

بسم الله الرحمن الرحيم



پروژه نهایی MATLAB سیگنال و سیستم
نیم سال دوم سال تحصیلی ۱۴۰۰-۰۱

توجه!!

- پاسخ های خود را در قالب یک فایل zip که نام آن فقط **شماره دانشجویی** تان است ارسال کنید.
- کد خود را جهت توضیح مختصر دستورات مورد استفاده، **کامنت گذاری** کنید و در نهایت به کمک امکان Publish در متلب، گزارش کار خود را همراه با پاسخ سوالات تکمیل کرده و در پوشه نهایی قرار دهید.
- رعایت تمام موارد بالا برای کسب نمره الزامی است.

مهلت تحویل : پنجشنبه ۹ تیر ماه ۱۴۰۱

۱.

متلب دارای مجموعه گفتاری از یک نفر است که می گوید ” متلب ” . ابتدا با استفاده از دستور load mtlb، سیگنال صدا را دریافت کنید. همچنین می توانید با استفاده از تابع soundsc صدای آن را پخش کنید. این مجموعه داده با فرکانس نمونه برداری $F_s = 7418Hz$ بدست آمده است.

(۱) سیگنال صوت را رسم کنید.

(۲) طیف این سیگنال را با کمک دستور dtft در متلب رسم کنید. کدام فرکانس ها غالب هستند ؟

(۳) به وسیله تابع rand به این سیگنال نویز اضافه کنید و آن را رسم کنید.

(۴) تعیین کنید که کدام بخش از این طیف ، مربوط به نویز است ؟

(۵) توضیح دهید که چرا یک فیلتر پایین گذر می تواند بخش خوبی از نویز را حذف کند ؟ فرکانس قطع این فیلتر را با توجه به نمودار بخش قبلی حدس بزنید.

(۶) به وسیله دستور elip یک فیلتر elliptical پایین گذر طراحی کنید. برای این کار لازم است که :

مقادیر مختلفی برای پارامترهای cut-off frequency ، passband ripple و stopband ripple در نظر بگیرید.

آیا می توانید اغلب نویز را از سیگنال نویزی حذف کنید ؟

(۷) حذف نویز بیشتر تحت تاثیر مقدار پارامتر passband ripple است یا مقدار پارامتر stopband ripple ؟ (

کدام مورد مهم تر است ؟ اینکه passband خیلی نزدیک به ۱ باشد یا اینکه stopband خیلی نزدیک به ۰ باشد ؟ طراحی های مختلف برای فیلتر را امتحان کنید و با توجه به نتایج بدست آمده ، نتایج را تحلیل کنید.)

۲.

در این بخش قصد داریم تا با مفهوم نمونه برداری و قضیه نایکویست آشنا شویم.

(۱) برای این بررسی سیگنال سینوسی با فرکانس 5° هرتز را در بازه زمان 0.1 ثانیه با فاصله های یک میلی ثانیه رسم کنید، این سیگنال به عنوان تقریب سیگنال پیوسته است.

(۲) سپس این سیگنال را با فرکانس 20° هرتز نمونه برداری کرده و نمونه ها را به صورت نقطه روی سیگنال اولیه نشان دهید.

(۳) با استفاده از دستور `pwelch` چگالی طیف توان سیگنال را که در قسمت ۲ نمونه برداری کردید رسم کنید تا فرکانس هایی که سیگنال در آن ها توان دارد را مشاهده کنید.

(۴) این بار سیگنال بخش یک را با فرکانس 4° هرتز نمونه برداری کنید و نمونه ها را به صورت نقطه روی سیگنال اولیه نمایش دهید.

(۵) با استفاده از دستور `pwelch` چگالی طیف توان سیگنال را دوباره ترسیم کنید و مانند گذشته، فرکانس هایی که سیگنال در آن ها توان دارد را مشخص کنید.

(۶) فرکانس های دارای توان را در بخش ۳ و ۵ را با یک دیگر مقایسه کنید.

(۷) برای نمونه برداری سیگنال ها در مرحله ی اول، رابطه زیر را تشکیل می دهند

$$x(t)_s = x(t) * \sum_{n=-\infty}^{n=\infty} \delta(t - nT)$$

رابطه ی تبدیل فوریه $x(t)_s$ را براساس تبدیل فوریه $x(t)$ بدست آورده و شرط آن را که طیف سیگنال $x(t)$ تحت هرشرایطی از روی طیف $x(t)_s$ با اعمال فیلتر، قابل حصول باشد را بیان کنید. (در اثبات فرض کنید که پهنای باند سیگنال $x(t)$ محدود است.)

(۸) ابتدا یک سیگنال سینوسی با فرکانس 20° هرتز در بازه ی زمان مشابه قسمت های پیشین رسم کنید. سپس آن را در یک سیگنال سینوسی با فرکانس 15° هرتز ضرب کنید. با دستور `pwelch` طیف توان سیگنال ضرب شده را بیابید.

باتوجه به نتایج، روشی را پیشنهاد دهید که بتوانیم یک سیگنال را باشرطی به غیر از شرط نایکویست که در بخش ۸ بدست آوردید، نمونه برداری کنیم.

(۹) مابین هریک از نمونه های سیگنال بخش قبل، ۴ صفر قرار داده و با رسم طیف توان، نتیجه را تحلیل کنید.

از مقایسه ی نتیجه با شکل حالت پیشین چه نتیجه ای می گیرید؟

(۱۰) باتوجه به مشاهده ی خود توضیح دهید که چگونه می توان نرخ نمونه برداری یک سیگنال را افزایش داد.

- (۱۱) از هر ۵ نمونه سیگنال بخش اول یکی را انتخاب کنید. (نمونه های ۰ و ۵ و ۱۰ و ...) سپس طیف این سیگنال را بکشید. مشاهده خود را با شکل حالت های قبلی مقایسه کرده و توضیح دهید.
- (۱۲) حال از هر ۱۰ نمونه یکی را انتخاب کنید و شکل را رسم کنید. آیا مشکلی پیش آمد؟ توضیح دهید.
- (۱۳) در صورت ایجاد مشکل در بخش پیشین راهی را پیشنهاد دهید تا بتوانیم بهبودی در نتیجه حاصل کنیم.
- (۱۴) متلب دارای دستوری به نام `resample` می باشد. این دستور می تواند نرخ نمونه برداری یک سیگنال را به یک عدد گویا تغییر دهد.
- با توجه به عملیات هایی که در بخش های ۹ و ۱۰ انجام دادید بیان کنید که این دستور چگونه عمل تغییر فرکانس را انجام می دهد.
- (۱۵) نرخ نمونه برداری سیگنال تعریف شده در بخش اول خود را به کمک دستور `resample` به مقدار $1.5kHz$ تغییر دهید.

۳. نویزدایی تصویر (Image Denoising) :

در این بخش با دو فیلتر دیجیتال مهم یعنی میانگین (mean) و میانه (median) آشنا می شوید. این فیلترها برای کاهش انواع خاصی از نویز در تصویر استفاده میشوند. ایده فیلتر میانگین عوض کردن مقدار هر پیکسل تصویر با میانگین مقادیر همسایگی شامل خود آن است. تاثیر این کار حذف مقادیر پیکسل هایی است که در اطراف وجود ندارند. فیلترینگ میانگین معمولا به عنوان فیلتر کانولوشن در نظر گرفته میشود. مانند دیگر کانولوشن ها این فیلتر بر مبنای یک هسته (kernel) می باشد که شکل و سایز همسایگی را هنگام محاسبه میانگین انتخاب می شود را مشخص می کند. همان طور که در شکل زیر نشان داده شده است معمولا از یک هسته مربعی 3×3 استفاده می شود، اگرچه در هموار سازی شدید هسته های بزرگتر مثل 5×5 نیز استفاده می شوند.

$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$

الف) تصویر با نام 'Image.jpg' را بارگذاری کنید.

ب) یک نویز گوسی با میانگین صفر و انحراف استاندارد برابر 20° را به آن اضافه کنید. می توانید از دستور متلب imnoise استفاده کنید. نتیجه را نمایش دهید.

ج) تابعی برای پیاده سازی فیلتر میانگین بنویسید. تابع شما باید تصویر اصلی و سایز پنجره را به عنوان ورودی بگیرد و تصویر اصلاح شده را به عنوان خروجی بدهد.

د) فیلتر میانگین 3×3 را اعمال کرده و خروجی را نمایش دهید. در مورد نتیجه بحث کنید.

ه) فیلتر میانگین 5×5 را روی آن اعمال کنید و تصویر منتجه را نمایش دهید. تفاوت بین این قسمت و قسمت قبل چیست؟ آیا یک بده بستان دارید؟ اگر پاسخ مثبت است، آن چیست؟

و) تصویر اصلی را در نظر بگیرید. این بار یک نویز نمک و فلفل با $p=10\%$ درصد با استفاده از دستور imnoise به آن اضافه کنید و تصویر خراب را نمایش دهید.

ز) یک فیلتر میانگین 3×3 به این تصویر خراب اعمال کرده و نتیجه را نمایش دهید. آیا این فیلتر برای کاهش نویز نمک و فلفل مفید است؟ چرا؟

همانطور که در بخش قبل دیدید فیلترینگ میانگین در کاهش نویز نمک و فلفل فایده ای ندارد، بنابراین ما از یک فیلتر دیگر استفاده می کنیم، فیلتر میانه. این فیلتر مانند فیلتر میانگین هر پیکسل تصویر را به نوبت در نظر میگیرد و به همسایگی اطراف آن نگاه میکند تا تصمیم بگیرد که آیا آن نماینده اطرافش هست یا نه. به جای تعویض

مقدار پیکسل با میانگین مقادیر پیکسل های همسایگی، آن را با مقدار میانه عوض میکند. میانه با ابتدا مرتب کردن عددی همه ی مقادیر پیکسل های همسایگی اطراف و سپس تعویض پیکسل مورد نظر با مقدار پیکسل میانی محاسبه میشود (در صورتی که همسایگی مورد نظر تعداد زوجی پیکسل داشته باشد، از میانگین دو عدد میانی استفاده میشود). شکل زیر یک محاسبه ی نمونه را نشان می دهد :

123	125	126	130	140
122	124	126	127	135
118	120	150	125	134
119	115	119	123	133
111	116	110	120	130

مقادیر همسایگی:

۱۱۵، ۱۱۹، ۱۲۰، ۱۲۳، ۱۲۴،

۱۲۵، ۱۲۶، ۱۲۷، ۱۵۰،

مقدار میانه: ۱۲۴

ح) تابعی برای پیاده سازی میانه بنویسید. تابع شما باید تصویر اصلی و سایز پنجره که طول و عرض آن فرد می باشد را به عنوان ورودی بگیرد و تصویر اصلاح شده را به عنوان خروجی بدهد و همچنین در صورتی که طول و یا عرض وارد شده عدد فردی نبود پیغام خطا ظاهر شود.

ط) فیلتر خود را با سایز ۳ به تصویر نویزی قسمت ”و” اعمال کنید. نتیجه را نمایش دهید. چگونه این فیلتر می تواند به شما کمک کند تا بر نویز نمک و فلفل غلبه کنید ؟

ک) اثر blurring فیلترهای میانه و میانگین را مقایسه کنید. آیا فیلتر میانه بدی هایی نیز دارد ؟