



دانشگاه صنعتی شریف

دانشکده مهندسی برق

درس پردازش سیگنال گسسته

پروژه پایانی : فشرده سازی صوت

استاد درس : دکتر کرباسی

تهیه کننده : علیرضا رفیعی ساردوئی

97101723

1- کپستروم یک سیگنال به دو صورت حقیقی و مختلط می‌باشد. در کپستروم مختلط به صورت زیر کپستروم سیگنال را پیدا می‌کنیم:

$$C_C = \text{Complex-cepstrum}(x(t)) \rightarrow C_C = \mathcal{F}^{-1}(\log(\mathcal{F}(x(t))))$$

اگر بخواهیم به فرم زیباتری کپستروم مختلط را نشان دهیم به صورت زیر عمل می‌کنیم:

$$x(t) \xrightarrow{\mathcal{F}} X(f) = |X(f)|e^{j\angle X(f)} \xrightarrow{\log} \log(|X(f)|) + j\angle X(f) \rightarrow C_C = \mathcal{F}^{-1}(\log(|X(f)|) + j\angle X(f))$$

در بدست آوردن کپستروم لزومی ندارد که لگاریتم در مبنای طبیعی گرفته شود اما اگر در مبنای طبیعی کپستروم بگیریم میتوانیم مطابق بالا رابطه را تا حدی ساده کنیم.

کپستروم حقیقی به صورت زیر حساب می‌شود:

$$C_R = \text{Real-cepstrum}(x(t)) \rightarrow C_R = \mathcal{F}^{-1}(\log(|\mathcal{F}(x(t))|))$$

یا به فرم زیر میتوانیم بدست آوریم:

$$x(t) \xrightarrow{\mathcal{F}} X(f) = |X(f)|e^{j\angle X(f)} \xrightarrow{\log} \log(|X(f)|) \rightarrow C_R = \mathcal{F}^{-1}(\log(|X(f)|))$$

واژه *Cepstrum* با وارون کردن چهار حرف اول *Spectrum* بدست آمده است که منظور از *Spectrum* در واقع طیف سیگنال است.

حال برای اینکه تفاوت دو کپستروم را دریابیم به روابط بالا توجه می‌کنیم. در روابط بالا برای پیدا کردن کپستروم مختلط هم اندازه و هم فاز تبدیل فوریه سیگنال در نظر گرفته می‌شود اما در کپستروم حقیقی فقط بخش اندازه در نظر گرفته می‌شود. پس چون تبدیل فوریه کپستروم حقیقی، حقیقی خالص و زوج است پس کپستروم حقیقی یک سیگنال حقیقی، حقیقی و زوج خواهد بود اما در کپستروم مختلط یک سیگنال حقیقی با توجه به اینکه بخش موهومی تبدیل فوریه فرد است و بخش حقیقی آن زوج است میتوان به حقیقی بودن نتیجه کپستروم مختلط پی برد اما درباره زوج یا فرد بودن آن نمیتوان نظری دارد.

برای اینکه رابطه بین کپستروم حقیقی و مختلط پیدا کنیم داریم:

$$\mathcal{F}(C_R) = (\log(|X(f)|))$$

$$\mathcal{F}(C_C) = (\log(|X(f)|) + j\angle X(f)) \rightarrow \mathcal{F}(C_R) = \Re\{\mathcal{F}(C_C)\}$$

پس در حوزه فرکانس، تبدیل فوریه کپستروم حقیقی بخش حقیقی کپستروم مختلط هست. برای بدست آوردن رابطه حوزه زمان دو تبدیل داریم:

$$c_R(t) = \mathcal{F}^{-1}(\Re\{\mathcal{F}(C_C)\}) = \mathcal{F}^{-1}\left(\frac{C_C(f) + C_C^*(f)}{2}\right) = \frac{c_C(t) + c_C^*(-t)}{2}$$

حال اگر فرض کنیم سیگنال ورودی حقیقی باشد داریم:

$$c_R(t) = \frac{c_C(t) + c_C(-t)}{2} = \text{Even}(c_C(t))$$

بازسازی سیگنال از روی کپستروم آن: الف - کپستروم مختلط: در این کپستروم اطلاعاتی را دور نریخته‌ایم پس بازسازی آن از روی کپستروم آن کاری ندارد و کافی است مطابق رابطه زیر پیش رویم:

$$\mathcal{F}(C_C) = (\log(|X(f)|) + j\angle X(f)) \rightarrow X(f) = |X(f)|e^{j\angle X(f)} = e^{\mathcal{F}(C_C)} \rightarrow x(t) = \mathcal{F}^{-1}(e^{\mathcal{F}(C_C)})$$

ب) برای بازسازی سیگنال از روی کپستروم حقیقی آن نمی‌توانیم به راحتی بالا اقدام کنیم زیرا در بدست آوردن این تبدیل اطلاعات مربوط به فاز را دور ریخته‌ایم پس برای بدست آوردن معکوس آن به طور کلی میسر نمی‌باشد اما با یک فرض میتوانیم معکوس آن را به طور یکتا حساب کنیم. میدانیم اگر یک سیگنال حقیقی و مینیمم فاز باشد آنگاه دامنه و فاز تبدیل فوریه آن از یک دیگر مستقل نیستند و در این حالت کپستروم مختلط آن علی و حقیقی است همچنین میدانیم کپستروم حقیقی در حوزه زمان بخش زوج کپستروم مختلط است در نتیجه با فرض علی بودن کپستروم مختلط میتوان آن را به طور یکتا از کپستروم حقیقی بدست آورد، حال با داشتن کپستروم حقیقی ابتدا با فرض مینیمم فاز بودن سیگنال اصلی، کپستروم مختلط را حساب کرده و سپس از روش گفته برای معکوس کپستروم مختلط سیگنال اصلی را بدست می‌آوریم. در حالت گسسته از روش گفته شده در کتاب اینپهایم که به صورت زیر است کپستروم مختلط را از روی کپستروم حقیقی بدست می‌آوریم:

$$\tilde{x}[n] = \begin{cases} 2\tilde{x}_e[n], & n = 1, 2, \dots, (N/2) - 1, \\ \tilde{x}_e[n], & n = 0, N/2, \\ 0, & n = (N/2) + 1, \dots, N - 1, \end{cases}$$

در رابطه داده شده x_e کپستروم حقیقی است و x کپستروم مختلط است. (بنده نیز از همین روش استفاده کردم). پس تفاوت در سیگنال بازگردانده شده از کپستروم حقیقی و مختلط به مینیمم فاز بودن آن است در کپستروم حقیقی سیگنال بازگردانده شده مینیمم فاز است اما برای کپستروم مختلط همچنین فرضی به صورت کلی درست نیست. برای اینکه کپستروم یک سیگنال متناوب چه شکلی خواهد شد ابتدا تبدیل فوریه آن را در نظر میگیریم:

$$x[n] = \sum_{k=0}^{N-1} a_k e^{jk\omega_0 n} \xrightarrow{\mathcal{F}} X(e^{j\omega}) = 2\pi \sum_{k=0}^{N-1} a_k \delta(\omega - k\omega_0)$$

تبدیل فوریه این سیگنال نیز متناوب با دوره 2π است پس اگر از تبدیل فوریه بالا لگاریتم بگیریم باز هم تابعی متناوب با دوره 2π خواهیم داشت. تابع بدست آمده تنها در نقاطی مقدار دارد که این نقاط دلتا ها موجود در تبدیل فوریه سیگنال اصلی می باشد و در بقیه نقاط مقدار نخواهیم داشت پس اگر از تبدیل فوریه یک سیگنال متناوب لگاریتم بگیریم به صورت زیر در خواهد آمد:

$$\hat{X}(e^{j\omega}) = \sum_{k=0}^{(N-1)} \log(2\pi a_k) \delta(\omega - k\omega_0)$$

تابع بالا در واقع تبدیل فوریه کپستروم مختلط است و همانطور که مشاهده می کنید این تبدیل فوریه به علت داشتن δ ، مربوط به یک تابع متناوب هست پس کپستروم مختلط هم متناوب است و با توجه به رابطه میتوان دریافت که دوره تناوب کپستروم هم همان دوره تناوب سیگنال اصلی است. تحلیل مشابه میتوان برای کپستروم حقیقی به کار برد و در این حال تبدیل فوریه به صورت زیر در می آید:

$$\hat{X}(e^{j\omega}) = \sum_{k=0}^{(N-1)} \log(2\pi |a_k|) \delta(\omega - k\omega_0)$$

به صورت خلاصه تر تبدیل کپستروم یک سیگنال متناوب، سیگنالی متناوب با دوره تناوب سیگنال اصلی است.

2- قبل از توضیح درباره گام صحبت ابتدا مدل های صحبت را بیان می کنیم:

صحبت انسان به دو مدل *Voiced*, *Unvoiced* تقسیم می شود و اگر بخواهیم این مدل ها را به روش ساده توضیح دهیم *Voiced* مربوط به زمانی است که انسان صحبت می کند و *Unvoiced* مربوط به زمانی است که سخن نمی گوید. حال اگر بخواهیم به صورت دقیق تر موضوع را بررسی کنیم داریم:

Voiced (a): در این حالت صوت انسان توسط یک قطار گوسی متناوب تولید می شود که یک سیستم تغییرپذیر با زمان را تحریک می کند البته این سیستم تغییرات کندی در زمان دارد. منظور از سیستم تغییرپذیر با زمان حرکات لب ها و دستگاه صوتی و حنجره است که میتوان آن را با یک سیستم تغییرپذیر با زمان که تغییرات کندی دارد مدل کرد. مدل این نوع صحبت به صورت زیر است:

$$s(n) = p(n) * g(n) * v(n) * r(n)$$

$$= \sum_{r=-\infty}^{\infty} h_v(n - rN_p)$$

$p(n)$ is a periodic pulse train of period N_p samples

$g(n)$ is a glottal wave shape

$v(n)$ is the vocal tract impulse response

$r(n)$ is the radiation impulse response

$h_v(n)$ is the impulse response of a linear system that

combines the effects of glottal wave shape, vocal tract IR, and radiation IR

Unvoiced (b) : در این حالت تحریک توسط سیگنال پالسی متناوب صورت نمی‌گیرد و بلکه تحریک به صورت یک سیگنال نویزی تصادفی مدل می‌شود. در مدل دستگاه صوتی در این حالت حنجره را در مدل در نظر نمی‌گیریم. دامنه صدای تولیدی در این حالت نسبت به حالت قبلی اندازه کمتری دارد و مدت زمان تحریک سیستم به اندازه زمان تحریک توسط پالس گوسی نیست.

$$s(n) = u(n) * v(n) * r(n) = u(n) * h_u(n)$$

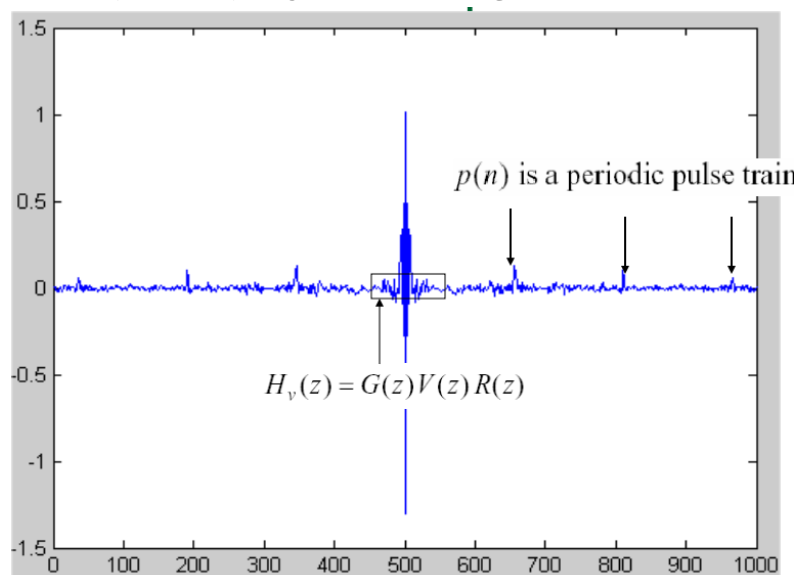
$u(n)$ is a random noise excitation

$h_u(n)$ is IR of system that combined the effects of the vocal tract and radiation

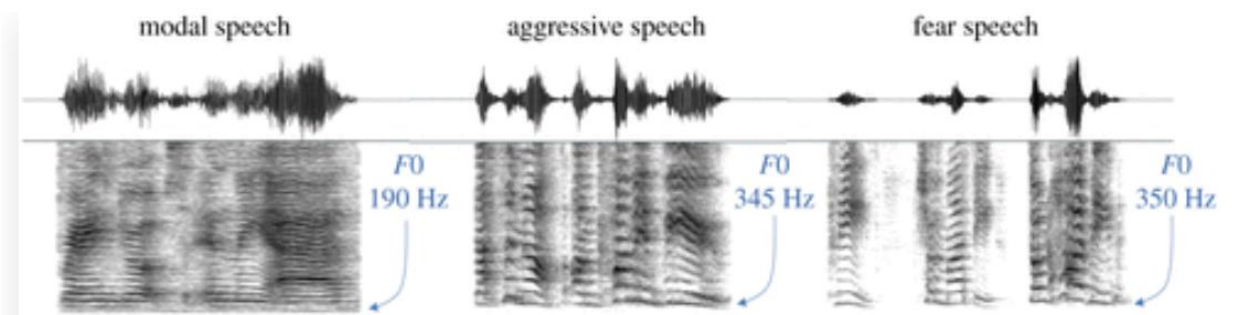
Transfer function for unvoiced speech

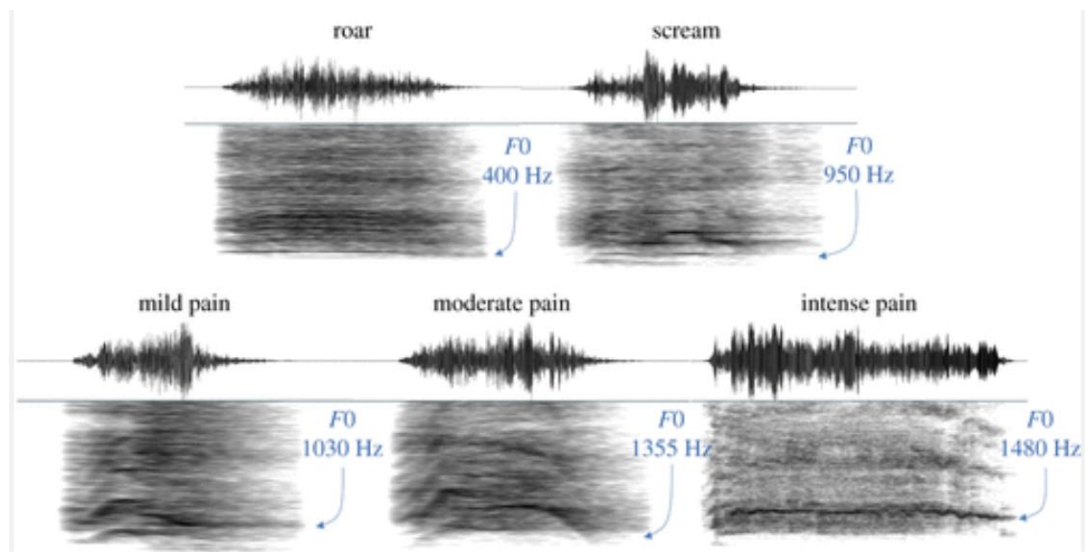
$$H_u(z) = V(z) R(z)$$

در حالت *Voiced* تحریک به صورت یک قطار پالس گوسی متناوب است پس توقع داریم که کپستروم آن نیز متناوب باشد :



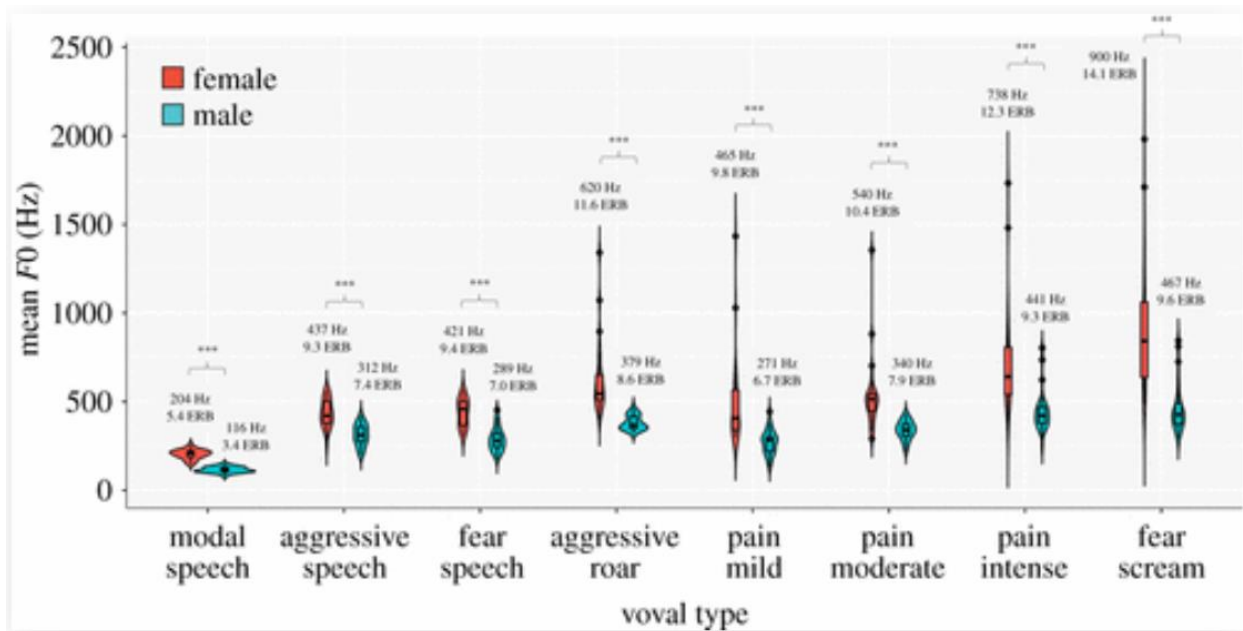
پس در کپستروم قله‌هایی داریم که مربوط به حالت *Voiced* هستند و این قله‌ها به صورت متناوب تکرار می‌شوند. به این قله‌ها گام صوت و یا *Pitch* می‌گوییم. این قله‌ها حاوی اطلاعات صوت هستند و با توجه به لحن صحبت و تن صدا تغییر می‌کنند همچنین در حالاتی که عواطف را در صدا منعکس می‌کنیم (مانند بغض و یا خشم) مقدار دامنه و فرکانس گام تغییر می‌کند. این گام با تغییر جنسیت و سن نیز تغییر می‌کند پس با توجه به گام می‌توانیم تخمینی از جنسیت فرد، حالت عاطفی آن و سن فرد دست بیاوریم. مانند شکل زیر: (در شکل زیر F_0 نشان دهنده فرکانس گام است).





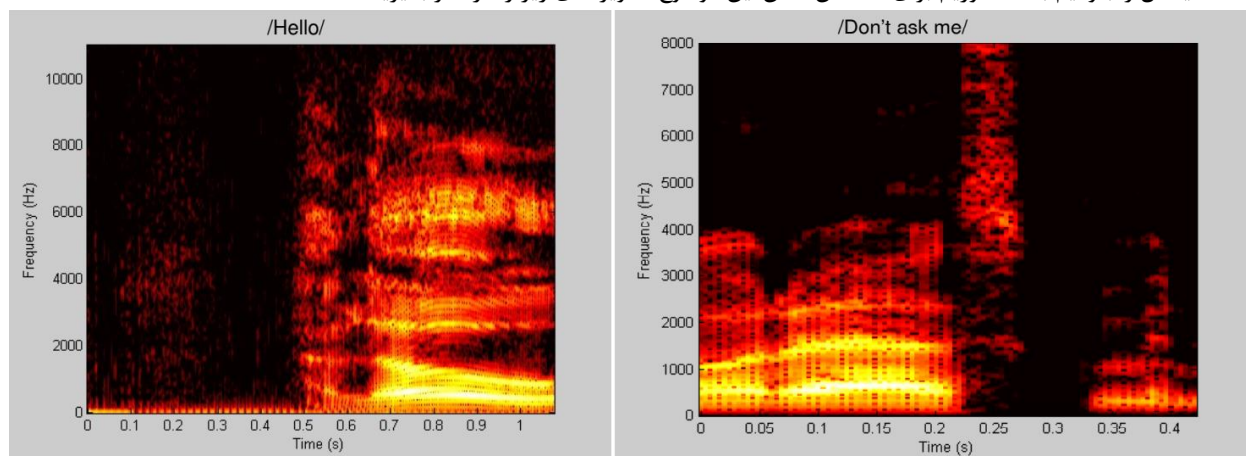
• دقت شود که گام با فرکانس شباهت دارند اما یکسان نیستند. زیرا فرکانس به طور مشخص اندازه گیری می شود اما گام یک پدیده حسی و درکی است و معیار مشخص ندارد.)

در حالت *Unvoiced* با توجه به اینکه تحریک توسط یک سیگنال تصادفی نویزی است و متناوب نیست پس در اصوات *Unvoiced* گام نداریم. پس نتیجه میگیریم که تمام اصوات انسان گام ندارد. حال بررسی میکنیم که گام زن و مرد چقدر متفاوت است. برای این کار یک جمله یکسان و با طول زمانی یکسان که توسط زن و مرد تلفظ شده را در نظر میگیریم: (مانند شکل زیر)



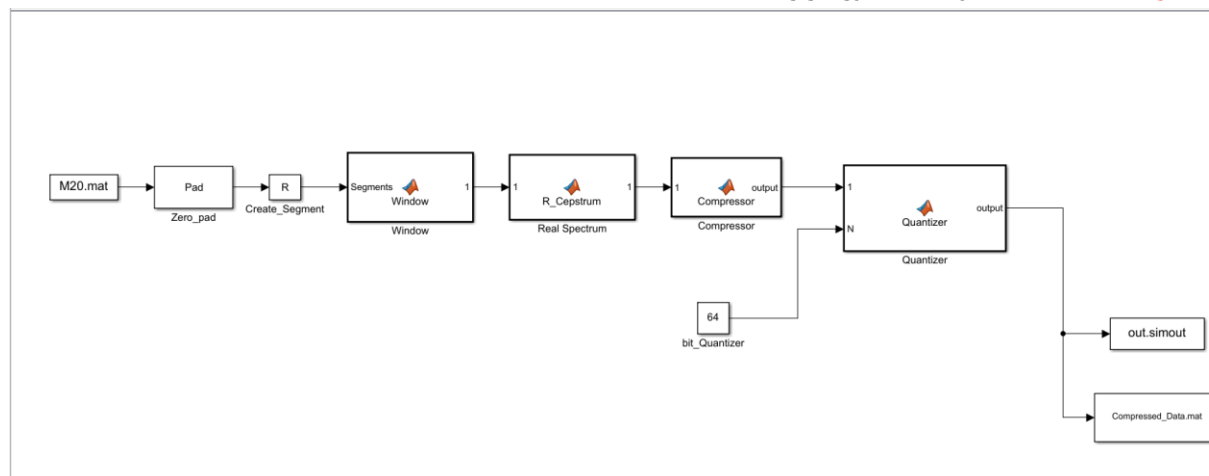
همانطور که مشاهده می کنید صوت زن در مقایسه با صوت مرد دارای فرکانس بیشتری است یعنی دوره تناوب پالس تولیدی کوتاه تر است و این باعث می شود که زن ها در مقایسه با مرد دارای صدای تیز تری داشته باشند و مولفه فرکانسی بالاتری داشته باشند. پس گام صدای زن در مقایسه با صدای مرد کوتاه تر است. همچنین زن ها طیف فرکانسی گسترده تری نسبت به مرد ها دارند.

3- سیگنال صوتی یک سیگنال تغییر پذیر با زمان است و نمی توان از آن تبدیل فوریه بگیریم پس باید از آن تبدیل $STFT$ بگیریم تا مشخصات سیگنال را بتوانیم بدست آوریم برای مشخص شدن این موضوع تصویر های زیر را در نظر بگیرید .



پس برای اینکه بتوانیم با همان تبدیل فوریه عادی کار کنیم مجبوریم فریم هایی با طول مناسب در نظر بگیریم . در صورت پروژه گفته شده است که $40ms$ در نظر بگیرید اما عدد $25ms$ نتیجه بهتری دارد و برای همین طول فریم را $25ms$ در نظر میگیریم . بنده دو فایل سیمولینک ساخته ام که فایل سیمولینک اول صوت را فشرده می کند و فایل سیمولینک دوم صدا را بازیابی می کند .

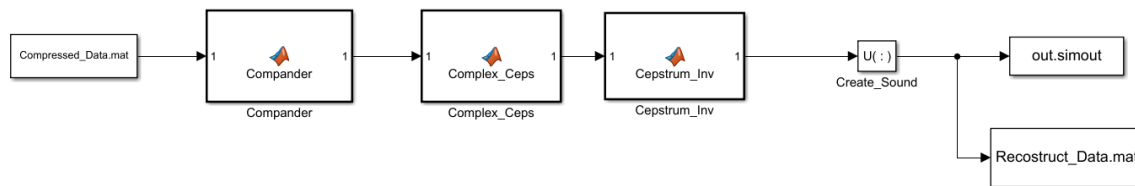
بخش فشرده ساز : محیط سیمولینک به صورت زیر است :



در بخش فشرده ساز بلوک ها کار های زیر را انجام می دهند :

- 1- بلوک *From File* صوت را وارد محیط سیمولینک می کند .
- 2- بلوک *Pad* صوت ورودی تا مقدار مورد نیاز *zero pad* میکند زیرا ممکن است اندازه صوت مضرب اندازه *Frame* نباشد پس برای همین با گسترش آن این مشکل را حل می کنیم .
- 3- بلوک بعدی که *Create_Segment* نام دارد صوت ورودی را به سگمنت ها میشکند و هر سگمنت را در یک آرایه عمودی میریزد و در آخر این آرایه را بهم متصل می کند و یک ماتریس دو بعدی تحویل می دهد.
- 4- بلوک *Window* هر ستون ماتریس ورودی را در پنجره *hamming* ضرب می کند .
- 5- بلوک *Real Cepstrum* از هر ستون ماتریس ورودی بلوک خود تبدیل کپستروم حقیقی میگیرد
- 6- بلوک *Compressor* به تعداد مطلوب ما داده از شروع کپستروم هر سگمنت برمیدارد همچنین مقدار دامنه *pitch* و مکان آن را نیز در فایل نهایی ذخیره می کند .
- 7- بلوک بعدی داده های ورودی را با تعداد بیت مشخص کوانتیزه می کند .
- 8- دو بلوک آخر یکی داده های فشرده شده را به محیط متلب و دیگری در یک فایل به نام *Compressed_Data* ذخیره می کند .

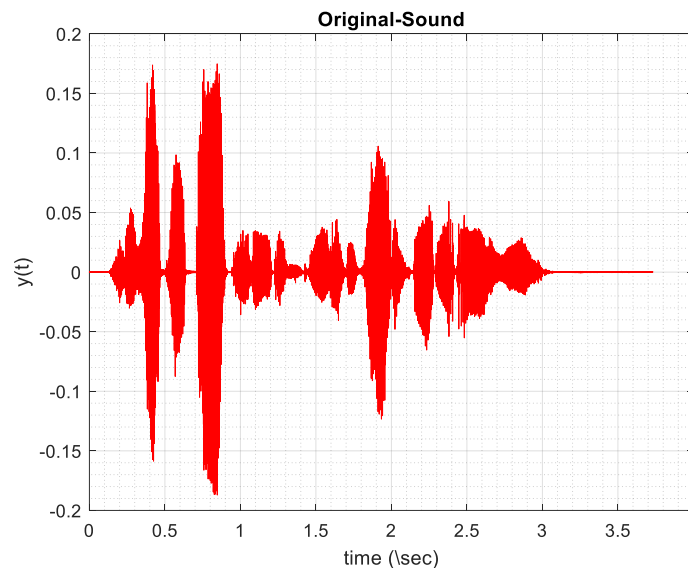
بازیابی داده‌های فشرده شده : ساختار زیر داده های فشرده شده را بازیابی می‌کند و سپس آن را در یک فایل ذخیره می‌کند.



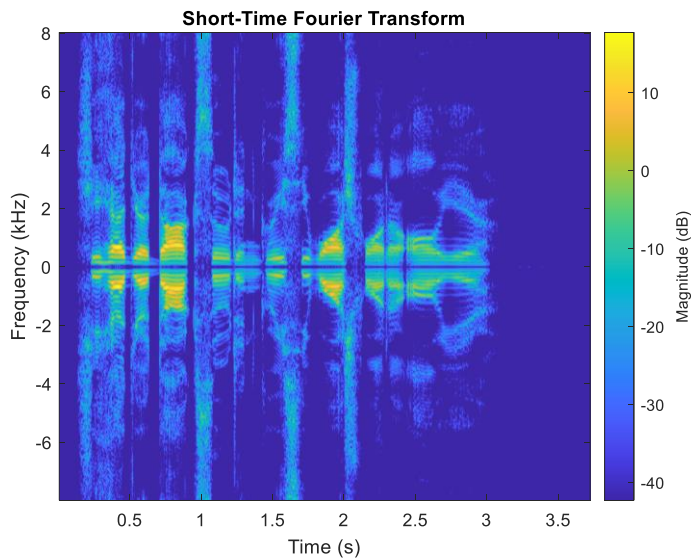
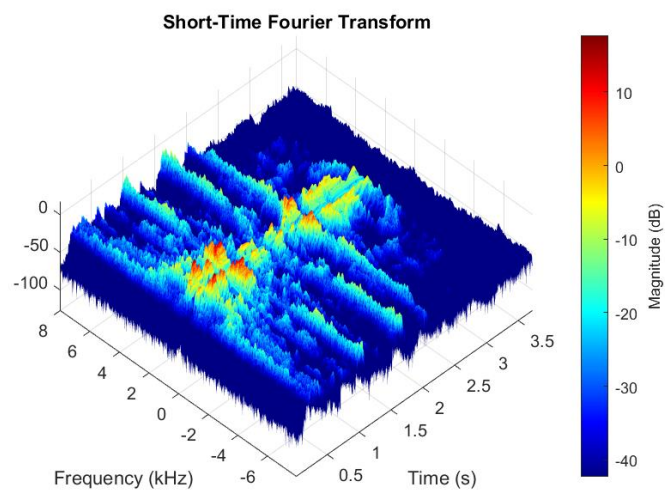
بلوک های این ساختار به صورت زیر است :

- 1- بلوک *Compander* داده های هر ستون به جز دو داده آخر که مربوط به گام است را در شروع ستون های ماتریس بازیابی قرار می‌دهد و داده مربوط به *Pitch* را در مکان آن قرار می‌دهد (دو داده آخر ذخیره شده هر ستون مربوط به *pitch* است).
- 2- خروجی بلوک *Compander* در واقع معادل کپستروم حقیقی صوت اولیه است و با توجه به رابطه ای که در بخش 1 برای تولید کپستروم مختلط از روی کپستروم حقیقی داشتیم ؛ معادل کپستروم مختلط صوت اولیه را می‌سازیم . این کار در بلوک *Complex - Ceps* صورت می‌پذیرد .
- 3- در بلوک *Cepstrum_Inv* معکوس کپستروم مختلط هر ستون را حساب میکنیم . با این کار سگمنت های صوت اولیه را بدست آورده ایم .
- 4- بلوک *Create_Sound* سگمنت های صوت را به هم متصل کرده و صوت بازیابی را شکل می‌دهد .
- 5- دو بلوک آخر صوت بازیابی شده را در محیط متلب و در *Reconstruct_Data.mat* ذخیره می‌کند .

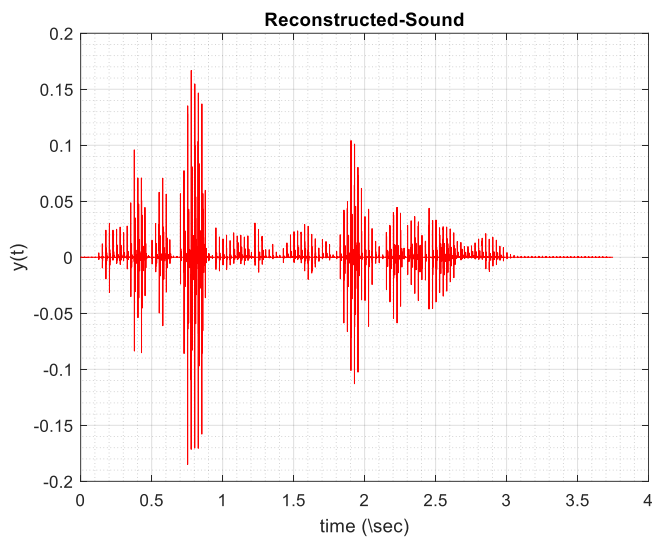
حال نتایج را برای صوت *M20* بررسی میکنیم . صوت اصلی به صورت زیر است :



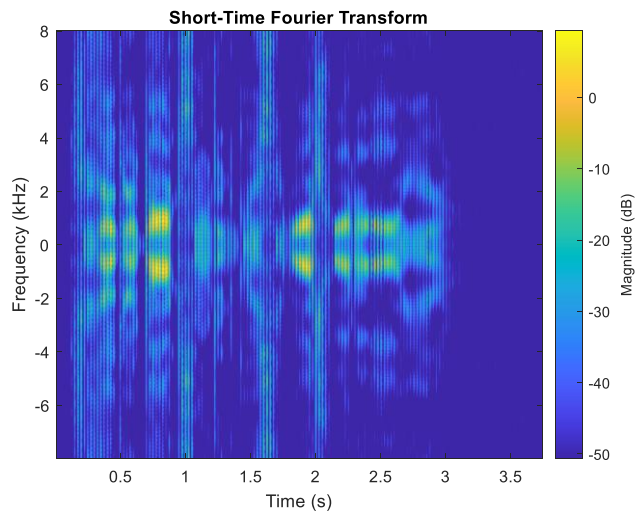
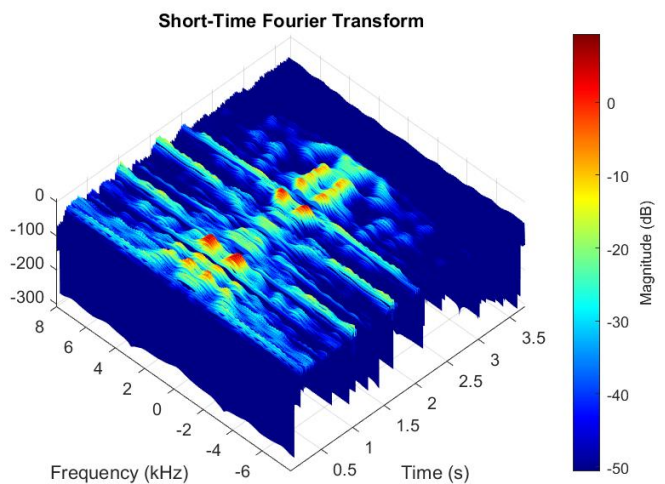
حال اگر صوت داده شده تبدیل *stft* بگیریم تا رفتار فرکانس آن برای ما مشخص شود شکل زیر بدست می‌آید :



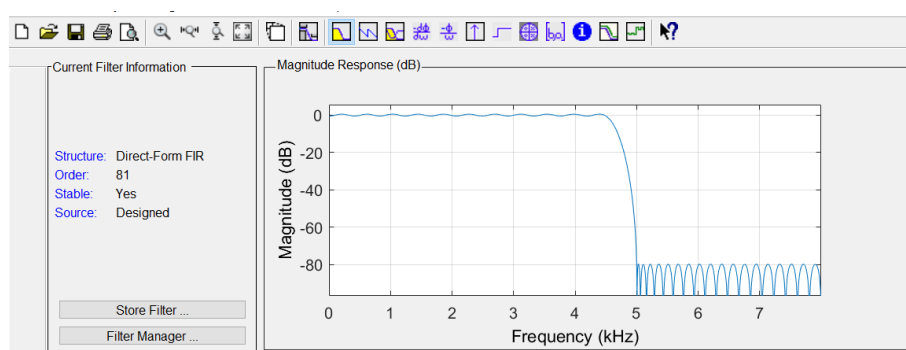
حال اگر سیگنال بازیابی شده را در نظر بگیریم :



در حوزه فرکانس نیز به صورت زیر می باشد :

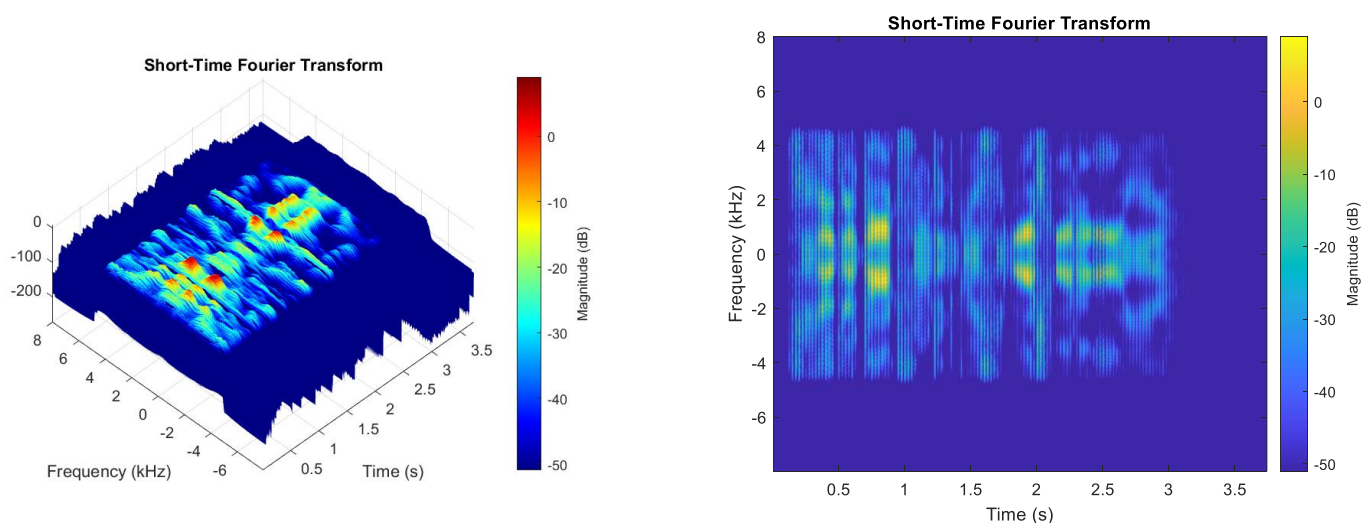
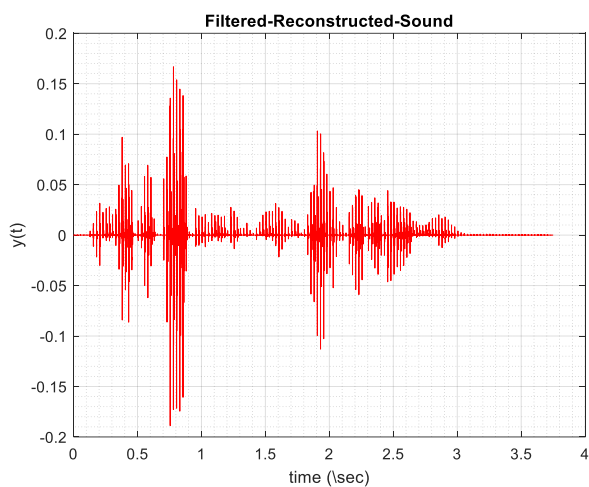


همانطور که مشاهده می کنید اثر حذف کردن داده و قرار دادن صفر به جای آن از روی نمودار کاملاً مشهود است . برای اینکه تغییر ناشی از متصل کردن سگمنت ها به هم را کم کنیم میتوانیم از یک فیلتر پایین گذر استفاده کنیم و فرکانس قطع را بر روی 4.5KHz قرار می دهیم: فیلتر به صورت زیر است :



• این فیلتر در محیط متلب است و در سیمولینک فیلتری اعمال نمی شود .

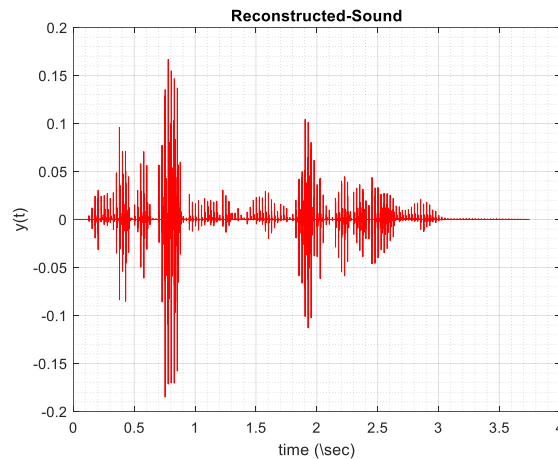
حال اگر سیگنال را فیلتر کنیم به صورت زیر در می آید:



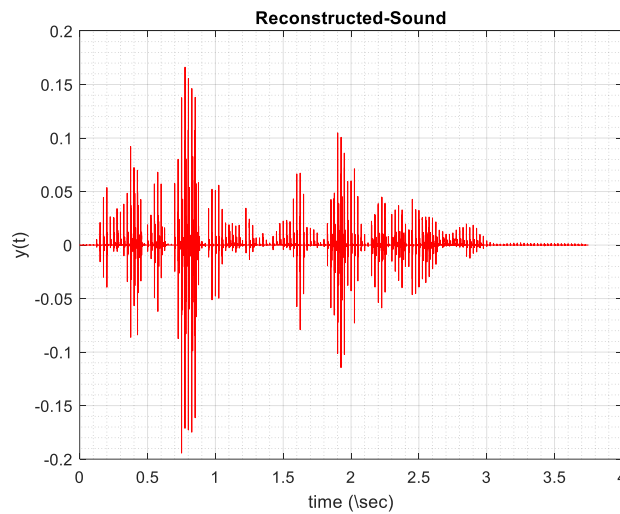
البته اعمال فیلتر پایین گذر برای صدای زن توصیه نمی شود زیرا صدای زن مولفه فرکانسی بالا دارد ، اما صدای مرد را بهبود میبخشد .

کیفیت صدای خروجی خیلی خوب نیست اما قابل فهم است . با فیلتر کردن صدا بهتر خواهد شد .
-4 در بخش قبلی پیاده سازی بلوک کوانتیزه کننده گفته شده است .

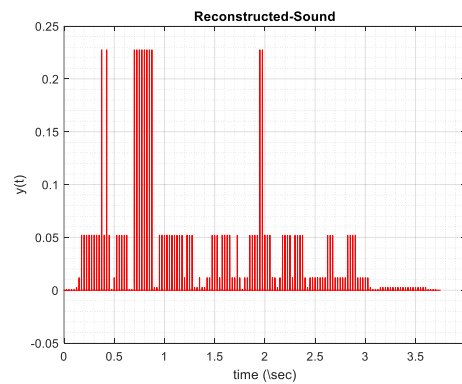
-5 در حالت عادی که قصد کوانتیزه کردن نداریم مقدار 64 بیت برای این بلوک در نظر میگیریم . خروجی بازسازی شده به صورت زیر است



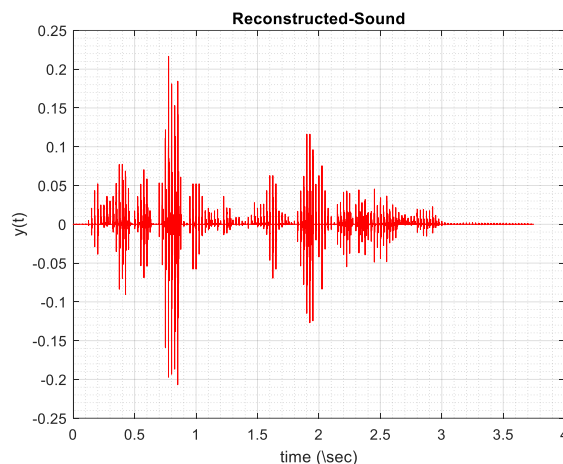
در این حالت صدای خروجی قابل فهم است . حال تعداد بیت را به 16 کاهش دهیم :



باز هم صدا قابل فهم است و تقریبا اصلا تغییر نکرده است . حال تعداد بیت را به 8 کاهش دهیم :



در این حالت صدا خراب می‌شود. حال با آزمون و خطا حداقل مقدار را برای تعداد بیت میابیم. به ازای حداقل 11 بیت صدای خروجی قابل فهم است و اگر از این کمتر شود صدا نامفهوم است. به ازای 11 بیت صدای خروجی به صورت زیر است.



6- در کوانتیزه کردن به روش عادی به کوانتایزر اسکالر شناخته می‌شود برای کوانتیزه کردن در هر لحظه یک عدد وارد می‌شود و این عدد به نزدیک ترین سطح کوانتایزر نگاشته می‌شود. در این حالت کوانتایزر در واقع در یک فضای یک بعدی است و مقدار های آن بر روی یک خط قرار دارد. حال اگر به جای این فضای یک بعدی از یک فضا با بعد بالاتر استفاده کنیم کوانتایزر برداری را استفاده کردیم. در واقع در کوانتایزر برداری به جای اینکه در هر لحظه از زمان یک نمونه را در نظر بگیرد و آن را کوانتیزه کند، دو و یا بیشتر نمونه در نظر می‌گیرد و آن‌ها را همزمان کوانتیزه می‌کند. در اینجا نیز نمونه های ورودی به نزدیک ترین سطح کوانتایزر نگاشته می‌شوند اما در اینجا سطوح کوانتایزر به جای یک عدد یک زوج مرتب و یا یک نقطه در فضا است. برای واضح تر شدن موضوع به طور مثال یک کوانتایزر دو بعدی با 16 سطح در نظر بگیریم:

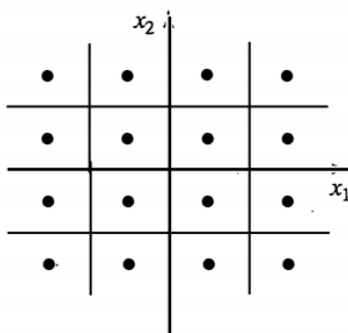


Figure 7.5 A scalar four-level quantization applied to two samples.

در اینجا در هر از زمان دو عدد وارد کوانتایزر می‌شوند و به نزدیک ترین سطح ممکن از میان 16 سطح موجود نگاشته می‌شود. روش بدست آوردن سطوح برای کوانتایزر و همچنین مرز ها در بهینه ترین حالت از طریق روش لوید - ماکس بدست می‌آید (این مطلب در کتاب سیستم های مخابراتی پراکیس صالحی به طور کامل بحث شده است.) و مثلاً اگر تعداد ناحیه های ممکن را افزایش دهیم دیگر انتخاب ناحیه مربعی بهینه نیست و بهتر است مثلاً سطوح کوانتایزر به صورت زیر انتخاب شوند.

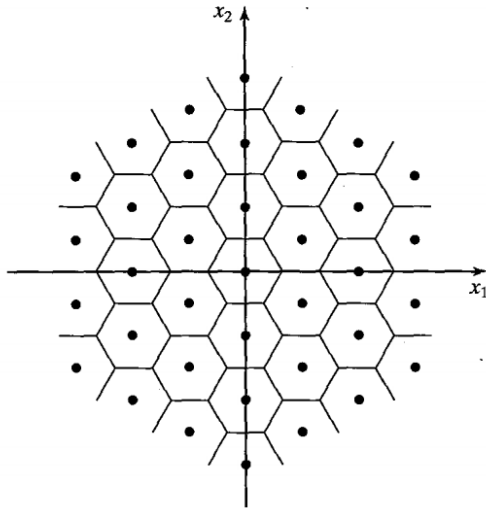


Figure 7.6 Vector quantization in two dimensions.

روش کوانتایزر برداری روش بهتر و بهینه تر از روش کوانتیزه کردن به صورت اسکالر با تعداد بیت یکسان است. از ارائه اثبات کلی اجتناب کرده اما روش اثبات با معرفی کردن اعوجاج کوانتیزاسیون برداری و اسکالر به صورت مربع فاصله اقلیدسی بین مقادیر کوانتیزه شده و مقدار های اصلی است و پس از یک سری اعمال ریاضی اثبات می شود که اعوجاج کوانتیزاسیون برای کوانتیزاسیون برداری کمتر است . اما به صورت شهودی اگر بخواهیم این موضوع را اثبات کنیم : در حالت اسکالر اگر بین نمونه های سیگنال همبستگی وجود نداشته باشد آنگاه کوانتایزر عددی نیز هیچگونه همبستگی بین آن ها ایجاد نخواهد کرد و در واقع با داشتن یک نمونه درباره مقدار نمونه بعدی هیچ اطلاعاتی نمی توان بدست آورد و اگر در حین انتقال داده در یک داده دچار خطا شویم امکان تشخیص خطا را نیز نداریم ! اما در عوض برای کوانتیزاسیون برداری حتی اگر نمونه ها از هم مستقل هم باشند آنگاه های خروجی کوانتیزاسیون از هم مستقل نخواهند بود و بین آنها همبستگی به وجود خواهد آمد . اگر مقدار یک نمونه را داشته باشیم نمونه بعدی باید مقدار مشخصی داشته باشد و نمی تواند هر مقدار داشته باشد و از این راه ما می توانیم اگر در انتقال داده در داده ای دچار خطا شویم آن را تشخیص دهیم .

پیوست : آموزش کار با فایل سیمولینک :

قبل از هر کار باید فایل *Initial_Set.m* را باز کنید در این فایل باید به صورت زیر عمل کنید .

```

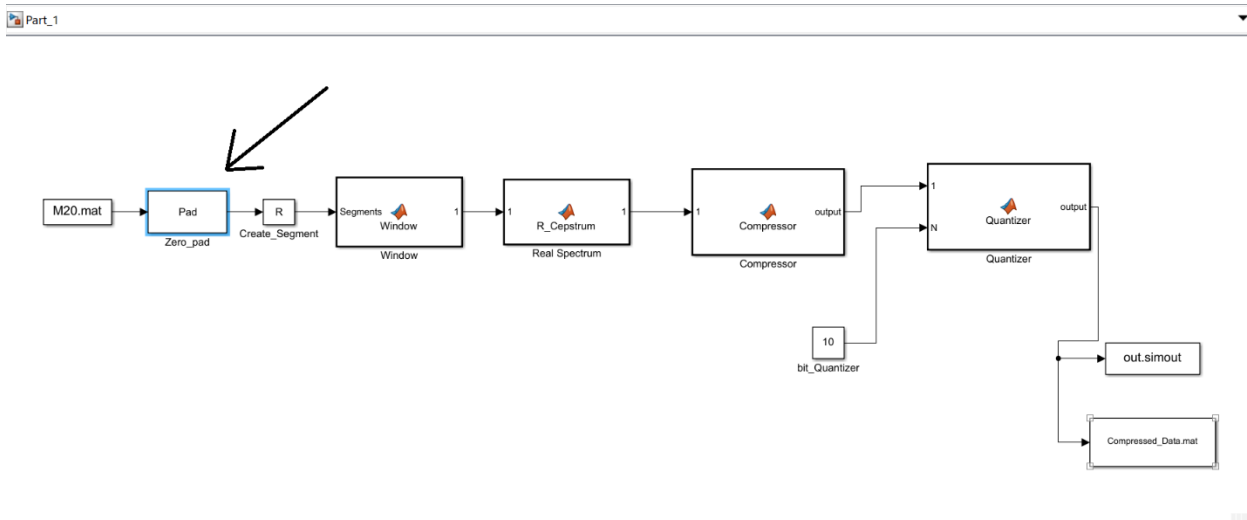
Editor - C:\Users\Alireza Rafiee\Documents\MATLAB\DSP\Project\Simulink\Initial_Set.m
Test.m x plotting.m x Initial_Set.m x +
1 - clc
2 - clear
3 - close all
4 - [Sound,Fs] = audioread('M20.WAV');
5 - M20=Sound;
6 - save('M20.mat','M20');
7 - Segment_time=25e-3;
8 - Frame=Fs*Segment_time;
9 - Frame
10 - Segment_num=ceil(length(Sound)/Frame);
11 - Segment_num
12 - Pad=Segment_num*Frame;
13 - Pad
14 - Frame=[0,Frame];

```

در این قسمت اسم صوت خود را وارد کنید

این قسمت زمان هر فریم هست . در صورت تمایل میتونید آن را تغییر دهید.

با اجرا کردن برنامه فوق سه داده در قسمت *Command Window* نمایش داده می شود به نام های *Segment_num* و *Frame* و *Pad* . باید این مقدار ها را در فایل سیمولینک اول که مربوط به پیاده سازی فشرده ساز هست در بلوک های مشخصی قرار دهید . مقدار *Pad* را در بلوک و در مکان مشخص شده قرار دهید .



Block Parameters: Zero_pad

Pad (mask) (link)
Append or prepend a constant value to the input along the specified dimensions. Truncation occurs when the specified output dimensions are shorter than the corresponding input dimensions.

Parameters

Pad over: Rows

Pad value source: Specify via dialog

Pad value: 0

Output row mode: User-specified

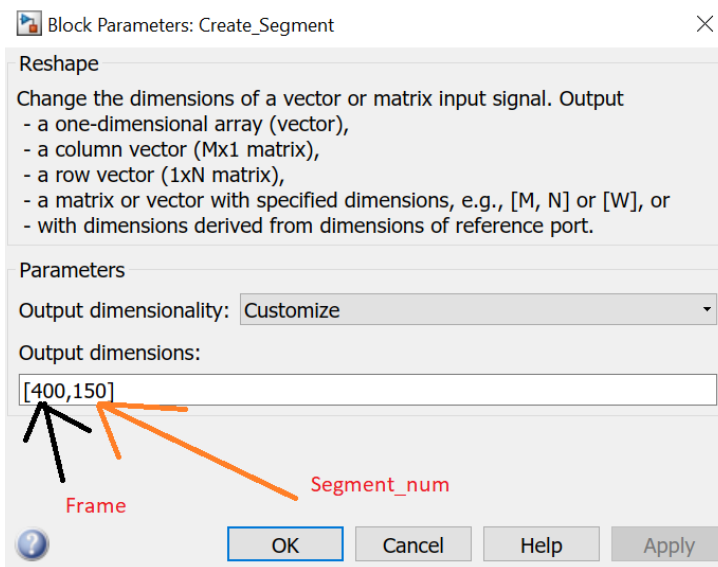
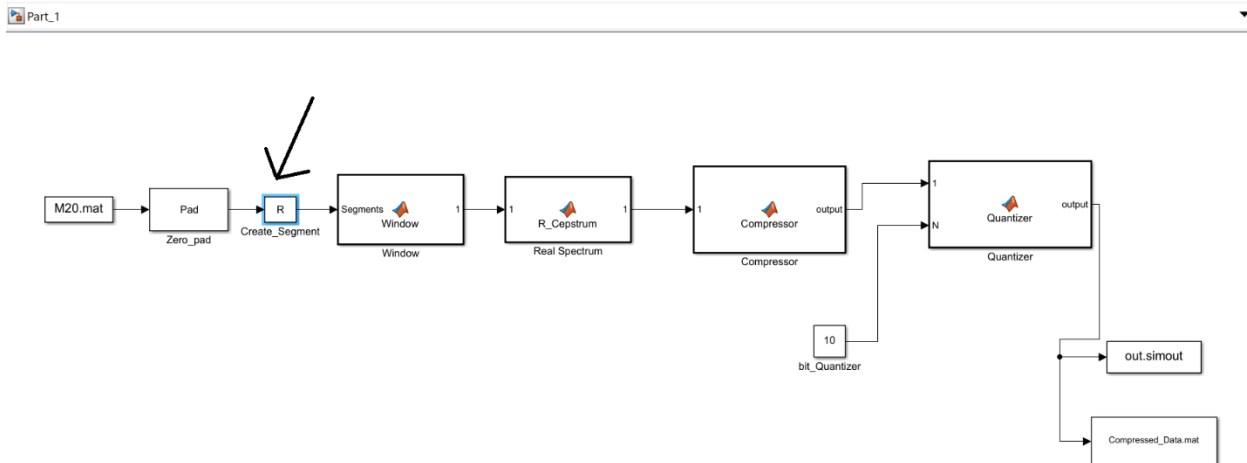
Row size: 60000

Pad signal at: End

Action when truncation occurs: None

OK Cancel Help Apply

مقدار های $Frame, Segnemt_num$ را در مکان های مشخص و در بلوک مشخص شده قرار دهید .



حال برنامه را اجرا کنید و صوت فشرده خواهد شد .

برای اجرای بخش بازسازی نیاز به جایگذاری چیز خاصی نیست و فقط باید دقت شود که حتما یک بار قبل از اجرای این بخش برنامه $Initial_Set.m$ را اجرا کردید و داده ها برنامه را پاک نکردید و یا برای اطمینان بیشتر و راحتی میتوانید مقدار $Frame$ بدست آوردید را به صورت زیر در $Command Window$ وارد کنید و سپس بخش دوم را اجرا کنید . (مقدار بدست آمده باید به جای $Frame$ موجود در ماتریس قرار گیرد) . البته باز راحت تر است که یک بار برنامه $Initial_Set.m$ را اجرا کنید و سپس بخش دوم سیمولینک را اجرا کنید .

$Frame = [0, Frame];$

در صورت تمایل می توانید پس از اجرای بخش دوم سیمولینک از طریق برنامه $Reasual.m$ فیلتر اعمال کرده و یا طیف ورودی و خروجی را مشاهده کنید .