

به نام خدا
سیگنال ها و سیستم ها
تمرین کامپیوتری اول

بخش اول:

در این بخش کارهای زیر را طبق توضیحات انجام دهید. احتیاجی به نوشتن گزارش برای این بخش نیست. صرفاً می خواهیم در Matlab با توابع مختلف و همچنین تعریف متغیرها به صورت های مختلف آشنا شویم.

For problems 1-7, write a script called `shortProblems.m` and put all the commands in it. Separate and label different problems using comments.

1. **Scalar variables.** Make the following variables

- a. $a = 10$
- b. $b = 2.5 \times 10^{23}$
- c. $c = 2 + 3i$, where i is the square root of -1
- d. $d = e^{j2\pi/3}$, where j is the square root of -1 and e is Euler's number¹ (use `exp`, `pi`)

2. **Vector variables.** Make the following variables

- a. $aVec = [3.14 \ 15 \ 9 \ 26]$
- b. $bVec = \begin{bmatrix} 2.71 \\ 8 \\ 28 \\ 182 \end{bmatrix}$
- c. $cVec = [5 \ 4.8 \ \dots \ -4.8 \ -5]$ (all the numbers from 5 to -5 in increments of -0.2)
- d. $dVec = [10^0 \ 10^{0.01} \ \dots \ 10^{0.99} \ 10^1]$ (Logarithmically spaced numbers between 1 and 10, use `logspace`, make sure you get the length right!)
- e. $eVec = \text{Hello}$ ($eVec$ is a string, which is a vector of characters)

3. **Matrix variables.** Make the following variables

a. $aMat = \begin{bmatrix} 2 & \dots & 2 \\ \vdots & \ddots & \vdots \\ 2 & \dots & 2 \end{bmatrix}$ a 9x9 matrix full of 2's (use **ones** or **zeros**)

b. $bMat = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & \ddots & 0 & \ddots \\ \vdots & 0 & 5 & 0 \\ \vdots & \ddots & 0 & \ddots \\ 0 & \dots & 0 & 1 \end{bmatrix}$ a 9x9 matrix of all zeros, but with the values
 $[1 \ 2 \ 3 \ 4 \ 5 \ 4 \ 3 \ 2 \ 1]$ on the main diagonal (use **zeros**, **diag**).

c. $cMat = \begin{bmatrix} 1 & 11 & \dots & 91 \\ 2 & 12 & \ddots & 92 \\ \vdots & \vdots & \ddots & \vdots \\ 10 & 20 & \dots & 100 \end{bmatrix}$ a 10x10 matrix where the vector 1:100 runs down the
columns (use **reshape**).

d. $dMat = \begin{bmatrix} NaN & NaN & NaN & NaN \\ NaN & NaN & NaN & NaN \\ NaN & NaN & NaN & NaN \end{bmatrix}$ a 3x4 NaN matrix (use **nan**)

e. $eMat = \begin{bmatrix} 13 & -1 & 5 \\ -22 & 10 & -87 \end{bmatrix}$

f. Make $fMat$ be a 5x3 matrix of random integers with values on the range -3 to 3 (First use **rand** and **floor** or **ceil**. Now only use **randi**)

4. **Scalar equations.** Using the variables created in 1, calculate x , y , and z .

a. $x = \frac{1}{1 + e^{-(a-15)/6}}$

b. $y = \left(\sqrt{a} + \sqrt[2]{b} \right)^\pi$, recall that $\sqrt[g]{h} = h^{1/g}$, and use **sqrt**. You can also use **nthroot** (refer to the MATLAB help to understand the difference between **nthroot** and a fractional power)

c. $z = \frac{\log(\Re[(c+d)(c-d)] \sin(a\pi/3))}{c\bar{c}}$ where \Re indicates the real part of the complex number in brackets, \bar{c} is the complex conjugate of c , and \log is the *natural* log (use **real**, **conj**, **log**).

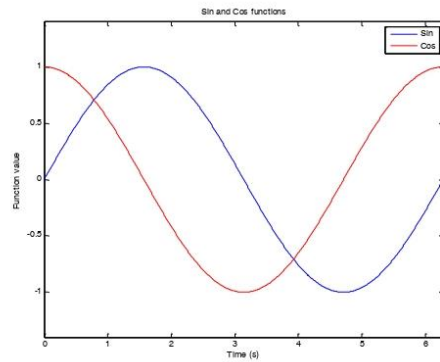
5. **Matrix equations.** Using the variables created in 2 and 3, find the values of `xMat`, `yMat` and `zMat` below. Use matrix operators.
 - a. $\text{xMat} = (\text{aVec} \cdot \text{bVec}) \cdot \text{aMat}^2$
 - b. $\text{yMat} = (\text{bVec} \cdot \text{aVec})$, note that this is *not* the same as $(\text{aVec} \cdot \text{bVec})$
 - c. $\text{zMat} = |\text{cMat}|(\text{aMat} \cdot \text{bMat})^T$, where $|\text{cMat}|$ is the determinant of cMat , and T again indicates the transpose (use **det**).

6. **Common functions and indexing.**
 - a. Make `cSum` the column-wise sum of `cMat`. The answer should be a row vector (use **sum**).
 - b. Make `eMean` the mean across the rows of `eMat`. The answer should be a column (use **mean**).
 - c. Replace the top row of `eMat` with $[1 \ 1 \ 1]$.
 - d. Make `cSub` the submatrix of `cMat` that only contains rows 2 through 9 and columns 2 through 9.
 - e. Make the vector $\text{lin} = [1 \ 2 \ \dots \ 20]$ (the integers from 1 to 20), and then make every even value in it negative to get $\text{lin} = [1 \ -2 \ 3 \ -4 \ \dots \ -20]$.
 - f. Make `r` a 1x5 vector using **rand**. Find the elements that have values <0.5 and set those values to 0 (use **find**).

7. **Plotting multiple lines and colors.** In class we saw how to plot a single line in the default blue color on a plot. You may have noticed that subsequent plot commands simply replace the existing line. Here, we'll write a script to plot two lines on the same axes.
 - a. Open a script and name it `twoLinePlot.m`. Write the following commands in this script.
 - b. Make a new figure using **figure**
 - c. We'll plot a sine wave and a cosine wave over one period
 - i. Make a time vector t from 0 to 2π with enough samples to get smooth lines
 - ii. Plot $\sin(t)$
 - iii. Type **hold on** to turn on the 'hold' property of the figure. This tells the figure not to discard lines that are already plotted when plotting new ones. Similarly, you can use **hold off** to turn off the hold property.
 - iv. Plot $\cos(t)$ using a red dashed line. To specify line color and style, simply add a third argument to your plot command (see the third paragraph of the **plot** help).

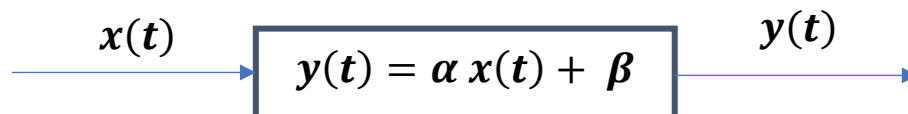
This argument is a string specifying the line properties as described in the help file. For example, the string 'k:' specifies a black dotted line.

- d. Now, we'll add labels to the plot
 - i. Label the x axis using **xlabel**
 - ii. Label the y axis using **ylabel**
 - iii. Give the figure a title using **title**
 - iv. Create a legend to describe the two lines you have plotted by using **legend** and passing to it the two strings 'Sin' and 'Cos'.
- e. If you run the script now, you'll see that the x axis goes from 0 to 7 and y goes from -1 to 1. To make this look nicer, we'll manually specify the x and y limits. Use **xlim** to set the x axis to be from 0 to 2π and use **ylim** to set the y axis to be from -1.4 to 1.4.
- f. Run the script to verify that everything runs right. You should see something like this:



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. α و β 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 α و β 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 α و β 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. β ، α ، 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 α و β 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.

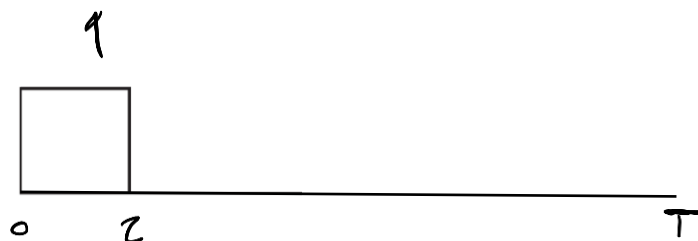
تمرین ۲-۵) در نوار APPS ، اپلیکیشن Cure Fitting را باز کنید. داده های x و y که در ماتریس p2.mat در اختیار شما قرار گرفته بود را وارد کنید. تخمین polynomial با درجه یک را انتخاب کنید. آیا نتایجی که در سمت چپ به شما ارائه می دهد، با نتایجی که در تمرین ۲-۴ به دست آوردید همخوانی دارد؟

بخش سوم:

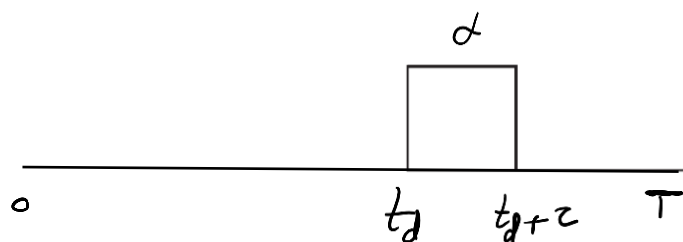
می خواهیم توسط یک رادار، فاصله ی یک جسم از رادار را بیابیم.



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.

نکات کلی:

- فایل نهایی شما باید به صورت یک فایل زیپ شامل گزارشکار به فرمت PDF و کدهای متلب باشد.