

بسمه تعالی



دستورکار هفتم (گروه یک)

تاریخ: ۱۳۹۹/۰۳/۱۸

مهلت تحویل گزارش: ۱۳۹۹/۰۳/۲۵

نحوه تحویل گزارشکار کارگاه:

هر گروه یک فایل گزارش آماده نماید. فایل اصلی گزارش در قالب ورد (word) باشد و pdf شده آن نیز ضمیمه گردد. در ابتدای فایل گزارش، شماره دستورکار، نام و شماره دانشجویی اعضای گروه و تاریخ آپلود فایل گزارش نوشته شود. در یک فولدر (که نام فولدر حاوی شماره دانشجویی‌های اعضای گروه است) بایستی کدهای M-file متلب (ذخیره شده با پسوند m) و متغیرهای احتمالی مورد نیاز (ذخیره شده با پسوند mat) و فایل‌های word و pdf گزارش قرار داده شود. سپس فولدر مذکور را در قالب rar. فشرده نموده و فایل فشرده شده را در درس افزار آپلود کنید.

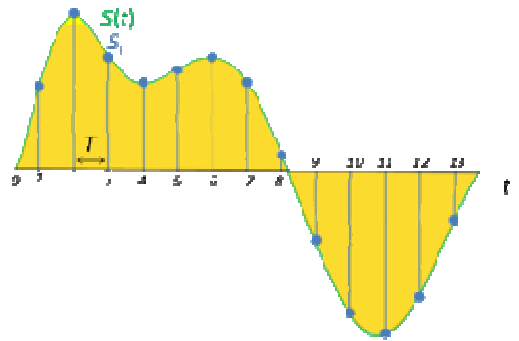
در متن گزارش و کدهای ارسالی، حتماً توضیح (comment) قرار دهید تا خوانایی کدهای برنامه افزایش یابد. ضمناً در متن گزارش، دستورات مورد استفاده و یا کدهای برنامه نویسی شده و توضیحات کافی را نیز قرار دهید.

متلب یک ماشین حساب گرافیکی قدرتمند محسوب می شود. از اینرو رسم شکل‌های مناسب در متلب یک هنر است. از حداکثر سلیقه خود و توانایی‌های نرم افزار متلب برای تهیه نمودارهای مناسب و دقیق استفاده کنید.

پردازش های ساده فایل های صوتی بوسیله متلب

با توجه به اینکه امروزه فایل های صوتی غالباً بصورت دیجیتال ذخیره می شوند، از اینرو بایستی در ابتدا با مفهوم نرخ نمونه برداری یا فرکانس نمونه برداری سیگنال آشنا شویم:

در شکل زیر سیگنال زمان-پیوسته $S(t)$ نمایش داده شده است که با نرخ نمونه برداری F_s نمونه برداری شده است. فرایند نمونه برداری منجر به تولید سیگنال گسسته در زمان می شود:



The sampling frequency or sampling rate, F_s , is the average number of samples obtained in one second (samples per second), thus $F_s = 1/T$.

به طور مثال دستورات زیر منجر به تولید و پخش سه سیگنال موج سینوسی با طول زمانی ۲ ثانیه و نرخ نمونه برداری 16000Hz می شود که هر کدام از این سیگنالها فرکانسهای مختلف (300Hz و 600Hz و 1500Hz) دارند:

```
Fs = 16000; % Hz
Length = 2; % Sec
t = 0 : 1/Fs : Length;
x1=sin(2*pi*300*t);x2=sin(2*pi*600*t);x3=sin(2*pi*1500*t);
soundsc(x1,Fs)
pause(3)
soundsc(x2,Fs)
pause(3)
soundsc(x3,Fs)
pause(3)
```

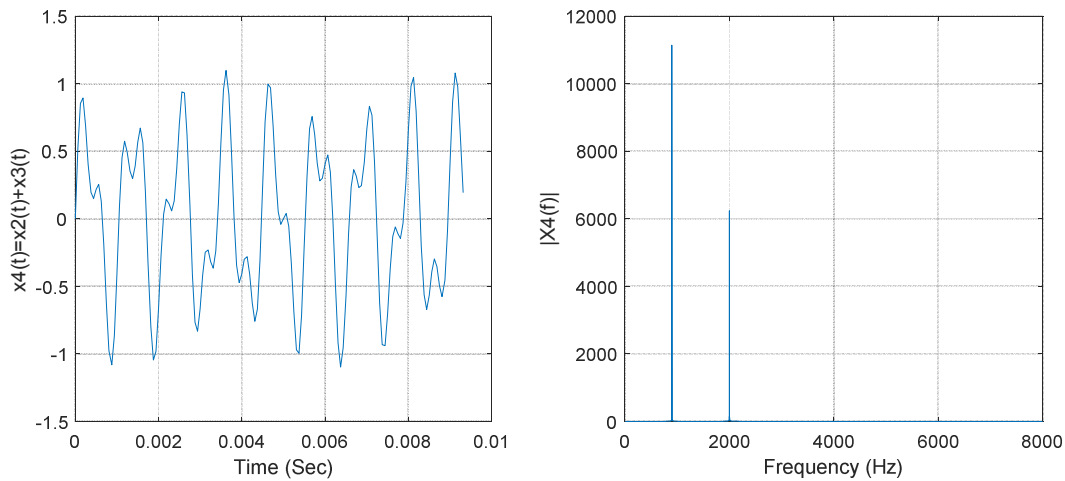
* توجه: هر یک از سیگنالهای سینوسی x_1 و x_2 و x_3 که منحصراً حاوی یک فرکانس مشخص هستند، تک تُن (Single Tone) نامیده می شوند.

* اگر یک سیگنال، حاصل جمع دو سیگنال تک تُن باشد، آن سیگنال Dual Tone نامیده می شود. (مانند سیگنال مجموع (x_3+x_2))

```
soundsc(x3+x2,Fs)
pause(3)
sound(0.1*(x3+x2),Fs)
pause(3)
soundsc(0.1*(x3+x2),Fs)
```

* معمولاً از ابزار تبدیل فوریه برای پردازش سیگنال صوتی استفاده می شود. به طور مثال دستورات ذیل منجر به نمایش محتوای فرکانسی سیگنالها می شود. برای این منظور می توان از اندازه تبدیل فوریه سریع (FFT: Fast Fourier Transform) استفاده نمود.

```
Fs = 16000; % Hz
Length = 2; % Sec
t = 0 : 1/Fs : Length;
x1=sin(2*pi*400*t);x2=0.7*sin(2*pi*900*t);x3=0.4*sin(2*pi*2000*t);
x4=x2+x3;
bin_frequency=1:floor(length(t)/2);
f=((bin_frequency-1)/length(bin_frequency))*Fs/2;
X1=abs(fft(x1));
X2=abs(fft(x2));
X3=abs(fft(x3));
X4=abs(fft(x4));
subplot(121)
plot(t(1:150),x4(1:150))
xlabel('Time (Sec)')
ylabel('x4(t)=x2(t)+x3(t)')
grid
subplot(122)
plot(f,X4(bin_frequency))
xlabel('Frequency (Hz)')
ylabel('|X4(f)|')
grid
```



در شکل بالا سمت راست، محور افقی بیانگر فرکانس و محور عمودی نشان دهنده اندازه محتوای فرکانسی سیگنال در هر فرکانس است. دقت شود سیگنالی که با نرخ نمونه برداری F_s نمونه برداری شود، طبق قضیه نمونه برداری، حداکثر حاوی فرکانس $F_s/2$ خواهد بود (در اینجا $F_s/2 = 8000\text{Hz}$). بنابراین در محور افقی، فرکانس در محدوده صفر تا ۸ کیلوهرتز پوشش داده شده است. همچنین تعداد نمونه های محور افقی (رزولوشن فرکانسی) نیز وابسته به تعداد نمونه های زمانی سیگنال اصلی است. هر چه تعداد نمونه های زمانی سیگنال اصلی بیشتر باشد، رزولوشن فرکانسی بهتری خواهیم داشت.

همانطور که در شکل فوق مشاهده می کنید برای سیگنال دو تن $x_4(t)$ که حاوی فرکانسهای ۹۰۰ و ۲۰۰۰ هرتز است، دو پیک فرکانسی در همان فرکانسها در نمایش فرکانسی سیگنال دیده می شود. اندازه پیک ها نیز وابسته به دامنه سیگنال سینوسی در آن فرکانس است.

* برای ذخیره سازی یک سیگنال بصورت یک فایل صوتی با نام دلخواه (مثلاً در اینجا نام `sin300.wav`) می توانید از دستور زیر استفاده کنید:

```
>>audiowrite('Sin300.wav',x1,Fs);
```

اکنون می توانید بوسیله پخش کننده صوتی ویندوز، فایل صوتی ذخیره شده را پخش (play) نمایید.

* با استفاده از دستور زیر می توانید اطلاعات مناسبی از مشخصات فایل صوتی (مثلاً تعداد نمونه، نرخ نمونه برداری، طول زمانی فایل و ...) به دست آورید:

```
>>info = audioinfo('Sin300.wav')
```

* برای خواندن محتوای یک فایل صوتی (استخراج سیگنال آن)، می توانید از دستور زیر استفاده نمود:

```
>>[y,fs] = audioread('Sin300.wav');
```

* دقت نمایید اگر فایل صوتی به صورت تک کاناله یا مونو (mono) ضبط شده باشد، y یک بردار خواهد بود و اگر فایل صوتی به صورت دوکاناله یا استریو (stereo) ضبط شده باشد، y یک ماتریس حاوی دو بردار است که هر بردار متناظر با سیگنال ضبط شده از یک کانال صوتی (میکروفون) است.

در پیوست این پروژه نمونه فایل‌های صوتی مونو و استریو قرار داده شده است.

* برای توقف پخش فایل‌های صوتی با طول زیاد از دستور clear all و یا clear sound استفاده کنید!!! در حالت کلی برای توقف یک برنامه در حال اجرای متلب، می توانید از Ctrl+C یا Ctrl+Break استفاده کنید.

* برای محاسبه طول زمانی فایل صوتی (برحسب ثانیه)، می توان از دستور زیر استفاده نمود:

```
>>DurationY = size(y,1)/fs
```

همچنین معادلاً می توان از دستور زیر استفاده نمود:

```
>>info = audioinfo('Sin300.wav');  
>>info.Duration
```

تکلیف ۷-۱- در مورد عملکرد سیستم‌های DTMF یا Dual-tone multi-frequency signaling تحقیق کنید؟ برنامه ای بنویسید که چند کاراکتر متوالی موجود در جدول ذیل را به عنوان ورودی از کاربر دریافت کند و به ازای هر کاراکتر، سیگنال Dual Tone متناظر با آن را به مدت ۳۰۰ میلی ثانیه پخش نماید؟ فاصله زمانی بین هر کاراکتر ۱۵۰ میلی ثانیه در نظر گرفته شود.

DTMF keypad frequencies (with sound clips)

	1209 Hz	1336 Hz	1477 Hz	1633 Hz
697 Hz	1	2	3	A
770 Hz	4	5	6	B
852 Hz	7	8	9	C
941 Hz	*	0	#	D

تکلیف ۷-۲- برنامه ای در متلب بنویسید که آدرس یک فایل صوتی (مانند فایل Test_DTMF.wav که در پیوست پروژه آمده است) که حاوی توالی سیگنالهای Dual Tone (متناظر با کاراکترهای DTFM) است را دریافت نماید و توالی کارکترهای متناظر با آن فایل صوتی را تعیین نماید؟ (معادل با دیکدینگ سیگنال DTMF)

تکلیف ۷-۳ (اختیاری)- با تعریف مناسب فرکانس نت های موسیقی، برنامه ای در متلب بنویسید که کیبورد پیانو را شبیه سازی نماید.

پیروز و موفق باشید

شکفته