

به نام خدا  
سیگنال ها و سیستم ها  
تمرین کامپیوتری اول  
مهلت تحویل: سه شنبه ۱۷ مهر ساعت ۱۷:۰۰

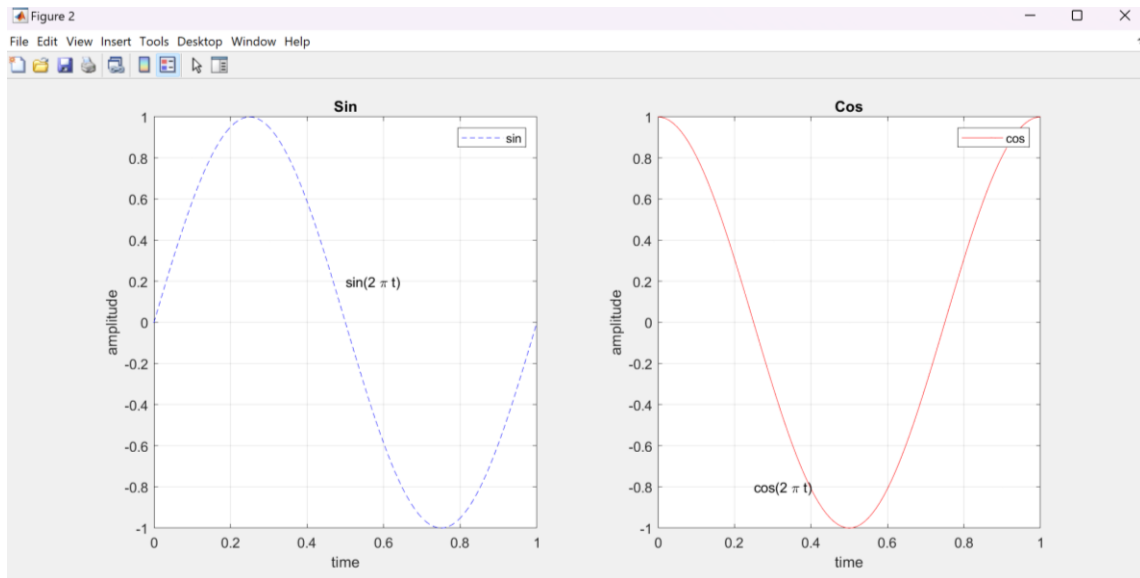
---

بخش اول:

تمرین ۱-۱) یک script جدید با نام p1\_1.m ایجاد کنید و کد زیر را در آن وارد کنید. کد را خط به خط Run کنید تا دقیقاً متوجه شوید هر خط چه کاری انجام می دهد. شکل نهایی که کد تولید می کند را در گزارش بیاورید. اگر دستور hold on را استفاده نکنید و یا به عبارت دیگر خط 7 را حذف کنید چه اتفاقی می افتد؟

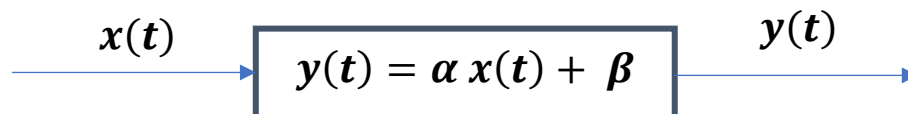
```
1 t=0:0.01:1;
2 z1=sin(2*pi*t);
3 z2=cos(2*pi*t);
4
5 figure;
6 plot(t,z1,'--b')
7 hold on
8 plot(t,z2,'r')
9
10 x0=[0.5;0.25];
11 y0=[0.2;-0.8];
12 s=['sin(2 \pi t)'; 'cos(2 \pi t)'];
13 text(x0, y0, s); % Add comment at (x0,y0)
14
15 title('Sin and Cos'); % Title
16 legend('sin','cos') % Add legend
17 xlabel('time') % the name of X-axis
18 ylabel('amplitude') % the name of Y-axis
19 grid on % Add grid
```

**تمرین ۱-۲)** در تمرین ۱-۱ اگر بخواهیم نمودار  $\sin$  و  $\cos$  را به جای این که روی هم رسم کنیم، مطابق شکل زیر در دو شکل کنار هم رسم کنیم، می توانیم از دستور subplot استفاده کنیم. در help راجع به دستور subplot مطالعه کنید. اسکریپتی که منجر به شکل زیر می شود را بنویسید و با نام p1\_2.m ذخیره کنید. تصویر اسکریپت را در گزارش بیاورید.



## بخش دوم:

سیستم زیر را در نظر بگیرید که یک رابطه ی خطی را نشان می دهد.



ورودی سیستم  $x(t)$  و خروجی آن  $y(t)$  است.  $\alpha$  و  $\beta$  نیز اعدادی ثابت هستند. متغیر p2.mat را load کنید. ورودی  $x(t)$  و خروجی  $y(t)$  به ازای  $t = 0:0.001:1$  به شما داده شده است.

تمرین ۱-۲ ورودی  $x$  را بر حسب  $t$  رسم کنید. حتماً شکل آن را در گزارش بیاورید.

تمرین ۲-۲ خروجی  $y$  را بر حسب  $t$  رسم کنید. حتماً شکل آن را در گزارش بیاورید. همان طور که مشاهده می کنید  $y$  نویزی است. این اتفاق در عمل بسیار محتمل است و نمی شود انتظار داشت همیشه به خروجی ایده آل دسترسی داشته باشیم.

تمرین ۳-۲ متغیر  $y$  را بر حسب  $x$  با استفاده از دستور زیر رسم کنید. آرگومان سوم دستور زیر باعث می شود که نقاط به هم وصل نشوند و بهتر بتوانید رابطه ی  $y$  و  $x$  را مشاهده کنید. حتماً شکل را در گزارش بیاورید.

`plot(x,y, ' . ')`

همان طور که مشاهده می کنید رابطه ی  $y$  بر حسب  $x$  تقریباً یک رابطه ی خطی است. شیب این خط چه پارامتری را به شما می دهد؟ عرض از مبدا این خط چه پارامتری را به شما می دهد؟

تمرین ۴-۲ روشی را پیشنهاد دهید که با استفاده از داده های  $x$  و  $y$  بتوان پارامترهای  $\alpha$  و  $\beta$  را استخراج کرد. روش خود را در یک فانکشن با نام p2\_4.m پیاده سازی کنید. این فانکشن داده های  $x$  و  $y$  را به عنوان ورودی گرفته و پارامترهای  $\alpha$  و  $\beta$  را به عنوان خروجی می دهد. پارامترهایی که به دست آوردید را گزارش کنید.

راهنمایی: می توانید با مینیمم کردن تابع هزینه زیر، پارامترها را استخراج کنید.

$$f(\alpha, \beta) = \sum_t (y(t) - \alpha x(t) - \beta)^2$$

**توجه:** حتماً چک کنید فانکشنی که نوشتید درست کار کند. برای این کار می توانید خودتان داده های جدیدی تولید کنید.  $\beta$  ،  $\alpha$  ،  $x$  دلخواه در نظر بگیرید و بعد طبق رابطه های زیر خروجی را تولید کنید:

$$y(t) = \alpha x(t) + \beta \quad (\text{حالت بدون نویز})$$

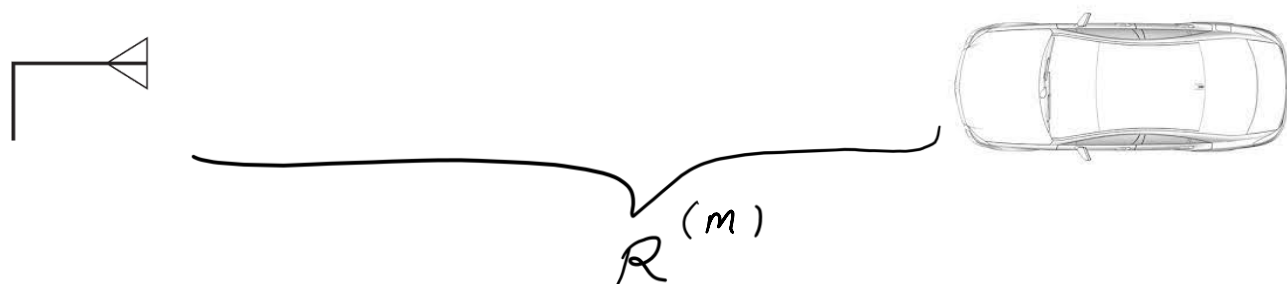
$$y(t) = \alpha x(t) + \beta + \text{Noise} \quad (\text{حالت با نویز})$$

بعد چک کنید ببینید آیا فانکشنی که نوشتید پارامترهای  $\alpha$  و  $\beta$  را به درستی تخمین می زند یا خیر؟ برای تولید نویز می توانید از دستور randn استفاده کنید. برای اطلاعات بیشتر راجع به این دستور به help مراجعه کنید. چون می خواهید چک کنید ببینید کدتان درست کار می کند یا نه، قدرت نویز را کم در نظر بگیرید.

**تمرین ۲-۵)** در نوار APPS ، اپلیکیشن Cure Fitting را باز کنید. داده های  $x$  و  $y$  که در ماتریس p2.mat در اختیار شما قرار گرفته بود را وارد کنید. تخمین polynomial با درجه یک را انتخاب کنید. آیا نتایجی که در سمت چپ به شما ارائه می دهد، با نتایجی که در تمرین ۲-۴ به دست آوردید همخوانی دارد؟

### بخش سوم:

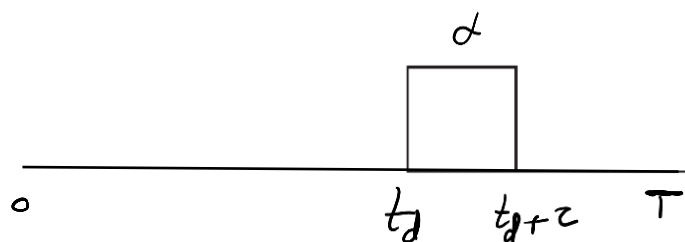
می خواهیم توسط یک رادار، فاصله ی یک جسم از رادار را بیابیم.



سیگنال ارسالی رادار به صورت زیر است:



این سیگنال پس از برخورد به جسم و بازگشت به سمت رادار، به صورت زیر دریافت می شود:



که در شکل بالا  $t_d = \frac{2R}{c}$  است و مقدار  $\alpha < 1$  اهمیتی ندارد. در این رابطه  $c$  سرعت نور با واحد متر بر ثانیه و  $R$  فاصله ی رادار تا جسم با واحد متر است.

پارامتر مجهول مساله  $t_d$  می باشد زیرا با به دست آوردن آن، فاصله نیز مشخص می شود. برای به دست آوردن این پارامتر، از همان ایده ی correlation یا template matching استفاده می کنیم.

تمرین ۱-۳ سیگنال ارسالی را با فرضیات زیر تولید کرده و رسم کنید (  $t_s$  فاصله ی دو نمونه ی زمانی و یا عکس فرکانس نمونه برداری  $f_s$  است).

$$t_s=1e-9; T=1e-5; \tau=1e-6;$$

تمرین ۲-۳ سیگنال دریافتی را با فرض  $R = 450 \text{ m}$  تولید کرده و رسم کنید (  $\alpha = 0.5$  ).

تمرین ۳-۳ حال به صورت برعکس به مساله نگاه کنید. یعنی فرض کنید سیگنال دریافتی را داریم و می خواهیم از روی آن فاصله را محاسبه کنیم. با ایده ی correlation یا template matching، این کار را انجام دهید.

تمرین ۴-۳ حال به سیگنال دریافتی کمی نویز اضافه کنید و تمرین ۳-۳ را تکرار کنید. قدرت نویز را کم کم و طی چندین مرحله افزایش دهید و ببینید تا کجا همچنان می توانید فاصله را به درستی تشخیص دهید؟ فرض کنید اگر خطای فاصله یابی کمتر از ۱۰ متر باشد، بگوییم فاصله درست تخمین زده شده است.

توجه: از آنجایی که نویز ماهیتی تصادفی دارد و ممکن است هر دفعه یک پاسخی به دست آورید، برای انجام دقیق این تمرین می بایست به صورت زیر عمل کنید:

قدرت نویز را کم کم زیاد کنید. برای هر قدرت نویز ثابت، یک حلقه ی for صدتایی (یا بیشتر) باید داشته باشید که در هر حلقه ی آن یک نویز جدید با همان قدرت ثابت تولید کرده و به داده اضافه کنید. حال فاصله را تخمین زده و خطای فاصله یابی را ذخیره کنید. میانگین خطا در این حلقه ی صدتایی (یا بیشتر) را به عنوان خطا در این قدرت نویز در نظر بگیرید. در پایان می توانید خطا را بر حسب قدرت نویز های مختلف رسم کنید. طبیعتاً هر چه قدرت نویز بیشتر باشد باید خطای تخمین فاصله بیشتر شود.

## بخش چهارم:

**تمرین ۴-۱)** یک بیت از شاعری که به او علاقه دارید را در مدت حداکثر ۱۰ ثانیه بخوانید و آن را به فرمت wav ضبط کنید (حتما می بایست صدای خود شما باشد!). فایل صوتی را با استفاده از دستور audioread داخل متلب وارد کنید و در بردار  $x$  ذخیره کنید. با استفاده از همین دستور، فرکانس نمونه برداری آن را مشخص کنید. در واقع، فایل مورد نظر از نمونه برداری سیگنال پیوسته زمانی بدست آمده است و فرکانس نمونه برداری نشان می دهد که در هر ثانیه، چند نمونه از سیگنال پیوسته گرفته شده است.

**تمرین ۴-۲)** سیگنال صوتی را رسم کنید و محور افقی را بر حسب ثانیه label گذاری کنید. توجه داشته باشید که در محیط متلب با استفاده از دستور sound می توانید فایل را گوش کنید. سیگنال  $x$  را با دستور audiowrite به صورت x.wav ذخیره کنید.

**تمرین ۴-۳)** تابعی با نام p4\_3.m بنویسید که دو ورودی  $x$  (همان فایل صوتی شما) و speed (سرعت پخش) را بگیرد و سپس  $x$  را با سرعت speed پخش کند.

### توجه:

- پارامتر speed فقط می تواند 0.5 یا 2 باشد. وقتی speed=2 است یعنی سرعت پخش باید دو برابر شود و وقتی speed=0.5 است سرعت پخش باید نصف شود.
- اگر speed عددی به غیر از بازه ی تعریف شده در پاراگراف قبل بود (مثلا 0.65 یا 1.24) تابع خطا بدهد و هیچ کار دیگری انجام ندهد.
- هنگامی که سرعت پخش برابر 2 است باید یکی در میان نمونه های سیگنال را دور بریزید.
- هنگامی که سرعت پخش برای 0.5 است باید بین هر دو نمونه ی سیگنال یک نمونه اضافه کنید. برای این کار می توانید میانگین دو نمونه مجاور را به عنوان نمونه ی جدید در نظر بگیرید.

**تمرین ۴-۴)** تابع بخش قبل را ارتقا دهید به گونه ای که با هر speed دلخواهی (مثلا عددی دلخواه بین 0.5 تا 2 و البته فرض کنید مضرب 0.1 باشد) فایل صوتی را پخش کند. تابع را با نام p4\_4.m ذخیره کنید.

## نکات کلی:

- در صورت وجود هرگونه پرسش و ابهام به نگار شمایی و استاد ایمیل بزنید.
- فایل نهایی شما باید به صورت یک فایل زیپ شامل گزارشکار به فرمت PDF و کد های متلب باشد.