به نام خدا

سیگنال و سیستم - دکتر صالح کلیبر

تمرین کامپیوتری سری اول

95-1

در این تمرین قصد داریم که ابتدا تا حدی با نرم افزار MATLAB آشنا شویم. این نرم افزار در حال حاضر کاربردی ترین ابزار مورد استفاده برای پردازش سیگنال ها است. زبان مورد استفاده این نرم افزار ساده و محیط آن بسیار کاربردی می باشد.

از ابتدا به این نکته توجه کنید که MATLAB با سیگنال های گسسته کار می کند و هرجا از شما سیگنال پیوسته خواسته شد منظور یک سیگنال گسسته با نمونه های زیاد است.

بخش اول:

۱) سیگنال پیوسته زیر را در بازه 20 ثانیه و با $\Delta t=0.001$ در محیط MATLAB رسم کنید (اندازه و فاز).

$$x_a(t) = 1 + cos(4\pi t) + e^{jt}(cos(2\pi t) + cos(\pi t))$$

۲) حال سیگنال های زیر را در محیط MATLAB رسم کنید.

$$h_1 = e^t$$
$$h_2 = e^{-t}$$

۳) حال سیگنال x_a را یک بار از سیستمی با پاسخ ضربه h_1 و بار دیگر از سیستمی با پاسخ ضربه h_2 عبور دهید. برای عبور سیگنال از این سیستم ها از دستور conv استفاده نمایید. (دقت نمایید که برای استفاده از دستور conv حتما راهنمای MATLAB را مطالعه فرمایید.)

راهنمایی: ممکن است خروجی conv تماما خروجی مورد نظر ما نباشد.

۴) حال دو خروجی قسمت قبل را رسم نمایید و با هم مقایسه کنید. چه نتیجه ای میگیرید؟ پاسخ را از دید مشخصات دو سیستم مذکور بررسی نمایید . در گزارش خود به صورت تئوری صحت حرف خود را اثبات نمایید.

بخش دوم:

۵) یکی از کاربرد های پردازش سیگنال آماده کردن داده های خام حاصل از آزمایش به داده های نهایی میباشد. در این بخش قصد داریم داده های عملی حاصل از یک آزمایش را پردازش کنیم. بدین منظور ابتدا فایل accel_data.txt را بخوانید. برای این کار از دستورات fopen ، fclose و textscan استفاده کنید. نحوه خواندن فایل به صورت کامل در راهنمایی دستور textscan بیان شده است. خروجی textscan را در آرایه ی C ذخیره کنید. حال مقادیر C را درون چهار بردار x, y, z ,w بریزید.

راهنمایی:

```
C=textscan(fileID, '%f %f %f %f');
x=C{1,1};
y=C{1,2};
z=C{1,3};
w=C{1,4};
```

۶) در قسمت قبل مشاهده کردید که اعداد به دست آمده فوق العاده نویزی هستند. برای محدود کردن نویز میتوان از یک فیلتر میانگین_متحرک (moving average) استفاده کرد. این فیلتر ها به مانند یک فیلتر پایین گذر نویز را محدود میکنند. در باره این نوع فیلتر ها و شکل پاسخ ضربه آنها یک تحقیق کوتاه(حداکثر 5 خط) کرده و در گزارش خود ذکر کنید.

۷) حال فرض کنید پاسخ ضربه این فیلتر به صورت زیر باشد:

```
h= 0.1*ones(1,ma_length);
```

حال کانولوشن x, y, z را با h محاسبه کنید و خروجی را برای حالات زیر رسم کنید.

الف) ma_length=10

ب) ma_length=100

ج) ma_length=1000

در هر مورد سیگنال خروجی و ورودی فیلتر را با استفاده از دستور subplot در کنار هم بگذارید.

۸) تو ضیح دهید که سیگنال بعد از گذر از فیلتر چه تغییری کرده است؟ با افزایش طول فیلتر عملکرد آن بهبود یافته است یا خیر؟

بخش سوم:

در این بخش قصد داریم با کمک MATLAB آهنگ بسازیم. همانگونه که قطعا میدانید یک موسیقی مجموعه ای از تعداد زیادی سینوسی با فرکانس معلوم میباشد که به آنها note های موسیقی میگویند. برای اینکه موسیقی ما به یک آلت موسیقی آنالوگ شبیه شود باید هر note با تعدادی از هارمونیک های بعدش جمع شود. (حداقل ده هارمونیک) (هارمونیک موج سینوسی با فرکانسی مضرب فرکانس اصلی می باشد، برای مثال اگر فرکانس اصلی f باشد، هارمونیک های آن $2f, 3f, \ldots$ می باشند.)

در نهایت این مجموع را در یک تابع نمایی میرا ضرب میکنند تا میرایی که در ابزار آلات موسیقی وجود دارد دیده شود. حال با کمک هم سعی در پیاده سازی یک آهنگ کوچک داریم.

مدل ریاضی یک note به صورت زیر است:

$$x(j) = na(j) * sin(2 * pi * j * nf(j) * tt)/j;$$

که در آن j شماره هارمونیک (هارمونیک چندم است؟)، na دامنه note، nf فرکانس note و tt بردار زمان ما است. دقت شود که در اینجا برای سادگی فرض میکنیم دامنه هارمونیک jاُم همان دامنه note است که بر j تقسیم شده است.

حال برای ساخت مدل یک note و هارمونیک های آن کافیست j را از یک تا 10 تغییر داده و هر بار x را با مقدار قبلی اش جمع کنیم.

بحث مهمی که در اینجا وجود دارد نحوه ساخت tt می باشد؛ به دستور زیر توجه کنید:

```
tt= 0: 1/FS: TD/nd;
```

که در آن tt بردار زمان مرتبط با tone است. FS فرکانس نمونه برداری ما است که در طول برنامه ثابت است. TD مدت زمانی است که یک note کامل طول میکشد و nd بیانگر نسبت note موجود با یک note کامل است. برای مثال ممکن است یک note در موسیقی نیاز باشد که نصف TD پخش شود. برای این حالت nd=2 در نظر میگیریم و ... . طول بردار بالا (TD/nd)*FS میباشد.(چرا؟)

حال با عوض کردن فرکانس و سایر موارد بالا می توان note های مختلف را تولید کرد. از کنار هم قرار دادن note های مختلف یک موسیقی پدید می آید.

۹) فایل notes.m به شما داده شده است. این فایل را باز کنید. مشاهده میکنید که اعدادی در این فایل موجودند. این اعداد فرکانس های مورد استفاده می باشند. دقت کنید که در هر بخش یک فرکانس پایه وجود دارد و باقی فرکانس ها از ضرب $2^{1/12}$ در فرکانس قبلی به دست می آید. این فایل را بررسی کنید.

در گزارش خود خلاصه ای از مطالب توضیح داده شده در توضیحات بالا را ذکر کنید. همچنین بیان کنید که tt چگونه به دست آمده است و چرا طول آن FS*(TD/nd) میباشد؟

۱۰) یک فایل جدید MATLAB بسازید. دقت کنید که در ابتدای کد حتما از دستورات clc و clear استفاده کنید. حال دقت کنید که فایل notes.m در پوشه جاری شما باشد. اکنون با استفاده از دستور notes() متغیر های تعریف شده در فایل notes.m را فراخوانی کنید.

۱۱) حال متغیر های زیر را در محیط MATLAB تعریف کنید.

```
FS=16000;

nf=Notes to play

nd=Duration of each note: 8,4,2,1= 1/8,1/4,1/2,1

na=Relative amplitude of each note

TD=Time duration of one whole note in secs
```

مقادیر nfو naو nd و TD را از فایل odetojoy.m بردارید و در محیط خود کپی کنید. دقت کنید که nf و ... بردارند و nf(i) نشان دهنده نت iأم میباشد.

۱۲) حال طول سیگنال نهایی باید محاسبه شود. با استفاده از یک حلقه به ازای i های مختلف طول هر نت را با استفاده از فرمول ارایه شده در توضیحات بیابید و با جمع کردن همه مقادیر طول سیگنال نهایی را بیابید. (آن را SUM بنامید.) حال سیگنال نهایی را به صورت زیر تعریف کنید.

```
x=zeros(1, SUM);
```

۱۳) حال به پیاده سازی بخش اصلی کار می پردازیم. روش کلی بدین صورت است که با تغییر i هر بار یک نت تولید میکنیم. برای تولید نت از فرمول ارایه شده در توضیحات استفاده شود. دقت کنید که در اینجا یک حلقه درونی (برای j های مختلف است، که با جمع کردن هارمونیک های 1 تا 10 نت iأم note نهایی iأم را تولید میکند.) و یک حلقه بیرونی که با تغییر i بردار nf را جارو میکند و هر بار یک هارمونیک را تولید میکند. دقت کنید که اولین دستوری که درون حلقه بیرونی قرار میگیرد تعریف tt می باشد.(به همان صورتی که بالا گفته شد.)

نکته بسیار مهم این بخش این است که هر بار که note کامل j ام تولید میشود مقدار آن در جایش در بردار x قرار داده شود. برای این کار می توانید از همان روشی که برای به دست آوردن SUM در بخش قبل استفاده شد ، استفاده کنید.

راهنمایی:

کد زیر ممکن است به کارتان آید.

```
SUMtemp=SUMnew;

SUMnew= SUMnew+ (FS*(TD/nd(i));
```

*xtemp=xtemp.*exp(-tt);*

x(SUMtemp+1:SUMnew+1)=xtemp;

خط سوم دستور های بالا به همان بحث میرایی صدا باز میگردد و باقی خطوط نحوه جایگذاری tone کامل در فایل موسیقی (x) را نشان میدهد. (دقت کنید که حتما متغیر SUMnew را همان ابتدای برنامه با مقدار صفر تعریف کنید.)

شما میتوانید از هر دستوری برای جایگذاری استفاده کنید. حتما در گزارش خود ذکر کنید که کد بالا (یا کد خودتان) چگونه عمل جایگذاری را انجام میدهد.

۱۴) *بعد از اجرای برنامه و رفع اشکالات آن صدای تولید شده نهایی را با استفاده از دستور audiowrite ذخیره کنید.*

فایل های note دیگری نیز در پوشه وجود دارد که میتوانید از آنها نیز استفاده کنید و برنامه خود را تست کنید.

۱۵) *امتیازی: در صورتی که برنامه خود را بدون حلقه و یا با یک حلقه اجرا کردید(با استفاده از خواص ماتریس ها)، نمره امتیازی به شما تعلق میگیرد.*

نکته1: گزارش خود را تا حد امکان کامل بنویسید. در فایل گزارش می بایست تمامی نمودارها و توضیحات خواسته شده قرار گرفته باشد به طوری که بدون مراجعه به کد، قابل فهم باشد.

نکته2: کل نمره تمرین کامپیوتری شما به چند قسمت دسته بندی می شود که تنها یک قسمت آن به کدهای زده شده اختصاص می یابد و بخش های دیگر نمره مربوط به گزارش و تحویل حضوری می باشد. پس اگر گزارش شبیه سازی ناقص یا نامفهوم باشد نمره گزارش کسر خواهد شد. دقت کنید که هر فرد باید به تنهایی گزارش خود را تحویل دهد. در صورتی که گزارش های یکسان توسط افراد مختلف تحویل داده شوند، علاوه بر نمره گزارش از نمره تحویل حضوری تمامی آن افراد کسر خواهد شد.

نکته3: از ارجاع دادن به کد جدا خود داری فرمایید.

در پایان در صورتی که سوالی برای شما ایجاد شد با ایمیل javadfallah@ut.ac.ir در میان بگذارید.

با آرزوی موفقیت