

تمرین کامپیوتری سری اول درس سیگنال و سیستم

موعد تحویل: چهارشنبه 13 اسفند 1393 (11:59PM)

- پاسخ تکلیف به صورت یک فایل فشرده zip (یا rar) در درس افزار cw.sharif.ir قبل از موعد تعیین شده آپلود کنید. این فایل باید حاوی گزارش در یک فایل pdf، M-file و سایر موارد خواسته شده باشد.
- نام فایل خود را CHW1_StudentNumber قرار دهید که StudentNumber شماره دانشجویی شماست.
- برای هر مساله M-file جداگانه استفاده کنید و زیر مساله ها را با comment های مناسب از هم تفکیک کنید. کدهای شما باید کامنت گذاری مناسب داشته باشد.

بخش اول: آشنایی و کار با سیگنال

(۱) سیستم های زیر را در نظر بگیرید :

$$1. y_1[n] = x[n] + x[n - 2] + x[n - 4] + x[n - 6]$$

$$2. y_2[n] = \begin{cases} x\left[\frac{n}{2}\right] & n = 2k \\ 0 & o.w. \end{cases}$$

$$3. h[n] = \begin{cases} \left(\frac{1}{2}\right)^n & n = 0, 2, 4, 6, 8, \dots, \infty \\ 0 & otherwise \end{cases}$$

$$4. y[n] - \frac{3}{4}y[n - 1] + \frac{1}{8}y[n - 2] = 2x[n - 1]$$

$$5. h(t) = \sin\left(\pi/6 t\right)u(t)$$

الف. به کمک دستورهای `plot` (برای رسم سیگنال های پیوسته) و `stem` (برای رسم سیگنال های گسسته) پاسخ ضربه ی هر یک از سیستم های فوق را در بازه زمانی مناسب رسم کنید. (در قسمت `help` توضیحات مربوط به این دستور و خواص و قابلیت های هر کدام را بخوانید. هر یک از نمودار های شما باید دارای عنوان مناسب ، نام گذاری مناسب برای هر یک از محورها و تنظیم مقیاس های مناسب برای هر دو محور باشد).

ب. پاسخ پله هر یک از سیستم ها را با استفاده از انتگرال گیر و به کمک دستور `subplot` در کنار پاسخ ضربه رسم نمایید.

حال سیگنال های زیر را به عنوان ورودی به این سیستم ها در نظر بگیرید :

$$1. y[n] = \begin{cases} \cos\left(\frac{n\pi}{6}\right) & n \text{ odd} \\ e^{-n} & n \text{ even} \end{cases}$$

$$2. y[n] = \sin\left(\frac{n\pi}{6}\right)$$

$$3. y[n] = \left(\frac{1}{4}\right)^n \sin(n)$$

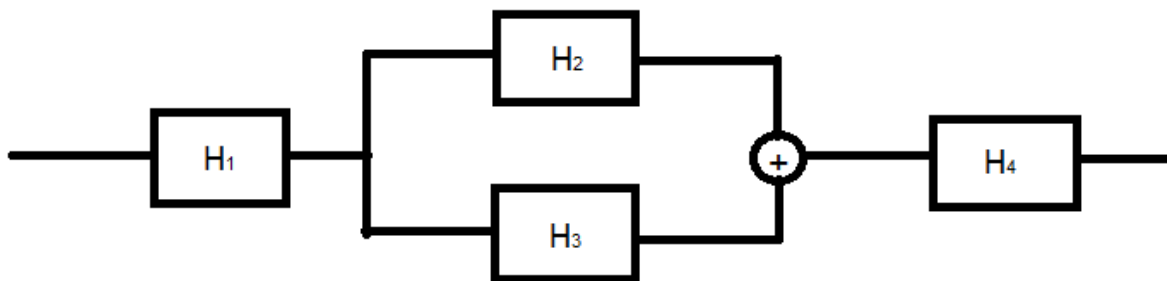
$$4. y(t) = e^{-3t} + 2e^{-t}$$

$$5. y(t) = \sin\left(\frac{2}{3}t\right) + \cos(\pi t)$$

ج. سیگنال ها را در بازه زمانی `[-5,5]` رسم کنید.

د. خروجی سیستم 5 را به سیگنال های `CT` داده شده و سیستم های 1 تا 4 به سیگنال های `DT` داده شده را محاسبه و رسم نمایید. در صورت نیاز برای محاسبه خروجی سیستم از دستور `conv` استفاده کنید. (با حفظ نکات مطرح شده در قسمت اول)

(۲) سیستم زیر را در نظر بگیرید:



$$h_1(t) = e^{-t}u(t)$$

$$h_2(t) = -2e^{-5t}u(t)$$

$$h_3(t) = 5e^{-14t}u(t)$$

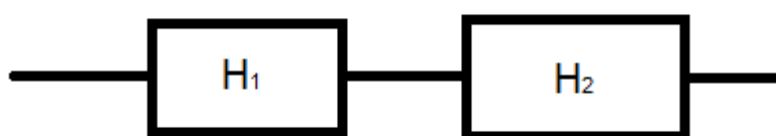
$$h_4(t) = (\sin \pi \alpha t) / \pi t$$

الف) پاسخ ضربه سیستم را به ازای $\alpha = 30$ محاسبه و رسم نمایید. با توجه به شکل پاسخ ضربه در مورد علّیت، حافظه و پایداری سیستم اظهارنظر کنید.

ب) بخش الف را به ازای $\alpha = 1, 5, 25$ تکرار نمایید.

ج) از بخش ب) چه نتیجه ای در مورد عملکرد بلوک H_4 می گیرید؟

۳) در سیستم زیر به موارد خواسته شده پاسخ دهید :



$$h_1[n] = 5\delta[n]$$

$$h_2[n] = (1/3)^n u[n]$$

الف) پاسخ ضربه سیستم را بدست آورده و رسم کنید.

ب) با تغییر $h_2[n] = (1/2)^n u[n]$ ، $h_2[n] = (1/10)^n u[n]$ ، $h_2[n] = (1/3)^{2n} u[n]$ بخش الف را تکرار کنید.

ج) پاسخ های ضربه الف و ب را در یک شکل رسم نموده و مقایسه نمایید (برای رسم این قسمت از دستور hold استفاده کنید).

د) با تغییر $h_1[n] = 5\delta[n-3]$ ، $h_1[n] = 5\delta[n-10]$ ، $h_1[n] = 5\delta[n+1]$ بخش الف را تکرار کنید.

ه) پاسخ های ضربه الف و د را در یک شکل رسم نموده و مقایسه نمایید.

بخش دوم: پردازش صوت ابتدایی

در این تمرین پدیده ی پژواک (اکو) صوت در یک محیط را شبیه سازی می کنیم. در حالت کلی مدل پیشنهادی برای این پدیده یک سیستم به صورت زیر است :

$$y(t) = a_0x(t) + a_1x(t - T) + \dots + a_nx(t - nT)$$

که در آن x صوت خارج شده از منبع صوت و y سیگنال دریافتی گیرنده (مثلاً گوش انسان) است. در یک محیط ساده (مثل یک کلاس) ضرایب a_i به صورت $a_i = \alpha^i$ در نظر گرفته می شود که α عددی کوچکتر از یک است و گین اکو نام دارد.

فایل `sound.wav` را از سایت درس دانلود کنید .

الف. تابعی بنویسید که فایل صوتی، تعداد تکرار اکو، فاصله ی زمانی بین هر بازتاب ، گین اکو و فرکانس نمونه برداری را به عنوان ورودی گرفته و صوت اکو شده را به عنوان خروجی تابع تولید کند. (اگر Fs فرکانس نمونه برداری از صوت پیوسته توسط کامپیوتر باشد، در مدل اکو که رابطه ی آن را در بالا نوشتیم به جای T ، $T = Fs * T$ بگذارید).

فایل `sound.wav` را بخوانید و پخش کنید (به کمک دستورات `audioread` , `wavread` می توانید فایل صوتی را ذخیره کنید و با استفاده از دستور `wavplay` صوت ذخیره شده قابل پخش است.) صوت ذخیره شده را نیز به کمک دستور `plot` رسم کنید.

نمونه ی خروجی تابع با 3 تکرار و 1 ثانیه تاخیر بین هر تکرار :

$$y(t) = x(t) + x(t - Fs) + x(t - 2Fs) + x(t - 3Fs)$$

فایل تولید شده را پخش کرده و گوش کنید . توضیح دهید. به کمک دستور `subplot` صوت اکو شده را در کنار صوت اصلی رسم کرده و تفاوت این دو سیگنال را بیان کنید.

ب. به فایل صوتی اولیه نویز سفید گاوسی را با استفاده از دستور `wgn` یا `awgn` اضافه کنید.
(توان و واریانس نویز را تغییر دهید و تاثیرات تغییر هر کدام را شرح دهید.)

فایل را پخش کنید و گوش کنید. توضیح دهید. مجدداً سیگنال نویزی را در کنار سیگنال اولیه رسم کرده و تفاوت این دو را مشاهده و بیان کنید.

هر کدام از سیگنال های خروجی در قسمت های الف و ب را با عنوان های مناسب ذخیره کنید.

بخش سوم: پردازش ابتدایی تصویر

الف. دو تصویر `pic1.jpg` و `pic2.jpg` را از سایت درس دانلود کنید و در متلب هر کدام از تصویر ها را بخوانید و ذخیره کنید (به کمک دستور `imread` و `imwrite`). میدانیم که هر تصویر از ترکیب سه تصویر با رنگ های آبی، سبز و قرمز ساخته میشود. در متلب هر تصویر با یک ماتریس $n \times m \times 3$ ذخیره میشود که هر سه کانال `r` و `g` و `b` در تصویر به این صورت قابل بازیابی است. مولفه های `r` و `g` و `b` هر تصویر را به صورت جداگانه نمایش داده و با عنوان مناسب ذخیره کنید.

برای تصویر دوم میانگین سه مولفه `r` و `g` و `b` هر نقطه را محاسبه کنید و در ماتریس جدیدی ذخیره کنید. عکس خروجی را مشاهده کنید. حال با دستور `rgb2gray` تصویر را سیاه سفید کنید و با عنوان `pic1_grayscale` ذخیره کنید. نتایج را با یکدیگر مقایسه کنید. (دستور `imshow` را مطالعه و برای نمایش هر تصویر از آن استفاده کنید.)

ب. همانطور که مشاهده میشود ، تصویر اول پس زمینه ی تصویر دوم است . تصویر اول را که پس زمینه است از تصویر دوم کم کنید و قدرمطلق این تفاضل را در ماتریس جدیدی ذخیره کنید. تصویر به دست آمده را مشاهده کنید و با عنوان difference ذخیره کنید. توضیح دهید.

ج. تفاضل دو تصویر را به تصویر باینری تبدیل کنید و نتیجه را مشاهده کنید. برای انجام این کار میتوانید از دستور `im2bw` در متلب استفاده کنید. تصویر به دست آمده را با عنوان `binary`

ذخیره کنید. به نظر شما این تصویر چه کاربردی میتواند در `image processing` داشته باشد؟

د. تصویر دوم را `blur` کنید. برای انجام اینکار، مقدار هر پیکسل تصویر را برابر میانگین خودش و 9 نقطه ی دورش قرار دهید. نتیجه را مشاهده کرده و با عنوان `blurred` ذخیره کنید. توضیح

دهید.

مثال :

$$F[x, y]$$

| | | | | | | | | | |
|---|---|----|----|----|----|----|----|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 90 | 90 | 90 | 90 | 90 | 0 | 0 |
| 0 | 0 | 0 | 90 | 90 | 90 | 90 | 90 | 0 | 0 |
| 0 | 0 | 0 | 90 | 90 | 90 | 90 | 90 | 0 | 0 |
| 0 | 0 | 0 | 90 | 0 | 90 | 90 | 90 | 0 | 0 |
| 0 | 0 | 0 | 90 | 90 | 90 | 90 | 90 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 90 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

$$G[x, y]$$

| | | | | | | | | | |
|--|----|----|----|----|----|----|----|----|--|
| | | | | | | | | | |
| | 0 | 10 | 20 | 30 | 30 | 30 | 20 | 10 | |
| | 0 | 20 | 40 | 60 | 60 | 60 | 40 | 20 | |
| | 0 | 30 | 60 | 90 | 90 | 90 | 60 | 30 | |
| | 0 | 30 | 50 | 80 | 80 | 90 | 60 | 30 | |
| | 0 | 30 | 50 | 80 | 80 | 90 | 60 | 30 | |
| | 0 | 20 | 30 | 50 | 50 | 60 | 40 | 20 | |
| | 10 | 20 | 30 | 30 | 30 | 30 | 20 | 10 | |
| | 10 | 10 | 10 | 0 | 0 | 0 | 0 | 0 | |
| | | | | | | | | | |