



سیستم‌های چندرسانه‌ای (۱-۴۰۳۴۲)

نیم‌سال دوم ۹۸-۹۹

استاد: مهدی امیری

تمرین سری اول

آشنایی با Sampling و Aliasing و Quantization

تاریخ تحویل: ۱۲ اسفند

به موارد زیر توجه کنید:

- پاسخ تمرین را به همراه تمامی فایل‌ها به صورت یک فایل فشرده، که نام آن در قالب MMS_HW1_LastName_StudentID باشد، به آدرس ایمیل درس به نشانی mms2020spring@gmail.com بفرستید. لطفا عنوان ایمیل خود را همانند قالب فوق قرار دهید.
- مهلت ارسال پاسخ تمرین تا ساعت ۲۳:۵۹ روز اعلام‌شده است. بهتر است نوشتن تمرین را به ساعات پایانی موکول نکنید.
- همکاری و هم‌فکری شما در حل تمرین مانعی ندارد، اما پاسخ ارسالی هر کس حتما باید توسط خود شخص نوشته شده باشد.
- مبنای درس، اعتماد بر پاسخ ارسالی از سوی شماست؛ بنابراین ارسال پاسخ به این معناست که پاسخ آن تمرین، توسط شما نوشته شده‌است. در صورت تقلب یا اثبات عدم نوشتار پاسخ حتی یک سوال از تمرین، برخورد شدیدی صورت خواهد گرفت.
- پاسخ سوالات مطرح شده در صورت تمرین‌ها در قالب یک گزارش با فرمت PDF، نمودارها و شکل‌های خروجی m فایل‌ها و خود m فایل‌ها می‌باشد.

همان‌گونه که در درس با مفهوم سیگنال دیجیتال آشنا شدید، دیجیتال کردن سیگنال شامل دو مرحله نمونه‌برداری^۱ و کوانتیزه کردن^۲ سیگنال است. شما در این تکلیف با نمونه‌برداری سیگنال پیوسته در زمان و پدیده aliasing و نیز کوانتیزه کردن سیگنال آشنا خواهید شد. همچنین اثرات این پردازش را روی سیگنال‌های صوتی و تصویری بررسی خواهید کرد.

سوال ۱. نمونه‌برداری

در این بخش از تمرین تمرکز ما بر روی نمونه‌برداری از یک سیگنال به منظور ذخیره‌سازی آن سیگنال و پخش آن در متلب است. یک سیگنال متناوب مانند سیگنال زیر را به عنوان پایه در نظر می‌گیریم.

$$f(t) = \cos(\omega_c t + \theta) = \cos(2\pi F_c t + \theta)$$

این سیگنال را در زمان ۰ تا ۳ ثانیه با ویژگی‌های زیر در نظر بگیرید.

$$F_c = 25\text{Hz}, \quad \theta = \frac{\pi}{6}$$

برای این سیگنال نرخ نمونه‌برداری مناسبی (F_s) انتخاب کنید تا بدون رخ دادن پدیده Aliasing بتوان از سیگنال نمونه‌برداری کرد.

۱.۱ ورود به متلب

پس از در نظر گرفتن ملاحظات تئوریک نمونه‌برداری در قسمت قبل، حال نوبت آن رسیده تا به صورت عملی وارد کار شویم! در این بخش شما باید F_s مناسبی را انتخاب کنید تا بتوانید سیگنال نمونه‌برداری شده را به خوبی نمایش دهید. پیشنهاد می‌شود برای این کار از فرکانس ۳۳۰ هرتز استفاده نمایید. به منظور نمونه‌برداری همانطور که می‌دانید باید ابتدا سیگنال $f(t)$ را به صورت دنباله $F_n = f(nT)$ در بیاورید که در آن داریم $T = F_s^{-1}$. خروجی خواسته شده از شما در این بخش یک کد متلب است که این سیگنال را پس از تبدیل به دنباله‌ی مربوطه به تصویر بکشد. این کار را برای حداقل ۴ فرکانس مختلف انجام دهید و تمام عکس‌ها و را کنارهم قرار داده و تحلیل کنید.

۲.۱ بدست آوردن طیف فرکانسی

برای بررسی بهتر سیگنال‌ها، ما همواره به طیف فرکانسی و تبدیل فوریه آن‌ها دقت می‌کنیم. در این بخش از تمرین شما باید دنباله‌ای را که در بخش قبلی به دست آوردید به حوزه فرکانس ببرید و سپس آن را مشاهده و بررسی کنید. برای گرفتن تبدیل فوریه از توابع fft و fftshift استفاده کنید (fft(x) و fftshift(x) که x تابع در حوزه‌ی فرکانس است) پس از به دست آوردن تبدیل فوریه اندازه آن را برای فرکانس‌های مختلف در غالب single-side spectrum در متلب نمایش دهید. اگر این عمل را برای فرکانس‌های مختلف نمونه‌برداری انجام دهیم چه تغییراتی را مشاهده خواهیم نمود؟

¹Sampling

²Quantization

۳.۱ اصوات طبیعی

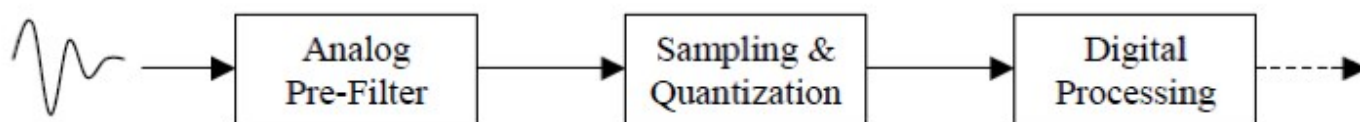
در این بخش برنامه audacity را بر روی سیستم‌های خود نصب کنید. سپس صدای خود را با فرکانس‌های نمونه‌برداری مختلف ضبط کرده و نتایج را تحلیل نمایید. این تغییر فرکانس چه تاثیری در صدایی که از خود می‌شنوید دارد؟ (فرکانس‌های ضبط صدا: 5KH, 10KH, 20KH, 100KH می‌باشند)

۴.۱ اصوات غیرطبیعی

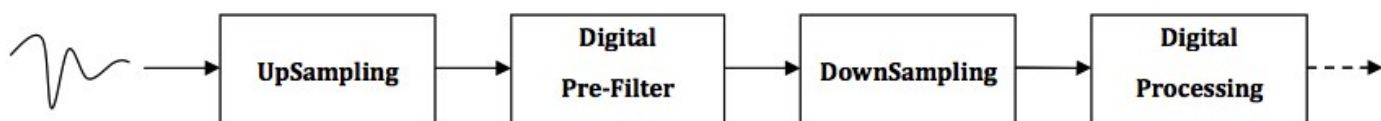
سیگنالی که در قسمت اول سوال داده شده بود مفروض است. از آن با Fs های ۱.۰، ۵.۰، ۱، ۵، ۵.۱۱ و ۵.۱۳ کیلوهرتز نمونه‌برداری کنید. حال دنباله‌های بدست آمده را با فرکانس ۵.۱۲ کیلوهرتز پخش کنید و تفاوت صداها را بدست آمده را با یکدیگر مقایسه کنید و دلیل این تفاوت‌ها را ذکر کنید.

سوال ۲. پدیده Aliasing در سیگنال‌ها

همان‌گونه که می‌دانید، aliasing زمانی رخ می‌دهد که فرکانس نمونه‌برداری کمتر از نرخ نایکوئیست باشد. این پدیده در حوزه فرکانس، به صورت نمایش فرکانس‌های بالا در محدوده فرکانس‌های پایین ظاهر می‌شود. در حوزه زمان نیز، این پدیده باعث از بین رفتن اطلاعات سیگنال می‌گردد. لذا aliasing یک پدیده مزاحم بوده و باید رفع شود. راه‌های مختلفی برای مقابله با aliasing وجود دارد که رعایت نرخ نایکوئیست یکی از آنهاست. متأسفانه بسیاری از سیگنال‌ها در محیط اطراف، محدوده فرکانسی معینی ندارند و برای این سیگنال‌ها، امکان رعایت نرخ نایکوئیست وجود ندارد. یک راه حل مناسب برای ذخیره‌سازی این نوع سیگنال‌ها، حذف فرکانس‌های بالای سیگنال به وسیله یک فیلتر پایین‌گذر آنالوگ پیش از ذخیره‌سازی سیگنال است.



در بسیاری اوقات، طراحی یک فیلتر آنالوگ دقیق کار مشکلی است. لذا برای حل این مشکل از یک فیلتر دیجیتال استفاده می‌شود. در این صورت مراحل ذکر شده در بالا برای نمونه‌برداری سیگنال به صورت زیر تغییر می‌یابد.



ابتدا سیگنال پیوسته در زمان، نمونه‌برداری افزایشی می‌شود. سپس یک فیلتر دیجیتال پایین‌گذر روی آن اعمال شده و دوباره سیگنال نمونه‌برداری کاهش می‌گردد. با توجه به آنچه در درس سیگنال‌ها و سیستم‌ها خوانده‌اید، توضیح دهید که این کار چگونه می‌تواند پدیده aliasing را حذف نماید؟

۱.۲ اثر aliasing روی سیگنال صوت

۱. یک سیگنال صوتی نمونه‌برداری شده در فرکانس ۲۰ کیلوهرتز و ۸ بیت را در نظر بگیرید. این سیگنال، سیگنال upsample شده شماست. برای مشاهده اثر پدیده aliasing روی این سیگنال، بدون پیش‌فیلتر کردن، روی آن نمونه‌برداری کاهشی انجام دهید و نتیجه آن را بررسی کنید. برای این کار می‌توانید از کدی که در ادامه آمده است استفاده کنید.

```
function sp(infilename, dsr)
% infilename is your signal name and dsr is downsampling rate
display([13 'the origional sound' 13]);
[y,Fs]= audioread(infilename);
% Fs is sample rate in Hz

% play it
sound(y,Fs); pause(length(y)/Fs);

% Downsample
display([13 'the downsampled sound' 13]);
x = y(1:dsr:end);
dFs = Fs/dsr;
sound(x, dFs); pause(length(x)/dFs);

% save down sample as down.wav audiowrite('down.wav', x, dFs);
```

۲. یک سیگنال صوتی نمونه‌برداری در این قسمت، برای حذف اثر aliasing، پیش از نمونه‌برداری کاهشی، از یک فیلتر پایین‌گذر برای فیلتر کردن سیگنال استفاده می‌شود. برای فیلتر کردن سیگنال پیش از نمونه‌برداری کاهشی، می‌توانید از دستور زیر کمک بگیرید.

```
filter_coeff = fir1(n, 1/dsr);
filteredSound = filter(filter_coeff, 1, infilename);
```

دستور fir1، یک فیلتر پایین‌گذر را در حوزه زمان به طول n برای شما شبیه‌سازی می‌کند. دستور filter نیز سیگنال ورودی را با فیلتر طراحی شده در حوزه زمان کانالو می‌نماید. برای آگاهی از نحوه عملکرد این دستورات، از help نرم افزار MATLAB کمک بگیرید. سیگنال حاصل را گوش کنید. چه تفاوتی میان این سیگنال و سیگنال قسمت (۱) وجود دارد؟ این مقایسه را برای مقادیر مختلف dsr که در قسمت قبل، امتحان کردید، انجام دهید. آیا عمل پیش‌فیلتر کردن بر روی کیفیت سیگنال موثر بوده است؟ اگر $dsr > 5$ باشد خروجی چگونه خواهد بود؟ آیا هنوز پیش‌فیلتر کردن موثر است؟ چرا؟ در کدام حالت، پیش‌فیلتر کردن بیشترین تاثیر مثبت را روی خروجی این مرحله داشته است؟

سوال ۳. کوانتیزاسیون

نمونه برداری یکی از گام‌ها در جهت ذخیره سازی و پردازش سیگنال‌ها است. گام بعدی، کوانتیزاسیون این سیگنال‌های نمونه برداری شده است.

۱.۱ خطای کوانتیزاسیون

در صورتی که بازه اعداد $[a, b]$ را به n بخش تقسیم کنیم. حداکثر خطای کوانتیزاسیون هر بخش از سیگنال و اندازه هر پله را به صورت نمادین محاسبه نمایید.

۲.۱ کاهش خطای کوانتیزاسیون

برای کمتر کردن تاثیرات منفی کوانتیزاسیون بر روی سیگنال روش‌های مختلفی وجود دارد چند مورد را نام ببرید و یکی از آن‌ها را به دلخواه انتخاب کرده و راجع به آن توضیح دهید. (برای راحت تر شدن پاسخ، فرض می‌کنیم سیگنال مورد بحث، سیگنال صوتی است)

۳.۱ کوانتیزاسیون سیگنال صوتی

با استفاده از فرمول زیر می‌توان کوانتیزاسیون را شبیه سازی کرد. با استفاده از این فرمول یک تابع quantization بنویسید که مقادیر a ، b و N و یک ماتریس مقادیر سیگنال را دریافت کرده و عمل کوانتیزیشن را بر روی آن‌ها شبیه سازی کند و خروجی را به صورت یک رشته از نمونه‌های کوانتیزه شده برگرداند. در فرمول زیر Δ برابر سائز هر پله و a ، b و N به ترتیب برابر با حداقل amplitude سیگنال، حداکثر amplitude سیگنال و تعداد level های کوانتیزاسیون با فواصل یکسان می‌باشد. به طوری که $\Delta = \frac{b-a}{N}$ خواهد بود.

$$Q(x) = \Delta \left\lfloor \frac{x}{\Delta} + \frac{1}{2} \right\rfloor$$

حال از تابعی که نوشته‌اید استفاده کنید و فایل صوتی ضمیمه شده را کوانتیزه کنید و نمودارهای مربوط به اصل سیگنال و کوانتیزه شده سیگنال و همچنین نمودار میزان خطا در لحظات مختلف را رسم کرده و در گزارش خود تحلیل کنید.

موفق باشید

موضوع تمرین بعد: موضوع تمرین بعد: پردازش صوت