمان بازگشت سیگنال و فاصله شی مورد نظر نیز ساده و کم خطاست.

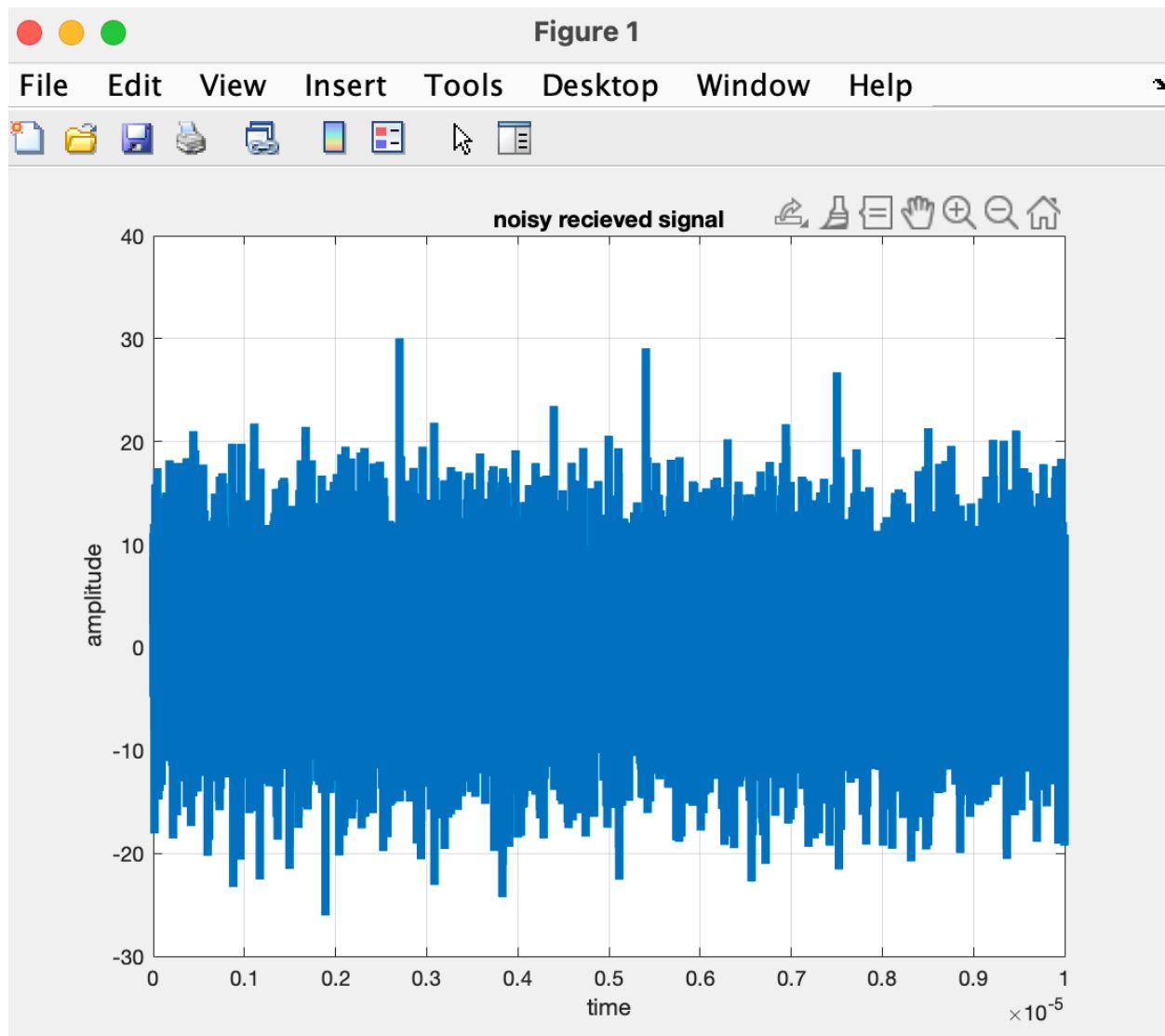
با افزایش قدرت نویز به ۱ شکل سیگنال دریافتی به این شکل خواهد بود:



در این شکل سیگنال دریافتی قابل تشخیص با چشم نیست اما با گرفتن correlation آن با موج تست و رسم dot product می‌توانیم به نتیجه خوبی از زمان بازگشت موج برسیم:

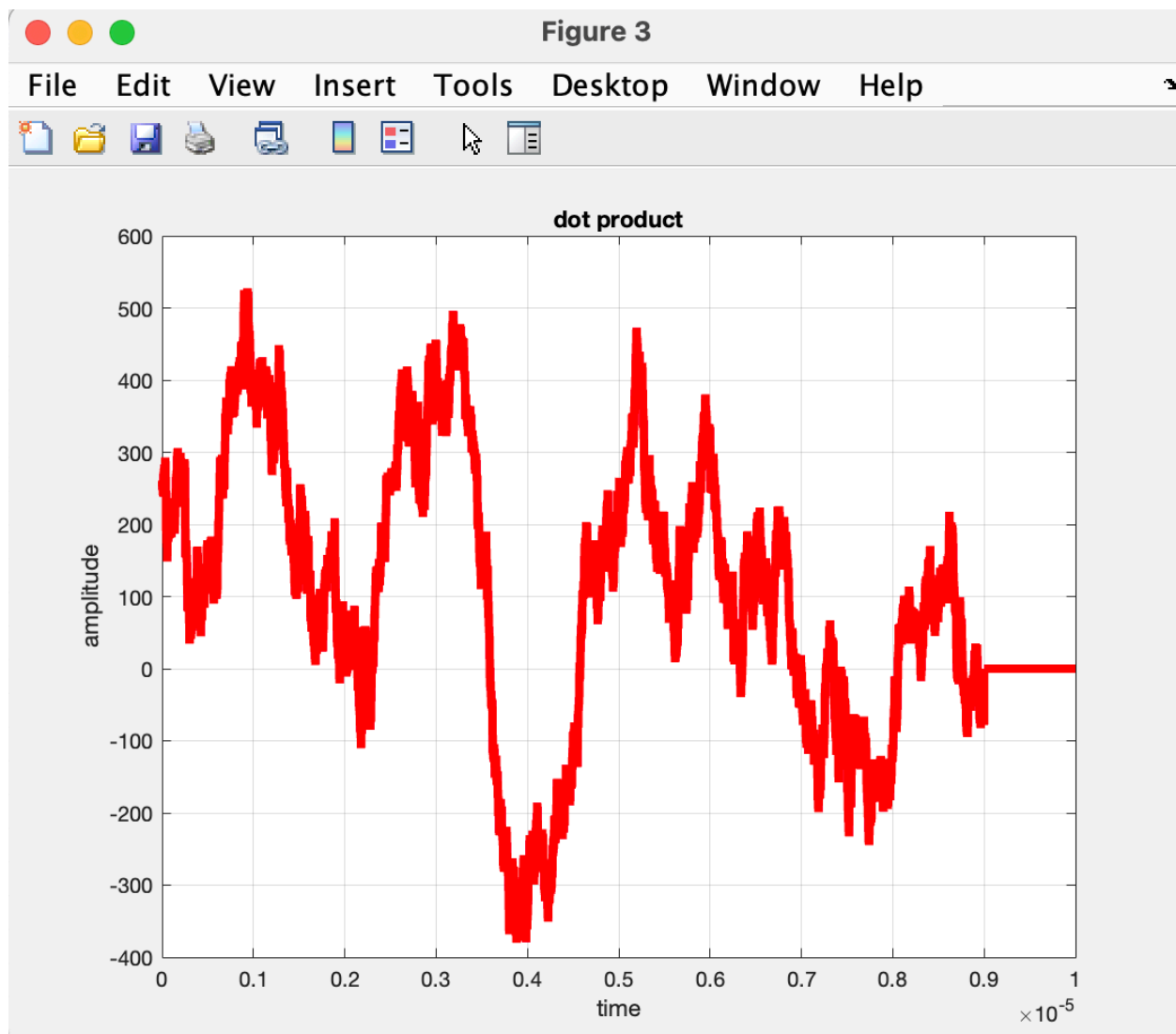


اما اگر قدرت نویز را به ۷ افزایش دهیم شکل سیگنال دریافتی به این شکل می شود:



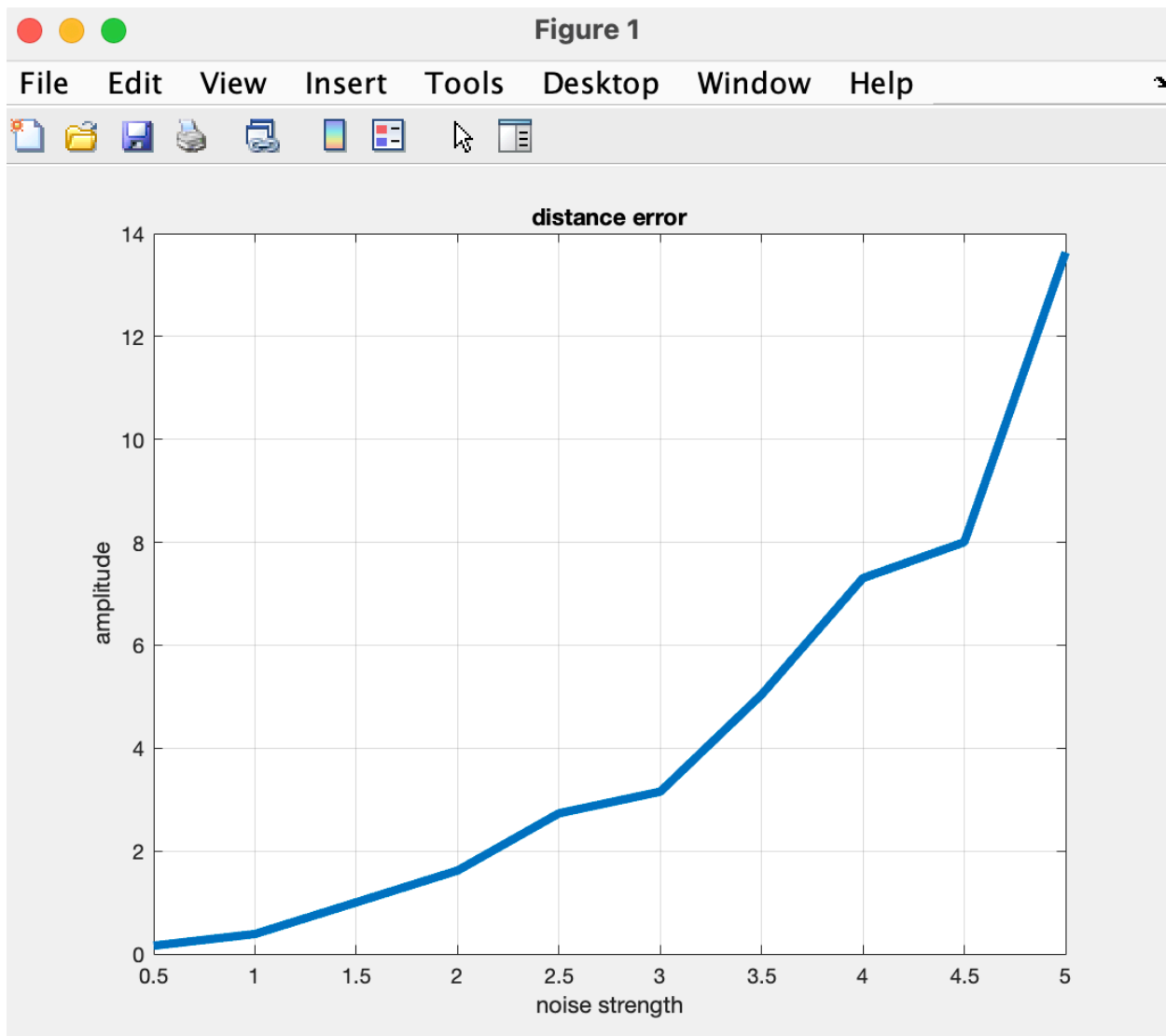
در این حالت دیگر سیگنال دریافتی قابل تشخیص نیست و تشخیص زمان بازگشت سیگنال و فاصله شی با خطای زیادی مواجه است.

شکل زیر dot product این موج بازگشتی با موج تست را نشان می دهد:



همانطور که مشخص است تشخیص زمان بازگشت موج دیگر ممکن نیست و خطای فاصله محاسبه شده بسیار بالا می‌رود.

شکل زیر میانگین خطای تشخیص فاصله شی بر حسب قدرت نویز را نمایش می‌دهد:



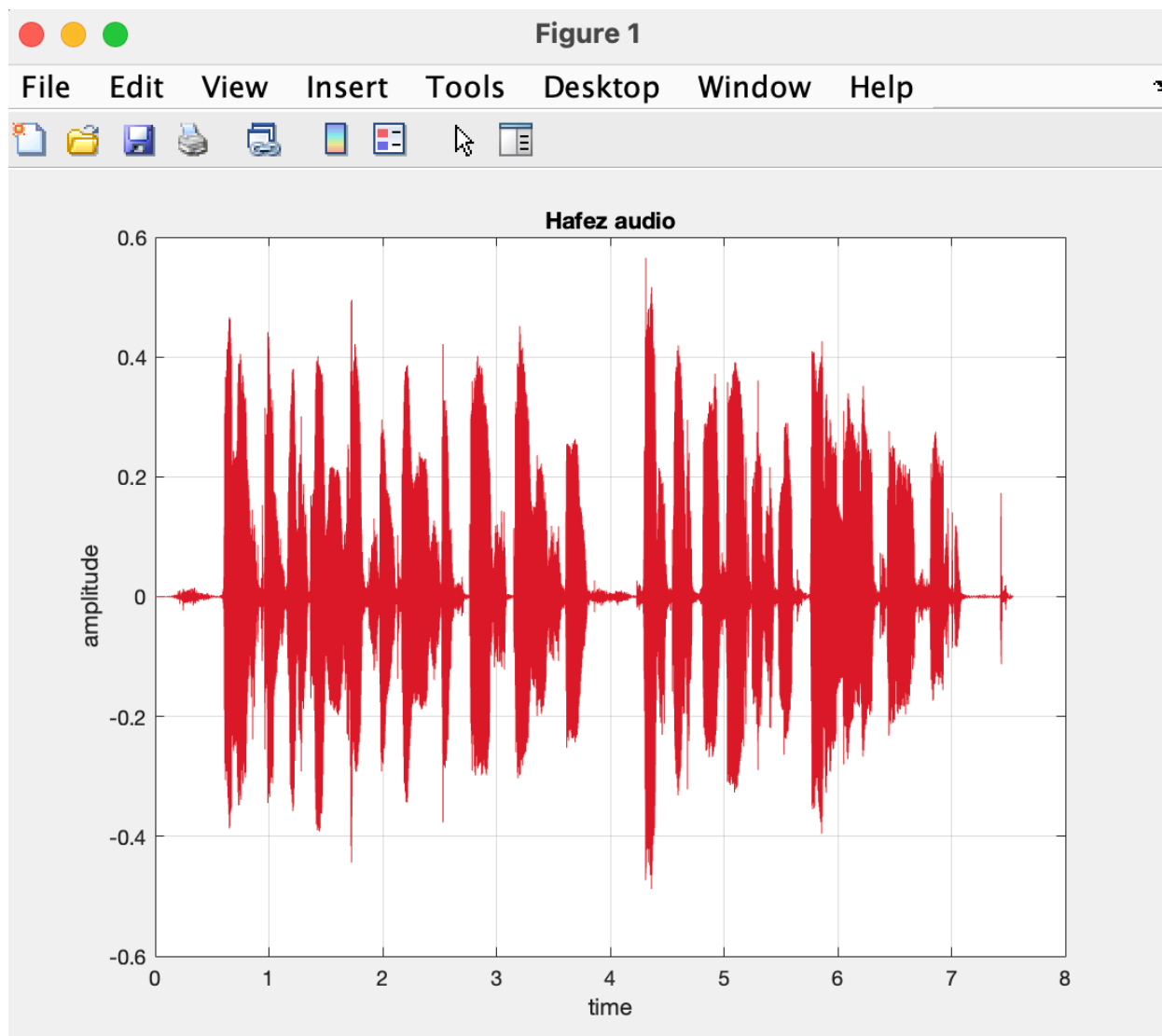
مشخص است که با افزایش قدرت نویز خطای فاصله به دست آمده هم بیشتر می شود به طوری که با قدرت نویز بیشتر از حدود ۳ خطای فاصله به دست آمده به بالای ۱۰ متر می رسد و می توان گفت فاصله به اشتباه تخمین زده شده است.

تمرین ۴-۱)

پس از استفاده از دستور `audioread` فرکانس نمونه برداری ۴۴۱۰۰ است که یعنی در هر ثانیه این تعداد نمونه از سیگنال پیوسته فایل `Hafez.wav` گرفته شده و نمودار گسسته آن در `X` کشیده شده است.

تمرین ۴-۲)

سیگنال صوتی رسم شده بر حسب ثانیه به این شکل است:



تمرین ۳-۴

تابع در صورتی که سرعت پخش ۲ باشد یکی درمیان نمونه های سیگنال x را دور می ریزد و باقی را در ارایه ای دیگر ریخته و با فرکانس f_s پخش می کند.

در صورتی که سرعت پخش ۰.۵ باشد در یک حلقه for همه نمونه های سیگنال x را به همراه حدسی از نمونه میانی بین هر دو نمونه مجاور (میانگین دو نمونه) در ارایه ای دیگر ریخته و سپس آن را با فرکانس f_s پخش می کند.

تمرین ۴-۴

این فانکشن در سرعت های کمتر از ۱ اگر سرعت را m در نظر بگیریم به طور میانگین از هر نمونه سیگنال به تعداد $1/m$ نمونه جدید می سازد. (میانگین آن نمونه با نمونه بعدی) سپس نمونه های تولید شده را در ارایه دیگری می ریزد و آن را با فرکانس نمونه برداری f_s پخش می کند.

در سرعت های بیشتر از ۱ اگر سرعت را m در نظر بگیریم به طور میانگین از هر m تا سیگنال فقط یک نمونه از آنها را استخراج می کند و در آرایه جدیدی می ریزد و آن را به فرکانس نمونه برداری f_s پخش می کند.