

ب نام خدا

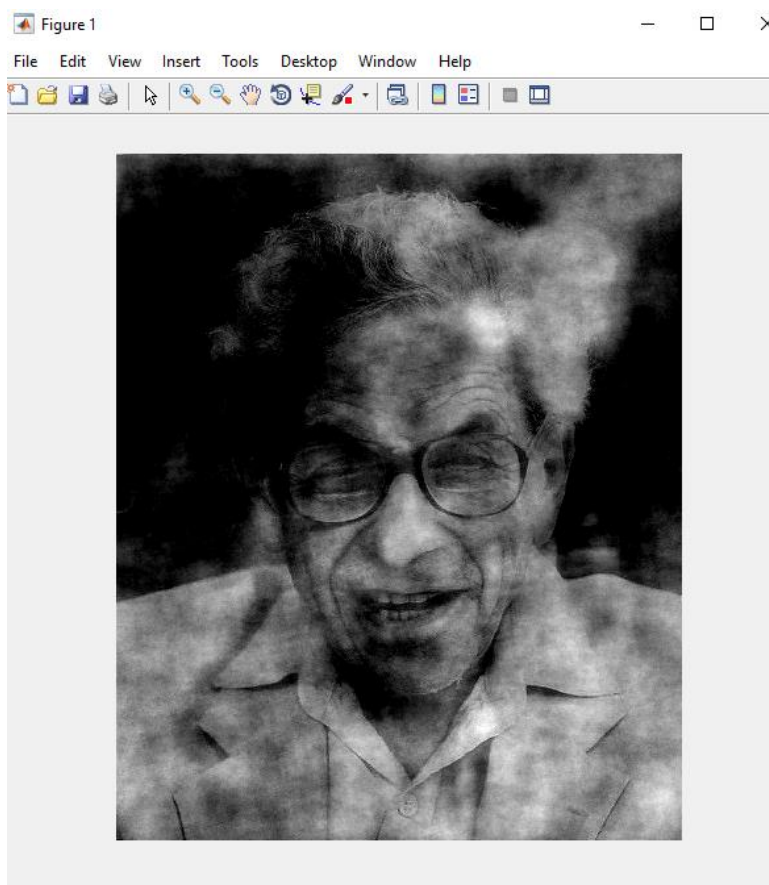
گزارش تمرین سوم متلب دکتر کربلایی

نگین اسماعیل زاده ۹۷۱۰۴۰۳۴

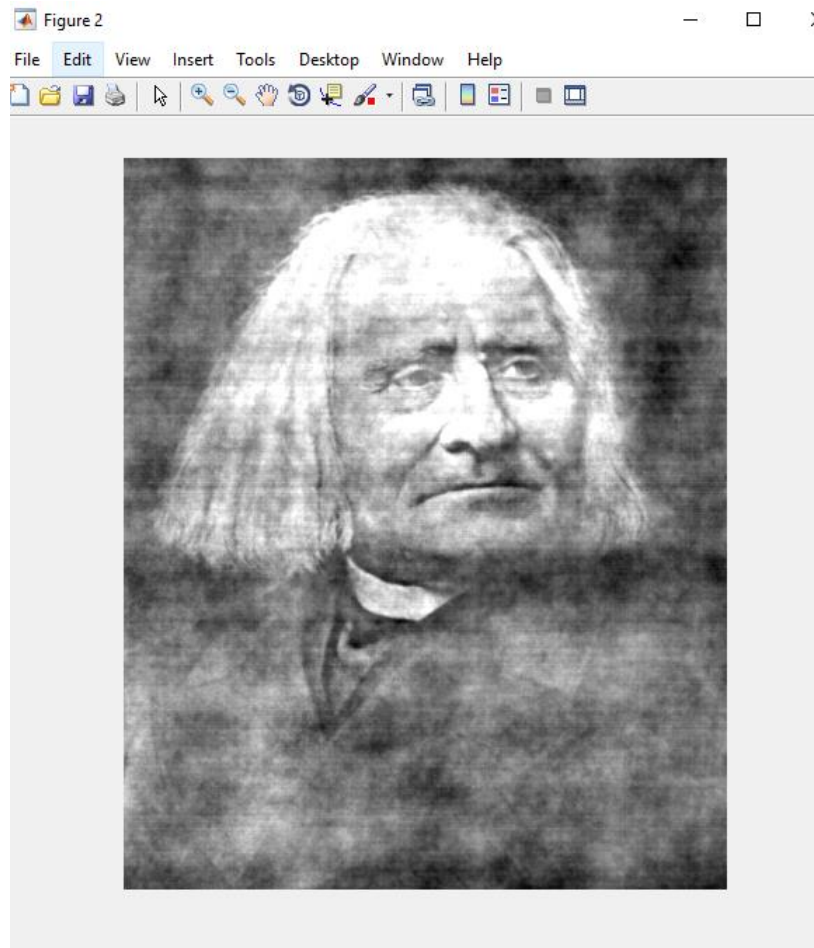
۱) توسط `imread` تصاویر را در متلب به صورت ماتریس دریافت میکنیم. توسط `fft2` تبدیل فوریه دو بعدی آنها را حساب میکنیم سپس اندازه `(abs)` و فاز `(angle)` آنها را جابجا کرده و توسط دستور `ifft2` از آنها وارون میگیریم و توسط `imshow` دوباره نمایش میدهیم.

نتیجه این بخش این است که تصویر نهایی به صورت کلی شبیه تصویری خواهد شد که فاز مشابه دارد و اندازه تبدیل فوریه اثر خود را خیلی کمتر (اطلاعات مربوط به کیفیت تصویر، میزان سایه ها یا روشنایی ها) میگذارد. نتیجتاً به نظر میاید محتوای اصلی در فاز تبدیل فوریه قرار دارد.

تصویر با فاز تصویر اول و اندازه تصویر دوم:



تصویر با اندازه تصویر اول و فاز تصویر دوم:



در واقع برای بازسازی کامل یک تصویر هم به فاز هم به اندازه نیاز داریم اما تاثیر فاز در شکل کلی بیشتر است اندازه به نوعی مقدار سیاه سفید بودن را مشخص میکند. همانطور که در تصاویر بالا هم میبینیم اثر تخریبی که استفاده از دامنه تصویر دیگر گذاشت کمی تغییر رنگ است به طوری که انگار کمی از طیف رنگی (هاله ها) مربوط به تصویری که از اندازه ی آن استفاده کردیم وارد عکس جدید شده است. در اکثر کاربرد ها معمولا تغییرات را روی اندازه اعمال میکنیم چون اطلاعات درون فاز اطلاعات اصلی مربوط به تصویر است و اطلاعات درون اندازه مربوط به کیفیت است

۲) در این سوال از حلقه استفاده شده تا میزان اوپتیمم چرخش و انتقال را پیدا کنیم ، اما به دلیل طولانی شدن زمان اجرا حلقه اول که با دقت کمتری بین مقادیر حرکت میکند و حلقه دوم که اعشار را تعیین میکند در چند section جدا زده شده اند (برای راحتی در توقف در هر مرحله از دقت مطلوب) که کافیت پشت سر هم اجرا شوند تا به نتیجه مطلوب برسیم. در مرحله آخر هم با اجرا کردن section آخر تصاویر قابل نمایش اند.

در این بخش از توابع `imshowpair` برای نشان دادن دو تصویر در کنار یا روی هم در خروجی ، `imref2d` برای ذخیره ابعاد تصویر و ارجاع آن به عنوان ورودی به تابع `imwarp` (که تغییرات انتقال یا چرخش را روی تصویر ایجاد میکند) به این علت که ابعاد تصویر بعد از اعمال تغییر حفظ شوند و تابع `corr2` برای محاسبه مقدار کورلیشن تصویر دو بعدی استفاده شده است. همچنین ساخت ماتریس تبدیل برای تصاویر دو بعدی (در اینجا انتقال یا چرخش) توسط `affine2d` شده است.

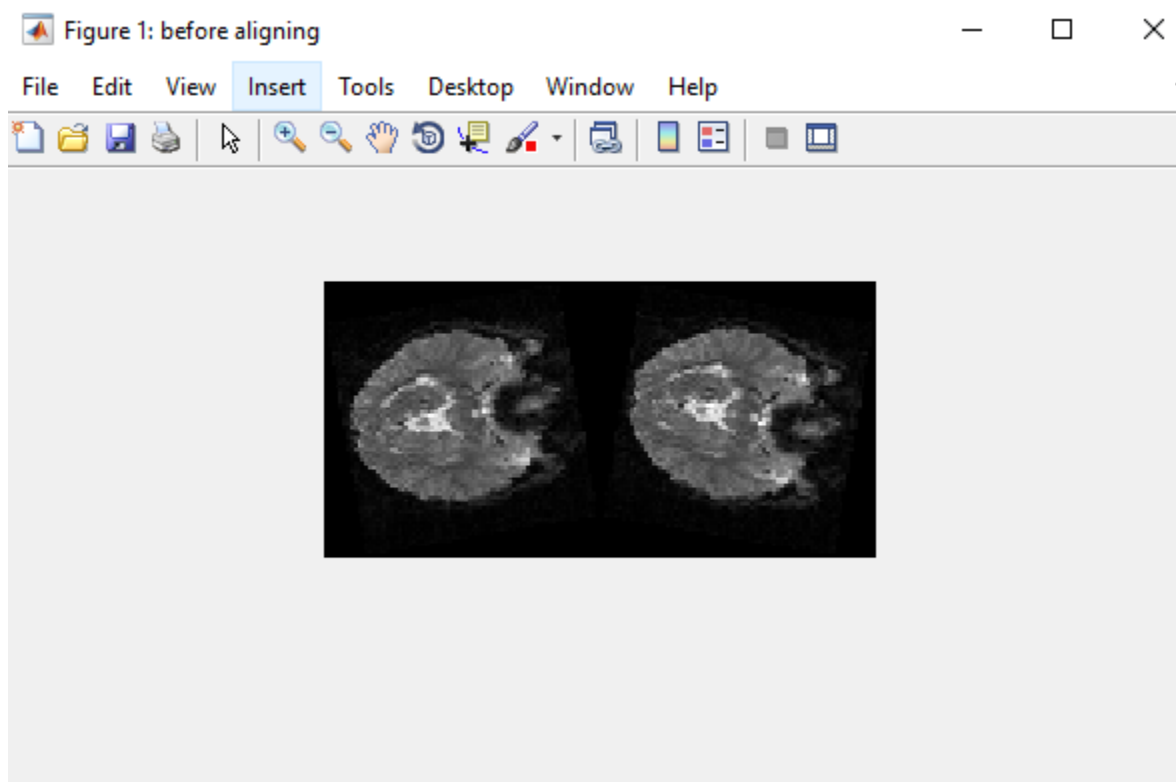
چون با دقت به تصویر میبینیم که نسبت انتقال لازم به ابعاد تصویر احتمالا کم است این محدوده را کوچک در نظر گرفتم تا زمان اجرا کم شود ، از انجایی که کورلیشن ماکزیمم در این محدوده پیدا شد نیز اطمینان پیدا میکنیم که خارج از این محدوده شباهت رو به کاهش بوده است. (مسیر کوچک کردن بازه به این شکل که ابتدا بازه کوچکی حول صفر میگیریم و میبینیم که به کدام حد(بالا یا پایین) محدود میشود سپس حول آن حد بازه ی دیگری را میگیریم) . نتایج مقدار X و Y انتقال و θ چرخش (ساعتگرد و به درجه) به این صورت اند :

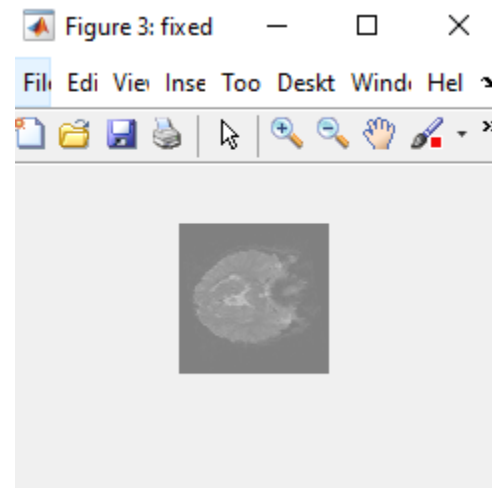
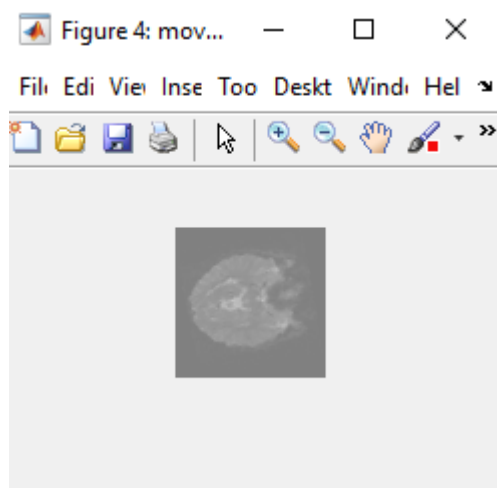
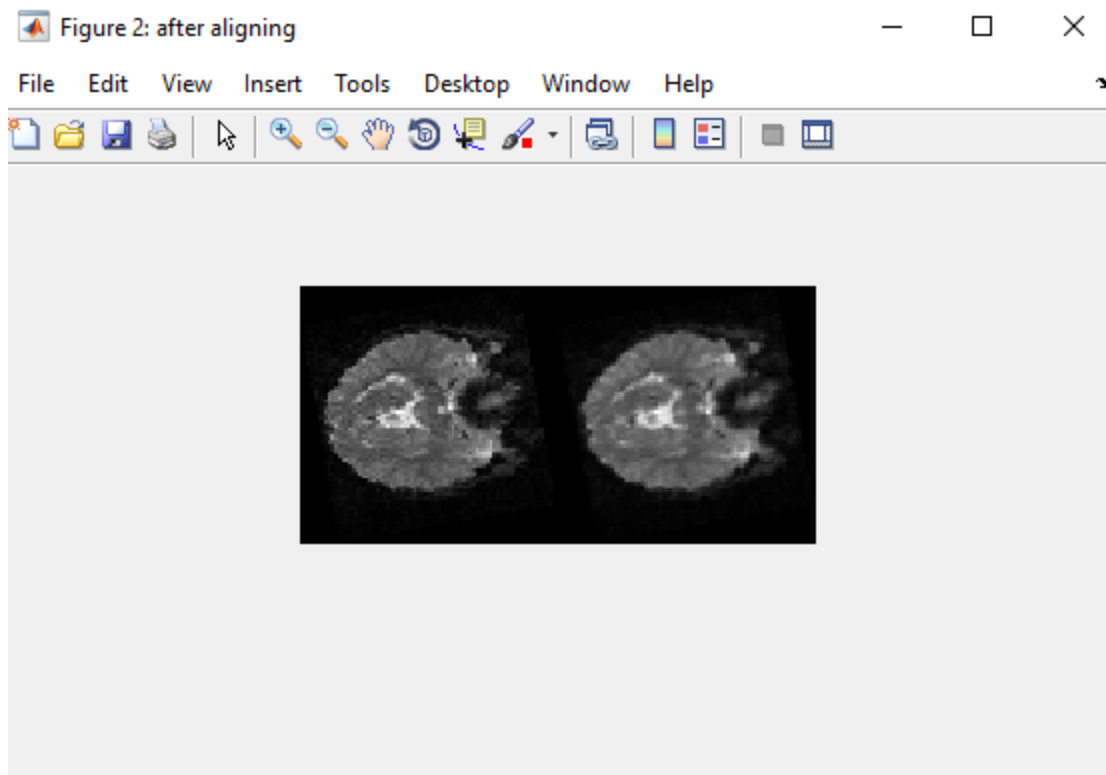
$$X = -10.909$$

$$Y = 15.449$$

$$\theta = 339.834$$

$$\text{Correlation} = 0.9783$$



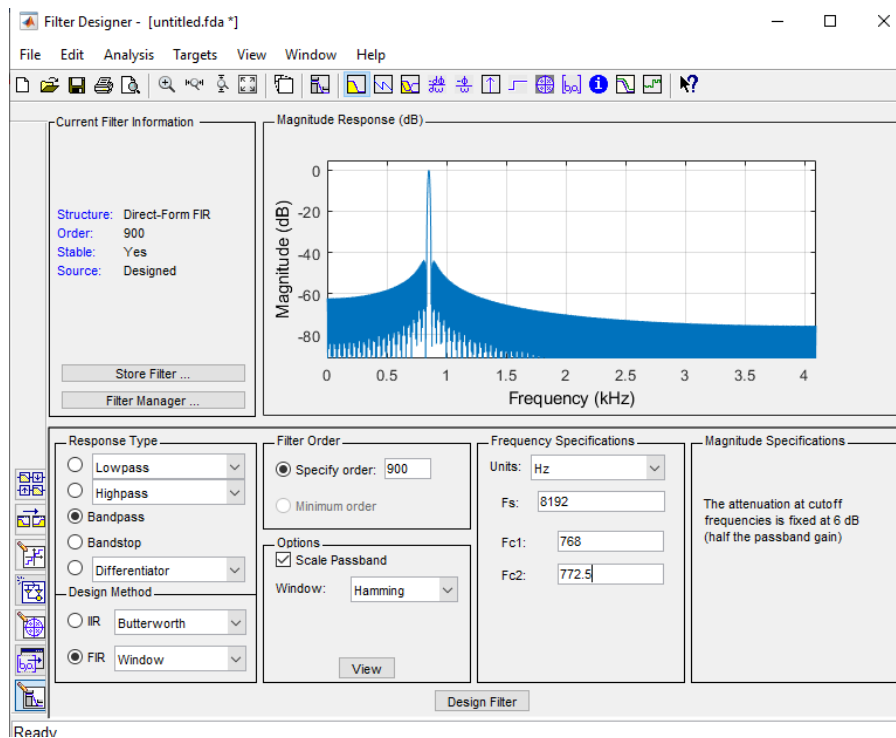
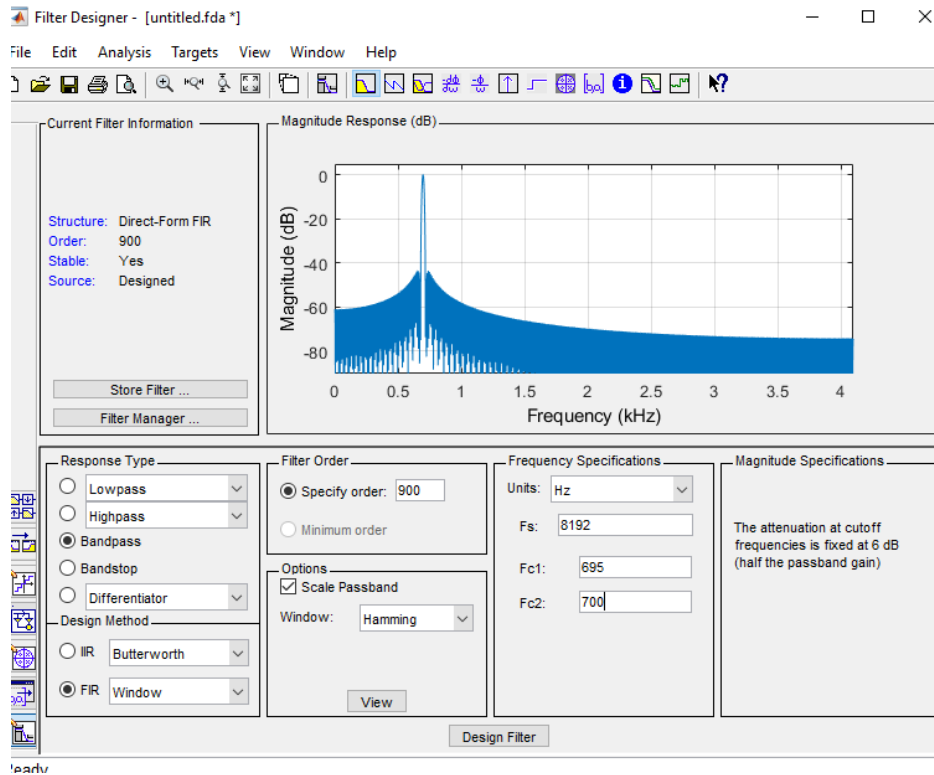


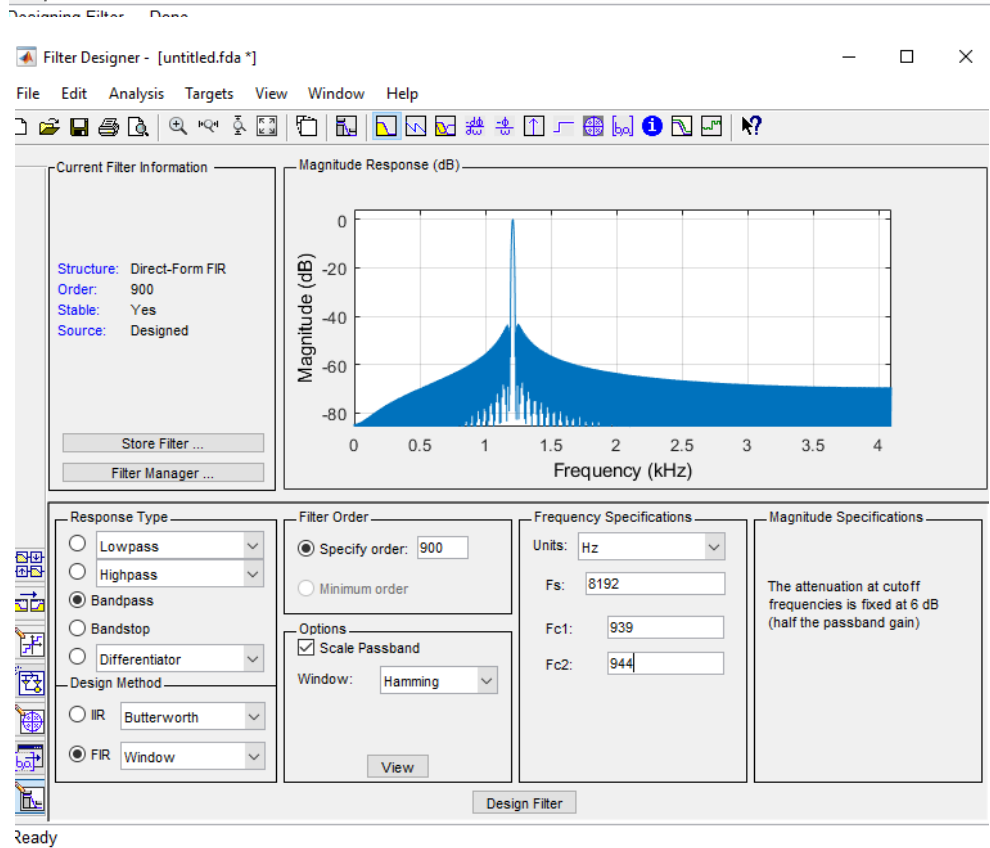
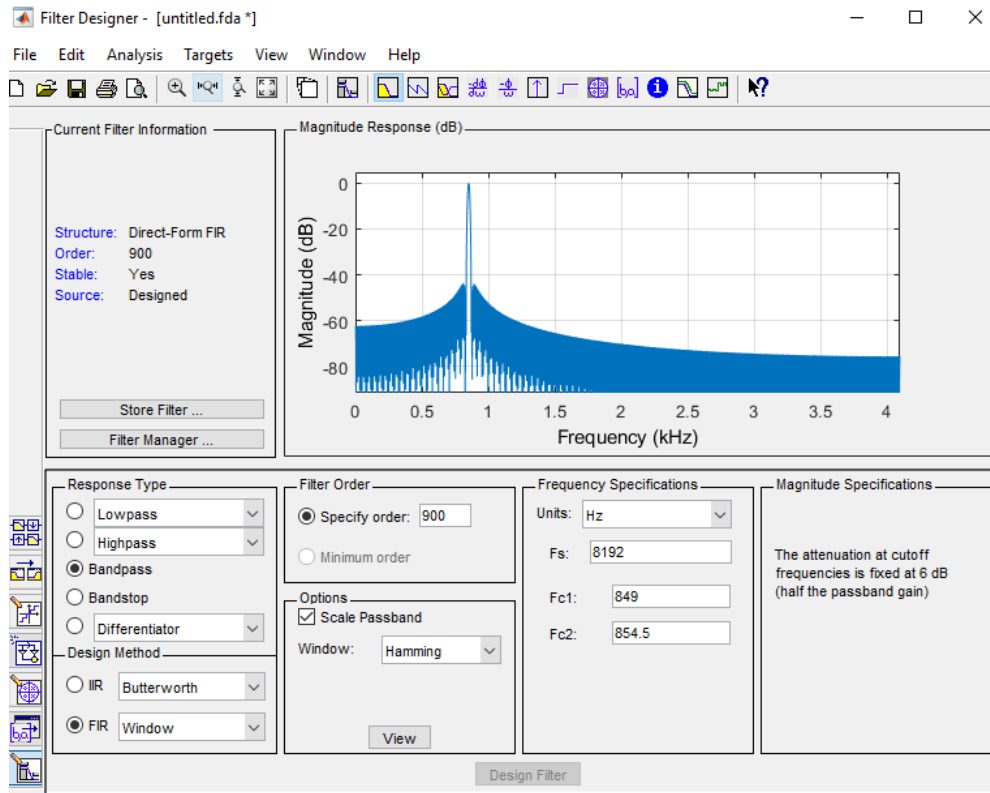
ماتریس شکل دوم بعد تغییرات نیز ذخیره شده است .

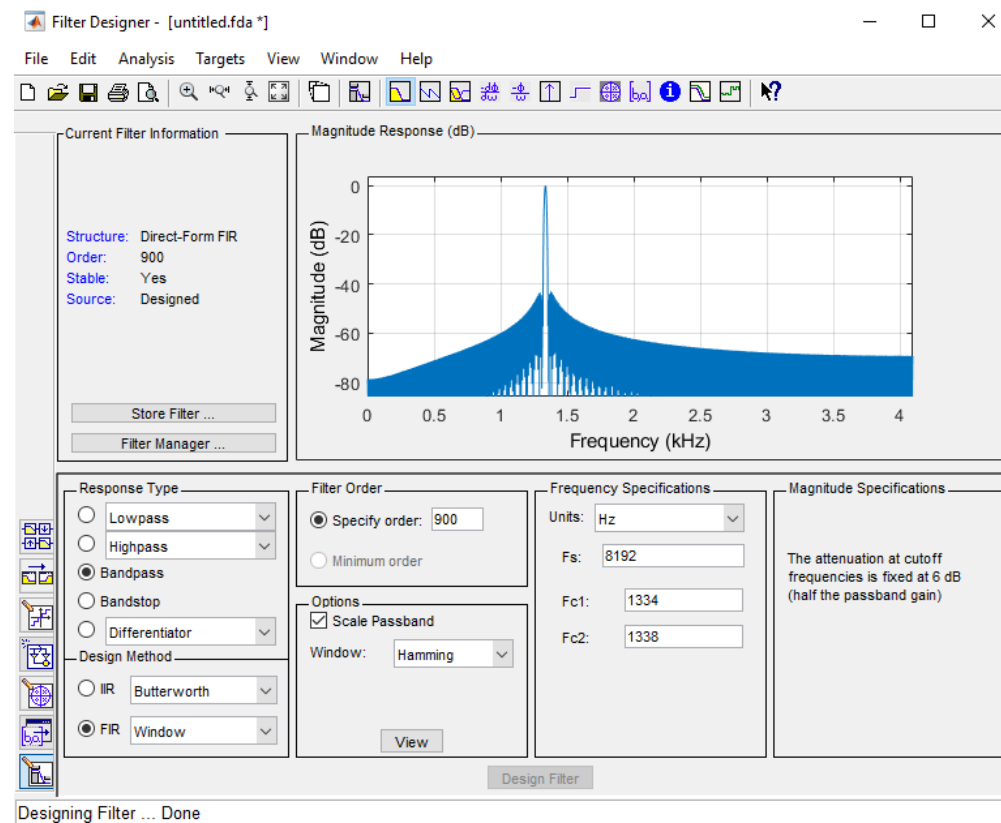
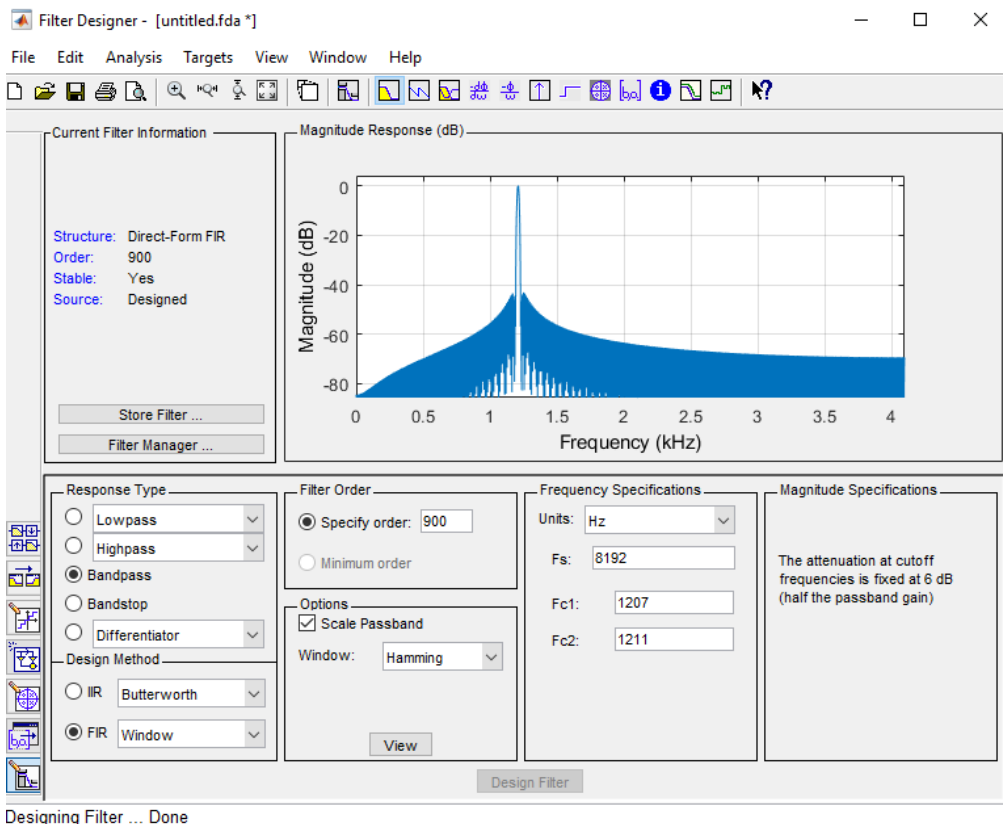
ماتریس چرخش و انتقال :

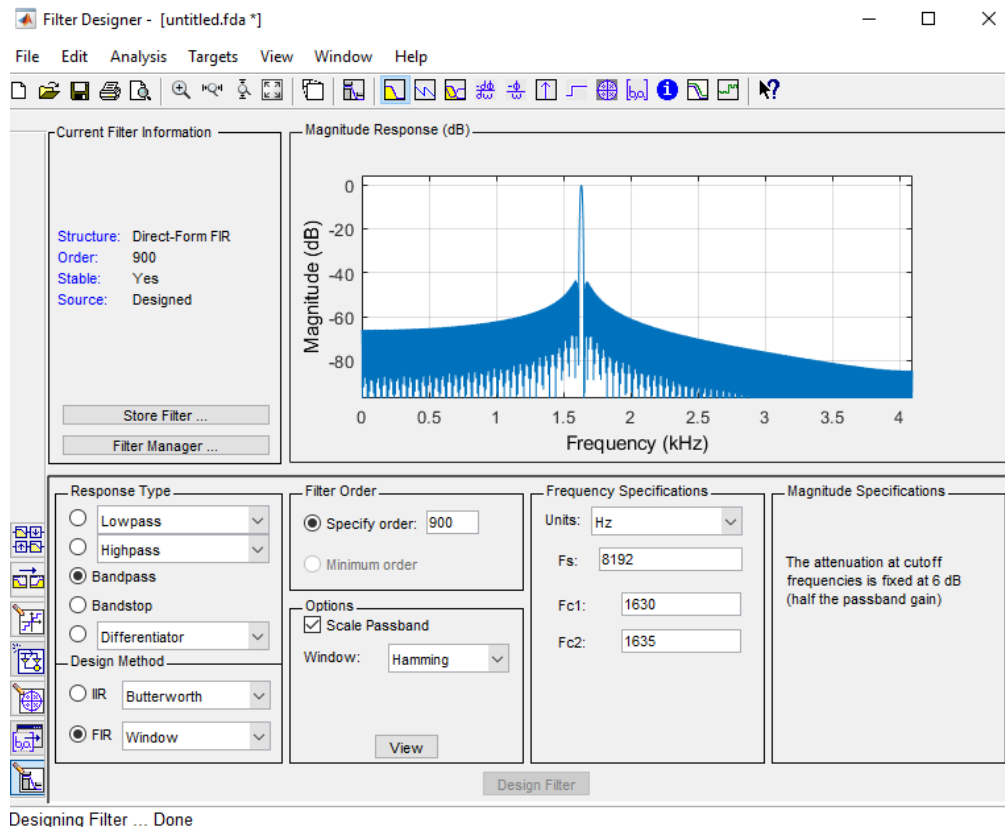
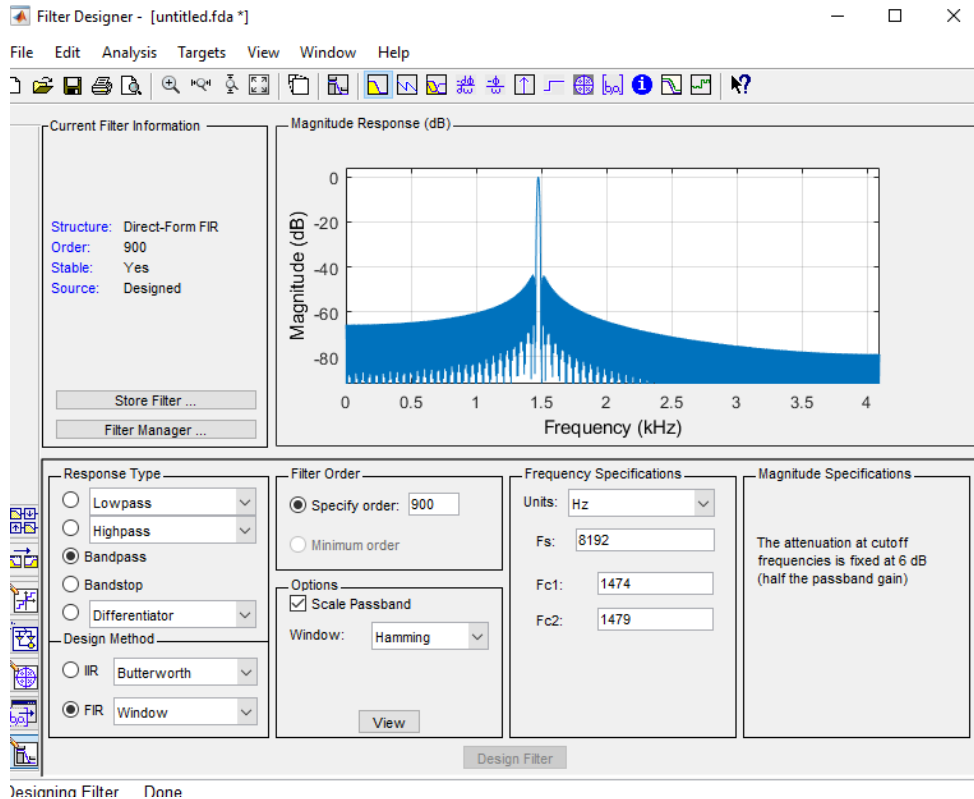
$$\begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ \sin(\theta) & -\cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ X & Y & 1 \end{bmatrix}$$

۳) برای طراحی فیلتر در این بخش از فیلتر های bandpass با پهنای باند کم نوع همینگ با طول ۹۰۰ استفاده شده که تصاویر آن آمده است



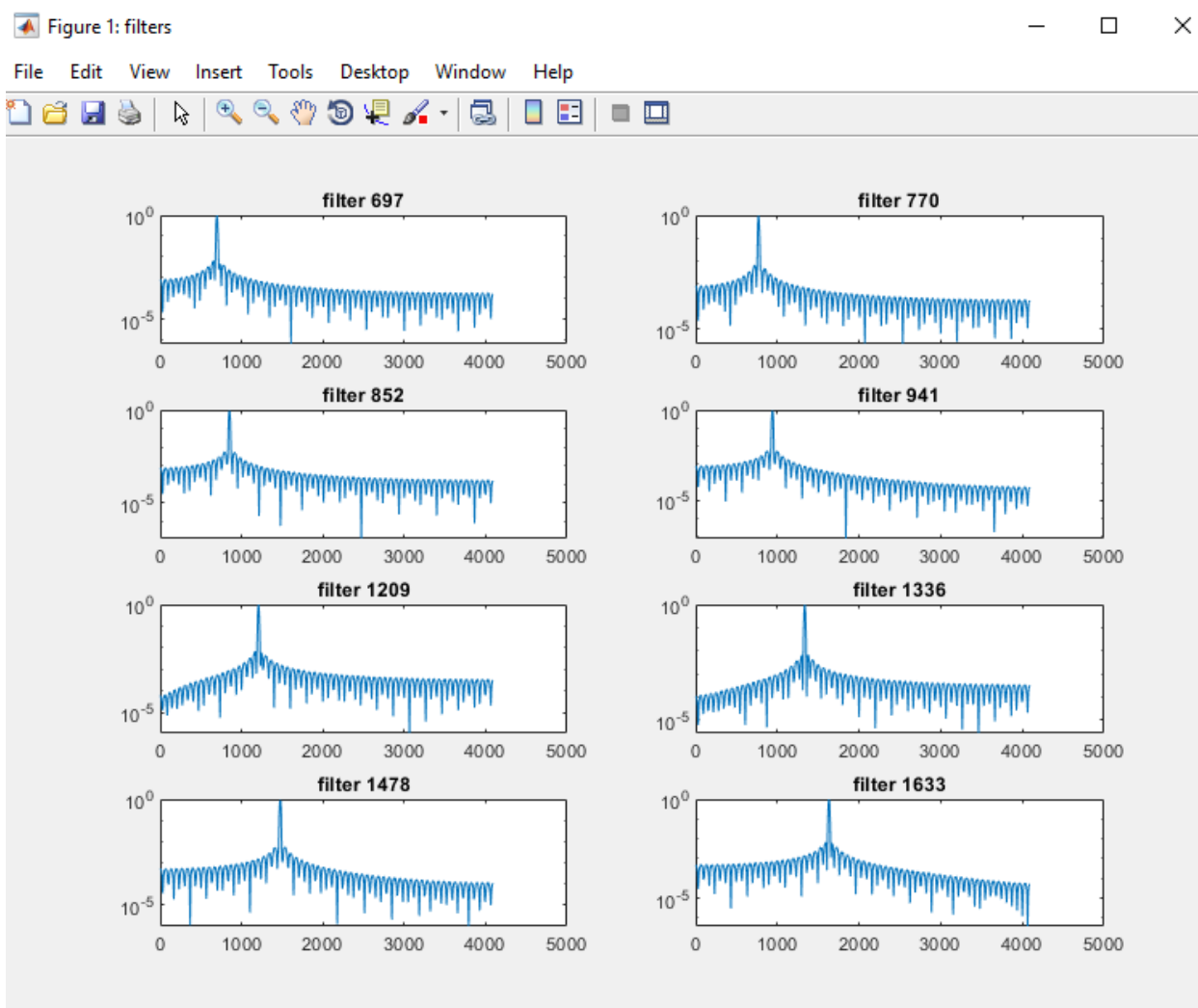




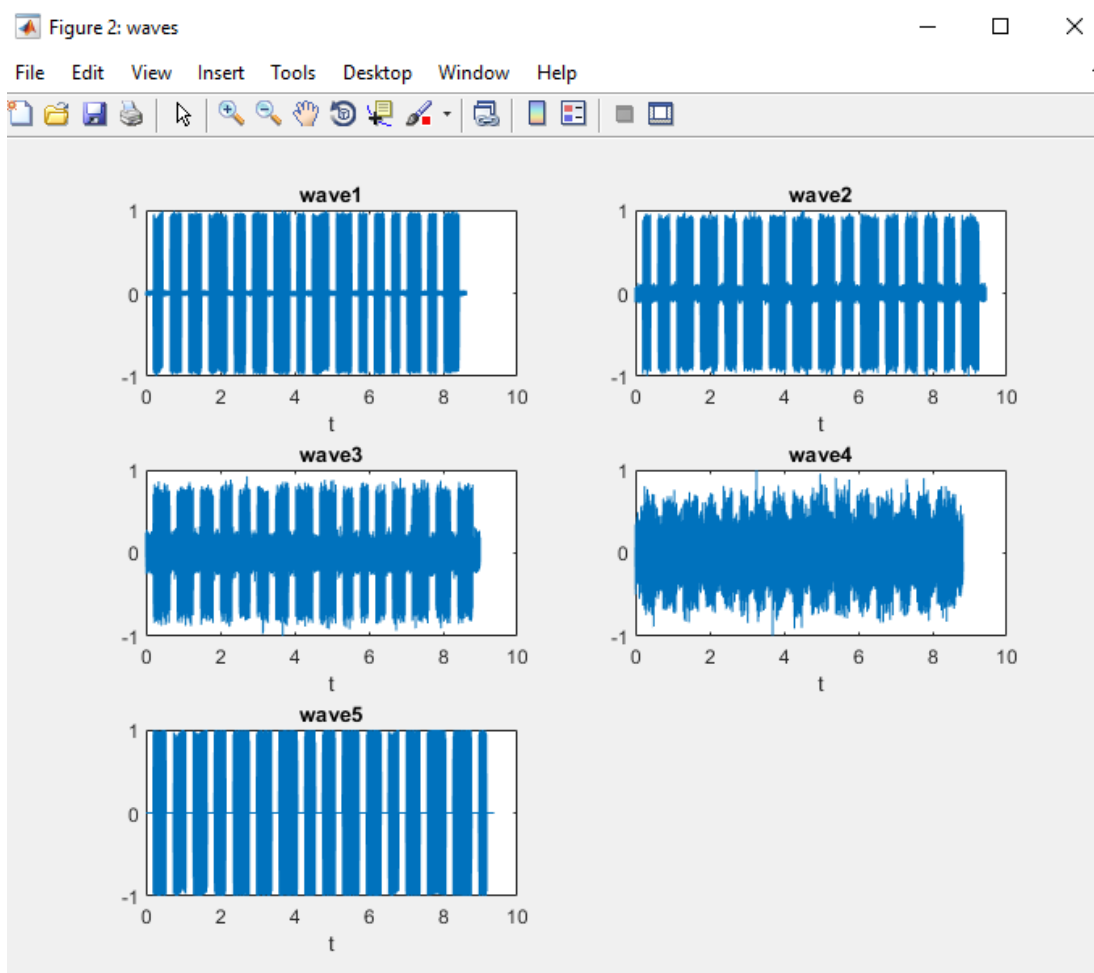


فرکانس های مربوط به رخ دادن قله narrow filter ها (به ترتیب) :

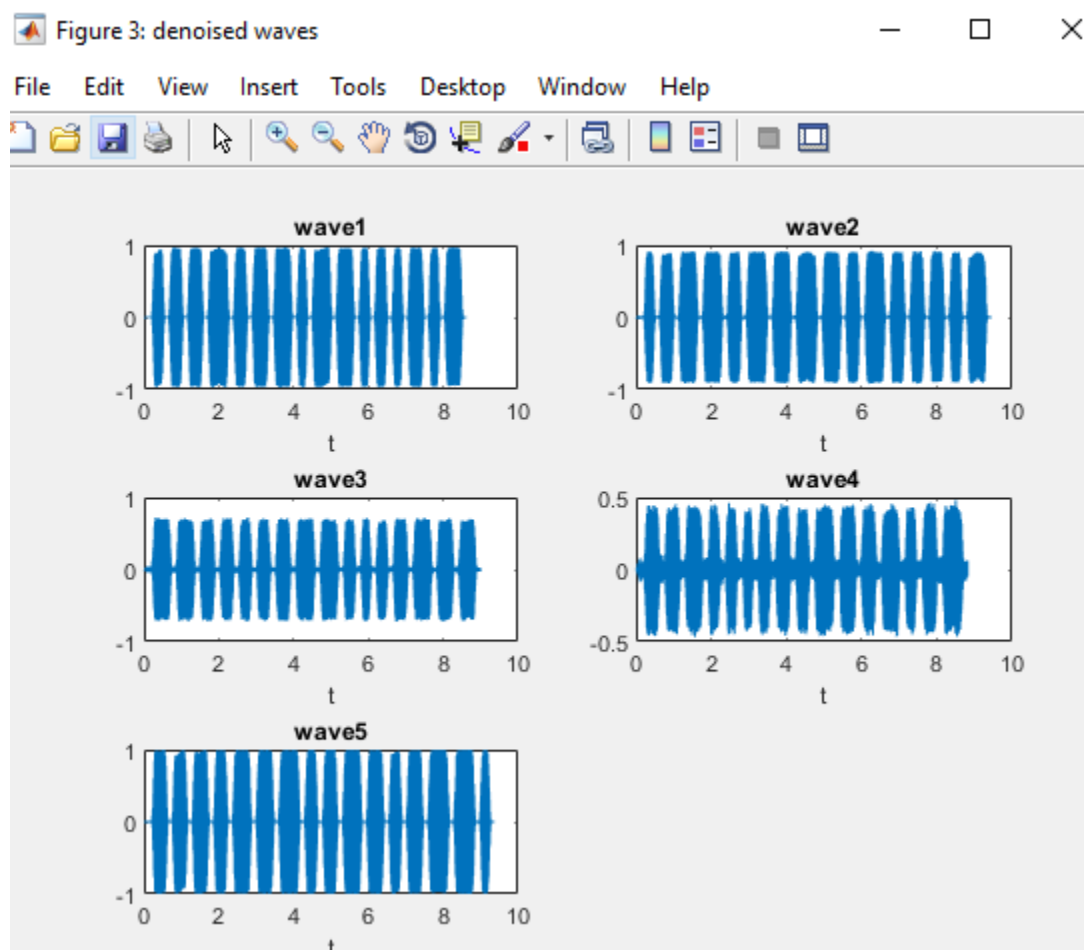
m1	696
m2	768
m3	848
m4	944
m5	1208
m6	1.3360e+03
m7	1480
m8	1632



پاسخ های زمانی صوت ها قبل از اعمال فیلتر :

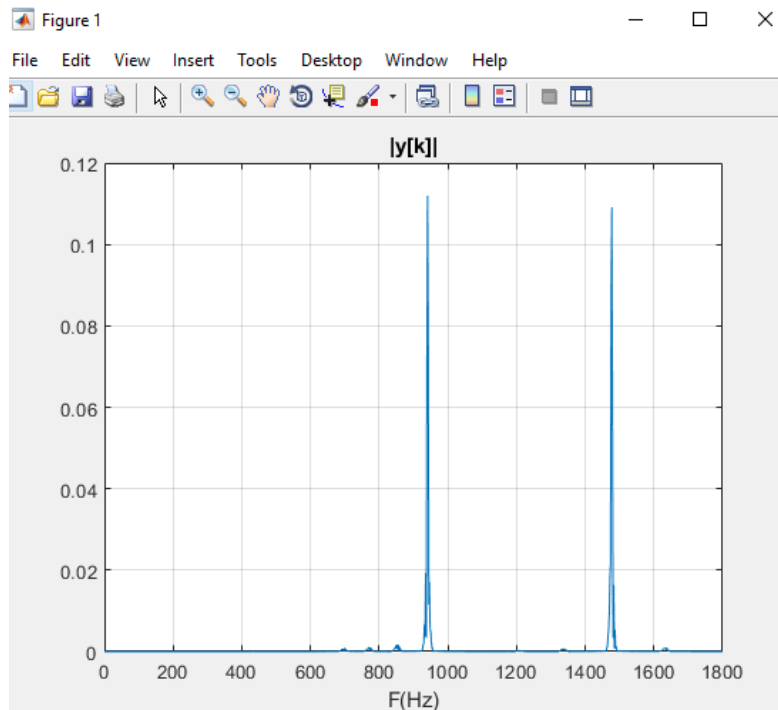


بعد از فیلتر :



در این بخش ابتدا توسط تابع **mute** زمان های شروع دکمه فشردن را حساب میکنیم (تشخیص سکوت ها) سپس به بخش های کوچک ترتبدل کرده و هر بار فشردن دکمه را دی کد میکنیم.

شکل زیر خروجی یکی از این بخش ها در حوزه فرکانس است همانطور که میبینیم رقم اولی که در موج اول فشرده شده جمع ۱۶۳۳ و ۶۹۷ هرتز است که دکمه ی # است .



شکل ها برای هر موج در بخش دیکود آن نمایش داده میشوند

خروجی ها به این صورت است :

برای موج اول :

#A7*60452918C3BD

موج دوم :

325640B*9187ADC#

سوم :

D176A354C90#8B2*

چهارم :

A13D475BC296#80*

پنجم :

4#206A78513*9DBC