

بسمه تعالی

گزارش تمرین شماره 1 درس سیگنال و سیستم
دکتر کربلایی

طاها انتصاری 95101117

امیرحسین طالبی 94101426

تابع مذکور ساخته شده و به پیوست در فایل زیپ قرار دارد.

با چند آزمون و خطا مشاهده میشود که در $F_s=150k$ برای تابع sound هم نیز هنوز صدای خوبی شنیده میشود. در واقع هر چه قدر که ما فرکانس سمپلینگ را برای تابع sound کم کنیم مدت زمان بیشتری طول میکشد که بردار y را پرمایش کرده و صدای آن را تولید کند. البته اگر فرکانس سمپلینگ را برای تابع SoundMaker هم تغییر دهیم میبینیم که همان آهنگ پخش میشود (البته گوش قادر به تشخیص تفاوت ها نیست). هر قدر که فرکانس سمپلینگ را برای تابع sound کم کنیم مدت زمان بیشتری طول میکشد که آهنگ پخش شود. البته این مدت زمان بیشتر دقیقاً برابر نسبت 2 فرکانس سمپلینگ در 16.334 (که این عدد 16.334 جمع درایه های interval است). اگر فرکانس سمپلینگ تابع sound را بیشتر کنیم تابع در واقع بردار y ورودی اش را با سرعت بیشتر پرمایش میکند. البته این کار به نظر مشکل هایی دارد چرا که از آنجا که بردار y ما گسسته است، ممکن است در بعضی جا ها بردار y برای آن نقطه ای که تابع sound داده میخاهد، هیچ داده ای نداشته باشد (و شاید خود متلب برای این مشکل میانگین گیری کند).

در فرکانس 20 کیلو آهنگ خیلی کند بوده و در 100 کیلو کند. فرکانس 400 کیلو را متلب به عنوان فرکانس سمپلینگ قبول نمیکند و ارور دستگاه میدهد.

با اضافه شدن dc به note اگر مقدار اضافه شده مثبت باشد، پس فرکانس بالا میرود و صدا زیرتر میشود همانگونه که مشاهده شد، و اگر مقدار اضافه شده منفی باشد، صدا بم تر میشود.

برای interval نیز اگر چیزی اضافه کنیم مدت زمان هر یک از نت ها به همان مقدار افزایش می یابد.

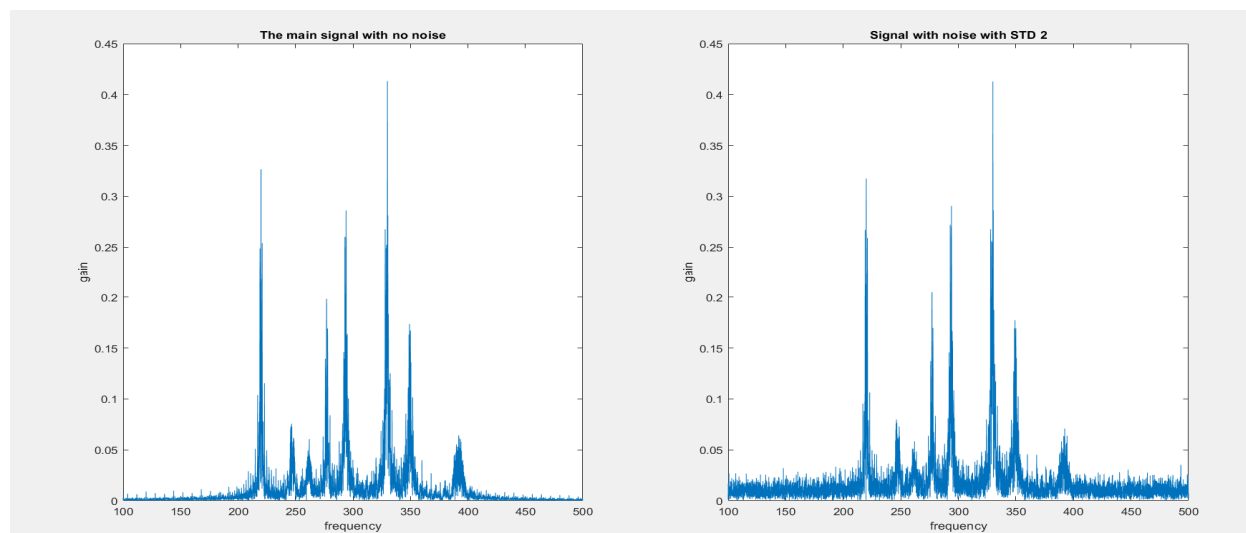
افزودن نویز:

در بین 3 نویز افزوده شده به interval note, gain, کمترین تأثیر به نظر من نویز روی دامنه است. نویز بر روی note بعضی جا ها کلن آهنگ فرق میکند و قابل تطابق با فایل اصلی نیست. نویز بر روی interval هم مخرب بود و مدت زمان هر نت که فرق میکرد آهنگ قابل تشخیص نبود و در بسیاری جا ها نت ها خیلی دراز تر بودند.

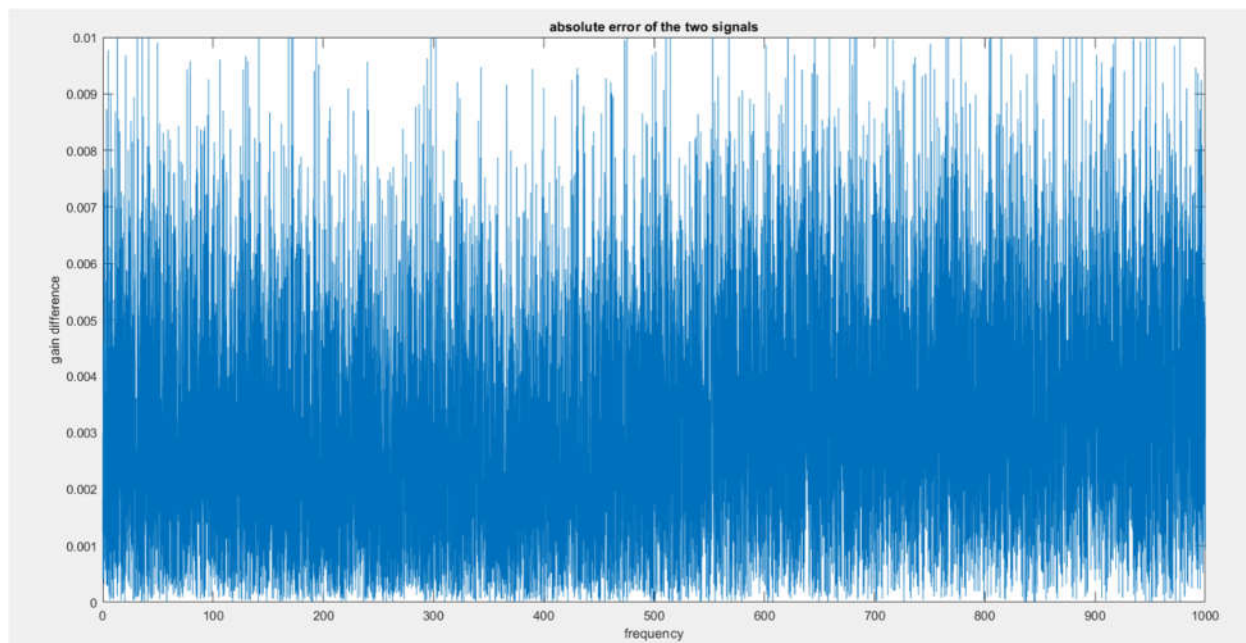
در افزودن نویز به بردار نهایی مشاهده میشود که ریتم و آهنگ موسیقی فرق نکرده و تنها یک خش خش پس زمینه اضافه شده.

در مورد بیشینه توان نویز میتوان گفت که تا زمانی که انحراف از معیار نویز 20 است هنوز صدای موسیقی شنیده میشود اگر چه بسیار کم و تشخیص آن سخت است. پس بردار m2 را اضافه کردن نویزی با واریانس 16 به بردار صدای نهایی m1 میسازیم. فایل صوتی این بردار صوتی تحت عنوان m2 music در پوشه به پیوست است.

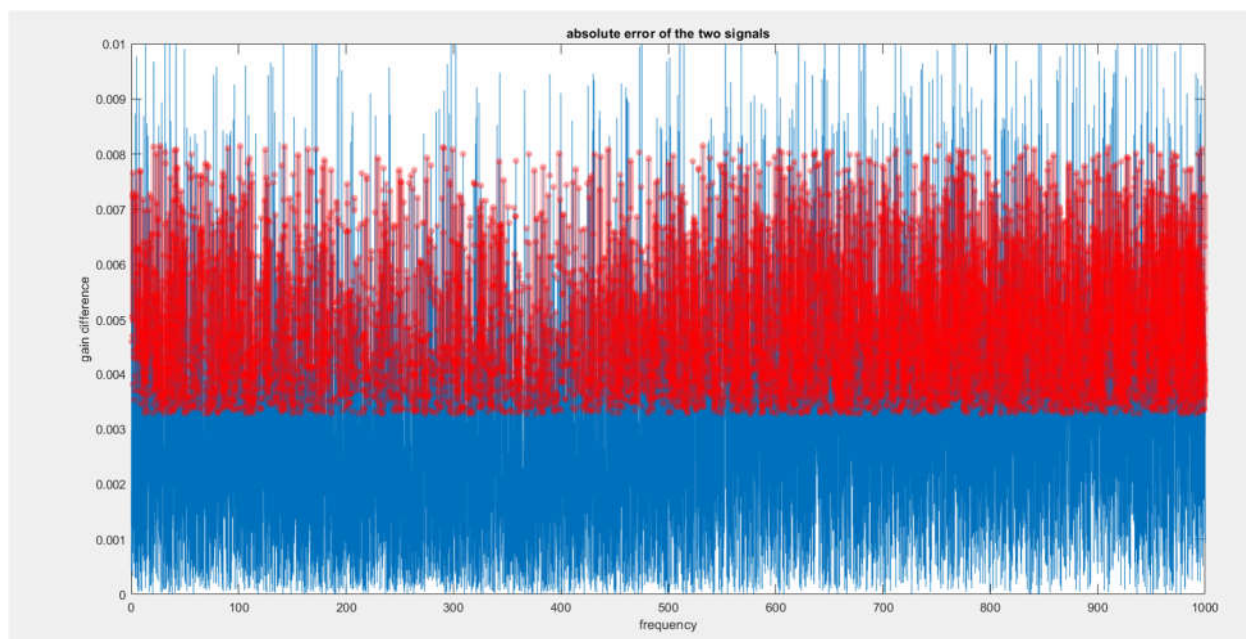
تبدیل فوریه دو بردار را با دستور fft میگیریم و نمودار خروجی را بر حسب فرکانس میکشیم.



مشاهده میشود که بر روی سیگنال اصلی ما فرکانس های دیگری اضافه شده این و بر کل سیگنال دامنه فرکانس ها با فرکانس اصلی فرق میکند که این تفاوت در فرکانس های بالا و پایین مشهود تر است. نمودار تفاوت این 2 بردار به شکل زیر است:



با کمی دقت در نمودار بالا میتوان دریافت که اثر نویز در محدوده 200 تا 400 که مقادیر بردار اصلی نیز در بازه بیشتر قرار دارند، کمتر است (با دقت در چگالی خطوط). این تفاوت در شکل زیر مشهود تر است که چگالی در بازه ی 200 تا 400 کمتر است.



نمودار تبدیل فوریه بردار اصلی، را میتوان به صورت مجموعی از دلتا های شیفته دانست. از ریاضی مهندسی به یاد داریم که تبدیل فوریه سینوس و کسینوس به صورت دلتا میباشد پس شکل مذکور را میتوان به صورت تبدیل فوریه مجموع تعدادی سینوی و کسینوس دانست.

البته تبدیل سینوس و کسینوس هر 2 دارای 2 دلتا هستند که یکی در فرکانس های منفی میباشد که از آنجایی که ما اندازه ی بردار خروجی fft را استفاده کرده ایم فرکانس های منفی در شکل ما موجود نیستند.

تفاوت اول را شاید بتوان وجود فرکانس های دیگر به جز 8 فرکانس اصلی (که در بردار $note$ بودند) در طیف فرکانسی دانست. چرا که اگر بردارها و داده ها ایده آل بودند در شکل خروجی ما تنها بایستی 8 پیک دلتا داشته باشیم اما در خروجی ما بین این 8 نقطه بیشینه نیز فرکانس داریم. البته دامنه آن ها بسیار کم است.

برای فیلتر نیاز داریم که فرکانس هایی که در بردار اصلی ما نیستند را حذف کنیم. بردار فرکانس اصلی داده، شامل فرکانس از 220 تا 392 است. پس اگر فیلتر ما دارای فرکانس قطع 400 باشد به احتمال خواهیم توانست نویز را تا حدودی حذف کنیم.

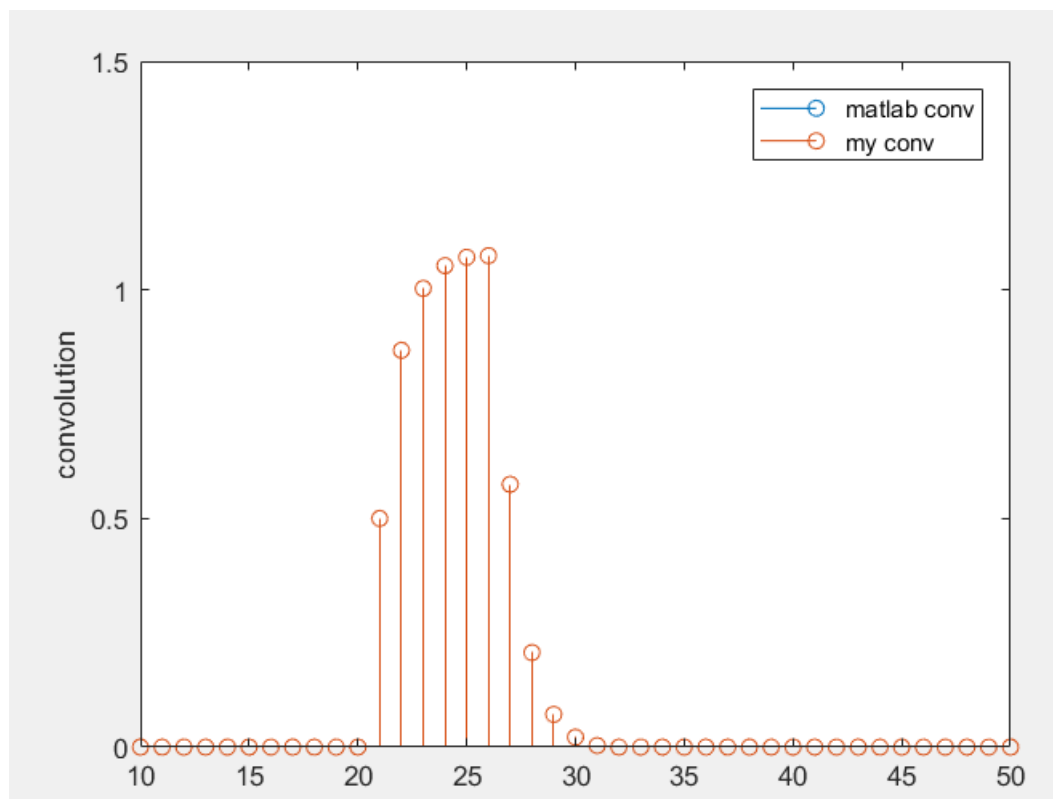
اگر البته بخواهیم که فیلتر برای هر موسیقی کار کند شاید بهتر باشد تا فرکانس قطع را 4000 هرتز انتخاب کنیم چرا که ریز ترین صدای پیانو در حد 3700 هرتز است.

از آنجایی که تبدیل فوریه تابع sinc تابع rect میباشد و convolution در حوزه زمان به ضرب در حوزه فرکانس میانجامد پس اگر ما بردار صدای خود را با تابع sinc کانوالو بکنیم پس در حوزه ی زمان بقیه ی فرکانس های ناخواسته را میتوان حذف کرد. برای این کار بردار تابع $\text{sinc}(400*t)$ را تشکیل میدهم و با استفاده از تابع درونی conv متلب، حاصل کانولوشن این 2 بردار می یابیم. البته بردار خروجی حدود 2 برابر اندازه بردار اصلی میباشد که ما تنها به مقدار تعداد اعضای بردار اولیه از آن را نیاز داریم. خروجی به پیوست آمده است.

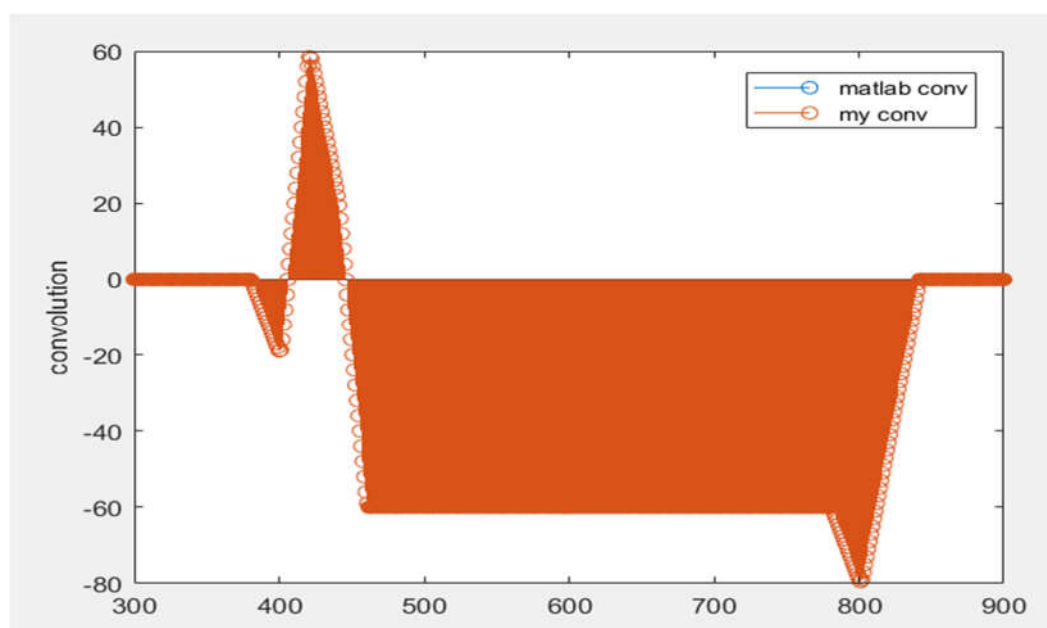
سوال 2

تابع `myconv` را با استفاده از تابع `convmtx` میسازیم. این روش کاملاً دقیق بوده و در تمامی نمودار های پیوست، 2 نمودار ناشی از تابع نوشته شده `myconv` و تابع درونی `conv` کاملاً بر هم منطبق اند.

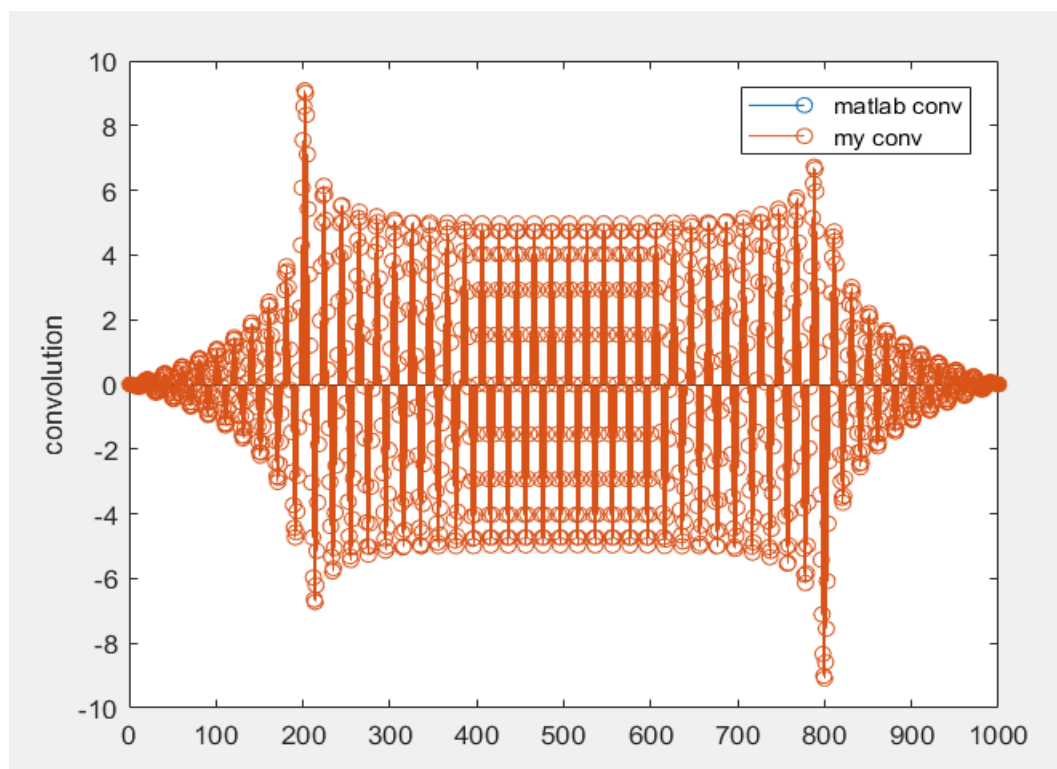
قسمت الف)



قسمت ب)



قسمت ج)



قسمت د)

