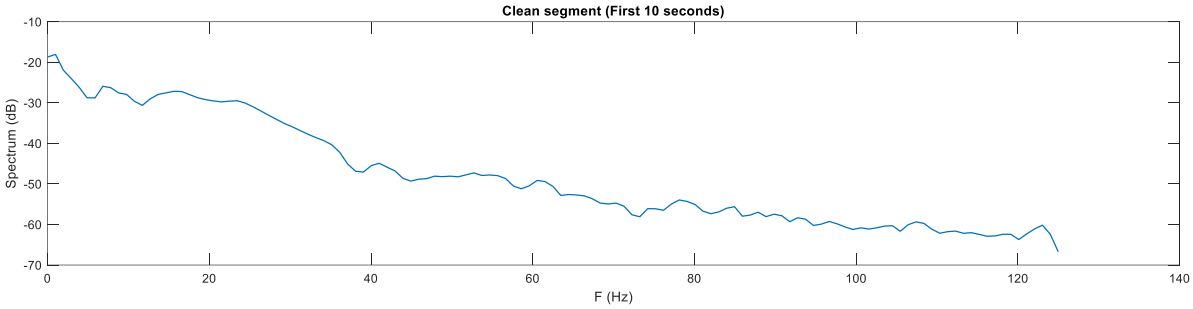
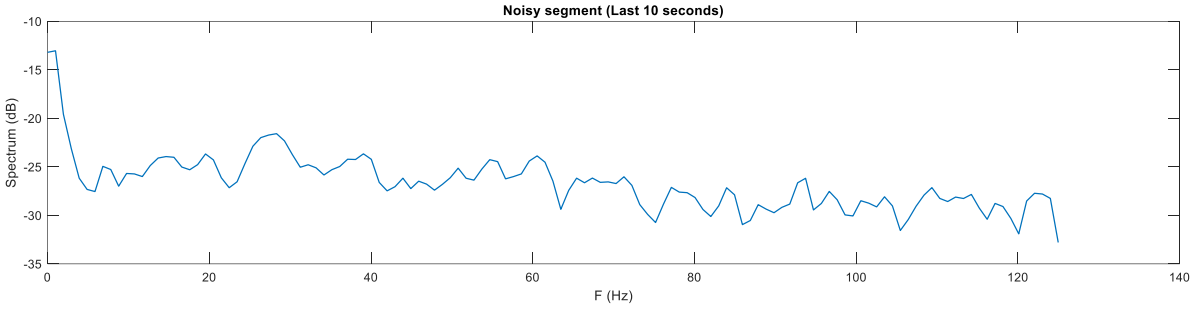
بخش اول: محدودسازی فرکانسی سیگنال/کاهش نویز

الف) برای سیگنال تمیز، ۱۰ ثانیه ی ابتدای سیگنال، و برای قسمت نویزی، ۱۰ ثانیه‌ی انتهای سیگنال انتخاب شده است (چون قسمت نویزی، یک دقیقه‌ی آخر داده گیری رخ داده است). با طول پنجره ی ۲۰۰، طیف های pwelch به صورت زیر برای سیگنال های تمیز و نویزی بدست می آید:



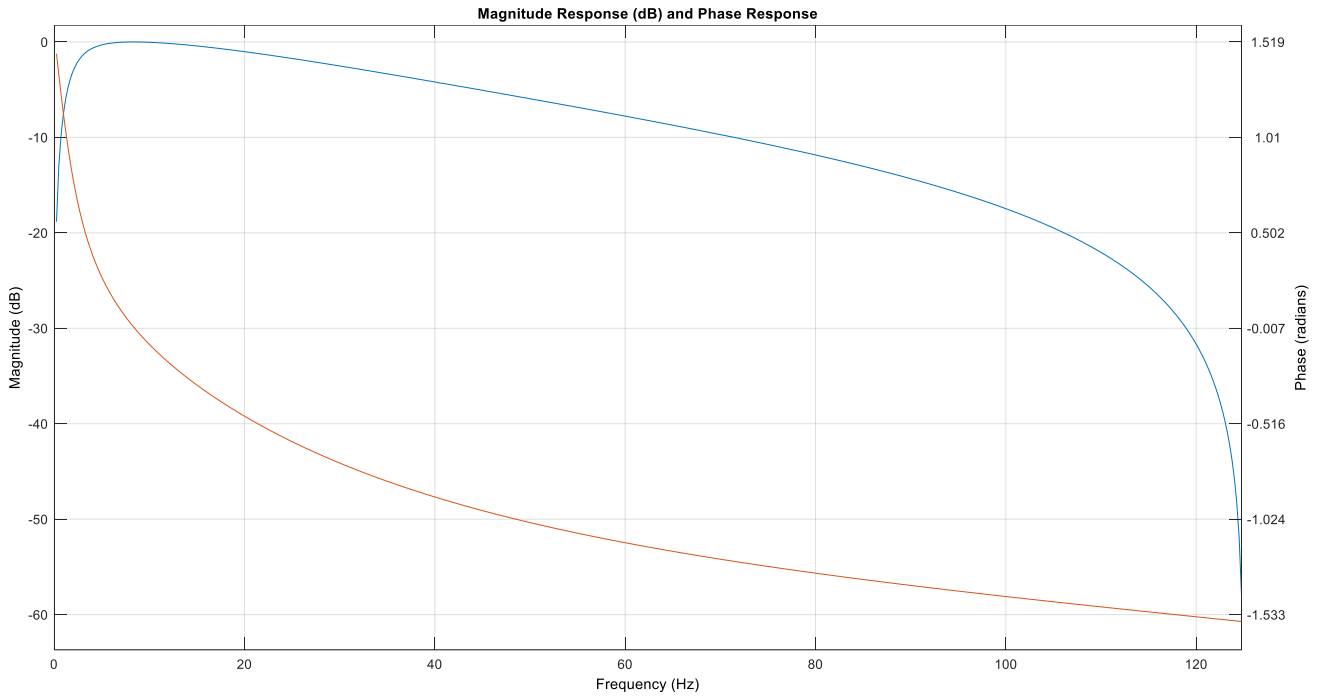


میتوان بخوبی افت توان در فرکانس های بالاتر را در سیگنال تمیز دید. اما در سیگنال نویزی، طیف بیشتر فرم flat دارد و از مقایسه مقدار توان نیز میتوان دید که توان نویزهای فرکانس بالا، به نسبت سیگنال تمیز، بیشتر است.

ب) در این بخش، فیلتر با استفاده از designfilt طراحی شده است. فرکانس قطع پایین برای حذف baseline، ۲ هرتز انتخاب شده است. برای تعیین فرکانس بالا، ابتدا

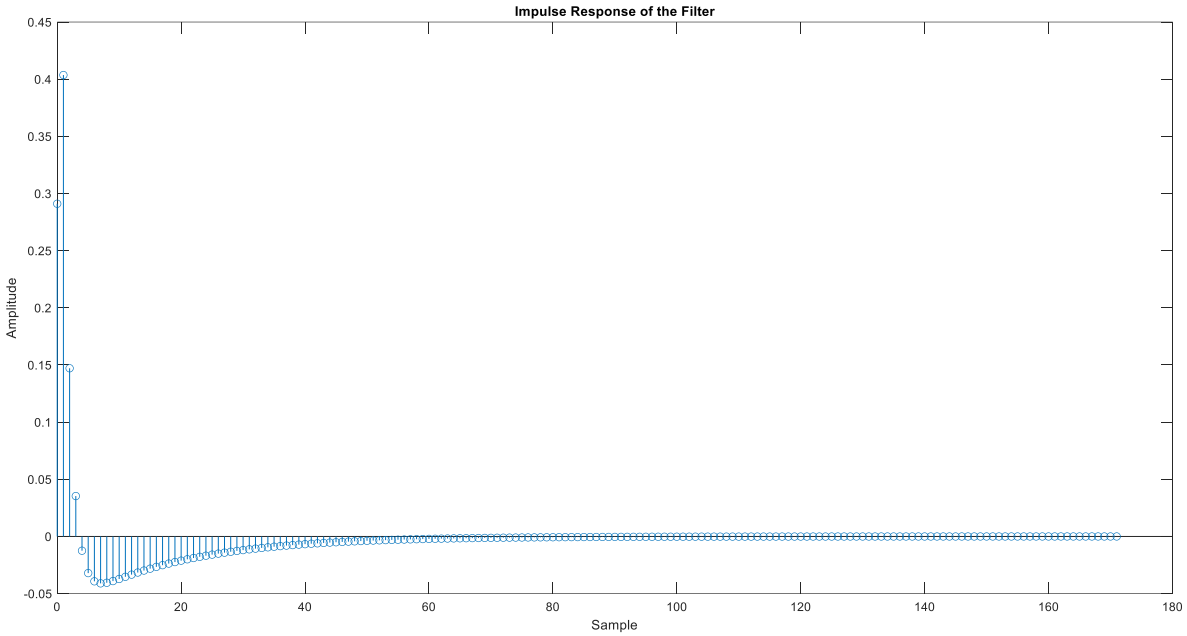
فرکانس بالا را بسایر بزرگ میگذاریم تا تنها baseline حذف شود. در این حالت انرژی سیگنال را حساب میکنیم. حال فرکانس قطع را در یک لوپ، از ۱۰ هرتز گام به گام افزایش میدهیم تا جایی که انرژی سیگنال فیلتر شده به ۹۰درصد انرژی سیگنال اصلی برسد.

در نهایت فرکانس قطع بالا برابر با ۳۳هرتز و پاسخ فرکانسی به صورت زیر بدست می آید، که نمودار آبی رنگ دامنه و نمودار قرمز رنگ فاز آن است:

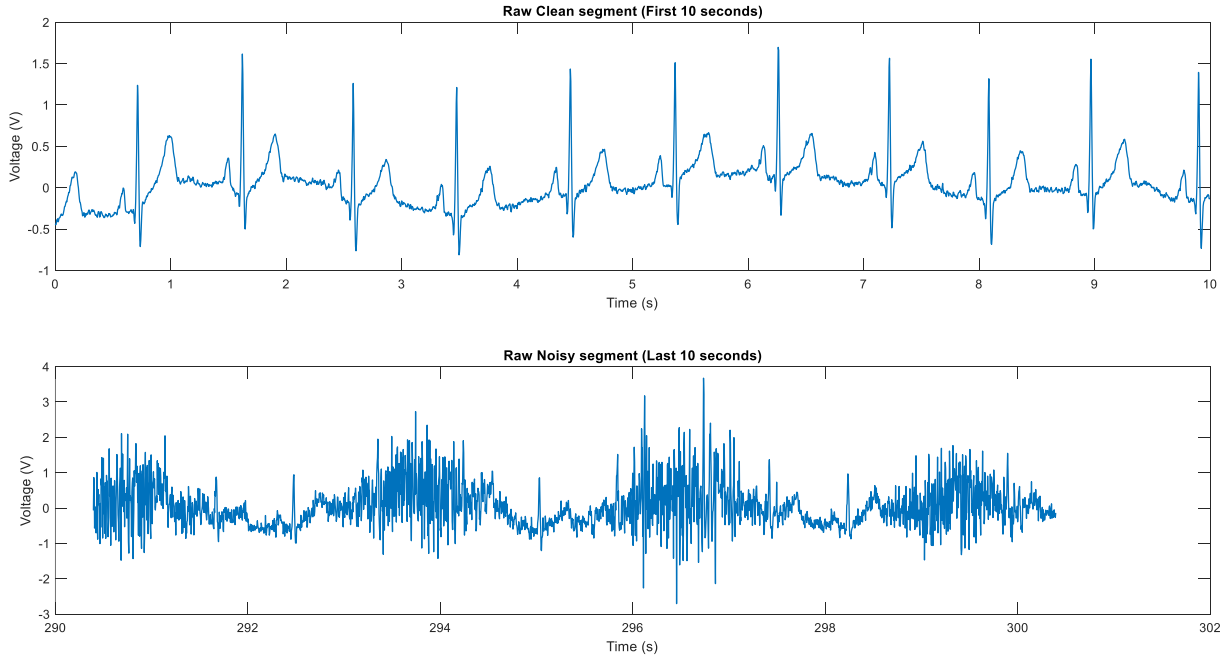


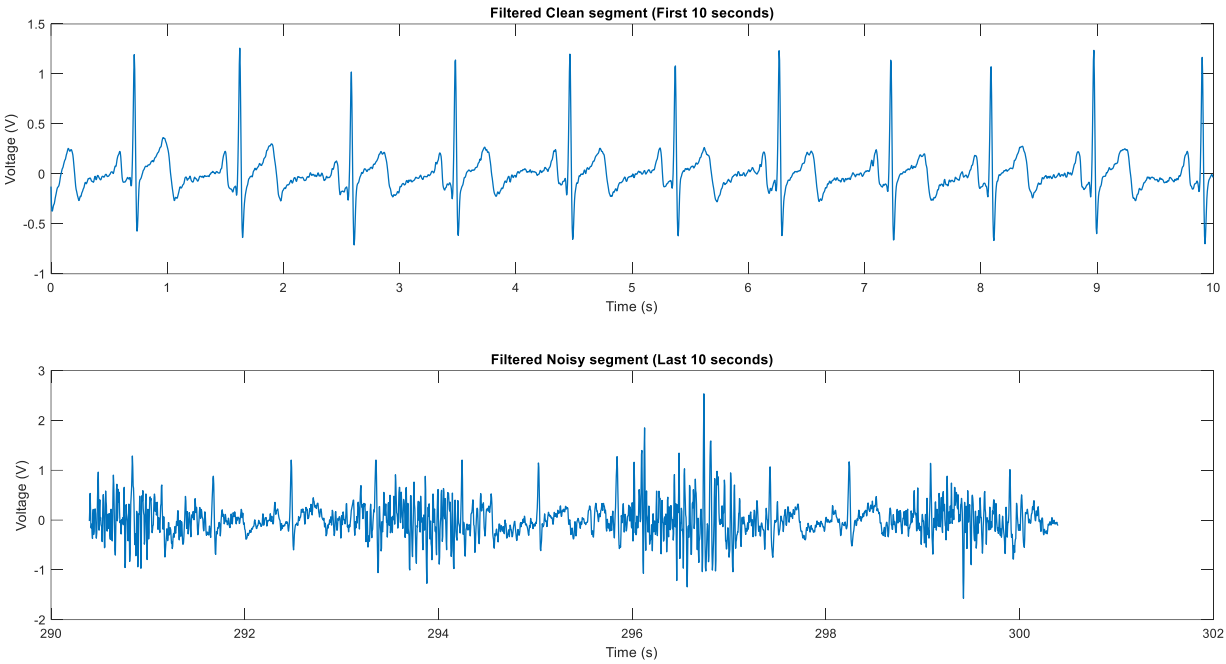
توجه داریم که فرکانس قطع های تعیین شده (۲ و ۳۳ هرتز)، فرکانس های قطع نصف توان هستند.

پاسخ ضربه فیلتر نیز به صورت زیر است:



ج) در زیر، نمودار اول مربوط به دیتاهای قبل از فیلتر (بخش تمیز و نویزی)، و نمودار دوم مربوط به دیتاهای فیلتر شده (بخش تمیز و نویزی) هستند:





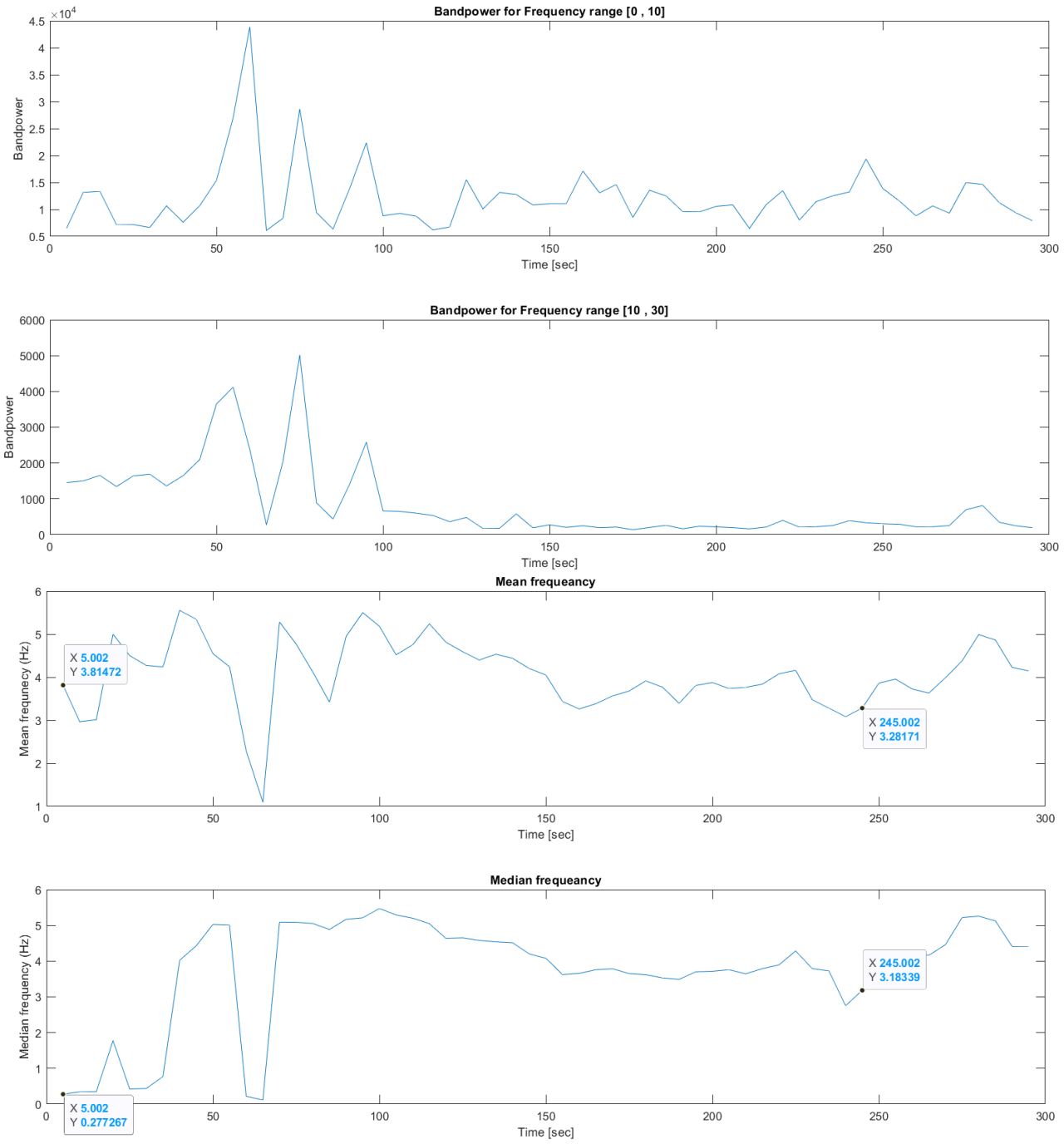
مشاهده میشود که به وضوح با اعمال فیلتر، هم در بخش نویزی و هم بخش تمیز، مقدار baseline حذف شده و سیگنال حول میانگین صفر مرتب شده است. از طرفی نوسانات فرکانس بالای پردامنه نیز حذف شده و یا کاهش یافته است (این مورد در سیگنال نویزی مشهودتر است). بدین ترتیب نویزهای فرکانس بالا نیز تا حد محسوسی کاهش می یابند.

بخش دوم: تشخیص آریتمی های بطنی

پ) در سکشن part3، زمان شروع ایونت ها و نوع آنها از روی فایل txt. مربوط به دیتای n422، خوانده شده و این مقادیر در دو آرایه ریخته شده اند. سپس پنجره بندی انجام شده و در هر پنجره، تعیین میشود که لیبل متناظر با آن پنجره چیست. برای وارد کردن اثر تداخل دو event در یک پنجره، تنها زمانی یک event را به پنجره مان نسبت میدهیم که آن پنجره بطور کامل در داخل بازه زمانی event قرار داشته باشد. لیبل ها در نهایت در labelsArr ریخته میشوند.

ت) در سکشن part4، سه ویژگی bandpower, mean frequency, median frequency را برای هر پنجره حساب میکنیم. با توجه به طیف های رسم شده در بخش ب، تفاوت سیگنال نرمال و VFIB، در فرکانس های پایین با فرکانس های بالا متفاوت است. بدین ترتیب bandpower را، یکبار در بازه ی [0, 10]Hz و یکبار در بازه ی [10, 30]Hz محاسبه میکنیم.

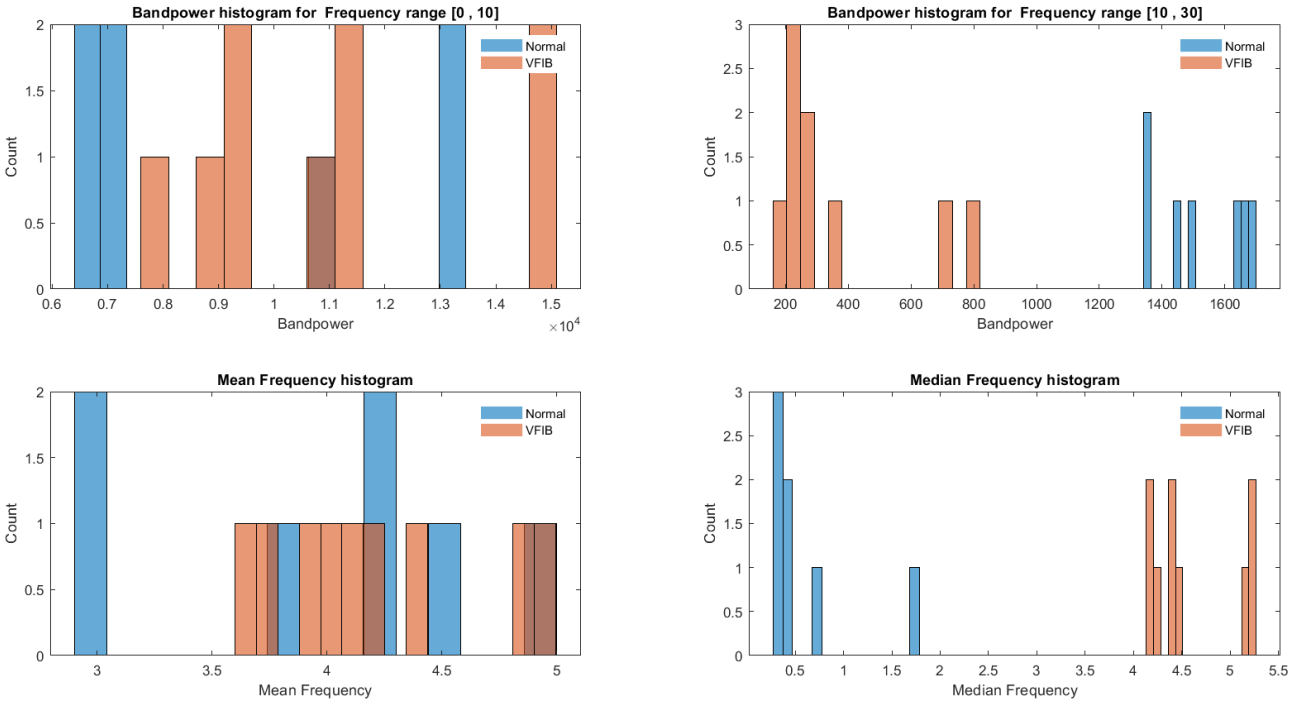
نمودار bandpower ها و نیز mean frequency و median frequency بر حسب پنجره ها، به صورت زیر است (توجه داریم برای هر پنجره که بازه ای از زمان را شامل میشود، سمپل میانی آن بازه را بعنوان زمان متناظر با پنجره در نظر گرفته ایم):



زمان های mark شده در شکل بالا، به ترتیب زمان شروع event های نرمال و VFIB هستند. همینجا میتوان تا حدی دید که در بازه کوچکی پس از این دو نقطه، برای Bandpower در بازه ی [10, 30] هرتز، مقدار توان VFIB کمتر از Normal است.

همچنین Median frequency نیز در حدود زمانی VFIB بیشتر از Normal است.
در مورد دو نمودار دیگر، مقادیر کمی به هم نزدیک اند و نمیتوان نظر قطعی داد.

ث) پس از رسم هیستوگرام چهار ویژگی بالا، به نتایج زیر میرسیم:



مشاهده میشود مشابه نتایج بخش قبل، برای Bandpower در بازه [10, 30]Hz، توان VFIB کمتر از Normal بوده و داده ها از هم تفکیک شده اند. میتوان مثلا توان 1000 را بعنوان یک threshold در نظر گرفته و پنجره هایی با bandpower کمتر از ۱۰۰۰ را VFIB و بیشتر از آن را Normal تشخیص داد.

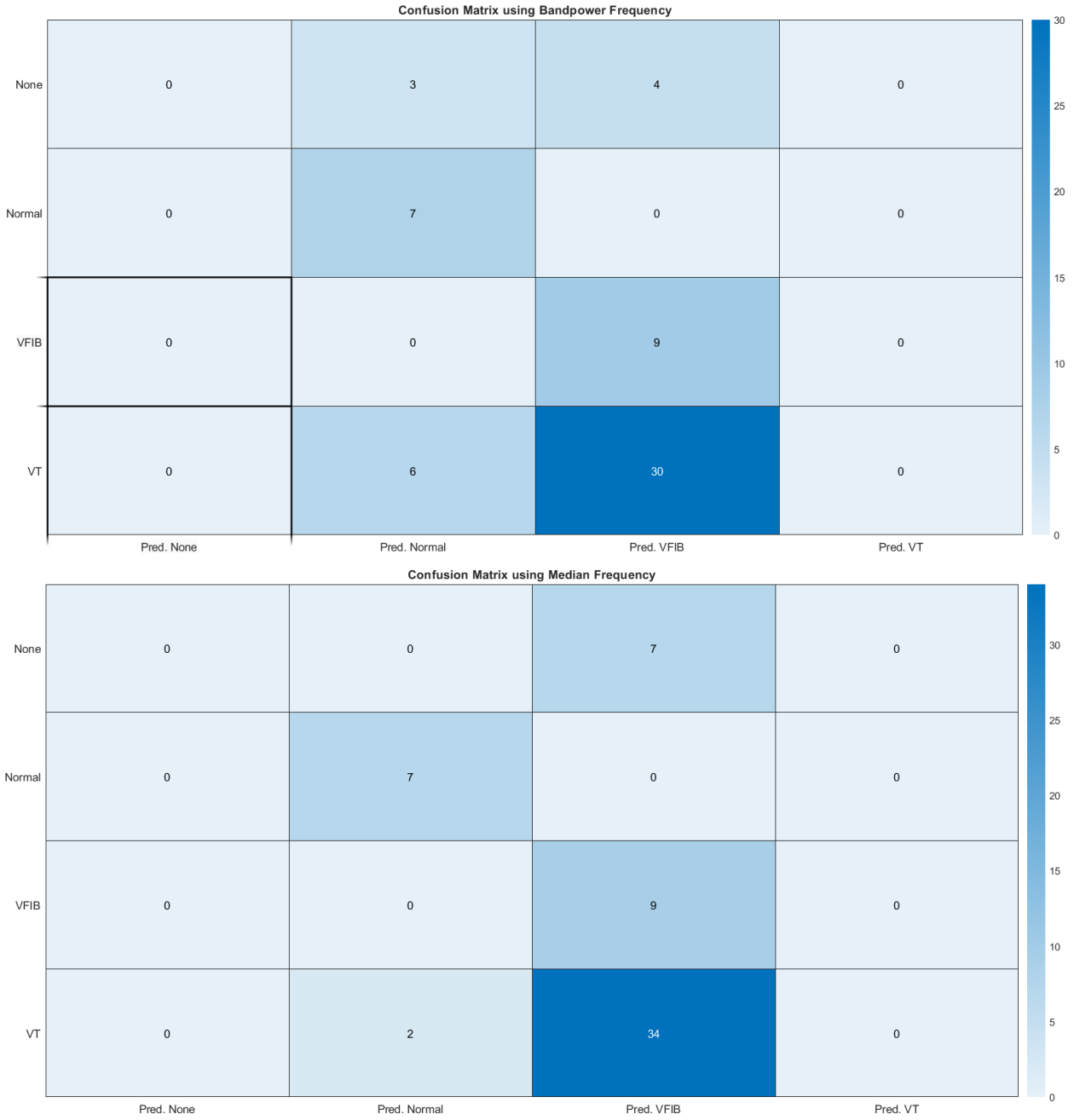
برای Medianfreq نیز برعکس است و VFIB فرکانس میانه‌ی بالاتری از نرمال دارد.
در اینجا هم داده ها قابل تفکیک اند. در اینجا نیز میتوان مثلا فرکانس میانه‌ی 2.5Hz

را بعنوان فرکانس threshold اعلام کرد که اگر در پنجره ای، فرکانس میانه بیشتر از 2.5 هرتز بود، VFIB و کمتر از آن را Normal تشخیص دهد.

اما در دو نمودار دیگر، داده های نرمال و VFIB با یکدیگر ادغام شده اند و قابل تفکیک نیستند. در نتیجه نمیتوان برایشان آستانه تعریف کرد.

ج) توابع va_detect_medfreq و ve_detect_bandpower، دو ویژگی منتخب در بالا را برای هر پنجره محاسبه کرده و بر اساس threshold های ذکر شده در بالا، در مورد VFIB یا نرمال بودن پنجره تصمیم گیری میکنند و اگر VFIB بود، در خروجی alarm=1 میدهند.

چ) ابتدا با استفاده از دو تابع مذکور در بالا، به دو روش کل پنجره ها label گذاری میشوند و سپس بردار alarm های حاصل از هر روش، همراه با لیبل های واقعی پنجره ها که در labelsArr داشتیم، یک confusion matrix را میسازند. این ماتریس 4*4 است زیرا کلا ۴ تا لیبل در دیتای n_422 موجود است (لیبل Noise در این دیتا وجود ندارد). این ۴ لیبل به ترتیب None, Normal, VFIB, VT هستند. سطر های این ماتریس، مقادیر حقیقی لیبل ها، و ستون های آن لیبل های پیش بینی شده توسط توابع بالا هستند. توجه داریم این دو تابع، فقط میتوانند بین VFIB و Normal تصمیم گیری کنند و در نتیجه در مورد دو لیبل None, VT تصمیمات درستی نمیگیرند. ماتریس Confusion با استفاده از روش bandpower و روش median frequency، به ترتیب به صورت زیر است:



اعداد روی قطر اصلی در ماتریس confusion، تشخیص های صحیح مدل را نشان میدهد. میبینیم که برای حالات نرمال و VFIB، تمام تضخیص ها درست بوده اما برای دو لیبل دیگر تشخیص های اشتباهی رخ داده است. خواسته صورت سوال، محاسبه پارامتر های sensitivity, specificity, accuracy برای دو کلاس نرمال و VFIB بوده است. پس تمرکز ما برای این کمیت ها، روی جدول 2*2 موجود در وسط شکل

های بالاست. با توجه به اینکه ما از روی یک دانش قبلی مقادیر threshold را به صورت ایده آل تعیین کردیم، برای کلاس های Normal, VFIB، در هردو روش لیبل گذاری، مقادیر زیر بدست آمده است:

$$sensitivity = specificity = accuracy = 1 \equiv 100\%$$

اما در حالت کلی از روابط زیر بدست می آیند:

$$sensitivity = \frac{TP}{P}, specificity = \frac{TN}{N}, accuracy = \frac{TP + TN}{P + N}$$

داده های کلاس None, VT نیز به ناچار بر اساس مقادیر فرکانس میانه و bandpower شان، در یکی از کلاس های Normal, VFIB قرار گرفته اند که تعدادشان را در ماتریس های بالا میبینیم.