

بسمه تعالی

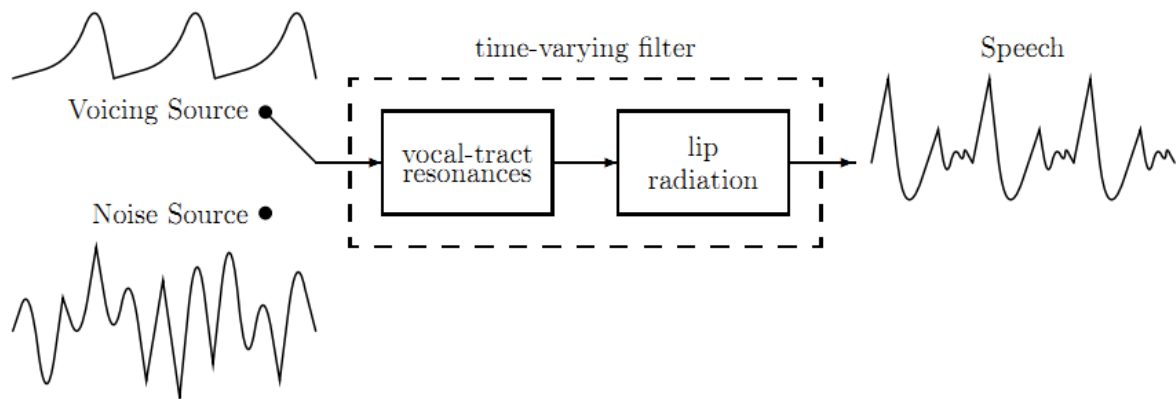
هدف از این تمرین آشنایی با یکی از سیستم های Analysis-Synthesis به نام Vocoder است.

## سیگنال صحبت

سیگنال صحبت ناشی از تغییرات فشار اکوستیک در فاصله ی مشخصی از لب های گوینده است که اندازه این فشار می تواند از 30dB ( نجوا کردن) تا 100dB (فریاد زدن) باشد. تقریبا اکثر انرژی سیگنال صحبت در فرکانس های پایین تر از 4 یا 5 کیلو هرتز متمرکز شده است به همین دلیل پهنای باند سیگنال صحبت حداکثر 4 یا 5 کیلوهرتز در نظر گرفته می شود.

## مدل سیگنال صحبت

سیگنال صحبت نتیجه عبور سیگنال صوت تولید شده بر اثر لرزش تار های صوتی است که از مسیر تنفسی بین تار های صوتی تا دهان عبور می کند. در بازه های زمانی کوتاه این سیستم را می توان با یک منبع ورودی (source) و یک فیلتر خطی متغیر با زمان (مدل کننده اثر مسیر تنفسی و لرزش لب ها بر سیگنال تولیدی تار های صوتی) مدل کرد (شکل 1).  
منبع ورودی با توجه به نوع حرفی که گوینده می گوید به دو نوع voiced source ( برای حروف صدا دار و بعضی حروف بی صدا) و unvoiced source تقسیم می شود. Voiced source با سیگنالی پرپودیک (شبیه به قطار ضربه) که مدل کننده لرزش تار های صوتی است مدل می شود. سیگنال نویز هم مدل کننده unvoiced source است  
فیلتر متغیر با زمان ، مدل کننده اثر مسیر عبور سیگنال صوت از تار های صوتی تا دهان و اثر لرزش لب ها بر روی صوت خروجی است.



شکل 1

هارمونیک اول voicing source ، pitch نام دارد. که در واقع شنوندگان صحبت، این فرکانس را می شنوند. به همین دلیل این فرکانس از اهمیت ویژه ای برخوردار است. برای مردان این فرکانس بین 80 تا 160 هرتز و برای زنان بین 160 تا 320 هرتز است.

## مقدمه ای بر vocoder ها

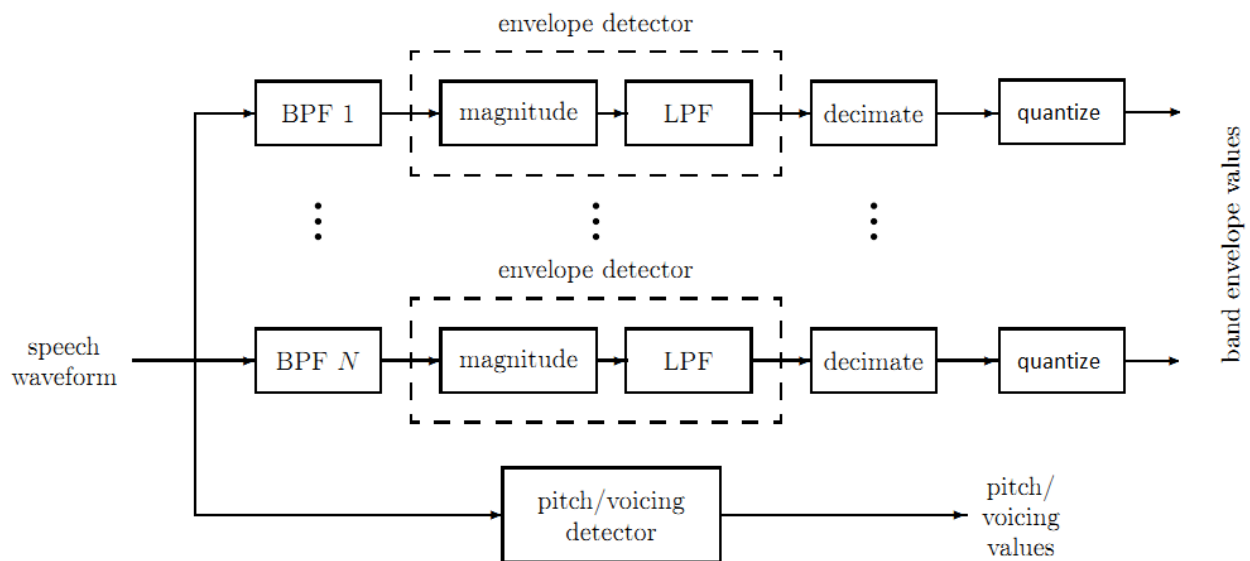
Vocoder ها دستگاه هایی هستند که برای ارسال و یا ذخیره کارآمد (efficient) سیگنال صحبت و سیگنال های صوتی استفاده می شوند. معیار ارسال کارآمد تعداد بیت در ثانیه است که برای مشخص کردن سیگنال مورد نظر لازم است. به طور مثال یک سیگنال صوتی با کیفیت، به طور معمول با نرخ 48KHz نمونه برداری می شود و با 16 بیت کوانتیزه می شود. در نتیجه برای ارسال سیگنال دو کاناله (استریو) نیاز به نرخ بیت 1.5M bit/s داریم و یا سیگنال صحبت تلفن که با نرخ 8KHz نمونه برداری می شود و با 10 بیت کوانتیزه می شود نیاز به نرخ ارسال بیت 80K bit/s دارد. Vocoder ها فقط با اندکی کاهش در کیفیت سیگنال صحبت این نرخ را به طور قابل ملاحظه ای کاهش می دهند (در حد 1200-9600 bit/s). یک vocoder از 3 قسمت اصلی تشکیل شده است:

(1) analysis stage : اطلاعات و پارامتر های سیگنال صحبت را استخراج کرده و کد می کند.

(2) Transmission channel : کانال ارتباطی بین فرستنده و گیرنده

(3) synthesis stage : کدگذاری اطلاعات و پارامتر های دریافتی و ساختن سیگنال صحبت

شمای کلی analysis stage به صورت زیر است :

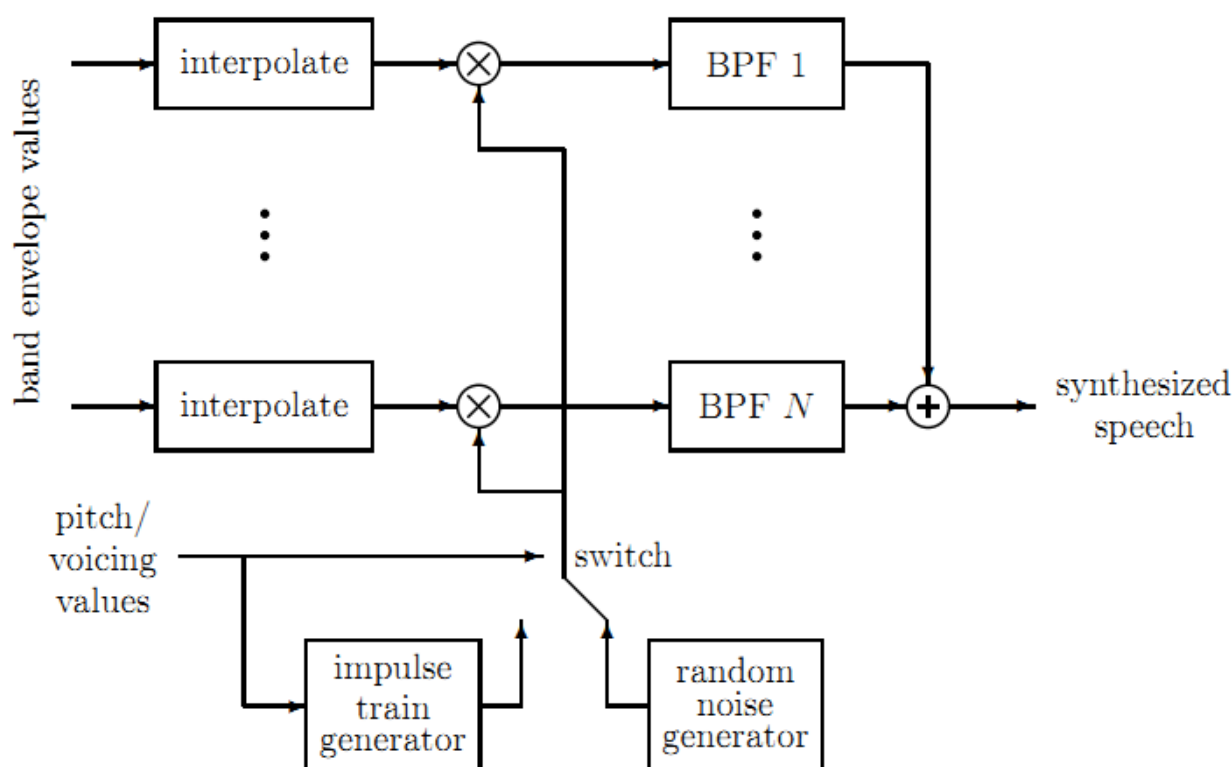


شکل 2

ابتدا سیگنال ورودی توسط N فیلتر میان گذر به N مولفه فرکانسی که همپوشانی با هم ندارند، تجزیه شده و اندازه هر مولفه محاسبه می شود (توجه کنید که گوش به فاز سیگنال حساس نیست پس این عمل مشکلی بوجود نمی آورد). سپس هر مولفه از یک فیلتر پایین گذر عبور کرده ، با نرخی down sample (در بلوک decimate) و کوانتیزه می شود (در بلوک quantize). در نهایت سیگنال های خروجی به گیرنده فرستاده می شود. گیرنده توسط این N مولفه اطلاعات مربوط به فیلتر مدل سیگنال صحبت را بازسازی می کند.

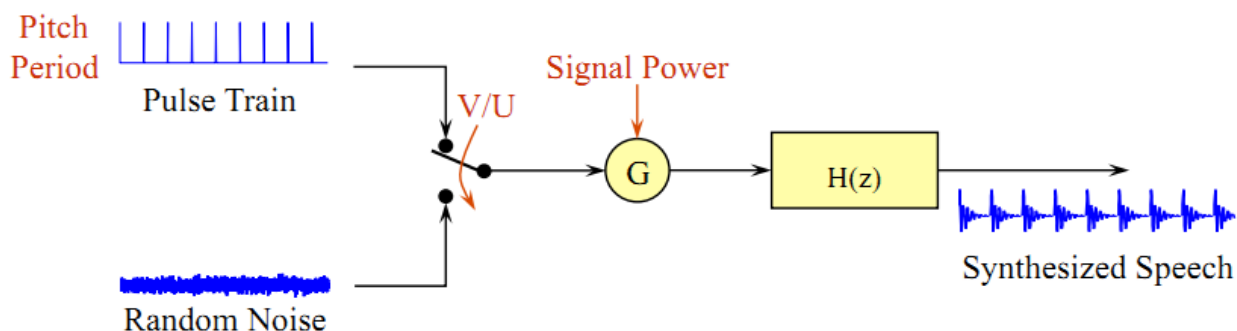
همان طور که در قسمت قبل توضیح داده شد، یکی از اطلاعات مهم که برای بازسازی سیگنال منبع (source) در سمت گیرنده لازم است، pitch است. قسمت pitch/voicing detector به این صورت عمل می کند که سیگنال ورودی را به بازه های زمانی کوچک (معمولاً 30ms) تقسیم کرده و برای هر قسمت مشخص می کند که voiced source است یا unvoiced source. در حالت voiced source مقدار pitch (هارمونیک اول آن) به گیرنده ارسال می شود (در غیر این صورت صفر فرستاده می شود). نحوه تشخیص voiced یا unvoiced بودن سیگنال به طور مختصر در توضیحات ابتدای فایل "pitch\_detect.m" بیان شده است.

شمای کلی synthesis stage به صورت زیر است :



شکل 3

برای بازسازی سیگنال صحبت عکس روند vocoding صورت می گیرد. ابتدا با interpolate کردن، اثر بلوک decimate خنثی می شود. در قسمت synthesize با توجه به مدل های آماری، سیگنالی مشابه صحبت اولیه تولید می شود. در قسمت پایین شکل 3 مشاهده می شود که با توجه به صفر و ناصفر بودن سیگنال pitch که از فرستنده دریافت شده است، ورودی ضرب کننده ها به صورت نویز سفید و یا قطار ضربه اختیار می شود. با ضرب شدن N مولفه دریافت شده، هر قسمت از سیگنال source در اندازه (amplitude) مناسب خود ضرب می شود (در باند فرکانسی مربوطه). در انتها سیگنال صحبت با عبور مولفه ها از یک فیلتر بانک گذر و محاسبه حاصل جمع بدست می آید. مدل دیگری از قسمت synthesize در شکل 4 آمده است.



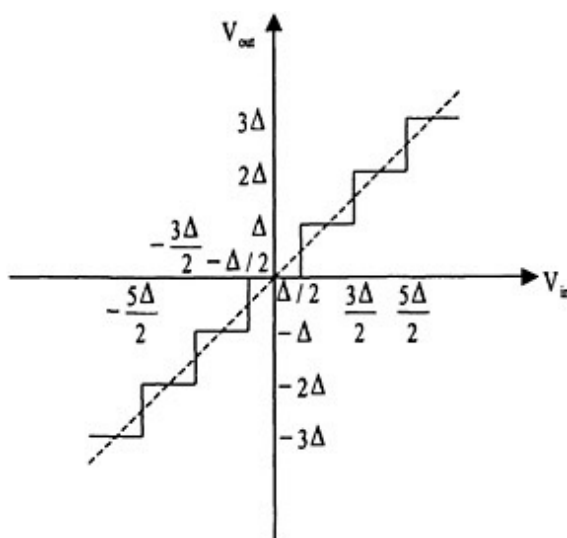
شکل 4

## نکات :

- قسمت های زیادی از کد MATLAB به صورت آماده در اختیار شما قرار گرفته است. وظیفه اصلی شما طراحی فیلتر بانک میان گذر و بلوک quantize است.
- دو فایل سیگنال صحبت (cw161\_8k,cw162\_8k) در پوشه "input voice" قرار دارد. با دستور load می توانید این دو فایل را در MATLAB باز کرده و از آن ها استفاده کنید. هم چنین با دستور soundsc می توانید سیگنال صوتی خود را پخش کنید (هر دو سیگنال با نرخ 8KHz نمونه برداری شده اند).
- در پوشه "function" دو تابع به نام chvocod\_syn.m و chvocod\_ana.m قرار دارد که به ترتیب پیاده کننده قسمت analyzer و synthesizer هستند. توضیحات مربوط به ورودی های توابع و نحوه عملکرد آن ها در خود توابع آمده است.
- با اجرای main.m یکی از دو فایل صحبت با توجه به مقدار متغیر speech\_selector انتخاب شده و با استفاده از پارامتر های اولیه ای که در کد موجود است، سیگنال صحبت را کد کرده و پاسخ فرکانسی بانک فیلتری با توجه به نوع پنجره و تعداد باند آن به طور خودکار در پوشه "filterbank figures" و در یکی از زیر پوشه های rectangular,Kaiser,P\_M ذخیره می شود (توجه شود که یکی از قسمت هایی که باید توسط شما نوشته شود طراحی بانک فیلتری توسط Parks-McClellan است و پوشه P\_M به همین دلیل نامگذاری شده است).
- سیگنال بازسازی شده بعد از هر بار اجرای main.m پخش شده و نمودار آن در حوزه فرکانس در کنار نمودار حوزه زمان سیگنال اولیه کشیده می شود.
- سیگنال باز سازی شده به صورت خودکار در پوشه "output voice" با نامی که توسط متغیر outputName مشخص شده است ذخیره می شود.
- تنها قسمتی از کد main.m که لازم است تغییر دهید، پارامتر های اولیه مربوط به vocoder و متغیر outputName است که محل آن در کد مشخص شده است.
- دو پوشه "filterbank figures" و "output voice"، به همراه دو تابع تکمیل شده خود (myownfilt\_bank.m و Quantization.m) و گزارش ۱ صفحه ای که شامل پاسخ مختصر پرسش ها است را در قالب یک فایل زیپ با نام FD\_HW\_StuNumber در سامانه CW آپلود کنید.

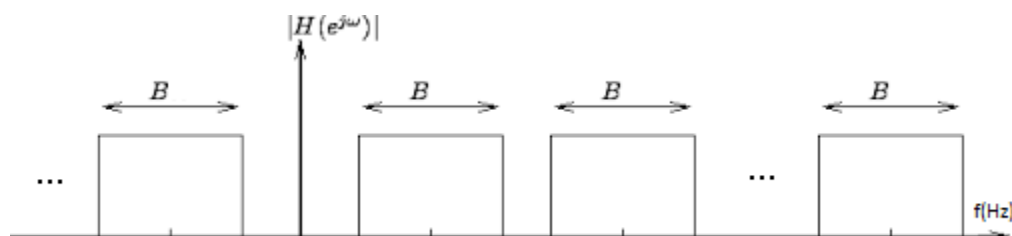
## پرسش ها:

1) تابعی با فرمت  $\text{output} = \text{Quantization}(\text{input}, B)$  در پوشه "functions" قرار دارد. این تابع وظیفه کوانتیزه کردن سیگنال ورودی خود را بر عهده دارد. کد مربوط به کوانتیزه کردن را در مکانی که درون تابع مشخص شده است، بنویسید. هدف شما پیاده سازی تابعی است که در شکل زیر نشان داده شده است و به این صورت که  $\Delta = \frac{1}{2B}$ . (فرض کنید که تابع کوانتیزه کننده اشباع نمی شود)



2) در پوشه "function" تابعی به نام "myownfilt\_bank.m" وجود دارد. در مکان مشخص شده دستورهای لازم برای پیاده سازی یک فیلتر بانک میان گذر  $N$  شاخه (بدون همپوشانی) با پهنای باند  $B$  و طول  $L$  را بنویسید. در قسمت اول، به این صورت باید عمل کنید که یک فیلتر پایین گذر تولید کرده سپس با شیفت دادن آن، فیلترهای میان گذر مختلف را تولید کنید. خروجی تابع شما یک ماتریس  $L \times N$  است که هر ستون آن یک فیلتر میان گذر  $L$  نقطه ای است. توضیحات مربوط به پارامترهای ورودی تابع در ابتدای فایل مربوطه آمده است. کد شما در قسمت اول باید حاوی طراحی فیلتر پایین گذر توسط 2 نوع پنجره Kaiser و Rectangular باشد که محل آن در کد مشخص شده است.

پاسخ فرکانسی فیلتر بانک شما باید به صورت زیر باشد:



\*\* می توانید از توابعی مانند fir1 برای طراحی فیلتر استفاده کنید.

**\*\* توجه کنید** که شما باید در این تابع بردار `freqAxis` را برای رسم درست پاسخ فرکانسی فیلتر بانک، به طور مناسب مقدار دهی کنید) محور افقی نمودار باید بر حسب هرتز باشد.

در قسمت دوم، شما می بایست بانک فیلتری را توسط روش Parks-McClellan طراحی کنید. محل نوشتن کد این قسمت هم در کد مربوط به تابع مشخص شده است.

**\*\* می توانید از توابعی مانند `firpm` برای طراحی فیلتر این قسمت استفاده کنید.**

**\*\* توجه کنید** که با تنظیم مقدار مناسب برای متغیر `is_PM` در کد می توانید انتخاب کنید که بانک فیلتری توسط روش اول یا روش دوم (Parks-McClellan) ساخته شود.

۳) حال که تابع های مورد نیاز تکمیل شد، دوسیگنال صحبت موجود در پشه "input voice" را توسط `vocoder` بازسازی کنید. تعداد باند های فیلتر بانک را 18، طول فیلتر ها را 65، نرخ `decimate` را 100 و تعداد بیت های کوانتیزاسیون را 16 قرار دهید. `outputName` را 'firstTest' قرار دهید.

این کار را برای دو نوع پنجره `Kaiser(β=3)` و `Rectangular` و هم چنین برای روش Parks-McClellan انجام دهید.

کیفیت سیگنال بازسازی شده توسط دو روش ( `window based` و `parks-mcClellan` ) را مقایسه کنید.

**\*\* توجه کنید** که برای ذخیره سازی سیگنال های بازسازی متفاوت باید محتوای متغیر `outputName` را تغییر دهید. به طور مثال می توانید از نام های 'firstTest\_PM' و 'firstTest\_rectangular' و 'firstTest\_Kaiser' استفاده کنید.

۴) در قسمت `analysis`، فیلتر `magnitude` چه اثری بر روی سیگنال ورودی خود در حوزه فرکانس می گذارد؟ با رسم نمودار پاسخ خود را شرح دهید.

۵) نرخ کوانتیزاسیون خود را به ۲ بیت کاهش دهید و بانک فیلتری خود را به صورت زیر تشکیل دهید:

طول فیلتر 61، تعداد باند ها 18 و نرخ `decimate` برابر 100.

متغیر `outputName` را با عبارت 'quantizeTest\_' شروع کرده و سیگنال صوتی را توسط دو پنجره `kaiser,rectangular` و روش Parks-McClellan باز سازی کنید.

۶) تعداد باند های بانک فیلتری را دو برابر کنید. متغیر `outputName` را با "bandNumTest\_" شروع کرده و با تعداد بیت کوانتیزاسیون 16 ( بقیه پارامتر ها مانند حالت قبل) سیگنال های صوتی را توسط دو پنجره `kaiser,rectangular` و روش Parks-McClellan را بازسازی کنید.

۷) آیا SNR معیار مناسبی برا کیفیت بازسازی سیگنال صوتی است؟ اگر خیر چه معیاری پیشنهاد می کنید؟