



سیستم‌های چندرسانه‌ای (۱-۴۰۳۴۲)

نیم‌سال دوم ۹۸-۹۹

استاد: مهدی امیری

تمرین سری دوم

پردازش سیگنال صوتی

تاریخ تحویل: ۱۵ فروردین

به موارد زیر توجه کنید:

- پاسخ تمرین را به همراه تمامی فایل‌ها به صورت یک فایل فشرده، که نام آن در قالب MMS_HW2_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 پاسخ‌های هر کس حتما باید توسط خود شخص نوشته شده باشد.
- مبنای درس، اعتماد بر پاسخ‌های 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.
- پاسخ‌های سوالات مطرح شده در صورت تمرین‌ها در قالب یک گزارش با فرمت PDF، نمودارها و شکل‌های خروجی m فایل‌ها و خود m فایل‌ها می‌باشد.

مقدمه

در این تمرین، شما برخی از مفاهیم پردازش صوت که در درس با آنها آشنایی پیدا نمودید را مورد آزمایش قرار داده و به بررسی علل آنها می‌پردازید. کرد.

سوال ۱. بخش تشریحی و تحلیلی

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

۱.۱

به وسیله نرم افزار Audacity صدای خود را که در حال گفتن کلمه ی stamp هستید ضبط کرده و spectrogram آن را رسم نمایید. جایگاه تک تک حرف‌ها را در شکل مشخص کنید و دلیل خود را در مورد انتخاب جایگاه حرف a توضیح دهید.

۲.۱

PCM را توضیح دهید و همچنین بیان کنید که PCM چگونه روی داده‌های صوتی به کار می‌رود.

۳.۱

توضیح دهید که چگونه با استفاده از کد گذاری پیشگویانه (DPCM) می‌توانیم متوسط نرخ بیتی را در مقایسه با کدگذاری مستقیم (PCM) کاهش دهیم؟

۴.۱

مفهوم ADPCM را توضیح دهید و دو نوع مختلف آن (backward-adaption و forward-adaption) را تشریح نمایید.

۵.۱

برتری روش CELP (Code Excited Linear Prediction) به روش LPC (Linear Predictive Coding) و همچنین راهکاری که برای این برتری استفاده شده است را بیان کنید.

سوال ۲. کار با متلب

۱.۲ فیلتر کردن سیگنال صوتی و نمونه برداری از صوت

در این بخش تمرین بایستی طبق مراحل زیر و به صورت گام به گام، جلو رفته و خواسته هر قسمت را انجام دهید. فایل ورودی این قسمت از تمرین Speech.wav می باشد که با نرخ 32000 نمونه در ثانیه ضبط شده است و ضمیمه تمرین می باشد. در تمام این بخش می توانید با استفاده از تابع wavread() فایل صوتی را بخوانید و با استفاده از تابع wavplay() آن را پخش کنید. ابتدا فایل صوتی تمرین را بخوانید و waveform آن را در یک متغیر متلب به نام x32 بارگذاری کنید. سپس بخش های زیر را به ترتیب انجام دهید. فراموش نکنید که خروجی های ذکر شده در مقدمه تمرین را برای تمامی قسمت های این بخش بایستی ارسال نمایید.

۱. سیگنال x32 را به عنوان تابعی از زمان رسم نمایید. محور اندازه در بازه [1.3, -1.3] و محور زمان با مقیاس ثانیه مدرج شده باشد. ۲. طیف فرکانسی سیگنال x32 را به صورت تخمینی و نرمالایز شده در بازه فرکانسی [0, Fs] با مقیاس Hz و محور اندازه در بازه [0, 0.008] رسم نمایید.

۳. حال سیگنال x32 را با فاکتور ۴ نمونه برداری کاهشی کنید و آن را در x8 قرار دهید و قسمت ۱ و ۲ را برای آن تکرار کنید. به صدای سیگنال s8 و s32 گوش کنید و این دو را با هم مقایسه نمایید. راهنمایی: برای نمونه برداری کاهشی گفته شده می توانید x8 را با انتخاب ۳ در میان از نمونه های x32 بسازید. $s8 = s32(1:4:\text{length}(s32))$

۴. قبل از نمونه برداری کاهشی استفاده از یک فیلتر FIR خطی می تواند مناسب باشد. علت را توضیح دهید. در این قسمت از تابع fir1 متلب برای طراحی این فیلتر به صورت زیر استفاده می کنیم. با این تابع در راهنمای متلب آشنا شوید. $\text{fir1}(64, 1/4)$ ۵. سیگنال x32f را با فیلتر کردن سیگنال x32 و استفاده از فیلتر قسمت قبل بسازید. راهنمایی: می توانید از دستور filter متلب استفاده نمایید.

۶. سیگنال x32f را به کمک قسمت ۲ نمونه برداری کاهشی با فاکتور ۴ نمایید و آن را در x8f قرار دهید و قسمت ۱ و ۲ را برای آن تکرار کنید. ۷. سیگنال x8 و x8f را در متلب پخش کنید و تفاوت این دو را با ذکر علت مشخص کنید.

۸. سیگنال x8f را با فاکتور ۴ نمونه برداری افزایشی کرده و آن را x8f32 بنامید و قسمت ۱ را برای آن تکرار کنید. سپس آن را پخش نموده و آن را با سیگنال اصلی ابتدایی مقایسه نمایید. انتظار می رود این دو مشابه باشند. این انتظار برآورده شده است؟ راهنمایی: $s8f32(4*(1:\text{length}(x8f))) = s8f(1:\text{length}(s8f))$

۹. برای بالاتر بردن کیفیت سیگنال x8f32 می توانیم از فیلتر پایین گذر طراحی شده در قسمت ۴ استفاده نماییم. سیگنال x8f32 را فیلتر نموده و آن را x8f32f نامیده و قسمت ۱ و ۲ را برای آن تکرار نمایید. سپس آن را پخش نموده و بهبود کیفیت آن را احساس کنید.

۱۰. در این قسمت می خواهیم با پیاده سازی CIC Moving Average Filter یک فیلتر پایین گذر ساده طراحی کنیم. این کار در متلب بدین صورت انجام می شود:

$\text{Acic} = [-1, 1]$, $\text{Bcic} = [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1]/8$; $\text{freqz}()$ سپس پاسخ فرکانسی این فیلتر و فیلتر طراحی شده در قسمت ۴ را با دستور رسم نمایید و مقایسه نتیجه این دو فیلتر با هم را بیان کنید.

۱۱. سیگنال x8f32cic را با اعمال فیلتر CIC بر سیگنال x8f32 ایجاد کنید. حال x8f32 را در ۴ ضرب کنید تا اندازه سیگنال به حالت اولیه برگردد و قسمت ۱ و ۲ را برای آن تکرار نمایید. حال آن را پخش نموده و بهبود کیفیت را احساس کنید. ۱۲. (امتیازی) در مورد پدیده انعکاس طیفی^۱ و علت آن جست و جو کنید و ارتباطش با قسمت قبل تمرین را توضیح دهید.

¹Replication Spectral

۲.۲ کوانتایزر غیرخطی

همانطور که می‌دانید برای اعمال کوانتیزاسیون غیرخطی از دو روش استفاده می‌شود. ابتدا فرمول زیر که مربوط به روش μ -law می‌باشد را به صورت یک تابع در متلب پیاده‌سازی نمایید. (مقدار $\mu=255$ و بازه $x=[-1,1]$)

۱. فایل ضمیمه همراه تمرین با نام speech.wav که با ۱۶ بیت کوانتایز شده است را در متلب بارگذاری نموده و تابع نوشته شده بالا را بر روی آن اعمال نمایید.

– راهنمایی ۱: برای بارگذاری فایل صوتی در متلب از دستور wavread با پارامتر native استفاده کنید. جزئیات این دستور را در راهنمای متلب دنبال کنید.

– راهنمایی ۲: برای بردن محدوده ورودی به بازه $[-1, 1]$ و اعمال تابع نوشته‌شده‌تان می‌توانید مقادیر ورودی را به ۳۲۷۶۸ تقسیم کنید و سپس تابعتان را اعمال نمایید. همچنین مقادیر خروجی را در ۱۲۷ ضرب کنید تا خروجی در محدوده ۸ بیت قرار گیرد.

۲. پس از انجام این کار به فایل اصلی و سیگنال خروجی گوش دهید. آیا تفاوتی حس می‌کنید؟ دلیل آن چیست؟

۳. نمودار این دو سیگنال (اصلی و کوانتایز شده) را در حوزه زمان رسم نمایید. در مورد تفاوت این دو نیز مختصر توضیح دهید.

۴. حال در این قسمت بایستی عکس تابع نوشته شده خود را نیز پیاده‌سازی نموده و آن را به فایل کوانتایز شده خود اعمال کنید. انتظار این است که پس از انجام این کار همان فایل اصلی به دست آید. آیا این انتظار تحقق می‌یابد؟ دلیل آن را تشریح نمایید.

۵. نمودار سیگنال اصلی و سیگنال بازیابی شده پس از دی‌کوانتایز را در حوزه زمان رسم نمایید. در مورد تفاوت این دو نیز مختصر توضیح دهید.

۶. نمودار خطای عمل کوانتایز نسبت به فایل اصلی و همچنین نمودار خطای عمل دی‌کوانتایز نسبت به فایل اصلی را در حوزه زمان رسم کنید.

۳.۲ کد کردن پیشگویانه

همانطور که می دانید در یک شکل موج صوت معمولی مثلاً صحبت و یا حتی آهنگ‌های ملایم نمونه‌های متوالی زیادی دارای مقادیر مشابهی هستند. یک راه برای فشرده‌سازی بیشتر استفاده از این تشابه‌ها و استفاده از کد کردن به روش پیشگویی خطی است. در این روش در قسمت کدکننده^۲ در ابتدا نمونه فعلی از روی ترکیب خطی نمونه‌های قبلی ساخته شده تخمین زده شده و سپس مقدار خطای بین نمونه اصلی و نمونه پیش‌بینی شده کوانتایز شده و کد می‌شود. در قسمت کدگشا^۳ نیز ابتدا همان مقدار پیش‌بینی شده از روی نمونه‌های قبلی کدگشایی شده ساخته می‌شود. سپس مقدار خطایی که در سمت کدکننده کوانتایز شده است به آن اضافه می‌گردد. در این قسمت می‌خواهیم سناریوی ذکر شده فوق را در متلب پیاده نموده و همان فایل قسمت ج را این بار با این روش مورد آزمایش قرار دهیم. (inputFile همان فایل speech.wav می‌باشد). ۱. ابتدا یک تابع بنویسید که یک فایل صوتی و تعداد بیت‌های خروجی را به عنوان ورودی دریافت کرده و دو بردار تفاضل نمونه‌ها و بردار خطا را باز می‌گرداند.

function [differ, err] = mydpcm(inputFile, n)

تابع شما باید سه کار زیر را به ترتیب انجام دهد:

– تفاوت هر دو نمونه متوالی از سیگنال ورودی را با کمک تابع diff متلب محاسبه نماید.

– این تفاوت‌ها را در محدوده $[-2n-1, 2n-1-1]$ کوانتایز خطی نمایید و خروجی آن را در برداری به نام differ (یکی از خروجی‌های این تابع) ذخیره نمایید.

– یک بردار با نام err ایجاد کنید که تفاوت هر درایه differ را با مقدار آن پیش از کوانتایز کردن نگه دارد.

۲. تابع mydpcm که عکس تابع فوق است را نیز پیاده‌سازی نمایید. این تابع، اولین نمونه سیگنال اصلی را به همراه بردار differ گرفته و سیگنال اولیه را بازسازی می‌نماید.

– راهنمایی: به منظور سادگی کار، فرض کنید که نمونه فعلی فقط از روی یک نمونه قبلی ساخته می‌شود.

۳. با قرار دادن مقدار ۴ برای n تفاوت سیگنال بازسازی شده و سیگنال اولیه را محاسبه نمایید و با بردار err مقایسه نمایید.

۴. به سیگنال اصلی و بازسازی شده گوش دهید. چه تفاوتی بین آن‌ها وجود دارد؟

مقدار n را به ۸ و ۲ تغییر داده و دوباره به سیگنال‌ها گوش دهید. آیا تغییر مقدار n روی کیفیت سیگنال بازسازی شده تأثیر قابل توجهی دارد؟ علت چیست؟

موفق باشید

موضوع تمرین بعد: پردازش صوت

²Encoder

³Decoder