

پروژه ی پایانی درس پردازش سیگنال های دیجیتال: Equalizer صدا

لطفا قبل از شروع، به نکات زیر توجه کنید:

۱. پروژه پایانی درس به صورت یک ساختار از پیش نوشته شده در محیط MATLAB در اختیار شما قرار می‌گیرد و تنها لازم است تا قسمت‌هایی از کد توسط شما مورد ویرایش قرار گیرند. در متن پیش رو، قسمت‌هایی که به ویرایش نیاز دارند توضیح داده شده‌اند. لطفا تغییری در سایر قسمت‌ها ایجاد نکنید. به علاوه، در قسمت‌هایی لازم است که به پرسش‌هایی پاسخ دهید. خروجی پروژه شما، علاوه بر کدهای تکمیل شده، گزارشی تایپ شده است که در آن به سوالات مورد نظر پاسخ گفته اید و نتایج حاصل از اجرای برنامه را در آن قرار داده‌اید. گزارش شما باید شامل توضیح نتایج حاصل نیز باشد. پس از تکمیل گزارش، کدهای MATLAB و نتایجی که باید در پوشه‌های در نظر گرفته شده ذخیره شوند، کل پوشه را به فرمت zip با نامی حاصل از ترکیب نام و شماره دانشجویی خود ذخیره کنید. این فایل zip را در سامانه CW در مهلت مقرر بارگذاری کنید. در صورتی که حجم فایل zip شما فراتر از حجم قابل قبول در سامانه CW شد، می‌توانید تعدادی از فایل‌های اولیه را (به غیر از کدها و نتایج خود) حذف کنید.

۲. برای تسهیل کار، یک رابط گرافیکی برای equalizer طراحی شد است که با اجرای فایل Equalizer.m پدیدار می‌شود. فایل‌های صوتی که در اجرا مورد استفاده قرار می‌گیرند در پوشه Audio_files قرار داده شده‌اند. در قسمت‌هایی که نیاز به بارگذاری یک فایل صوتی است، باید در این پوشه قرار داده شوند. با تنظیم هریک از Slider ها در محیط گرافیکی و یا بار گذاری EQ_files ها می‌توان مقدار دامنه هر یک از فیلترهای در بانک فیلتر Equalizer را تنظیم کرد. همچنین قابلیت اضافه کردن Effect های لرزش، بازتاب و Modulator سینوسی یا اره ای بر روی فایل صوتی وجود دارد. به دلیل استفاده از مفاهیم بانک فیلتر در این پروژه، ابتدا مرور مختصری بر این موارد خواهیم داشت. سپس بخش های مختلف Equalizer طراحی و تکمیل می‌شود.

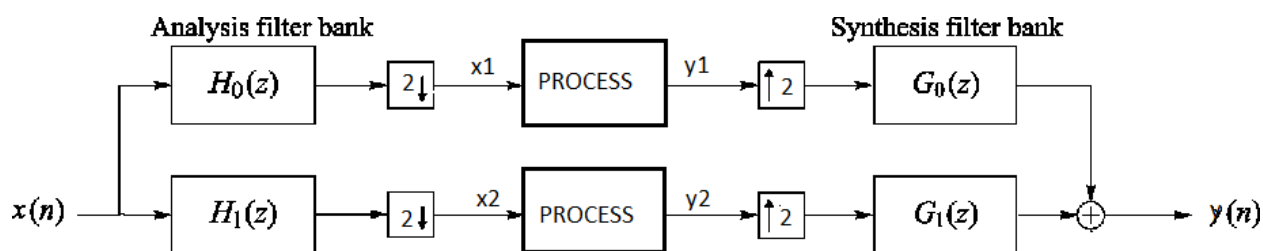
۳. این پروژه شامل نمره امتیازی نیز خواهد بود: پاسخ به سوالات امتیازی، نگارش مناسب گزارش، و رعایت اصول کد نویسی هر کدام به صورت جداگانه نمره امتیازی خواهند داشت.

۴. هدف اصلی از انجام پروژه، آشنایی بهتر و عملی با مباحث مطرح شده در کلاس درس است. با توجه به آماده بودن قسمت زیادی از کدها، تکمیل پروژه بسیار ساده‌تر از انجام کامل آن از صفر است. تعریف پروژه به صورتی است که به صورت فردی و در زمان مناسب قابل انجام باشد. اولویت پرسش سوال از دستیار آموزشی مرتبط با این پروژه است (j.maheri95@yahoo.com). با این حال، کمک گرفتن از سایرین برای رفع اشکال در صورتی که به کپی کردن (چه کد، چه نتایج و چه گزارش) نینجامد، مشکلی ندارد. اما در

صورت مشاهده هر نوع کپی، تمامی افراد خاطی با تخصیص نمره منفی جریمه خواهند شد (یعنی نه تنها نمره صفر در قسمت مربوطه دریافت میکنند، نمرای از قسمتهای دیگر نیز از دست خواهند داد).

قسمت اول: فیلتر بانک ها

ساختار بانک فیلتر برای اعمال فیلترها و احتمالاً عملگرهای غیر خطی مختلف بر روی بخش های فرکانسی متفاوت سیگنال ایجاد شده است:



در شکل بالا ساختار بانک فیلتر دوتایی (حالت M تایی نیز به راحتی قابل تعمیم هست) را مشاهده می کنید در این ساختار ابتدا به کمک فیلترهای H_0 و H_1 باند فرکانسی سیگنال ورودی به دو بخش تقسیم می گردد، سپس Down sampling انجام شده و در مرحله بعدی پردازش مورد نظر هر قسمت انجام می گیرد. دقت کنید که طراحی فیلتر پایین گذر یا بالا گذر ایده آل غیرعملی است. به همین دلیل، بعد از عمل Down sampling تداخل فرکانسی (aliasing) اجتناب ناپذیر است. لذا بر آنیم تا با طراحی مناسب فیلترهای G_0 و G_1 اثر تداخل را حذف کنیم؛ در حقیقت، اگر بتوان خروجی را به صورت نوشت یعنی $Y(z) = H(z)X(z)$ aliasing نداریم. در ابتدا با فرض نبودن هیچگونه پردازش مسئله را مورد بررسی قرار می دهیم.

۱. با فرض $x_1 = y_1$ و $x_2 = y_2$ شرط لازم و کافی برای این که فیلتربانک PR (یعنی بدون aliasing واعوجاج) باشد را بیابید. عبارتی که در بدست می آورید نشان می دهد ورودی پس از عبور از فیلتربانک دچار اعوجاج و aliasing می گردد. لذا زمانی می توان ادعا کرد که ورودی با خروجی برابر است که دچار اعوجاج و تداخل نشده باشد (شیفت اعوجاج محسوب نمی شود)

۲. نشان دهید $G_0(z) = H_1(-z)$ و $G_1(z) = -H_0(-z)$ یک شرط کافی برای نداشتن aliasing است. برای طراحی فیلتر بانک ها رویکردهای متنوعی وجود دارد. در اینجا با هدف آشنایی ابتدایی فقط روش QMF (Quadrature Mirror Filter) را مورد بررسی قرار می دهیم. در این روش ابتدا فیلترهای H_0 و H_1 طراحی می گردند. سپس با توجه به شرط کافی مطرح شده، فیلترهای G_0 و G_1 ساخته می شوند. فیلتر H_1 با رابطه $H_1(z) = H_0(-z)$ طراحی می گردد.

۳. با اعمال شرط کافی مطرح شده در قسمت قبل عبارت مربوط به اعوجاج را بیابید و نشان دهید در صورتی که $H_0(z) = c_0 z^{-2n_0} + c_1 z^{-2n_1+1}$ باشد می توان اعوجاج را حذف کرد (می توان نشان داد تنها حالت ممکن حذف اعوجاج همین است). واضحا با دو ضریب نمی توان فیلتر پایین گذر مناسبی طراحی کرد؛ پس عملا با این روش نمی توان فیلتر بانک PR ساخت. فیلترهای FIR با فاز خطی به دلیل عدم ایجاد اعوجاج فاز اهمیت قابل توجهی در پردازش سیگنال ها علی الخصوص سیگنال صدا دارند. در اینجا برای طراحی H_0 از آنها بهره می بریم.

۴. نشان دهید اگر $H_0(z)$ فیلتر FIR با فازخطی باشد $H(z)$ نیز FIR با فاز خطی است. حالت های مختلف را بررسی کنید. چرا از فیلترهای با تقارن فرد $(Y(z) = H(z)X(z))$ نمی توان برای طراحی $H_0(z)$ استفاده کرد؟

۵. نشان دهید اگر از فیلتر FIR با تقارن زوج و طول زوج استفاده کنیم اعوجاج شدید در فرکانس $\frac{\pi}{2}$ رخ می دهد. به علاوه، عبارت اعوجاج با انتخاب فیلتر FIR با تقارن زوج و طول فرد به صورت

$$H(e^{j\omega}) = \frac{1}{2} \{ |H_0(e^{j\omega})|^2 + |H_1(e^{j\omega})|^2 \} e^{-jM\omega}$$

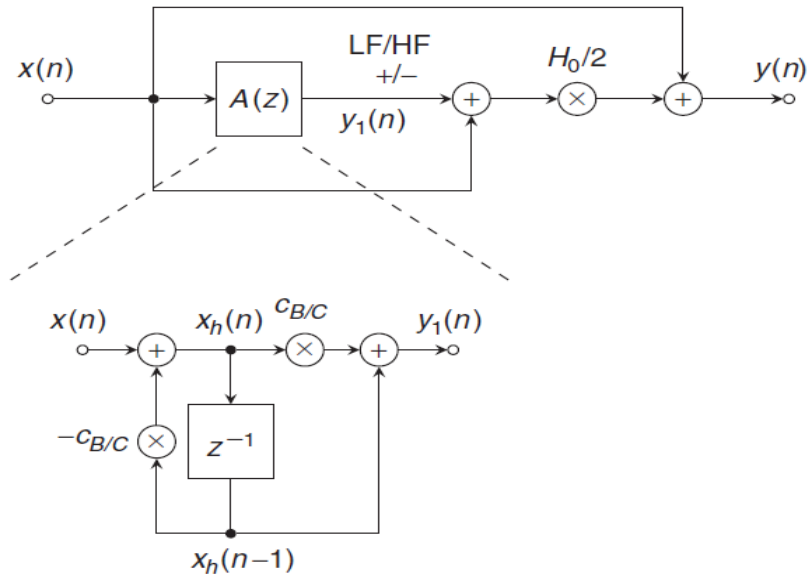
است. با توجه به خطی بودن فاز، اعوجاج در فاز نداریم. اما اعوجاج در دامنه داریم.

۶. (امتیازی) اگر $H_0(z)$ فیلترپایین گذر خوبی باشد تا حد خوبی در Pass Band و Stop Band اعوجاج دامنه نداریم. ولی مشکل در Transition Band به قوت خود باقی است. یک روش با جزئیات کامل جهت حل این مشکل مطرح کنید.

در ادامه کاربرد فیلتربانک ها در طراحی یک Equalizer صدا را بررسی میکنیم. جهت سادگی کار مباحث مطرح شده در بالا به طور کامل در فیلتر بانک Equalizer استفاده نشده است. در ادامه با چند نوع فیلتر پرکاربرد و Effect های صدا آشنا می شوید.

قسمت دوم: طراحی فیلترها

در این قسمت شما با طراحی چند فیلتر پر کاربرد جهت جداسازی بخش های مختلف طیف فرکانسی آشنا می شوید. برای جداسازی اولین و آخرین باند از فیلتری با مشخصات زیر استفاده می کنیم.



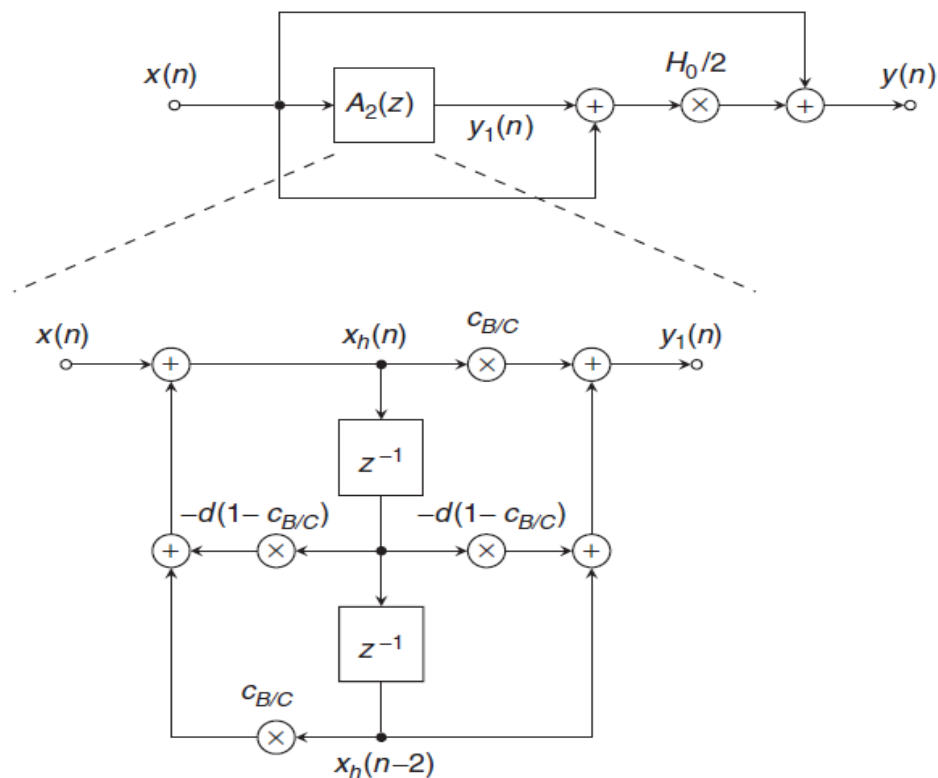
۱. در این قسمت فیلتر بالا را برای اعمال به باند اول و آخر طراحی میکنیم. با توجه به ساختار بالا و روابط زیر تابع

LOW_HIGH_filt.m را کامل کنید. ($V_0 = 10^{G/10}$, $H_0 = V_0 - 1$)

$$c_B = \frac{\tan\left(\frac{\pi f_c}{f_s}\right) - 1}{\tan\left(\frac{\pi f_c}{f_s}\right) + 1} \quad c_c = \frac{\tan\left(\frac{\pi f_c}{f_s}\right) - V_0}{\tan\left(\frac{\pi f_c}{f_s}\right) + V_0} \quad \text{related to Low pass filter} \quad \begin{cases} \text{if } G < 0, \text{ we use } c_c \\ \text{if } G > 0, \text{ we use } c_B \end{cases}$$

$$c_B = \frac{\tan\left(\frac{\pi f_c}{f_s}\right) - 1}{\tan\left(\frac{\pi f_c}{f_s}\right) + 1} \quad c_c = \frac{V_0 \tan\left(\frac{\pi f_c}{f_s}\right) - 1}{V_0 \tan\left(\frac{\pi f_c}{f_s}\right) + 1} \quad \text{related to High pass filter} \quad \begin{cases} \text{if } G < 0, \text{ we use } c_c \\ \text{if } G > 0, \text{ we use } c_B \end{cases}$$

پس از طراحی فیلتر پایین گذر و بالا گذر جهت اعمال به باند های اول و آخر، برای باندهای میانی 2 تا 9 از فیلتر Band pass با ساختار زیر استفاده می کنیم.



۲. با استفاده از ساختار بالا و روابط زیر تابع BAND_filt.m را کامل کنید.

$$. (d = -\cos(\frac{2\pi f_c}{f_s}) \quad V_0 = 10^{G/10} , \quad H_0 = V_0 - 1)$$

$$c_B = \frac{\tan\left(\frac{\pi f_b}{f_s}\right) - 1}{\tan\left(\frac{\pi f_b}{f_s}\right) + 1} \quad c_c = \frac{V_0 \tan\left(\frac{\pi f_b}{f_s}\right) - 1}{V_0 \tan\left(\frac{\pi f_b}{f_s}\right) + 1} \quad \text{related to Band pass filter} \begin{cases} \text{if } G < 0, \text{ we use } c_c \\ \text{if } G > 0, \text{ we use } c_B \end{cases}$$

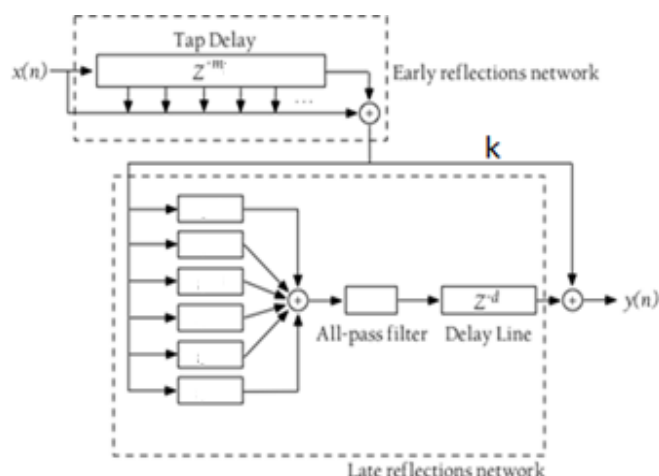
در رابطه بالا دقت شود به جای فرکانس قطع، از پهنا باند فیلتر میانگذر f_b استفاده شده است و $Q = \frac{f_b}{f_s}$ در تعیین مقدار Q برای هر باند (2 تا 9) دقت کافی صورت گیرد، در غیر این صورت بخش هایی از اطلاعات صوت از بین می رود.

قسمت سوم: اضافه کردن Effect های صدا

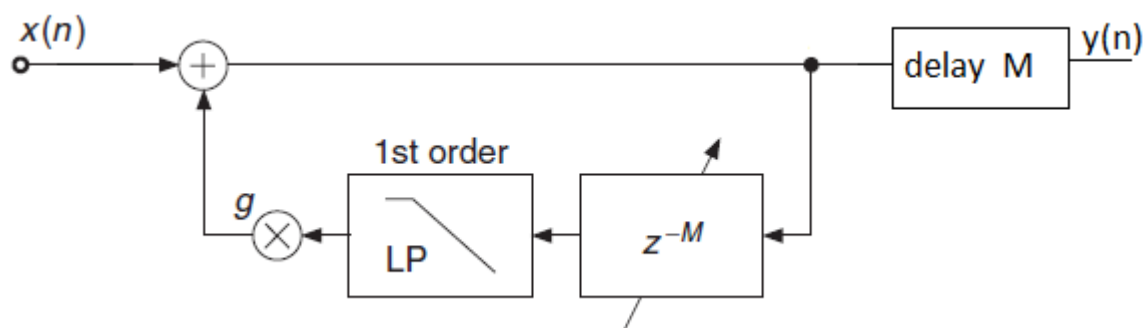
اولین Effect ی که اضافه می کنیم بازتاب صدا هست که با ساختار زیر پیاده می شود.

۱. توضیحات روش اعمال Effect بالا در شکل زیر و Specific_Design help قابل مشاهده است. ساختار زیر را در تابع Specific_Design.m پیاده کنید:

هریک از مسیر های شش گانه موازی در شکل زیر یک فیلتر خاص می باشد که پیاده سازی آن برعهده دانشجو است. همچنین در انتها، خروجی باید نرمالایز گردد.



هریک از فیلتر ها ی موازی دارای ساختار زیر می باشند.



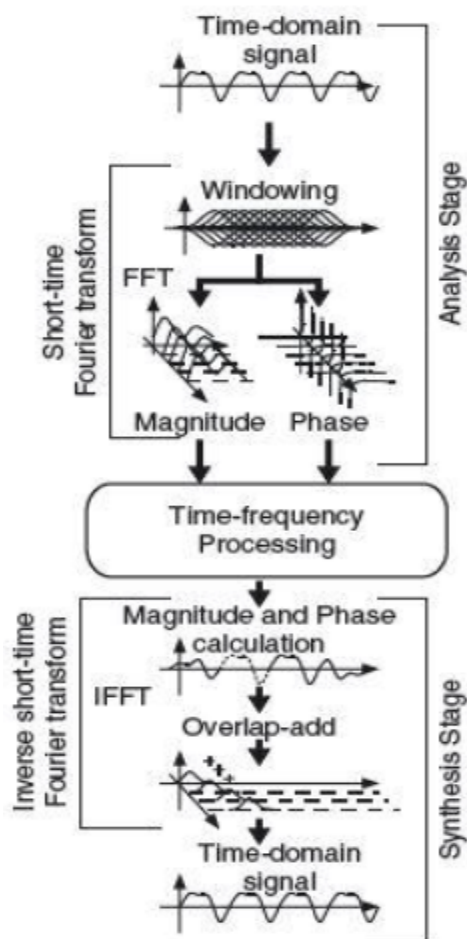
جهت سادگی کار، فیلتر Low pass را با رابطه $\frac{1 - g_1}{1 - g_1 z^{-1}}$ تقریب می زنیم. سپس Effect لرزش صدا را اعمال می کنیم.

قسمت چهارم: STFT

در این بخش می خواهیم دو تابع STFT و ISTF را که در بخش بعدی پروژه کاربرد دارد پیاده سازی کنیم. ابتدا تئوری را به صورت مختصر بررسی می کنیم. ساختار زیر به درک مطالب کمک زیادی میکند. STFT یا

Short Time Fourier Transform برای سیگنال $x(n)$ و با اعمال فیلتر $h(n)$ به صورت زیر تعریف می گردد.

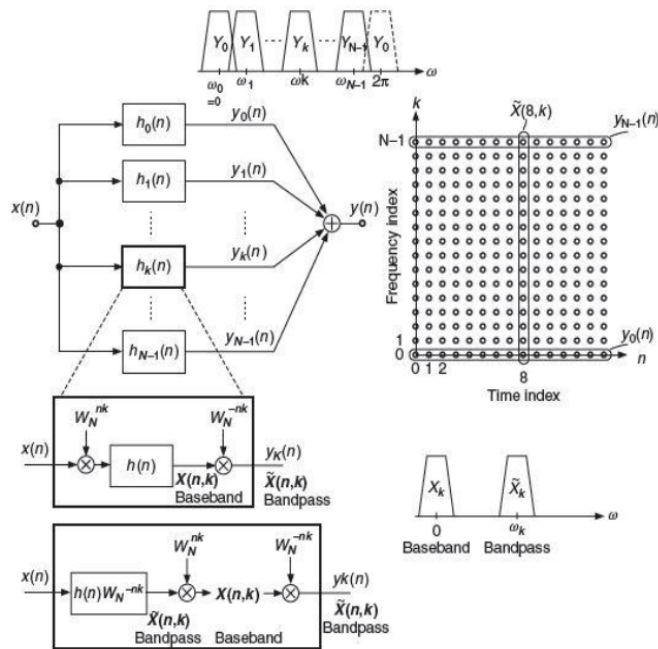
$$X(n, k) = \sum_{m=-\infty}^{\infty} x(m)h(n-m)e^{-j\frac{2\pi mk}{N}} = |X(n, k)|e^{j\varphi(n, k)}$$



پیاده سازی رابطه بالا به ازای n و k ثابت به کمک FFT امکان پذیر است. جهت سهولت پیاده سازی h_k را به صورت زیر تعریف می کنیم:

$$h_k(n) = h(n)e^{j\frac{2\pi nk}{N}}, \quad k = 0, 1, \dots, N-1 \rightarrow H_k(e^{j\omega}) = H_k(e^{j(\omega - \omega_k)}), \quad \omega_k = \frac{2\pi k}{N}$$

با تعریف بالا می توانیم STFT را به صورت فیلتر بانک پیاده سازی کنیم. $y_k(n)$ خروجی k امین فرکانسی bin است.



$$y_k(n) = \sum_{m=-\infty}^{\infty} x(m)h_k(n-m) = e^{j\frac{2\pi nk}{N}} \sum_{m=-\infty}^{\infty} x(m)h(n-m)e^{-j\frac{2\pi mk}{N}} = e^{j\frac{2\pi nk}{N}} \left| X(n, k) \right| e^{j\varphi(n, k)}$$

$$\left| y_k(n) \right| = \left| X(n, k) \right|, \hat{\varphi}(n, k) = \frac{2\pi k}{N}n + \varphi(n, k)$$

$$y(n) = \sum_{k=0}^{N-1} y_k(n) = \sum_{k=0}^{N-1} X(n, k) e^{j\frac{2\pi nk}{N}}$$

همچنین خروجی به صورت روبه رو است.

با توجه به حقیقی بودن سیگنال ورودی و پنجره، می توانیم روابط زیر را بنویسیم:

$$y_k(n) = X(n, k) e^{j\frac{2\pi nk}{N}} = X^*(n, k) = \left(e^{j\frac{2\pi n(N-k)}{N}} \right)^* = y_{N-k}^*(n)$$

تعریف می کنیم:

$$\hat{y}_k(n) = y_k(n) + y_{N-k}^*(n) = y_k(n) + y_k^*(n) = 2 \left| X(n, k) \right| \cos(\omega_k n + \varphi(n, k))$$

$$y(n) = \sum_{k=0}^{N/2} \hat{y}_k(n) = \sum_{k=0}^{N/2} A(n, k) \cos(\hat{\varphi}(n, k))$$

و در نهایت:

۱. با توجه به توضیحات بالا توابع stft.m و istft.m را تکمیل کنید.

قسمت پنجم: Modulator

در این قسمت برای سادگی کار فقط Modulation دامنه به صورت مختصر مورد بررسی قرار می گیرد.

۱. سیگنال Modulator سینوسی با فرکانس سه کیلوهرتز با انتخاب طول مناسب بسازید و به صورت فایل S1.wav در پوشه Modulator_sig ذخیره کنید.

۲. سیگنال Modulator اری با فرکانس سه کیلوهرتز با انتخاب طول مناسب بسازید و به صورت فایل S2.wav در پوشه Modulator_sig ذخیره کنید.

۳. جهت انجام Modulation لازم است پوش سیگنال را بیابیم. با مطالعه توضیحات تابع specenv.m آن را تکمیل نمایید.

۴. یکی از فایل های Audio_files را انتخاب کرده و عمل مدولاسیون را با هردو Modulator بالا انجام دهید و تفاوت را گزارش کنید.