PARKINSON'S TREMOR DETECTION IN PATIENTS RECEIVING DEEP BRAIN STIMULATION USING APPLE WATCH ACCELEROMETER DATA

A Thesis

Presented to

The Faculty of the Department of Electrical and Computer Engineering

California State University, Los Angeles

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in

Electrical Engineering

By

Luis Felipe Martinez Sandoval

May 2025

The thesis of Luis Felipe Martinez Sandoval is approved.

Deborah Won, PhD, Committee Chair

Marina Mondin, PhD, Committee Member

Curtis Wang, PhD, Committee Member

Charles Liu, PhD, Department Chair

California State University, Los Angeles

May 2025

TABLE OF CONTENTS

ABSTRACT

Parkinson's Tremor Detection in Patients Receiving Deep Brain Stimulation

Using Apple Watch Accelerometer Data

By

Luis Felipe Martinez Sandoval

The work presented seeks to investigate and develop an algorithm capable of accurately detecting tremor in individuals with Parkinsons disease (PD) undergoing deep brain stimulation (DBS), which could be implemented into a closed loop system to help manage tremor in real time. Data used for the development and testing of this algorithm was collected from six patients using accelerometers from both an Apple Watch and a Medtronic Summit RC+S implantable pulse generator (IPG) during clinical trials. DBS intensity was adjusted across multiple levels to vary tremor severity. The developed algorithm aims to classify tremor during various physical activity states such as sitting, standing, walking, and texting. The developed algorithm relies on a frequency spectrum analysis of the streamed accelerometry data using a continuous wavelet transform and focuses on the 4-7 Hz frequency band and its harmonics. The performance of the algorithm was compared against the current gold standard for tremor detection in the PD research community, developed by Rune Labs, which provides an estimate of the degree of tremor in 1-minute windows. Ground truth on a higher time resolution was established using a combination of trial logs and visual inspection of the raw accelerometry data. Results during testing prove that the developed algorithm improves upon current PD tremor detection by increasing temporal resolution and can detect tremor during high activity state.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisor Dr. Won. Without Dr. Won's guidance, mentorship, and unwavering support it would not have been possible for me to succeed to the extent that I have. I would also like to thank my thesis committee members Dr. Mondin and Dr. Wang for providing valuable feedback for the work done in this thesis and support. Also, thank you Dr. Liu for the work you do as department chair to support students throughout our academic journeys. Without your support along with the rest of the ECST faculty I would not be in the position I am in today.

I would also like to thank Orlando and Arabo for collaborating with me and providing valuable insight throughout this research effort. Not only were you outstanding research partners but were also two people I came to consider close friends.

Others I would like to specially thank are Fergus, Juan, and my wife Melina. Thank you, Fergus, for being my first friend at Cal State LA and for motivating me to pursue excellence throughout my academic work while earning my Bachelors and now Master's. Additionally, thank you Juan for being someone I could confide in and for sharing valuable advice and wisdom. But most importantly thank you Melina for helping me achieve my dream of going back to college and pursue a career in engineering.

Finally, I would like to acknowledge Dr. Stephen Schmidt, Dr. Dennis Turner and Jenny Peters and support from their NIH Grant #5UH3NS103468. I would also like to gratefully recognize the financial support of NSF Grant HRD-2112554. Without your support none of this research work would have been possible.

LIST OF TABLES

Table

LIST OF FIGURES

Figure

CHAPTER 1

Motivation

1.1 DBS Therapy and its Benefits

Deep brain stimulation (DBS) is a neuromodulation therapy that is used to manage symptoms in individuals with neurological disorders that are accompanied by movement related symptoms [1]. One of the DBS's primary uses is to help individuals with Parkinson's disease (PD) manage their motor symptoms [2]. Some of the most common motor symptoms in individuals with PD include tremor, bradykinesia, muscle rigidity, impaired posture, and changes in speech [1] [3]. This is caused by degeneration in the basil ganglia brain structure but more specifically in the subthalamic nucleus and globus pallidus [2] [4]. This degeneration and loss of neurons cause dopamine levels to drop and movement related symptoms to present themselves [4]. There is no cure for PD but current treatments can manage symptoms. Dopamine replacement therapy is already in clinical use and typically administered through medications such as Levodopa, which is a compound that is converted to dopamine in the brain [5]. However, long term use of Levodopa can become less effective over time, can lead to unmanageable dyskinesia (abnormal postures and movements), and may contribute to symptoms such as cognitive impairment, depression, and restlessness when medication levels in the blood stream drop between doses [6][10]. Another option is medication like apomorphine, which directly stimulates dopamine receptors to mimic the effects of dopamine [7]. But like Levodopa, long term use of apomorphine can also lead to side effects such as nausea, injection site reactions, visual hallucinations, and worsening dyskinesia between doses [8] [9]. Managing PD symptoms with medication can also be inconsistent due to the 'wearing-off

effect', where the medication's effectiveness decreases as its levels in the bloodstream drop between doses [10]. Figure 1 shows the relationship between Parkinsons related symptoms and medication concentration in the blood stream. When these kinds of drug therapies become ineffective DBS may be recommended as an alternative treatment to individuals with PD to help manage their symptoms [11].

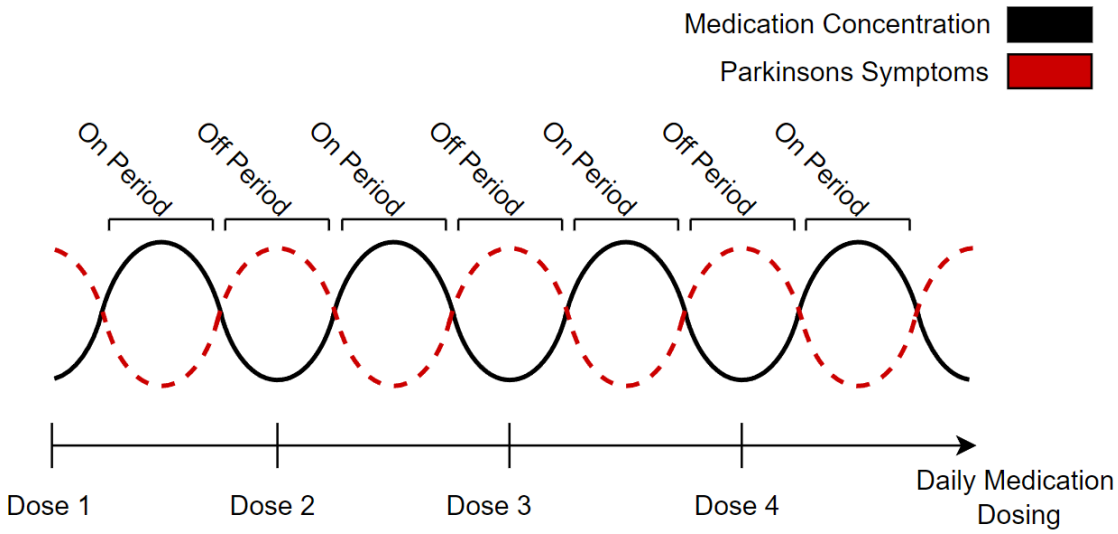

Figure 1. Relationship Between Medication Concentration in Blood Stream and Parkinsons Symptoms

PD symptom management through DBS involves surgical intervention and requires electrodes to be implanted which deliver electrical impulses to specific areas of the brain. The procedure also requires implanting a pulse generator (IPG) under the skin, usually on the chest wall [12]. The IPG continuously delivers electrical impulses to the targeted brain structures typically the subthalamic nucleus and globus pallidus [13]. Targeting the subthalamic nucleus or the globus pallidus with electrical stimulation has shown to effectively reduce bradykinesia, rigidity, and tremor in individuals with Parkinson's disease [14]. Unlike dopamine replacement therapy, which relies on increasing dopamine levels in the brain, DBS modifies neural signals and allows for a more consistent and personalized treatment option [15]. While DBS is primarily used to

treat neurological disorders and is FDA approved for managing Parkinson's disease, essential tremor, and epilepsy, it has also shown promising results in treating conditions such as Tourette's syndrome and treatment resistant depression in clinical trials [16] [17].

1.2 Current DBS Therapy Implementation

DBS therapy is applied to participants through dual bilaterally placed electrodes on the subthalamic nucleus and globus pallidus [18]. By applying electrical stimulation to these specific regions, DBS influences local field potential and modulates neural activity which in turn suppresses physical Parkinsons symptoms such as tremor and dyskinesia [18]. Biphasic pulses are applied to the implanted electrodes to maintain a safe charge balance between the electrodes and minimize risk to the tissue surrounding the electrodes [20]. The charge density present on the electrodes is designed to remain below $30\frac{\mu C}{cm^2}$ by the manufacturer [21]. Figure 2 shows the general shape of the biphasic pulse.

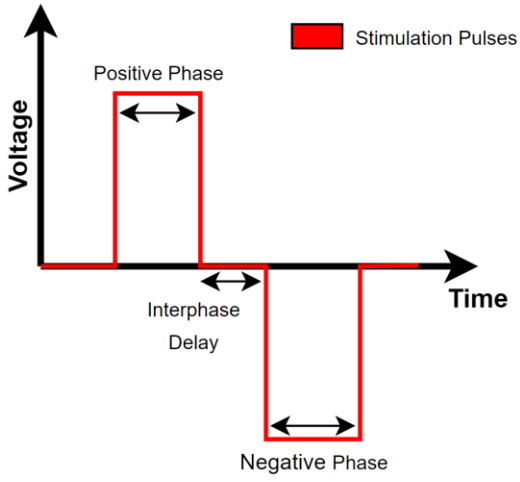

Figure 2. Biphasic Stimulation Pulse

While there are over 42,000 combinations of amplitude, frequency, and pulse width for this device, the IPG must be programed by a physician to a prescribed set of parameters and cannot be dynamically adjusted as Parkinson's symptoms present themselves [21]. This means that during a programming session it may appear the

individual has the optimal settings in place but if physical symptoms worsen or subside after the programming session the DBS cannot be adjusted to fit the individual's current needs.

While this type of therapy is revolutionary in managing Parkinsons symptoms, in its current implementation it is either on or off and does not have adaptive capabilities. Thus, the IPG needs occasional reprogramming, which is subjective, tedious, and has to be scheduled based on the physician's availability. Furthermore, applying DBS continuously with constant stimulation settings may be an inefficient use of energy at best and create unnecessary side effects at worst. For this reason, a possible implementation of a closed loop system is being investigated.

A recent study already demonstrated increased efficiency could be achieved while maintaining therapeutic efficacy by turning DBS on and off according to neural feedback [18]. Our long-term goal is to develop an adaptive DBS system that dynamically responds to physical Parkinson's symptoms like tremor.

Figure 3 shows the current open loop DBS implementation. Medtronic developed a closed-loop DBS platform, the Summit RC+S implantable pulse generator (IPG), which has the capability to both stimulate the brain and acquire neural local field potentials as well as accelerometry data to potentially be used as feedback for closed-loop control. Clinical trials have demonstrated observe the efficacy of using the new Summit RC+S [18].

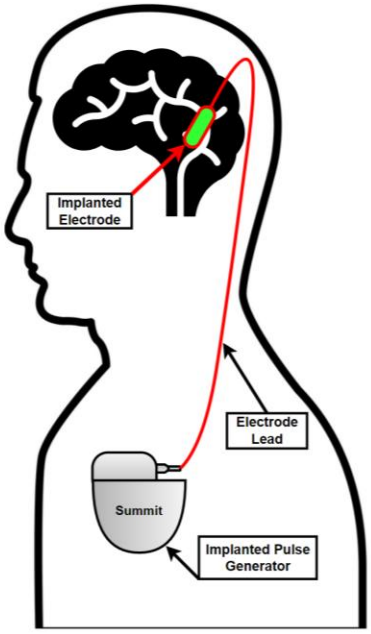

Figure 3. Open Loop DBS Implementation

The Summit RC+S pulse generator operates similarly to the FDA approved Medtronic Intellis rechargeable implantable neural stimulator [19] but has the additional neurofeedback capabilities. Our investigation of developing a tremor detection algorithm that could be used in a closed-loop DBS therapy utilizes this Summit RC+S pulse generator and its capability to stream accelerometry data to a cloud server.

1.3 Closed Loop DBS

Collaborators at Duke University have led clinical trials using the new Summit RC+S implantable pulse generator in Parkinson's disease patients. The RC+S offers new added functionality over the Medtronic Intellis neural stimulator, which is one of the FDA approved devices that is currently clinically in use. The new RC+S device offers 16 channels which could be set as either inputs or outputs, which means that electrodes can be used to record neural activity or apply DBS. Another added feature of the Summit RC+S is that it has a built-in accelerometer which can be used to monitor an individual's movement and potentially physical tremor episodes. The Summit RC+S also has

Bluetooth communication capabilities to transmit data to physicians and users [19]. With the device's capabilities in mind, we will explore the viability of using the Summit RC+S built in accelerometer to detect tremor by developing a custom tremor detection algorithm. Also, since the Summit RC+S also has built in Bluetooth capabilities we will also explore the viability of using an Apple Watch alongside the Summit RC+S, since tremor usually presents itself in the extremities such as arms, hands, feet, and legs [22].

Figure 4 shows the closed loop configuration for which tremor detection relies on accelerometry provided by the Summit IPG. By using the Summit RC+S IPG, acceleration caused by an individual's movement could be transmitted to an Apple Watch for processing through our tremor detection algorithm and detection results would be sent back to the Summit RC+S IPG. This would form a closed loop system where the Apple Watch provides constant feedback to the Summit RC+S IPG and indicates when the Summit RC+S IPG should apply DBS to suppress tremor. The tremor detection algorithm would detect key characteristics present in the accelerometry when tremor is present.

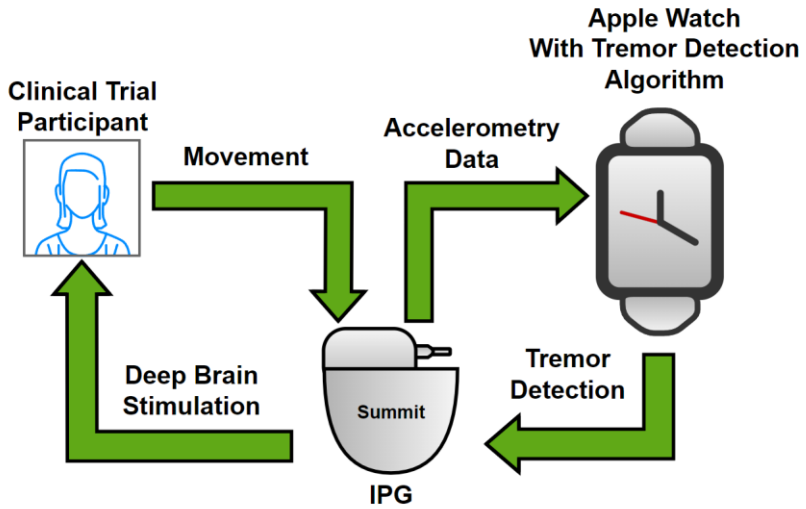

Figure 4. Proposed Closed Loop DBS Scheme Using IPG Accelerometry

A second closed loop configuration that could be implemented relies on using Apple Watch accelerometry as shown in Figure 5. In this second configuration an Apple Watch accelerometer is used to track hand movements. This configuration would depend on the detection algorithm being embedded in an iOS application specifically designed to analyze the accelerometry data. In this scheme, we would propose that the tremor detection is implemented in the Apple watch and tremor detection is transmitted to the Summit RC+S IPG via Bluetooth so that the Summit RC+S IPG would apply DBS as necessary. This scheme would consume more power due to the need for Bluetooth transmission. The integration of the accelerometer in the IPG in the first scheme would provide for a more efficient way to implement DBS using feedback from the integrated accelerometer and would also be more convenient for the user instead of the patient needing to wear an external device. However, since tremor is typically more notable in the limbs when compared to the body and so we expect using accelerometry from the Apple Watch will produce more accurate detection [23].

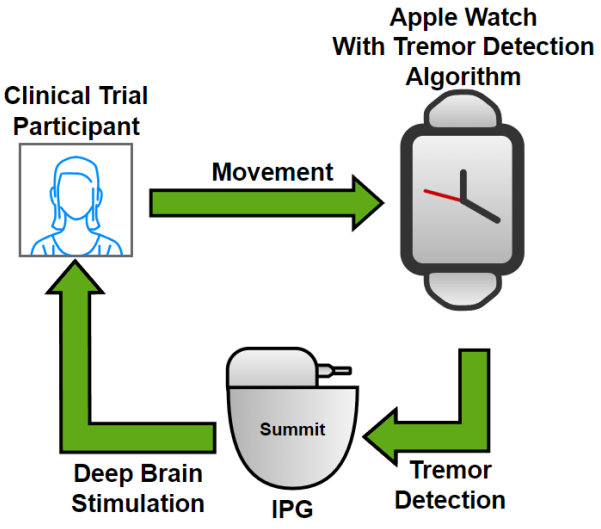

Figure 5. Proposed Closed Loop DBS Scheme Using Apple Watch Accelerometry

1.4 Thesis Objective Statement

In this thesis our efforts seek to develop and validate a novel algorithm for real-time tremor detection in individuals with Parkinson's disease undergoing deep brain stimulation using the Medtronic Summit RC+S implantable pulse generator. We will leverage accelerometry data from both a Medtronic Summit RC+S IPG and an Apple Watch to compare their viability in detecting tremor during various physical activity states including sitting, standing, walking, and texting. Furthermore, we will also compare the efficacy between both using short time Fourier transforms and continuous wavelet transforms as frequency analysis tools to detect tremor in the 4-7Hz range for both IPG and an Apple Watch accelerometry. This approach seeks to improve temporal resolution provided by the current gold standard established by StrivePD, which provides tremor percentage estimates over 1-minute windows, as well as improve StrivePD's limitation when it comes to tremor detection during high activity states. The algorithm's performance will be compared against StrivePD's established tremor metric, with the goal of achieving comparable detection accuracy during low activity states, higher detection accuracy during high activity states, and overall higher temporal resolution for tremor detection. This work aims to lay the foundation for an adaptive closed-loop DBS system that dynamically adjusts DBS intensity in response to detected tremor. This will improve symptom management and potentially minimizing side effects of the long-term DBS exposure.

CHAPTER 2

IPG and Apple Watch Accelerometry

2.1 Experimental Protocol

Our Duke collaborators have designed and been conducting clinical trials to investigate the use of the Summit RC+S in closed-loop control of DBS and conduct a number of investigations into developing adaptive DBS algorithms. All experimental methods used employed to conduct the research presented in this thesis were approved by the Duke University Internal Review Board. Six patient subjects with pharmaceutically refractive Parkinson's disease (PD) received bilateral Summit RC+S DBS implants. Participants attend four types of post-operative visits after the Summit RC+S IPG has been implanted. The first kind is a clinical programming session where settings are adjusted to find the optimal setting for stimulation of the subthalamic nucleus and globus pallidus. The second are virtual home visits which are attended by study personnel to observe participants in their home environment through a video conference. During these sessions DBS settings can be adjusted using a Surface tablet as a control unit. The virtual sessions can also be conducted as unattended home visits where the study personnel collect data through the Surface tablet control unit but do not hold a video conference. During these unattended sessions, participants are asked to keep the data streaming on, so that data can be acquired continuously up to 24 hours a day. Data collected during this time is transmitted then transmitted to the study personnel and uploaded to the Box.com servers [18] [21]. These data can then be analyzed offline to develop feedback control algorithms and understand how different stimulation protocols are affecting subjects.

The fourth kind of post-op visit is a day-long, in-person visit at a medical facility. These begin about six months after the Summit RC+S IPG has been implanted and the DBS settings have been optimized. The data collected during these visits is what was used for the development of our tremor detection algorithm. Figure 6 shows the data collection pipeline for the data used in the development our tremor detection algorithm [18] [21].

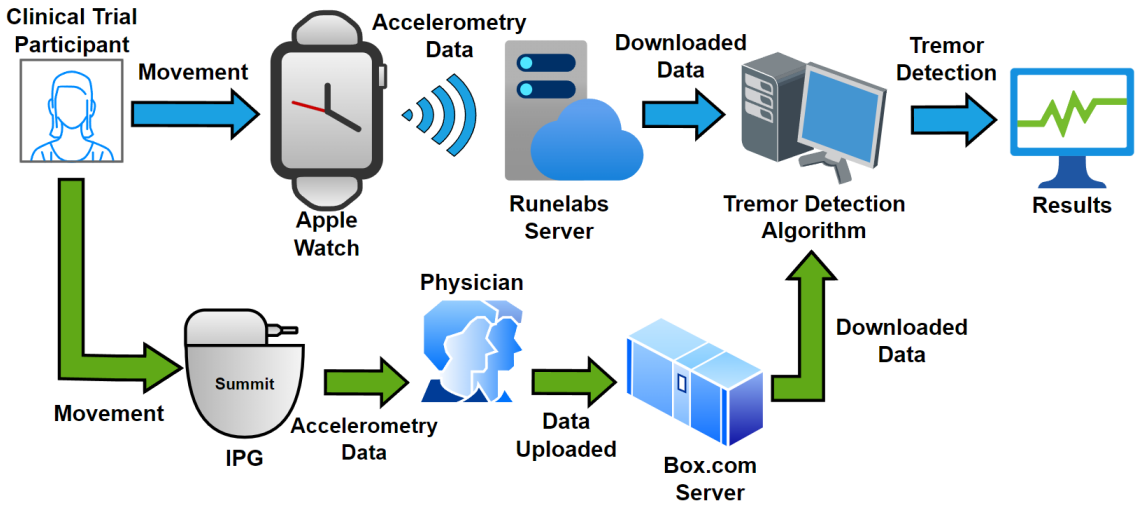

Figure 6. Data Collection Pipeline for Current Investigation

These day long visits occur on a monthly basis and multiple sessions are held on the same day, typically ranging from two to three sessions. Both Apple Watch and IPG accelerometry data is collected during these sessions. Each session of accelerometry data collection consists of 30 seconds of sitting, followed by 30 seconds of standing, then 30 seconds of walking, next is 30 seconds of sitting, and finally 30 seconds of texting. The order and time of execution of these motor activities is identical for each trial. The difference is that between trials the DBS intensity is lowered from 100% of the participants prescribed setting down to 80%, 60%, or 40% [18] [21]. Our tremor detection algorithm was developed using approximately 100 recorded sessions from five

participants in the trials. One of the six participants was unable to carry out the monthly in-clinic sessions.

Data collection and transfers from both the Apple Watch and Summit RC+S follow HIPPA compliant guidelines during all postop visits. In the case of the Apple Watch, the StrivePD application is used to anonymize the participants' personal health information and will be further discussed in section 2.2; Summit RC+S data collection is further discussed in section 2.3 [18] [21].

2.2 Apple Watch Hardware and Data Acquisition

The Apple Watch used during the clinical trial by participants was a typical Apple Watch that could be purchased on the consumer market. To collect the accelerometry data the Apple Watch's built in tri-axial accelerometer was used, the Apple Watch accelerometer axis orientation is shown in Figure 7 [27].

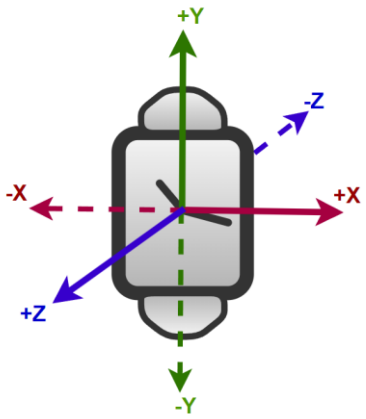

Figure 7. Apple Watch Accelerometer Axes

To collect the accelerometry data a custom application was created by Rune Labs called StrivePD and is available on the Apple AppStore. Transmission of the data from the Apple Watch to an iPhone is handled by the Apple security protocol and is encrypted before being streamed to the participant's iPhone [24]. Data transfer from the iPhone to the Rune Labs servers is handled by the StrivePD application and removes personal

identifiable health information before transmission to the Rune Labs servers. The StrivePD application also collects relevant patient-entered information such as medication, symptoms, and side effects [28]. The accelerometry data collected by the StrivePD application is the data used in the development of the tremor detection algorithm presented in this thesis.

To access the collected Apple Watch accelerometry data a script was developed using Python (Appendix A). This script utilizes the Rune Labs API to access the data after obtaining an access token. The raw Apple Watch accelerometry data available from Rune Labs is sampled at 50Hz and is stored using a Unix timestamp, also called Epoch time. Unix time refers to how much time has passed since January 1, 1970, 00:00:00 UTC [25]. The time stamps are formatted as follows: seconds.milliseconds. The Duke University research team provided timestamps for trial sessions and participants activity states, which were used to retrieve relevant data and synchronize the Apple Watch data with the IPG data using the Rune Labs API [26].

2.3 IPG Hardware and Data Acquisition

As previously mentioned, the Summit RC+S IPG has the added capability of recording and streaming accelerometry data when compared to the current Medtronic Intellis implanted neural stimulator. The Summit RC+S acquires and streams local field potentials and neural activity, along with accelerometry data from its built-in accelerometer [18]. For the development of our detection algorithm, we only focused on the streamed accelerometry data. Figure 8 shows the accelerometer axis orientation for the Summit RC+S IPG [19].

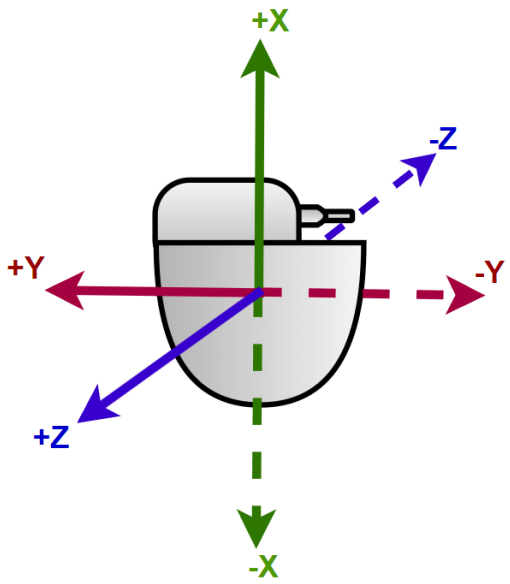

Figure 8. IPG Accelerometer Axes

Accelerometry data used in the development of our tremor detection algorithm was collected during day long research visits at a designated facility. Accelerometry data was collected using varying levels of DBS intensity, namely 100%, 80%, 60%, or 40% of the individuals DBS prescribed settings. These variations in DBS intensity evoke varying levels of tremor during different activity states since they are below the prescribed levels.

Data streamed by the Summit RC+S IPG is collected by the research staff and cleared of any personal identifiable before it is uploaded to a Box.com server where it can be accessed by individuals with the appropriate credentials. Accelerometry data generated by the Summit RC+S IPG however does not have a time stamp for each individual sample. Instead, the accelerometry data is stored in batches of 8 samples and is transmitted with a time stamp of when the whole 8 sample packet was generated. The time stamps given are also in Unix time like the Apple Watch data but is formatted as millisecond passed since January 1, 1970, 00:00:00 UTC [25]. To assign timestamps to individual samples within each packet, the time difference between sequential packet

timestamps was calculated. This difference is then divided by 8 to interpolate timestamps for each sample within the packet.

2.4 Clinical Trial Participant Pool

The participant pool for the clinical trials consists of 5 individuals with a range of Parkinsons related symptoms and prescribed DBS settings. Qualitative clinical assessments performed before the study will allow us to verify the accuracy of our tremor detection algorithm. We performed tremor detection on the data naïve to any clinical assessments; after performing tremor detection, we compared our results with the clinical assessments. Thus, here we describe the participants' symptoms and clinical observations of symptoms and amount and severity of tremor. Two participants (P1 and P2) experience dyskinesia as their dominant Parkinson's symptom and overall do not experience tremor even when the prescribed DBS intensity is reduced. For this reason, accelerometry data collected from P1 and P2 at the highest DBS intensity will be used to establish baseline accelerometry recordings without tremor.

Participant 3 (P3) experiences tremor as one of their dominant Parkinson's symptoms and experiences tremor during all physical states observed in the clinical trials. P3 also experiences varying levels of tremor where symptoms become more severe as DBS intensity is decreased. Participant 4 (P4) generally experiences tremor during the sitting, standing, and fine motor movement portions of the clinical trials. P4 typically does not exhibit tremor symptoms during physical activity. Participant 5 (P5) also experiences tremor during all physical activity stated but generally less during physical activity. Table 1 lists the 5 mentioned participants and their dominant Parkinsons

symptom. DBS intensity during the clinical trials varies as 100%, 80%, 60%, or 40% depending on the trial and are percentages of the participants prescribed DBS intensity.

Table 1. Participant Number and Dominant Parkinsons Symptom

Participant Number	Dominant Parkinsons Symptom
P1 (AC27)	Dyskinesia
P2 (E395)	Dyskinesia
P3 (RZCH)	Tremor
P4 (NU5U)	Tremor
P5 (6KOZ)	Tremor

2.5 DBS Intensity and Tremor Severity

By varying the level of DBS intensity as a percentage of the participants prescribed DBS settings, different levels of tremor intensity could be elicited during the clinical trials [21]. Tremor can be observed visually in the accelerometry signals with careful examination. Figure 9 and Figure 10 demonstrates the Apple Watch and IPG accelerometry data collected from P3 during clinical trials and DBS intensity set to 100% of the prescribed levels. As expected, the acceleration recorded during periods of sitting, standing, and texting are relatively flat for both the Apple Watch and IPG recordings. This means that the participants' hands and torso are not moving during these periods as expected since they are not moving and have DBS at its prescribed setting. Furthermore, during walking large clean regular oscillations at a low frequency of the walking gait can be observed.

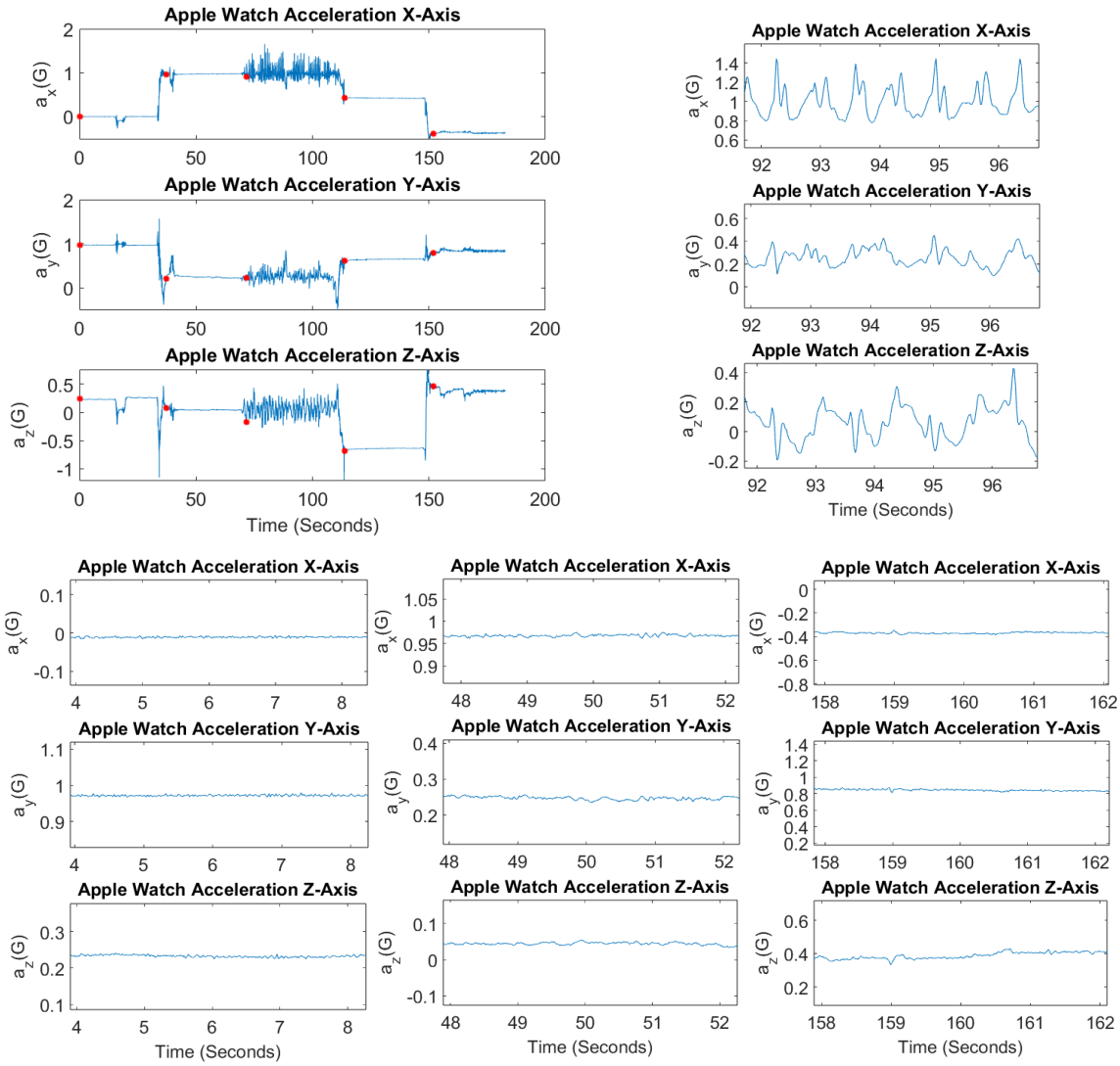

Figure 9. Apple Watch Accelerometry (P3, 100% DBS)

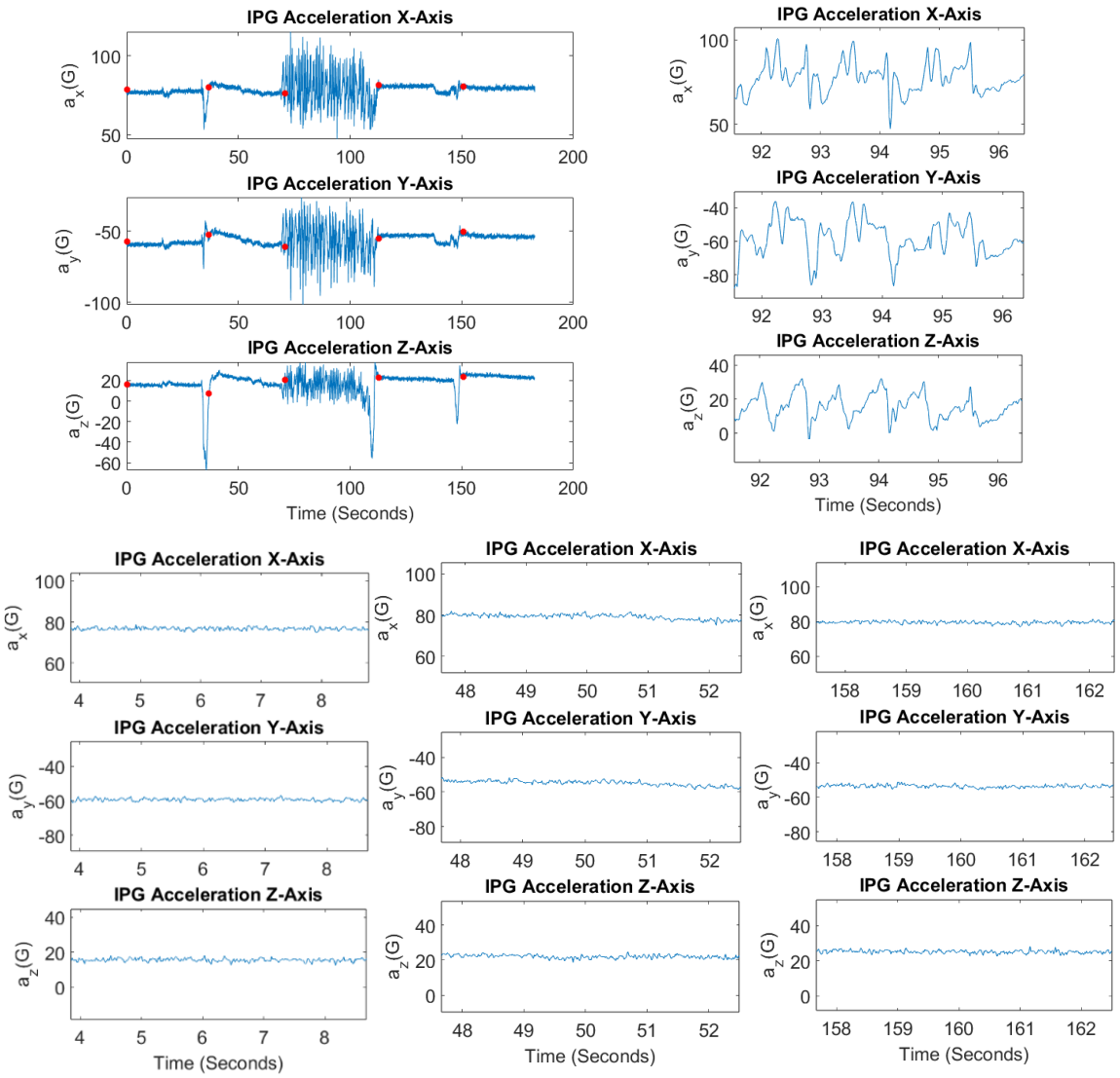

Figure 10. IPG Accelerometry (P3, 100% DBS)

By reducing the DBS intensity to 60% of the prescribed levels we can see in the Apple Watch accelerometry shown in Figure 11 that there is much more movement detected during the sitting, standing, and texting physical states since these portions of the recordings are no longer as flat as the recordings shown in Figure 9. While we don't see these significant difference when comparing the IPG accelerometry, Figure 10 and Figure 12, it would be expected considering tremor is most notable in the extremities.

This would suggest that DBS was indeed suppressing PD symptoms, and reducing DBS settings does in fact allow physical Parkinson's symptoms to reemerge.

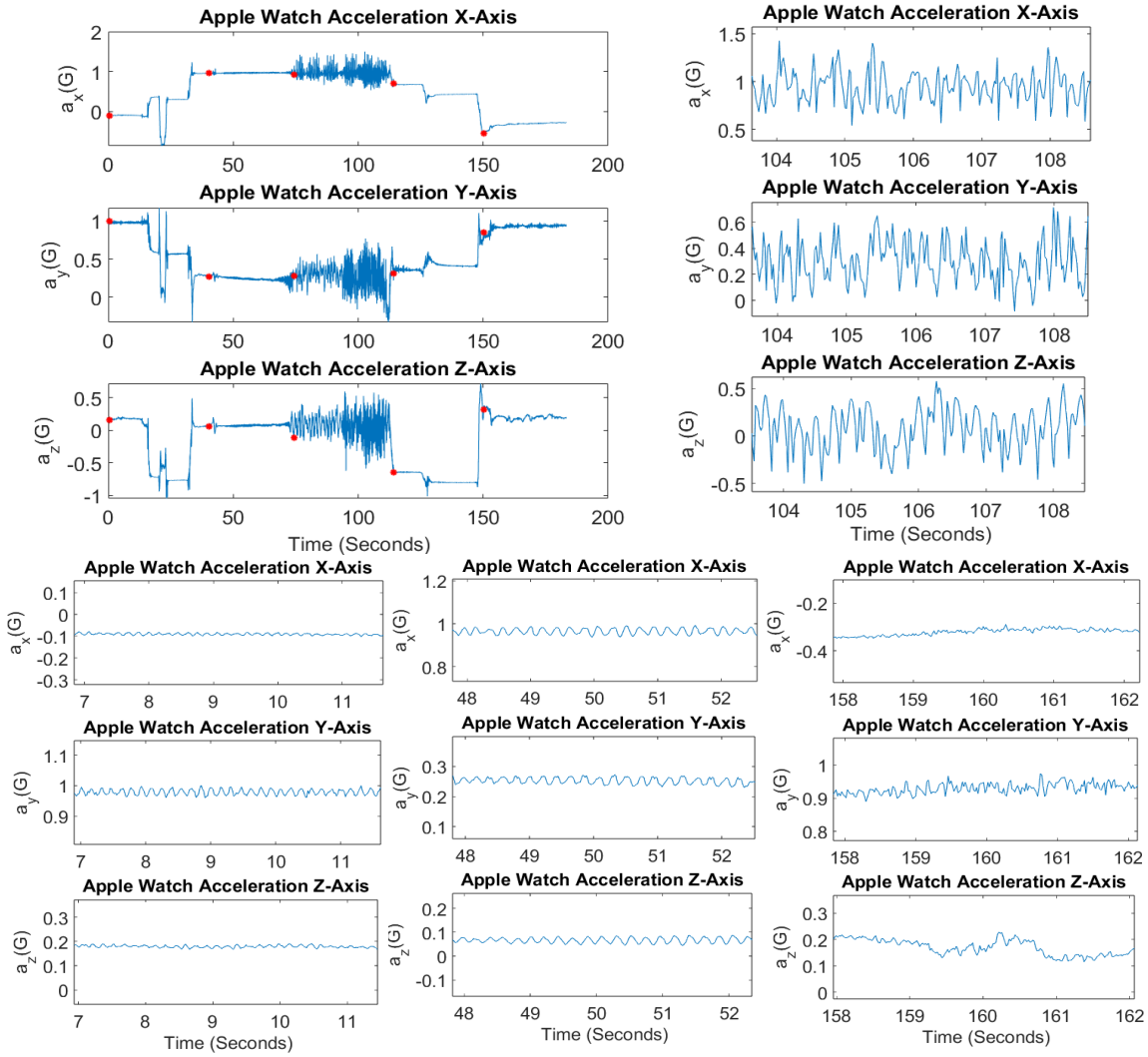

Figure 11. Apple Watch Accelerometry (P3, 60% DBS)

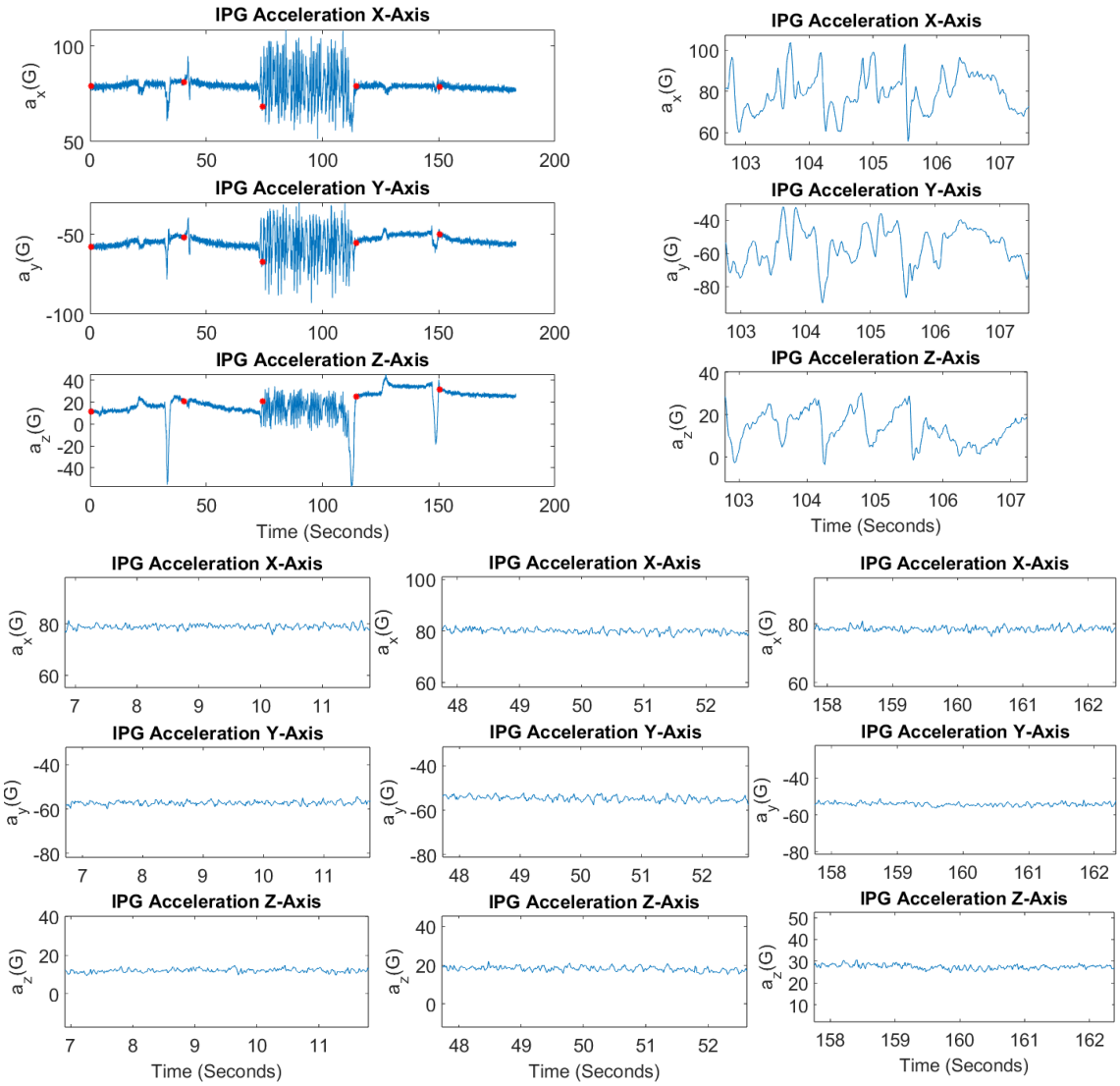

Figure 12. IPG Accelerometry (P3, 60% DBS)

When the DBS intensity is further reduced to 40% of the prescribed levels for P3 we can more effectively see the relationship DBS intensity has to physical Parkinson's symptoms and the accelerometry data recorded as shown in Figure 13 and Figure 14. When DBS intensity is reduced to 40%, we get much more movement when compared to 100% DBS intensity. In a further section we will discuss and show how this increase in acceleration is in fact tremor and what has been done to detect it using the developed detection algorithm. It should also be noted that the examples given in this section of

DBS intensity and acceleration recordings were taken during clinical trials occurring on the same day and in quick succession of one another.

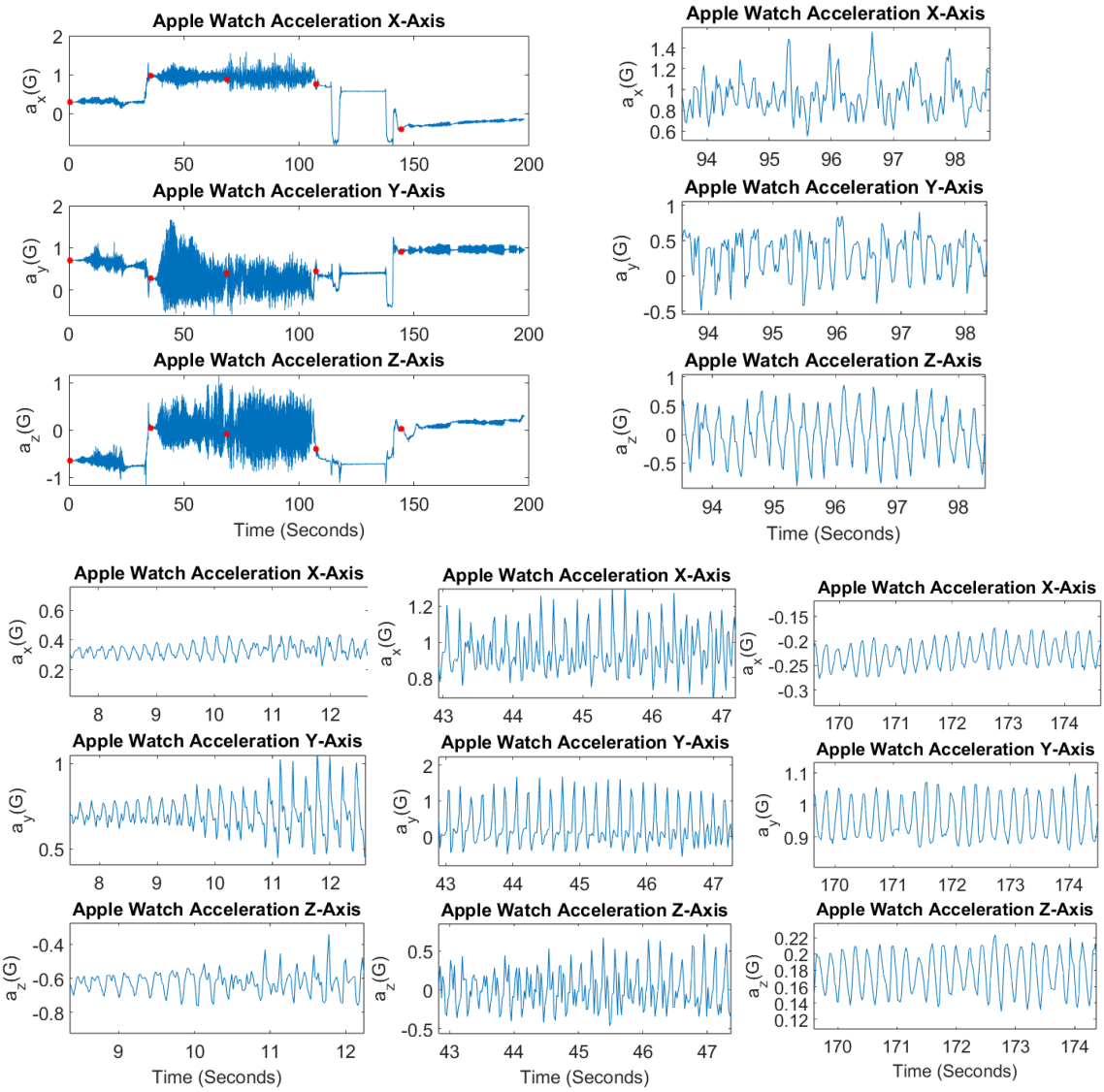

Figure 13. Apple Watch Accelerometry (P3, 40% DBS)

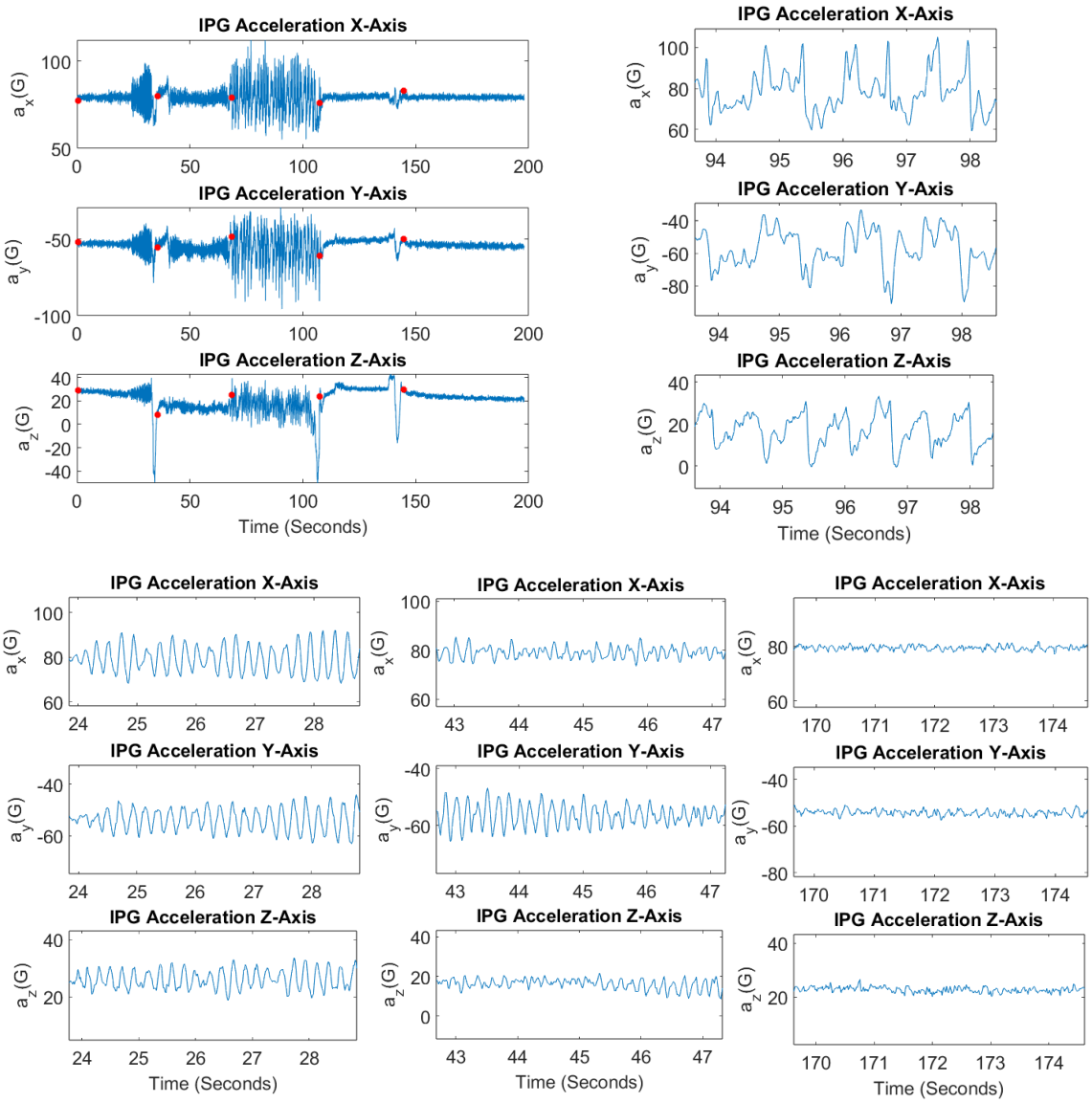

Figure 14. IPG Accelerometry (P3, 40% DBS)

CHAPTER 3

Tremor Detection Algorithm

3.1 Tremor Frequency Band

Our detection algorithm leverages our knowledge of spectral characteristics of tremor in PD. As outlined by Dongning Su in her work [29] characterizing tremor in individuals with Parkinson's disease and Multiple System Atrophy, individuals with Parkinson's disease who experience resting and postural tremor exhibit spectral peaks in the 4–7 Hz range in accelerometry data. Additionally, a frequency analysis revealed the presence of harmonics in 75% of cases [29]. In fact, other studies have also concluded that Parkinson's tremor occurs around the 4-7 Hz range and the frequency spectrum of the accelerometry includes harmonics of the tremor's fundamental frequency [30] [31] [32]. In general, Figure 15 shows an illustration of what we expect to see in a frequency spectrum analysis of our accelerometry data if tremor is present. As previously described if tremor is present there should be a major peak in the acceleration frequency spectrum within the tremor band, 4-7Hz, and smaller peaks at the tremor's fundamental frequency harmonics. Harmonics of the tremors fundamental frequency are defined as n multiples of first major peaks frequency.

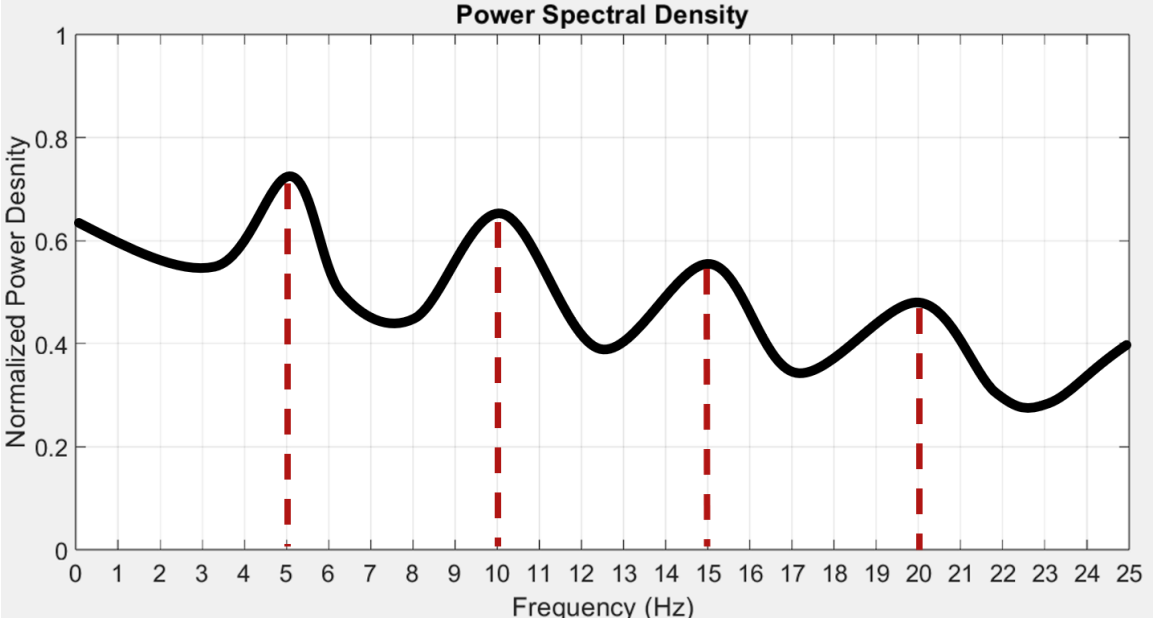

Figure 15. Example of Expected Frequency Spectrum When Tremor is Present

3.2 Frequency Analysis Methods

To perform the necessary frequency spectrum analysis on both the Apple Watch and IPG accelerometry data we explored two methods which are widely used for signal processing: Short Time Fourier Transforms (STFT) [33] [34] and Continuous Wavelet Transforms (CWT) [35] [36] [37] both implemented using the built-in MATLAB functions (stft and cwt).

The STFT was carried out with 1.5-second Hamming windows and 70% overlap with a frequency resolution of 0.39 Hz (or 65 frequency points). The Hamming window was selected over the rectangular window to reduce the amount of spectral leakage observed and measured in the signal's spectrogram [38]. A duration of 1.5 second windows was selected to give the spectrogram a fair balance of temporal resolution, making it easier to localize when tremor is occurring along the time axis. In MATLAB a Hamming window is defined with the following function [39].

$$w(n) = 0.54 - 0.46\cos\left(2\pi\frac{n}{N}\right), \qquad 0 \le n \le N$$

As mentioned, the window that is being used is a Hamming window due to its ability to reduce spectral leakage when compared to a rectangular window and could be seen in Figure 16. The frequency response of the Hamming window is shown in Figure 17.

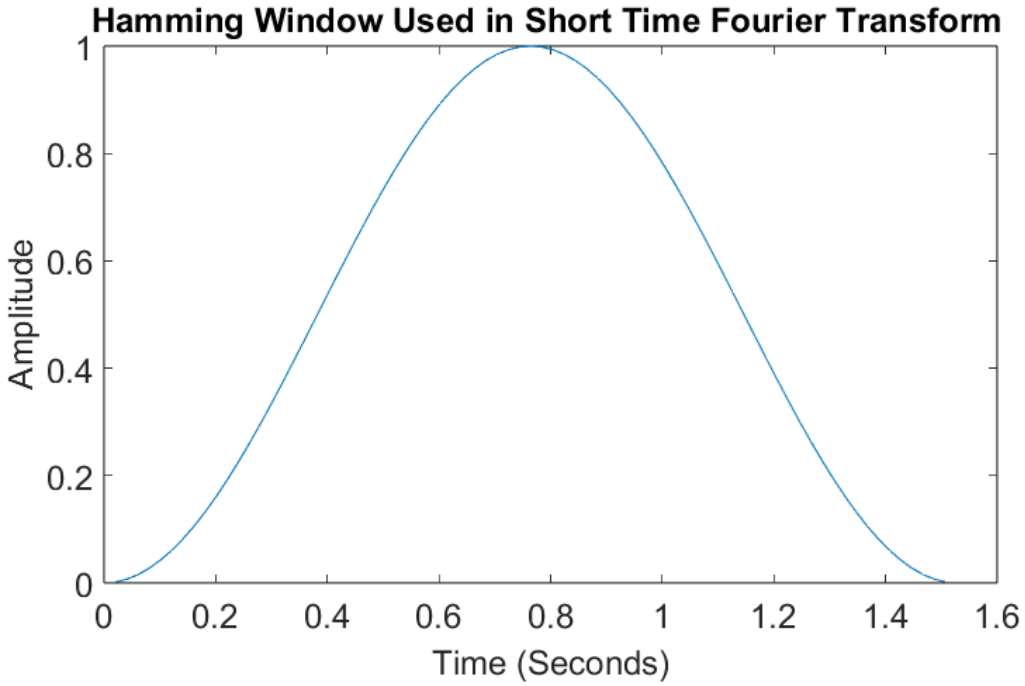

Figure 16. Hamming Window Used for STFT

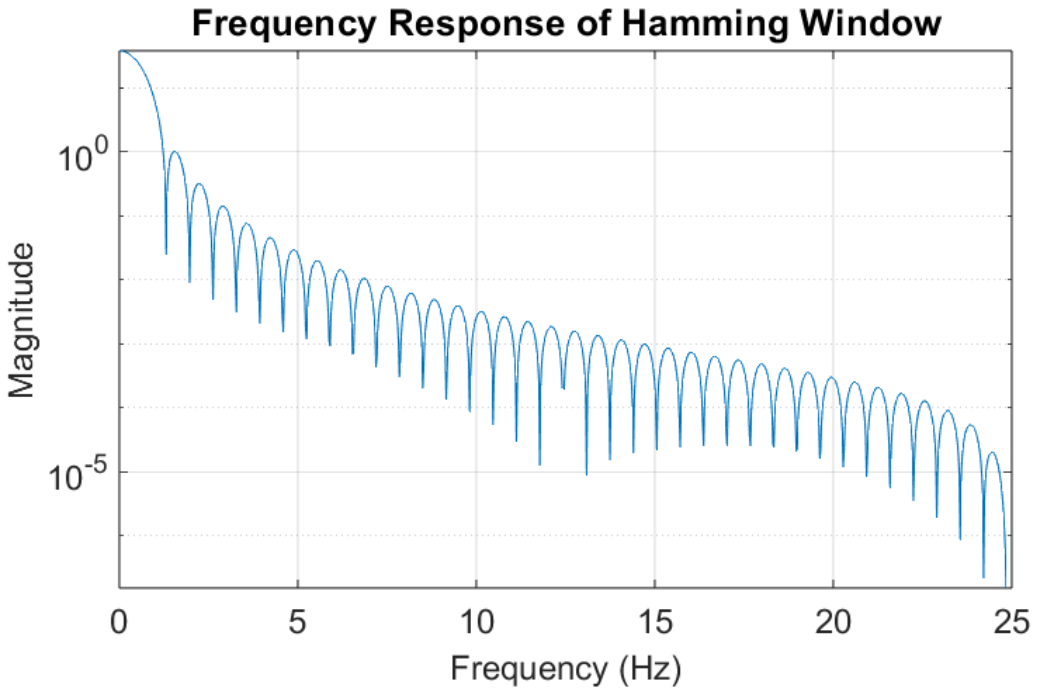

Figure 17. Hamming Window Frequency Response

An overlap of 70% was selected to further increase the temporal resolution of the STFT. We then computed the power in the 4-7Hz spectral band. With these parameters set, the estimate of spectral power in the accelerometry signal was updated roughly every 0.45 seconds. Figure 18 provides a visual representation of what this windowing process looks like in the time domain across the accelerometry data we are analyzing.

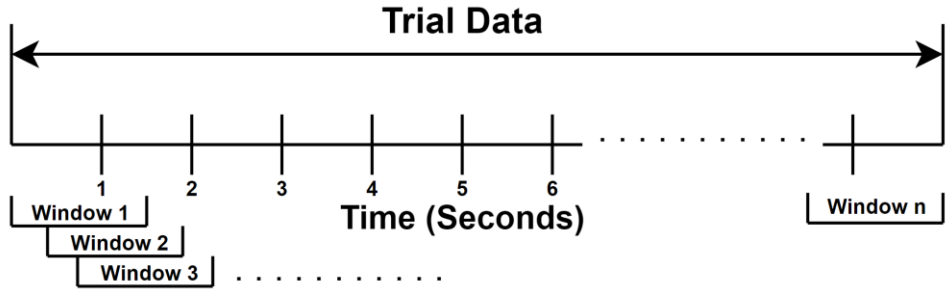

Figure 18. Short Time Fourier Transform Windowing

To perform the STFT, the typical function used is shown below. In MATLAB this is implemented using the **stft** function and is solved algorithmically [33] [34].

$$S(t,f) = \int_{-\infty}^{\infty} x(\tau)w(\tau - t)e^{-j2\pi f\tau}d\tau$$

While this formulation represents the STFT in continuous time we are dealing with discrete signals and a more accurate formulation would be as follows [34].

$$S[m,k] = \sum_{n=0}^{N-1} x[n]w[n-m]\, e^{-\frac{j2\pi kn}{N}}$$

In both the continuous and discrete STFT $x(\tau)$ and $x[n]$ represent the signal and $w(\tau - t)$ and $w[n-m]$ represent the window being used while $e^{-j2\pi f\tau}$ and $e^{-\frac{j2\pi kn}{N}}$ are the basis function in their respective domains.

In MATLAB the **cwt** function was used to calculate the CWT for our accelerometry data and calculates the wavelet transform using the following function [35].

$$CWT(t,s) = \frac{1}{s}\int_{-\infty}^{\infty} x(\tau)\psi^*\left(\frac{\tau - t}{s}\right)d\tau$$

Here $x(\tau)$ is the input signal, $\psi(\tau)$ is the mother wavelet used for the transform but in our case a Morse wavelet was selected and is shown as $\psi^*\left(\frac{\tau-t}{s}\right)$ since we will be using the complex conjugate of various shifted and scaled versions of the mother wavelet as the filter bank. The Morse mother wavelet is defined in the frequency domain by the following function [40].

$$\Psi_{P,\gamma}(\omega) = U(\omega)a_{P,\gamma}\omega^{\frac{P^2}{\gamma}}e^{-\omega^\gamma}$$

$$P^2 = \beta\gamma$$

In $\Psi_{P,\gamma}(\omega)$, $U(\omega)$ represents a unit step function also known as a Heavyside step function, $a_{P,\gamma}$ is the normalization constant defined by MATLAB's L1 normalization, P^2

is equal to the time bandwidth product where γ characterizes the wavelets symmetry and β characterizes the wavelets compactness [40]. For the Morse wavelet filter bank, a time bandwidth of 60 was selected. Using these parameters for the Morse mother wavelet and shifting and rescaling it we get the filter bank shown on Figure 19 in the frequency domain. Ultimately, the CWT will convolve this filter bank with our accelerometry data and should provide us with a scalogram showing us the power content of our signal at different frequencies.

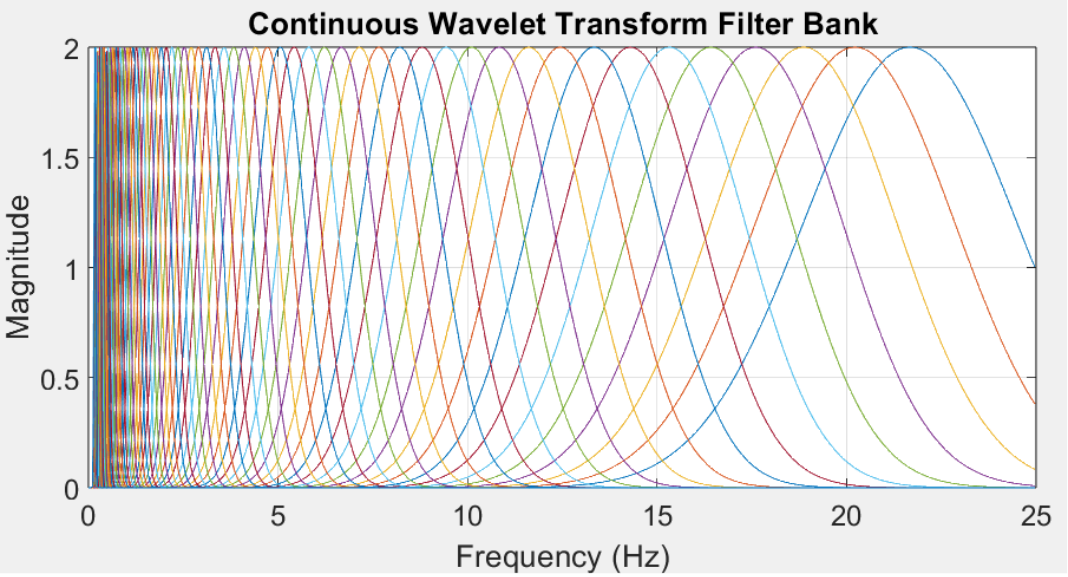

Figure 19. Wavelet Transform Filter Bank

3.3 Accelerometry Frequency Spectrum

To compare and analyze the frequency localization capabilities of the STFT and CWT previously mentioned we can perform both frequency spectrum analyses on the Apple Watch and IPG accelerometry. Figure 20 shows the data collected from P4 with a DBS intensity of 60%. From the trial logs provided we know that this participant experienced tremor during the sitting and standing portion of the trial. With this

information in mind, we expect frequency spectrum peaks in the 4-7Hz range during the sitting and standing portions of the trial.

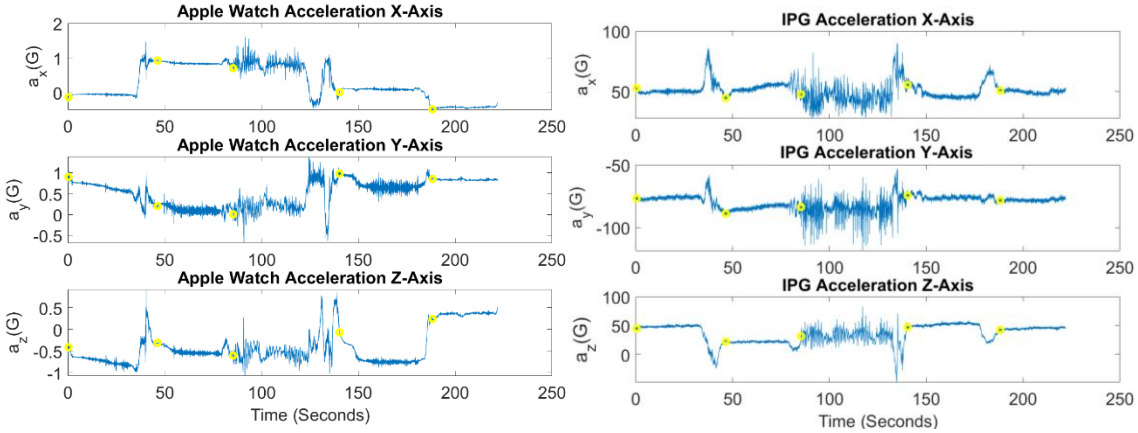

Figure 20. Apple Watch and IPG Accelerometry (P4, 60% DBS)

An STFT and CWT were applied to the three accelerometry axes of the Apple Watch and IPG. For each transform, the results were combined by calculating the L_2 norm across the axes using the following approach.

$$W_{L_2} = \sqrt{|W_x|^2 + |W_y|^2 + |W_z|^2}$$

W_x, W_y, and W_z are the results of the CWT for each accelerometry axis. After taking the L_2 norm of the CWT results across all three accelerometry axis the power was calculated on a decibel scale by using the following calculation.

$$W_{L_2,dB} = 20log_{10}(W_{L_2})$$

However, since we are comparing different frequency analysis techniques and different devices to collect the accelerometry data we will normalize $W_{L_2,dB}$ using the formulation shown below to compare peak prominence on the same scale. This process will normalize the dB scale to values between 0 and 1.

$$W_{L_2,dB,norm} = \frac{W_{L_2,dB} - W_{L_2,dB}(min)}{W_{L_2,dB}(max) - W_{L_2,dB}(min)}$$

Figure 21 shows the normalized power in P4's IPG accelerometry data using both STFT and CWT. Analyzing the normalized spectrogram and scalogram of the IPG accelerometry we can see that the normalized CWT scalogram appears to provide more distinct markers of power concentration across all frequency bands. However, there is no clear distinction of periods where tremor is present even though we know that there should be tremor during the sitting and standing phases of this trial. Periods in which we expect tremor to be present are outlined in red on Figure 21.

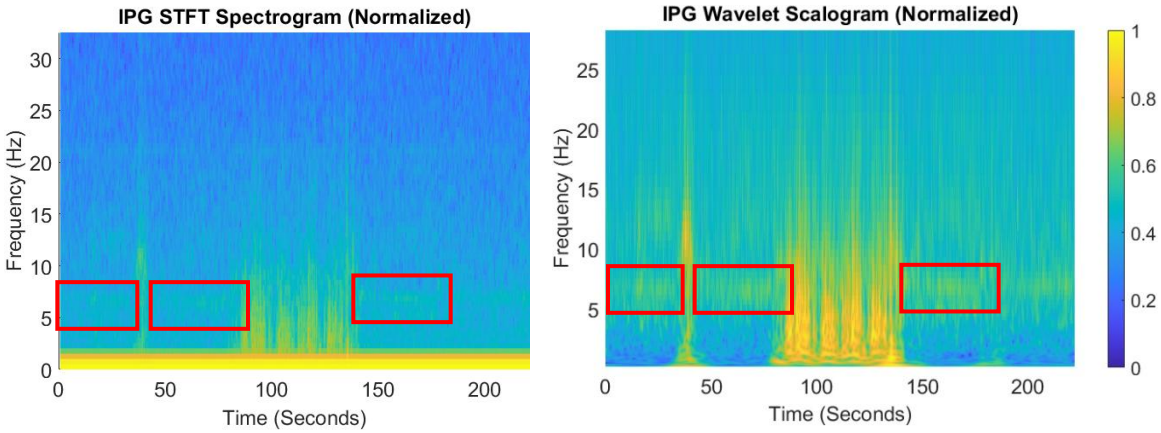

Figure 21. IPG, STFT vs CWT (P4, 60% DBS)

Now if we observe the spectrogram and the scalogram of the Apple Watch accelerometry, Figure 22, we can see there appears to be markers of power concentration within the tremor frequency band, 4-7 Hz. Initial observations would suggest that the Apple Watch is more effective at capturing tremor than the IPG. Furthermore, Figure 22 would also suggest that using a CWT is more effective at capturing and localizing the frequency spectrum of the collected data.

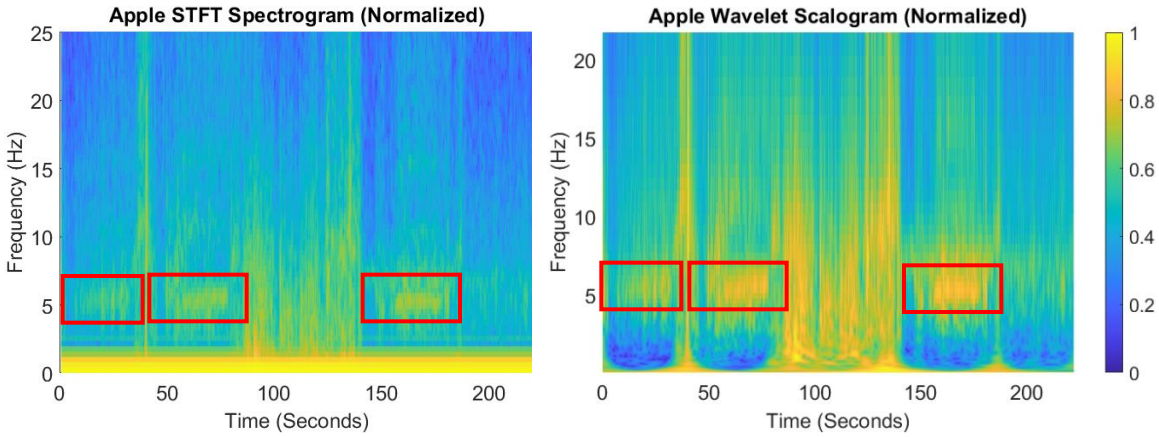

Figure 22. Apple Watch, STFT vs CWT (P4, 60% DBS)

Examining a time slice where tremor is present during standing in both the STFT spectrogram and CWT scalogram, Figure 23, we observe that the CWT scalogram offers better frequency localization due to its more distinct peak between the 4-7Hz range. In general, it appears peaks have a greater prominence and width when using a CWT compared to an STFT.

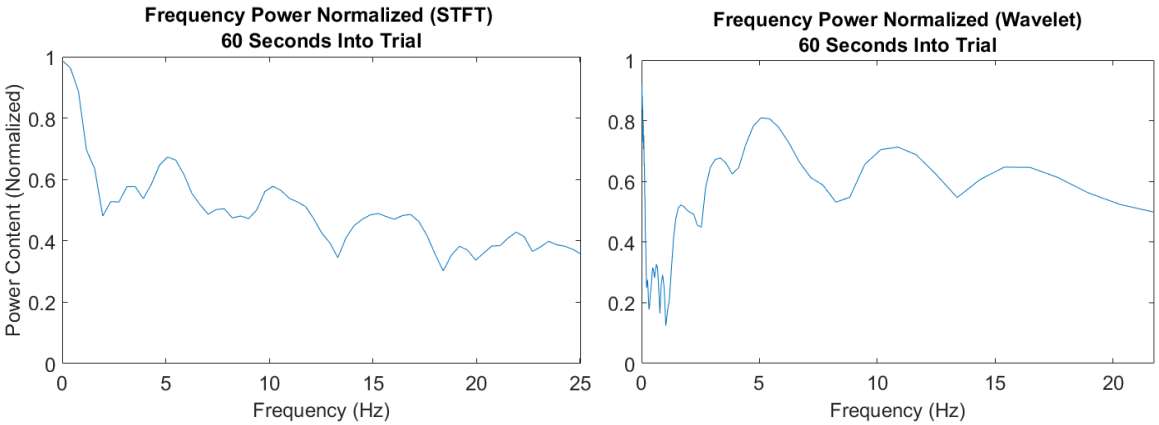

Figure 23. STFT vs Wavelet Frequency Spectrum at 60 Seconds (P4, 60% DBS)

Taking another time slice, Figure 24, shows a moment where P4 is sitting and experiencing tremor. We once again see the CWT scalogram provides a much more distinct peak between the 4-7Hz frequency band and the peak itself has a more regular shape when compared to a time slice taken from the STFT spectrogram at the same point in time. When using peak detection using an STFT may even present issues since there

are 2 peaks in the 4-7Hz range when compared to the CWT which almost appears to just have one peak with a greater width and prominence.

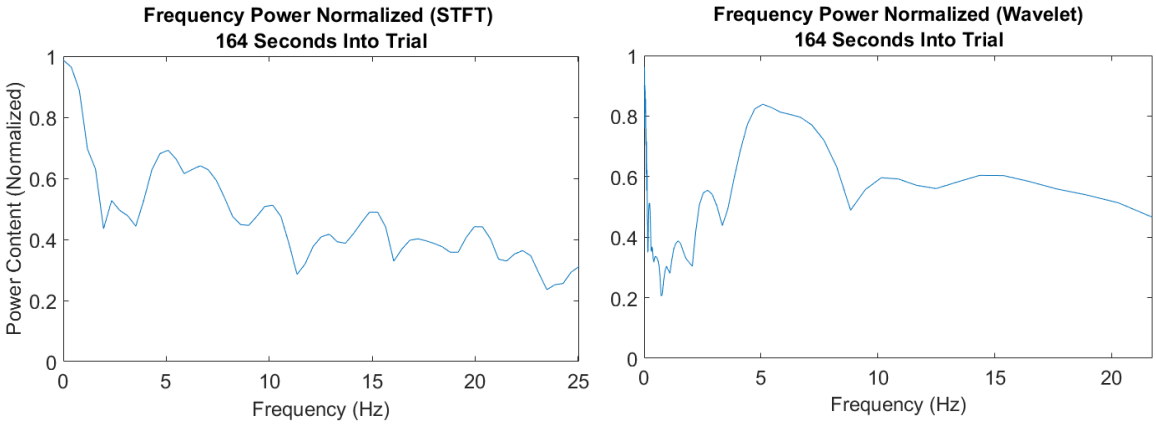

Figure 24. STFT vs Wavelet Frequency Spectrum at 164 Seconds (P4, 60% DBS)

Next, we selected a trial in which the logs indicate there was a slight tremor while standing. This is done to determine whether a STFT or CWT provides better frequency localization for the 4-7Hz band during slight tremor for both Apple Watch or IPG accelerometry. Figure 25 shows the accelerometry for both the IPG and Apple Watch collected from P3 while DBS was set to 60% of the prescribed levels.

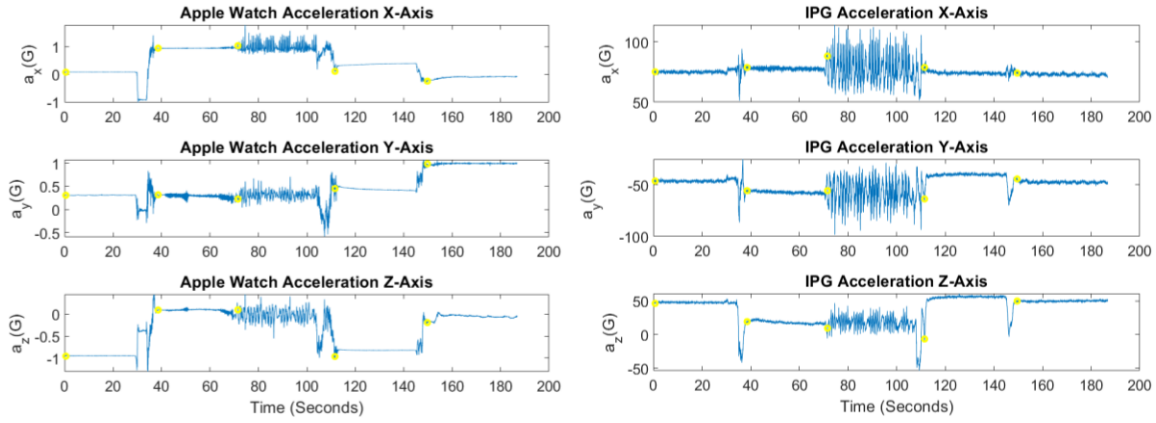

Figure 25. Apple Watch and IPG Accelerometry (P3, 60% DBS)

By conducting a visual inspection of the IPG STFT spectrogram and CWT scalogram we can once again determine that while there is some concentration of power

within the 4-7Hz band during the expected tremor period there it does not appear to be significant.

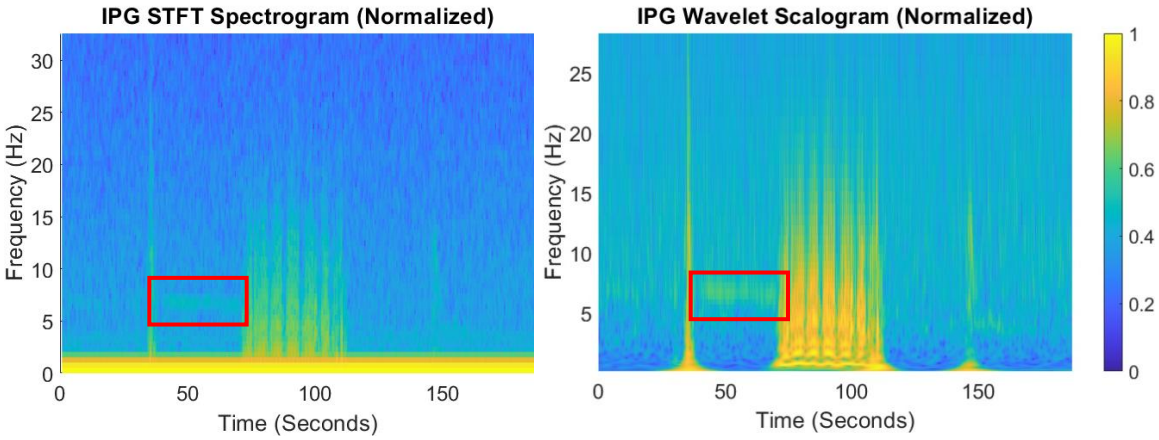

Figure 26. IPG, STFT vs CWT (P3, 60% DBS)

Moving over to the Apple Watch accelerometry collected during this trial we can immediately tell there are much more significant markers that tremor did occur during the time frame indicated by the trial logs, Figure 27. With a simple visual inspection of the region outlined in red, we can see that there is a significant amount of power concentration within the 4-7Hz frequency band in the Apple Watch accelerometry compared to the IPG accelerometry. Focusing on the Apple Watch accelerometry we can further determine that the CWT did a better job of localizing peaks within the frequency band of interest when compared to the STFT.

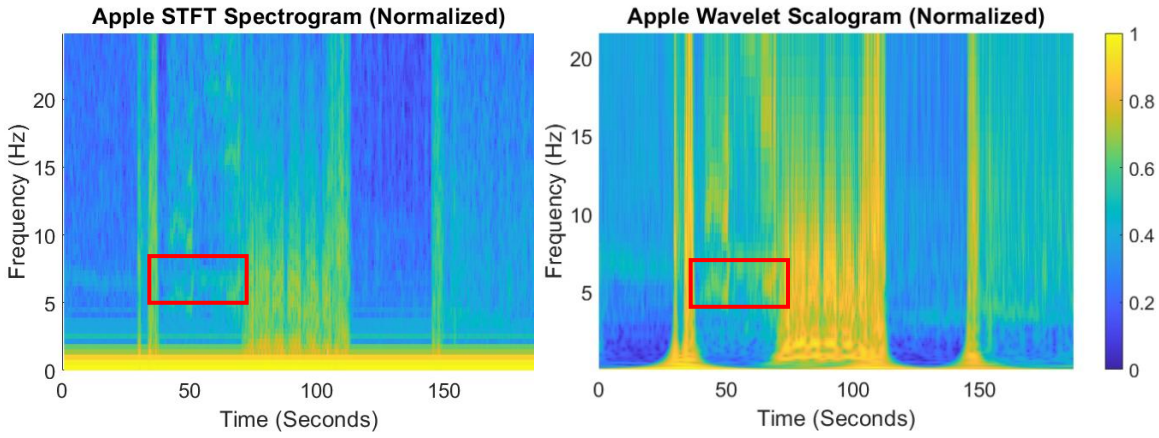

Figure 27. Apple Watch, STFT vs CWT (P3, 60% DBS)

By taking a closer look at one of the time slices where the P3 experienced tremor, Figure 28, we once again see the CWT provides a more distinct peak within the 4-7Hz band with a high peak width and prominence unlike the STFT slice and overall contains more power relative to the rest of the frequency spectrum.

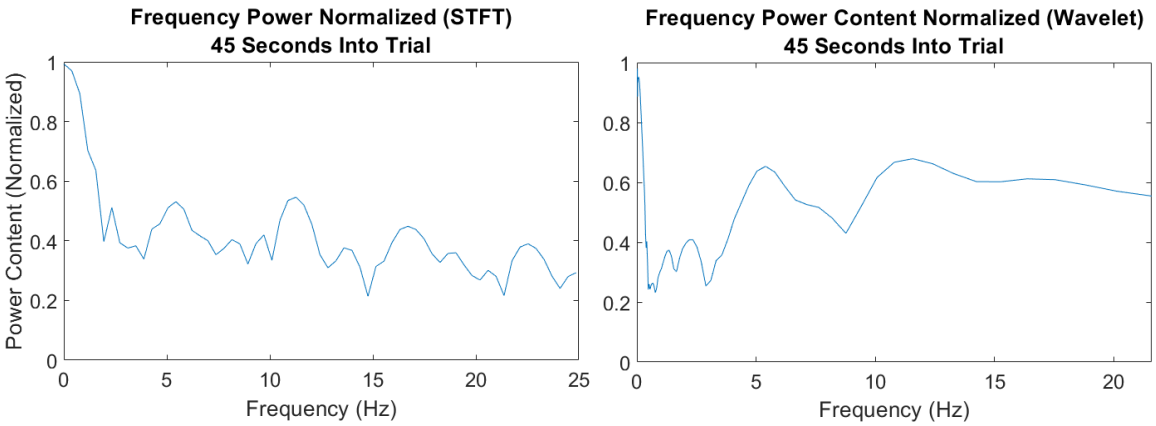

Figure 28. STFT vs Wavelet, Tremor at 45 Seconds (P3, 60% DBS)

Since we know P3 did not experience tremor in the rest of the trial we can also analyze a time slice where they should not be experiencing tremor to see what kind of characteristics the frequency spectrum has during these periods. Figure 29 shows a time slice from both the STFT spectrogram and the CWT scalogram previously shown in Figure 27 while P3 was sitting with no tremor. As expected, there are no significant peaks within the 4-7Hz frequency band and it generally holds a lower baseline at around 0.4 for both STFT and CWT.

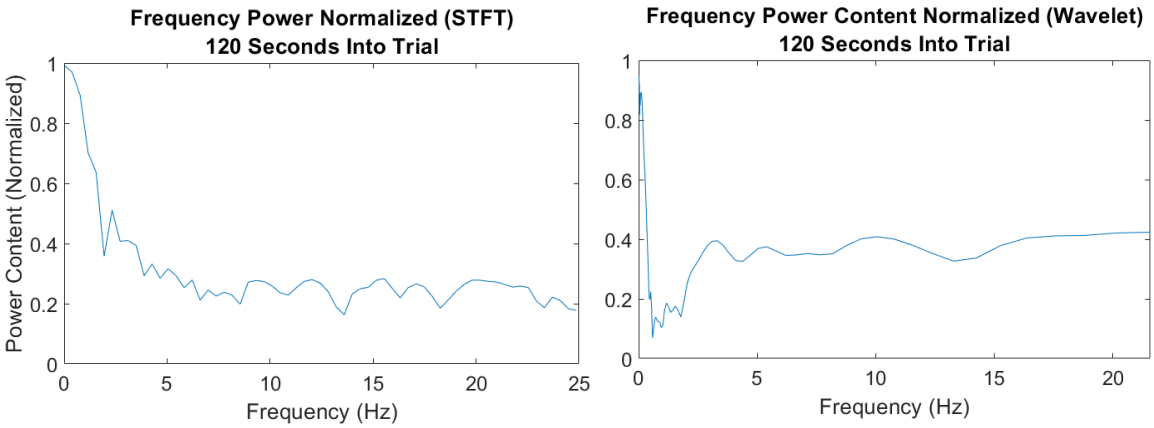

Figure 29. STFT vs Wavelet, No Tremor at 120 Seconds (P3, 60% DBS)

3.4 Presence of Harmonics

Finding peaks in the 4-7 Hz frequency band could be effective using Apple Watch accelerometry with CWT, but high-activity periods like walking can mask lower frequencies since walking shows high concentration in the 0.8-5 Hz range [41]. This interference would make it more difficult to identify significant peaks within the 4-7 Hz band. For this reason, we can leverage the fact that there are harmonics of the tremor's fundamental frequency also present to detect tremor during high activity states.

Figure 30, shows the IPG spectrogram and scalogram for a trial where P5 experienced tremor throughout the trial including while walking. If we look at the IPG accelerometry, Figure 30, there is no significant presence of harmonics while walking in the portion boxed in red. We would expect there to be concentrations of power at harmonics of the 4-7Hz frequency band if the STFT or CWT did pick up tremor. Observations were made across different participants and trial sessions, with consistently similar results. For this reason, we will no longer consider the IPG accelerometry as a viable means of detecting tremor during states of resting, high activity, or fine motor movements.

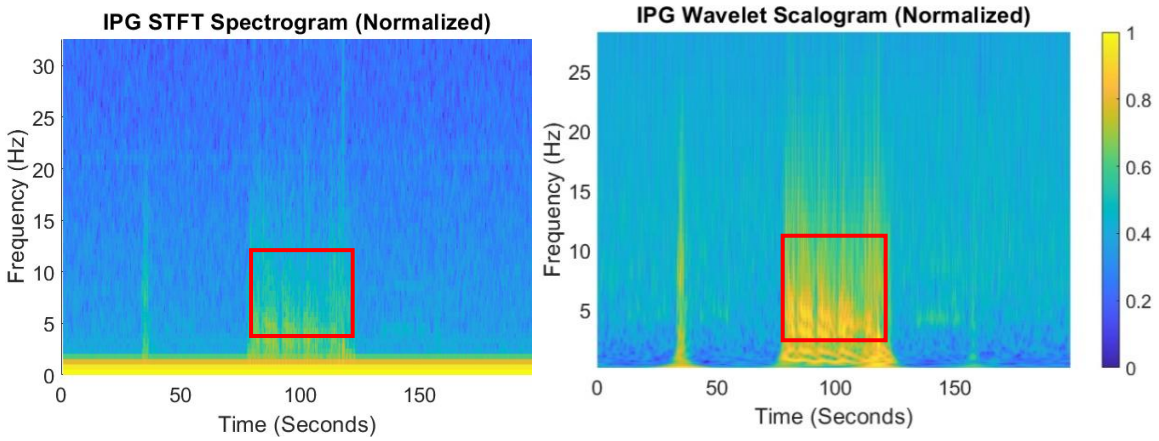

Figure 30. IPG, STFT vs CWT (P5, 60% DBS)

Contrary to the IPG accelerometry we can see the Apple Watch does a much better job at detecting periods where tremor is present. But more importantly the Apple Watch accelerometry also has the presence of harmonics while P5 is walking, this biomarker can be used to detect tremor during high activity states and is highlighted in red on Figure 31.

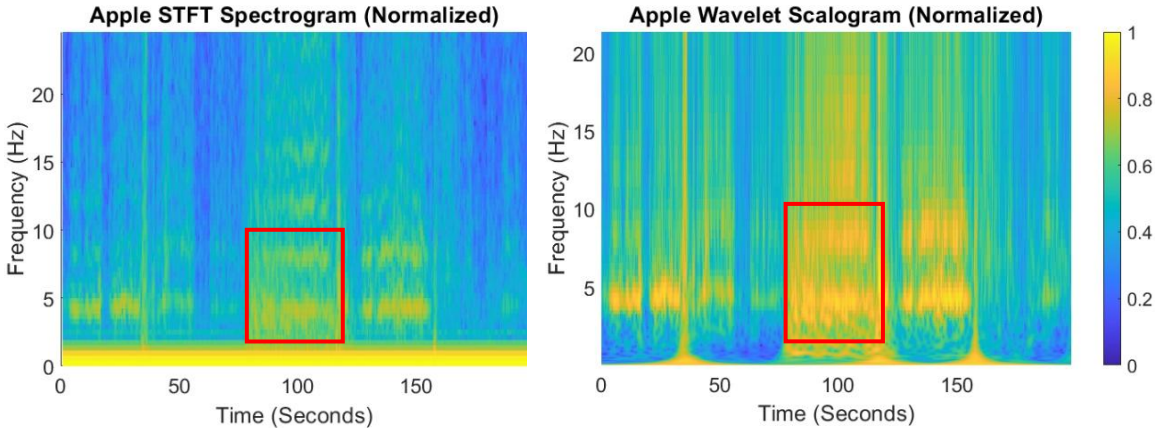

Figure 31. IPG and Apple Watch, STFT vs Wavelet (P5, 60% DBS)

If we take a time slice during the walking stage of the trial, Figure 32, we can see there are in fact harmonics present when tremor is present. Furthermore, we can also see that unlike the STFT using a CWT provides a frequency spectrum with higher peaks, and larger widths and prominences within the 4-7Hz frequency range and the associated harmonics.

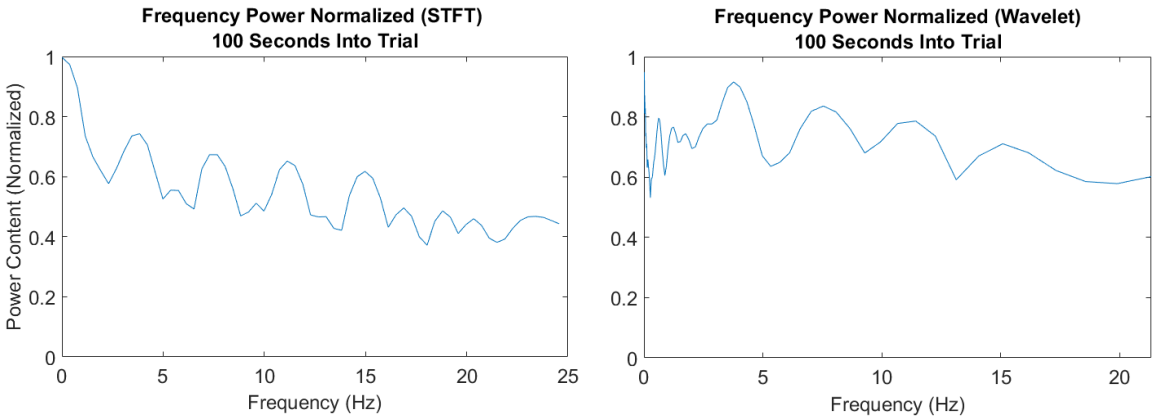

Figure 32. STFT vs Wavelet, Harmonics at 100 Seconds (P5, 60% DBS)

Taking a closer look at another time slice from both the STFT and CWT during sitting with tremor, we again see that when using a CWT, the tremors fundamental frequency and its harmonics are better defined. Ultimately, having more consistently shaped and prominent peaks allows the detection algorithm to identify spikes within the frequency spectrum more reliably.

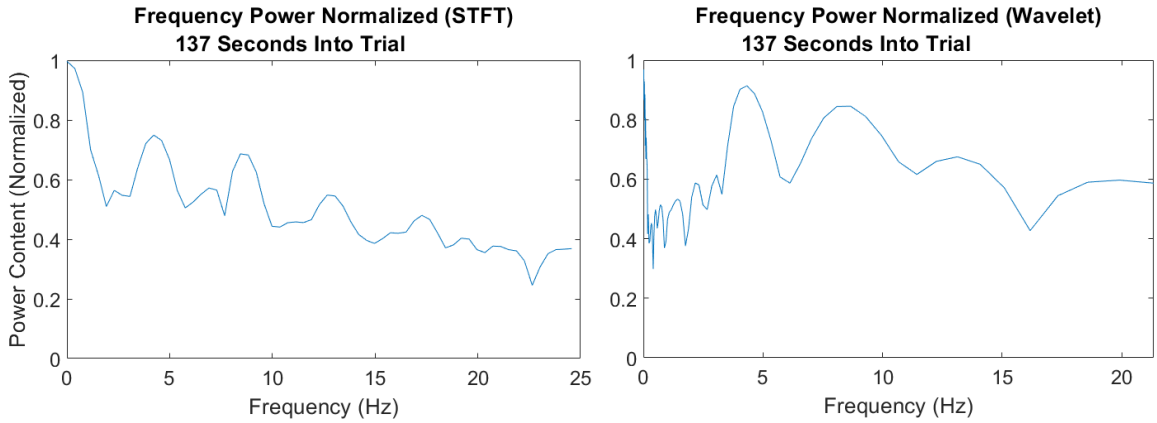

Figure 33. STFT vs Wavelet, Harmonics at 137 Seconds (P5, 60% DBS)

3.5 Thresholding

When determining what thresholds to set for tremor classification it first had to be determined how sensitive the detection algorithm would be to varying intensities of tremor. For this we consulted with the tremor severity results provided by StrivePD, the application used to collect and store the Apple Watch meta data. While StrivePD does

provide us with tremor intensity classifications ranging from “none” to “strong” it provides this as a percentage over a sixty second window and would not be suitable for our closed loop application since we seek to detect tremor at a much finer scale. For us to be able to implement a closed loop system we require tremor to be detected in real time not post hoc. To determine what thresholds to use, a side-by-side comparison was done with the results available on StrivePD for 60 second windows and it was found and that the power content present within the 4-7Hz frequency range is around -40dB for slight tremor, mild tremor contained about -34dB of power, moderate tremor was about -25dB, and strong tremor was more than -10dB of power. But considering our goal is to classify all levels of tremor in general as tremor, a threshold of -40dB was set to detect all levels of tremor ranging from slight up to strong. It should also be again noted that all plots past and presented in this thesis show the power content of the frequency spectrum as normalized values to facilitate comparisons between STFT, CWT, and between Apple Watch and IPG accelerometry. Table 2 summarizes the applied thresholds and their corresponding tremor intensity. Tremor displacement at different tremor intensities were provided by the Rune Labs API [26].

Table 2. Tremor Displacement and Intensity

Tremor Intensity	Tremor Displacement	Power Content (dB) Thresholds
Slight	Less than 0.1 cm	-40dB to -34dB
Mild	Between 0.1 cm and 0.6 cm	-34dB to -25dB
Moderate	Between 0.6 cm and 2.2 cm	-25dB to -10dB
Strong	Greater than 2.2 cm	More than -10dB

CHAPTER 4

Tremor Characterization and Detection

4.1 Tremor Intensity and DBS Settings

While developing the tremor detection algorithm, accelerometry from P1 and P2 was used to establish a baseline of different activity states while tremor is not present since these two individuals do not present tremor as their leading Parkinsons symptom. Trial logs were provided by clinical professionals present during the trials which describe if Parkinson's symptoms were observed during the session. The baselines shown in Figure 34 are a good generalization of the Apple Watch accelerometry collected from P1 and P2 and account for DBS settings at 60% and 40% of prescribed DBS intensity. Sections from these two trials will be used to compare activity states with and without tremor.

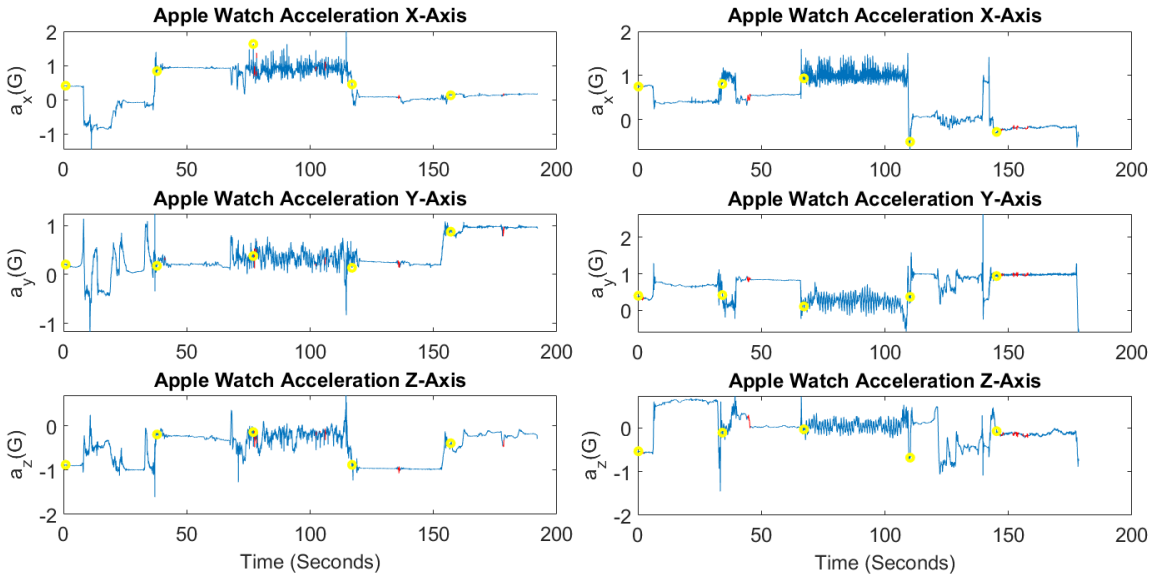

Figure 34. Apple Watch Tremor Detection Baseline (P1, 60% DBS) & (P2, 40% DBS)

Figure 35, shows some general results given by the tremor detection algorithm. Periods plotted in red are moments in which tremor was detected, and blue is no tremor. The detection algorithm results presented for P3 and P4 in Figure 35 show the

performance of the algorithm for sessions where DBS intensity is lowered from its prescribed settings in individuals with tremor as one of their leading Parkinson's symptom's. From the trial logs we also know that there was tremor present during these sessions and show much more activity when compared to the baseline shown in Figure 34.

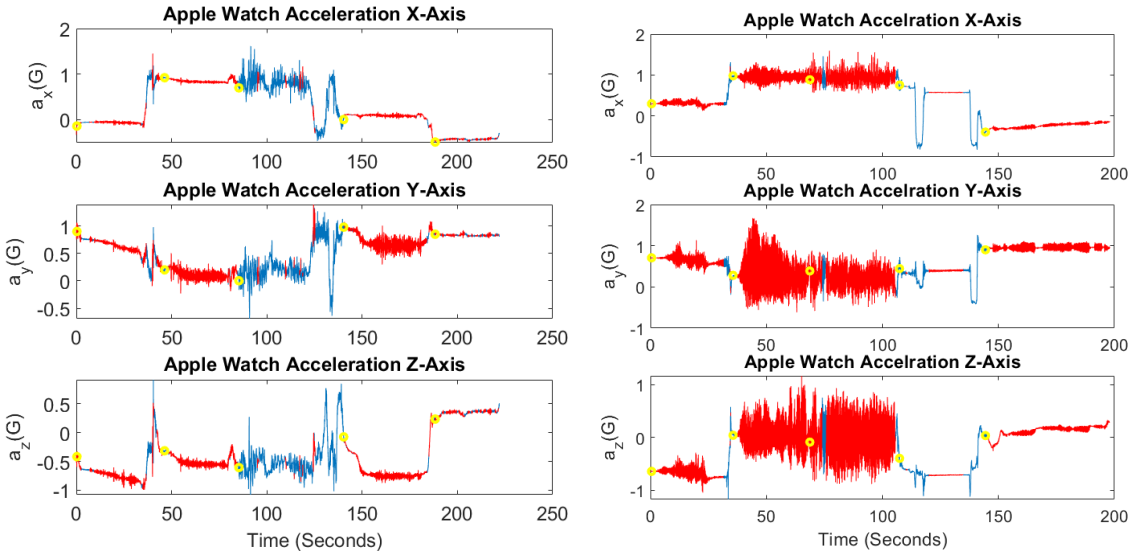

Figure 35. Apple Watch Tremor Detection (P4, 60% DBS) & (P3, 40% DBS)

The accelerometry presented in Figure 36 aims to further show the relationship between DBS intensity and tremor symptoms since DBS for P3 was reduced to 60% of its prescribed settings for the session shown. Accelerometry from P5 is also shown to give a representation of accelerometry in all the participants of the clinical trials and how tremor manifests when DBS settings are lowered. By analyzing accelerometry from all individuals who participated in these clinical trials we aim to show that the developed detection algorithm is not just tailored to one individual but can be applied to a wide range of individuals. Sessions shown in Figure 36 are also known to have had tremor present based on the clinical trial logs.

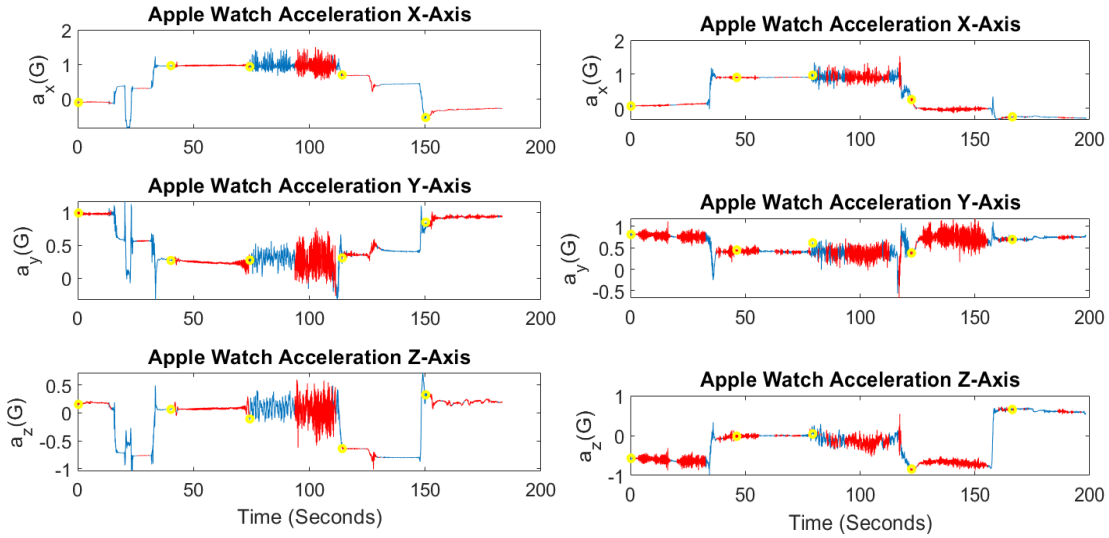

Figure 36. Apple Watch Tremor Detection (P3, 60% DBS) & (P5, 60% DBS)

To further test the performance of the developed detection algorithm we will also analyze its performance in trials where there was slight tremor was present as described by the trial logs. Figure 37 shows instances where P3 and P5 experienced slight tremor. These trials along with the others mentioned in this section demonstrate in general the detection algorithms versatility in detecting tremor across a wide range of tremor intensities and physical activities.

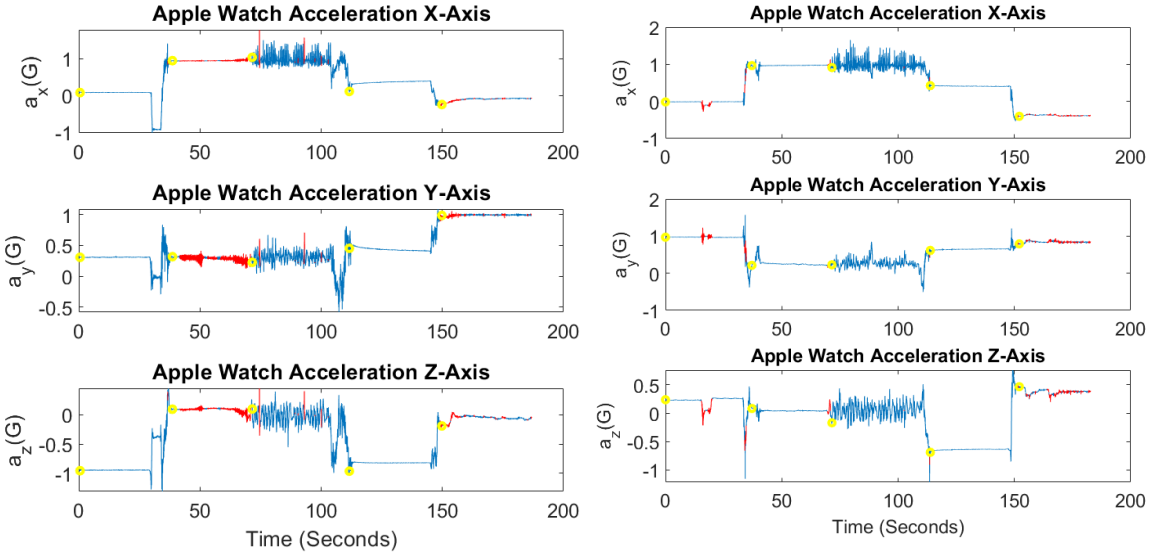

Figure 37. Apple Watch Tremor Detection (P3, 60% DBS) & (P5, 100% DBS)

4.2 Tremor During Sitting

Focusing on specific activity state, we will demonstrate typical recordings of Apple Watch accelerometry while sitting with no tremor. A 7 seconds period is shown in Figure 38. The segment shown is between 11 and 18 seconds into the trial and corresponds to the segments boxed in red. As demonstrated in the scalogram there are no notable concentration of power in the 4-7Hz frequency band, boxed in red. Also, for the accelerometry time segment shown for sitting with no tremor there appears to be no significant 4-7Hz oscillations. P1 in this trial also had their DBS settings set to 60% and was noted as not having tremor at any point during the trial by physicians present during the trial.

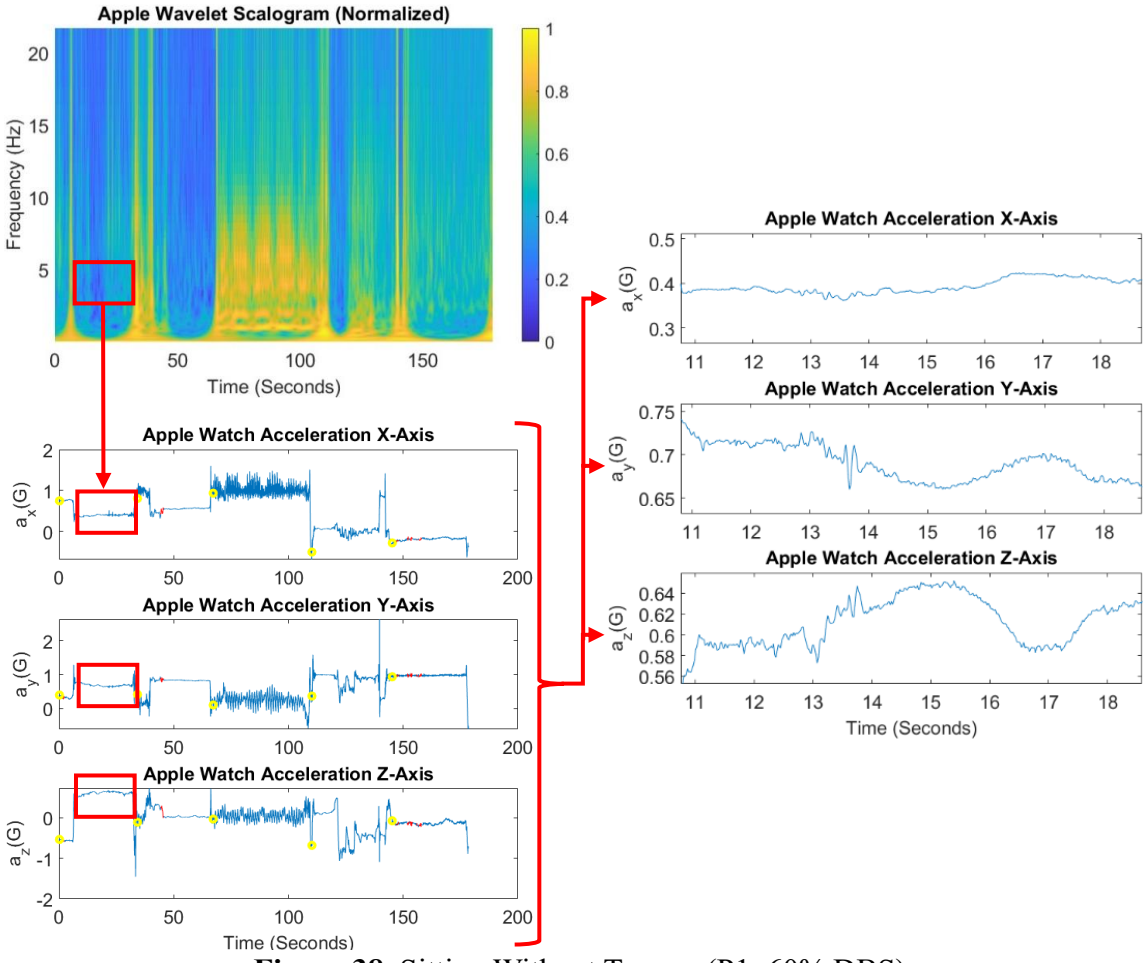

Figure 38. Sitting Without Tremor (P1, 60% DBS)

To verify our observations in the scalogram and accelerometry, Figure 39 shows the frequency spectrum at 15 seconds into the trial. As seen, there are no significant peaks in the 4-7 Hz range, and most of the frequency band's power content remains below a normalized value of 0.4.

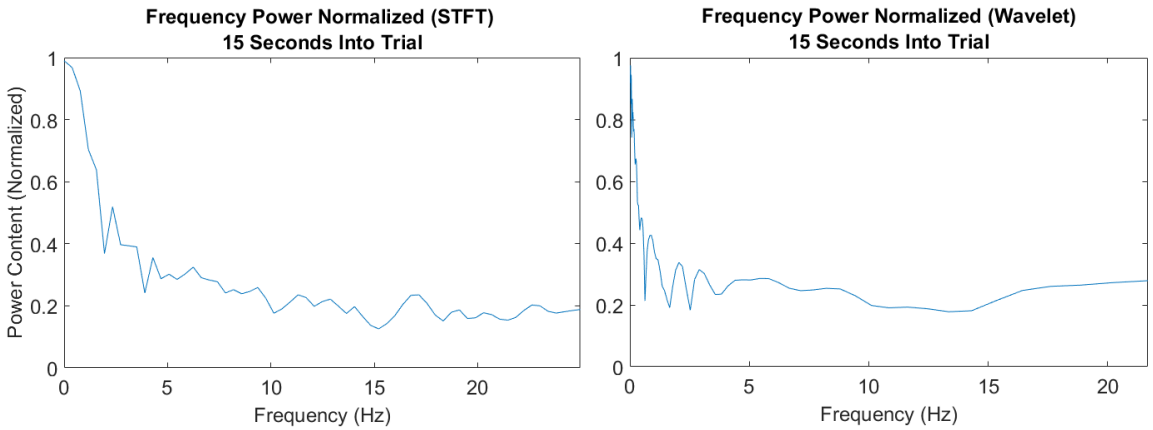

Figure 39. Sitting With No Tremor, Frequency Spectrum at 15 seconds (P1, 60% DBS)

Similarly, we can take a 7 second segment from P2's accelerometry during a period of sitting with no tremor to analyze the data's frequency and time characteristics. Figure 40 shows the selected segment and is boxed in red. From a visual inspection of P2's scalogram there does not appear to be significant spectral peaks within the 4-7Hz range in the boxed section. Taking a closer look at the time plots of the signal it remains relatively flat throughout the period of interest. Also, in the trial logs there was no tremor present throughout the trial and P2 has their DBS settings at 40% of the prescribed settings.

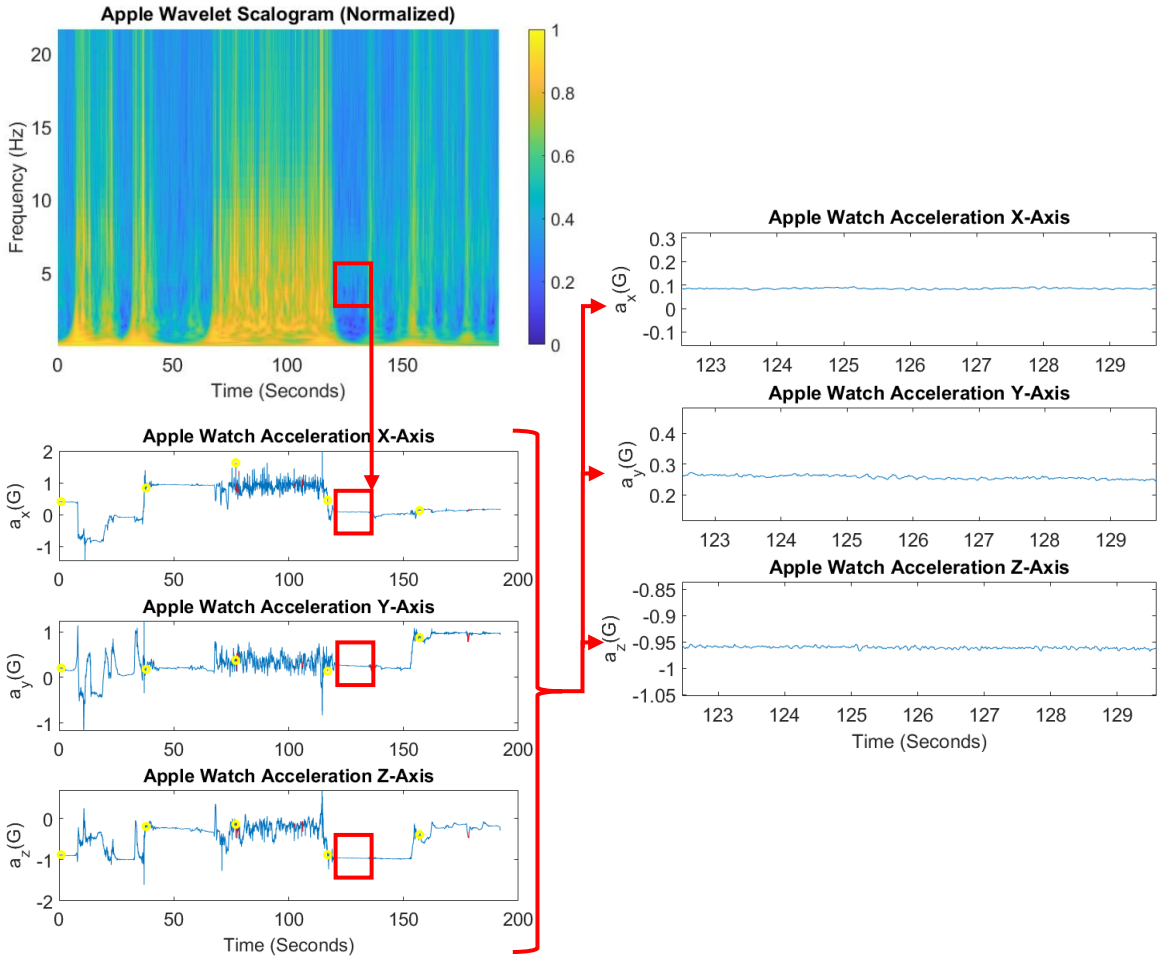

Figure 40. Sitting With No Tremor (P2, 40% DBS)

Performing a closer inspection of the frequency spectrum at 125 seconds we can verify there are no peaks within the 4-7HZ frequency range and most frequency band's power content remains at or below a normalized value of 0.4 on Figure 41. These results are expected considering there was no tremor detected and the trial logs provided by physicians that were present agree with our results.

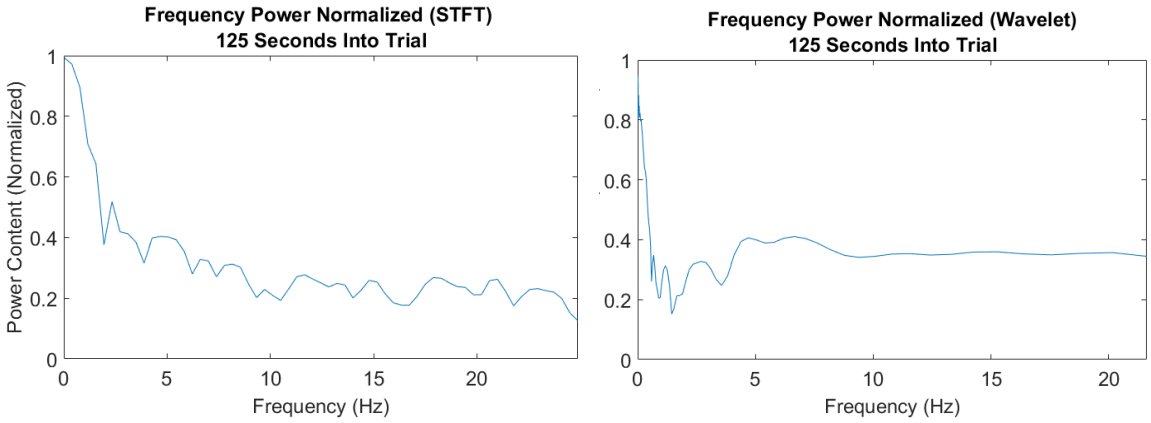

Figure 41. Sitting With No Tremor, Frequency Spectrum at 125 Seconds (P2, 40% DBS)

Moving onto trial data from P5, we analyzed periods where tremor was known to be present while sitting. The results for data fed into our detection algorithm are shown in Figure 42. Here, we will focus on analyzing the results for periods of tremor during sitting and comparing them to the baseline previously established using P1 and P2. From a visual inspection we can see that the boxed regions on the scalogram in Figure 42 does appear to show concentrations of power in the 4-7Hz frequency band. Taking a look at the accelerometry time plots we can also see that there is a significant amount of accelerometry data that was plotted in red, indicating tremor, during the sitting portion of the trial. Furthermore, a closer examination of two periods, one between 3 to 10 seconds and another between 134 to 141 seconds into the trial reveal oscillations in the 4-7Hz range. These oscillations were accurately detected by our algorithm, confirming the presence of tremor. These results agree with the information provided by the clinical trial logs.

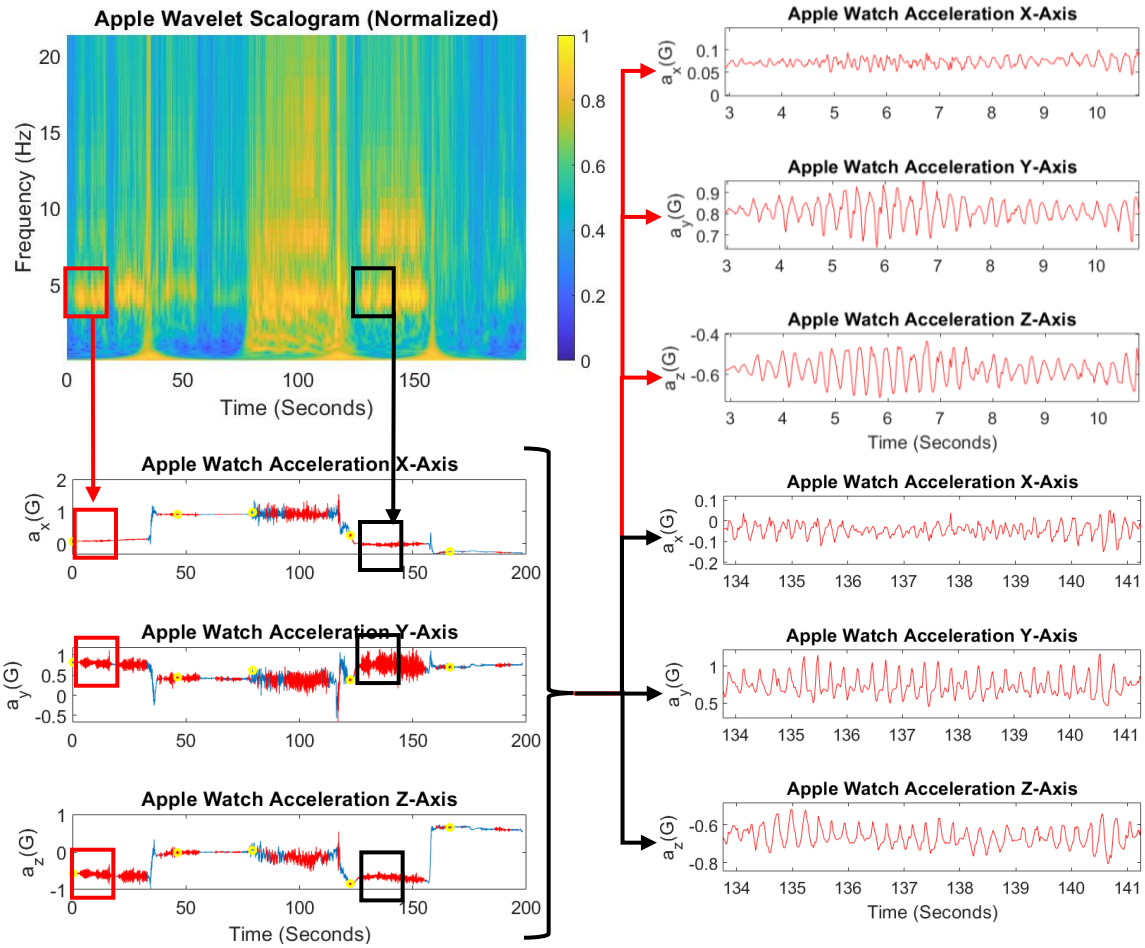

Figure 42. Sitting With Tremor (P5, 60% DBS)

Analyzing the frequency spectrum characteristics at 8 seconds into the trial, Figure 43, we also see that there is a peak present in the 4-7Hz range. However, we again see that using a CWT we get a more distinct and prominent peak which facilitates peak detection. Alternatively, the STFT frequency spectrum at 8 seconds provides a less pronounce and more irregularly shaped peak in the 4-7Hz range.

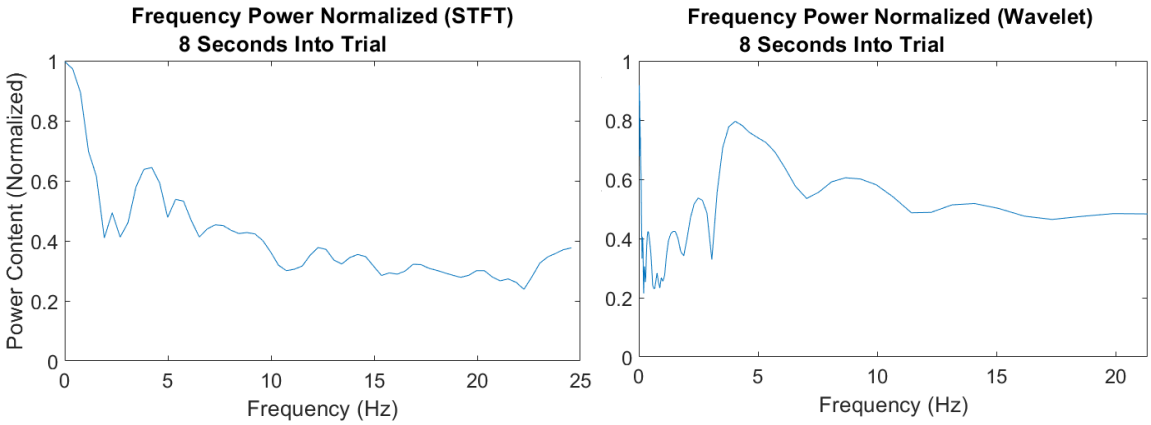

Figure 43. Sitting With Tremor, Frequency Spectrum at 8 Seconds (P5, 60% DBS)

Also, taking a closer look at 137 seconds into the trial, Figure 44, we see the frequency spectrum provided by the CWT also provides more distinct peaks in the 4-7Hz range and its harmonics when compared to using an STFT. While the STFT appears to show additional peaks between the tremor's fundamental frequency and harmonics these additional peaks could lead to less reliability in the tremor detection algorithm and missed detection if they are pronounced enough.

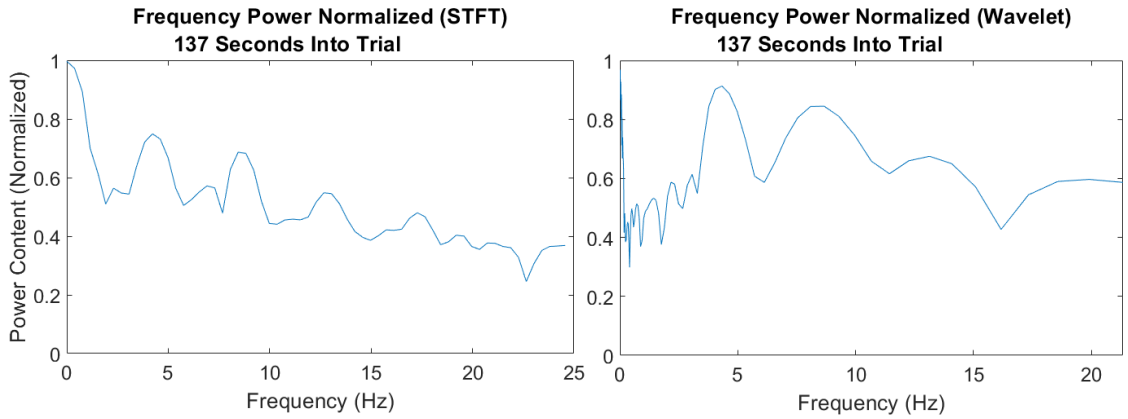

Figure 44. Sitting with Tremor, Frequency Spectrum at 137 Seconds (P5, 60% DBS)

Feeding accelerometry data collected from P4 while their DBS settings were set to 60% allows us to see how well the detection algorithm performs using a different participant's Apple Watch accelerometry data. In the accelerometry time plot shown in Figure 45, detected tremor is plotted in red and moments with no tremor are plotted in

blue. Two periods of interest where P4 was sitting are boxed in black and red. Zooming into these 7 second segments between 17 to 24 and 160 to 167 seconds into the trial we can clearly see 4-7Hz oscillations in the accelerometry data. Also, via a visual inspection it appears there are spectral peaks in the 4-7Hz frequency bands as indicated by bright yellow on the CWT scalogram also shown in Figure 45. Corresponding time segments are boxed in red and black. Clinical logs for this trial also indicate there was tremor during these periods.

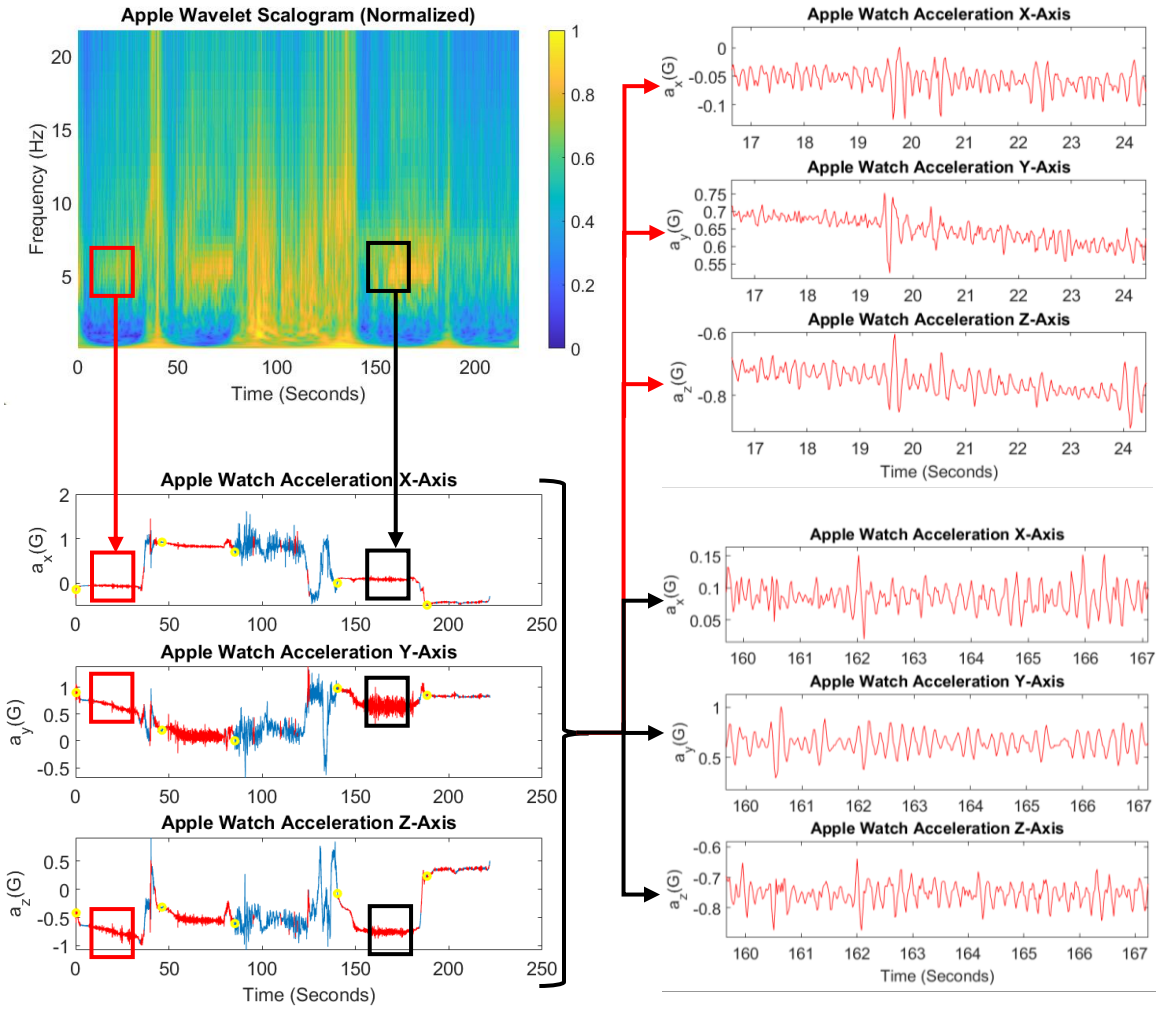

Figure 45. Sitting With Tremor (P4, 60% DBS)

A closer analysis of the frequency spectrum at 19 seconds into the trial, shown in Figure 46, allows us to directly compare the results produced by the STFT and CWT. The

CWT shows to be more effective for peak detection in the 4 to 7 Hz range, as even small irregularly shaped peaks appear more prominently within the tremor frequency band.

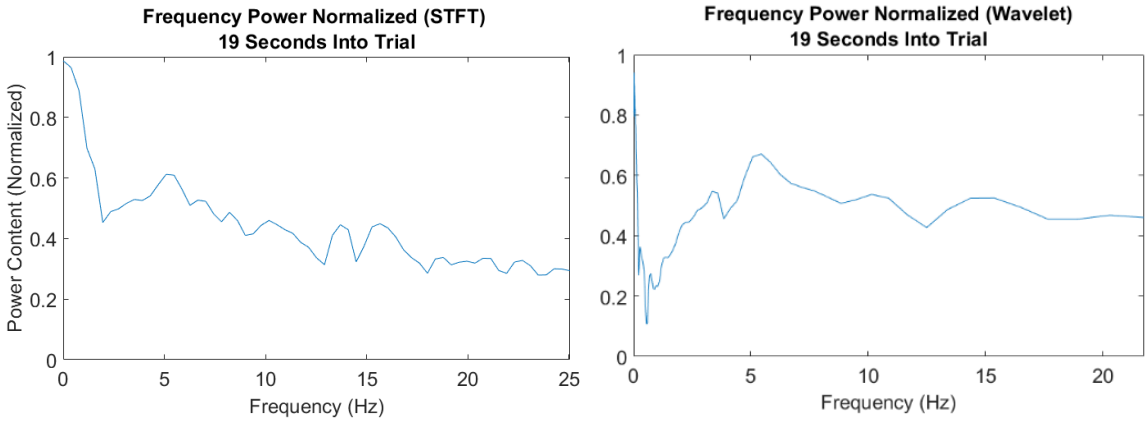

Figure 46. Sitting With Tremor, Frequency Spectrum at 19 Seconds (P4, 60 % DSB)

Similarly, if we take a look at the frequency spectrum at 164 seconds into the trial, Figure 47, we can see that the CWT generally does a better job of smoothing out peaks and makes them more distinguishable and thus more reliably to detected. Tremor during these periods were also verified to be present based on clinical trial logs provided from the study.

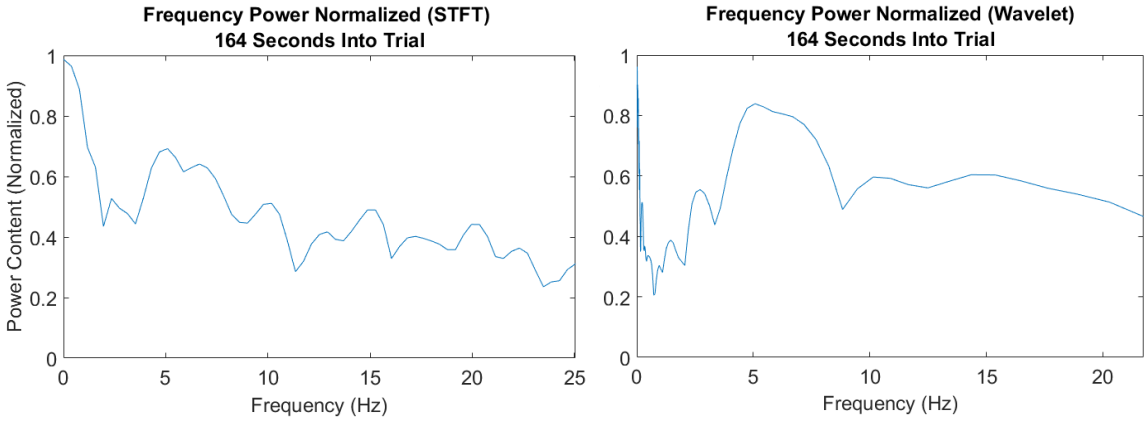

Figure 47. Sitting With Tremor, Frequency Spectrum at 164 Seconds (P4, 60% DBS)

4.3 Tremor During Standing

Like the previously analyzed physical activity state, we must first establish a baseline of the Apple Watch accelerometry characteristics while the participants are standing using recordings from P1 and P2. Figure 48 presents some baseline accelerometry recordings of P1 while they are standing with no tremor. Similar to before, a combination of trial logs, visual inspection, and peaks in the frequency spectrum's 4-7Hz frequency band was used to determine if tremor was correctly detected.

For the first set of accelerometry used to establish a baseline of standing without tremor, P1 had their DBS settings set to 60%. As shown in Figure 48, the accelerometry remains relatively flat with minimal movement across all three axes in the selected 7 second segment. This is expected, since the participants would be standing with their arms at rest by their sides. Also, the CWT scalogram doesn't present any significant spikes in the 4-7Hz range as shown in Figure 48 and boxed in red. The detection algorithm also did not detect any tremor during standing as the accelerometry was plotted in blue, no tremor.

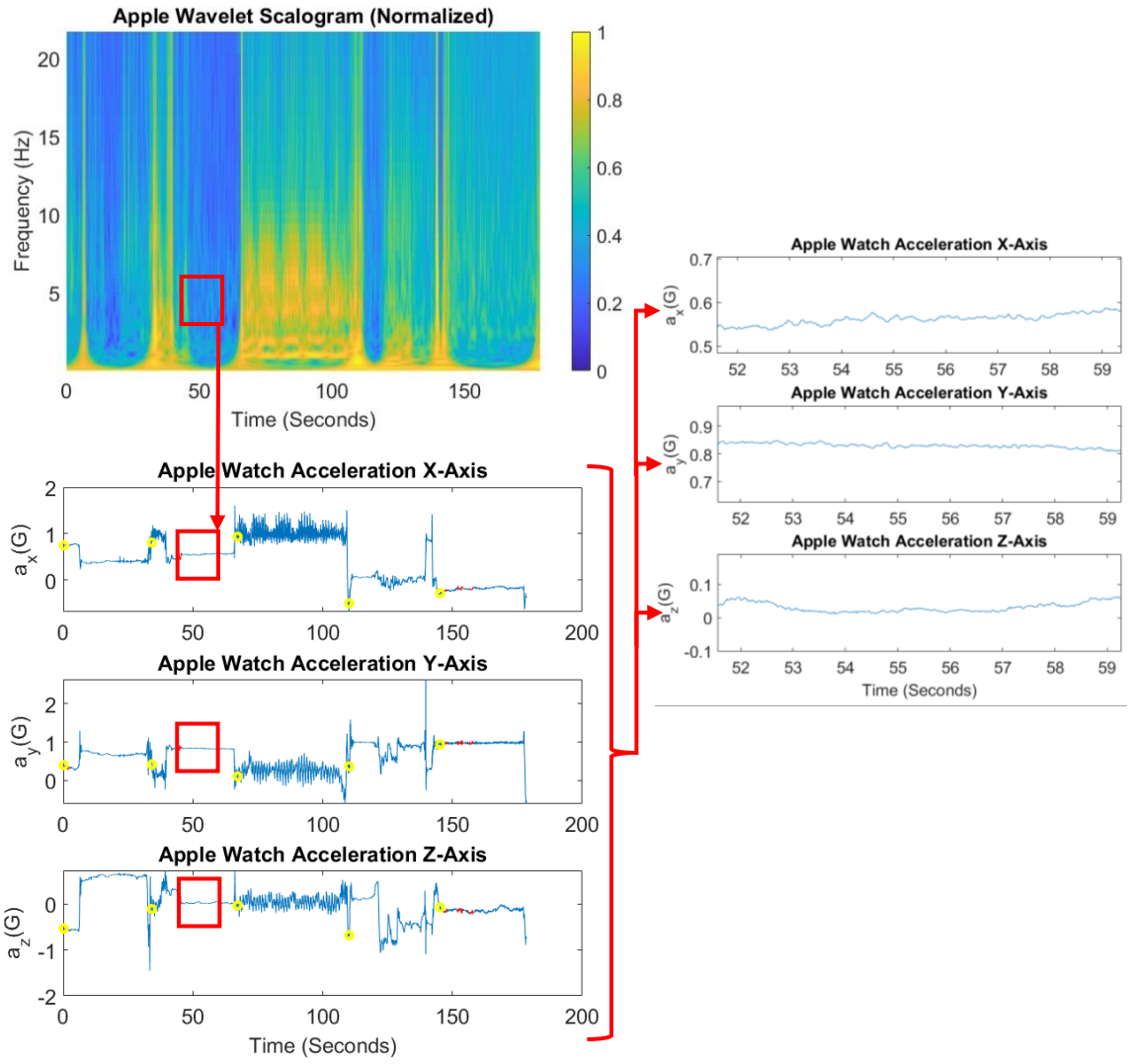

Figure 48. Standing With No Tremor (P1, 60% DBS)

With the Apple Watch accelerometry in Figure 48 showing in general a steady level while P1 is standing, the spectral measurements in Figure 49 behave as expected, showing little power concentration across all frequency bands.

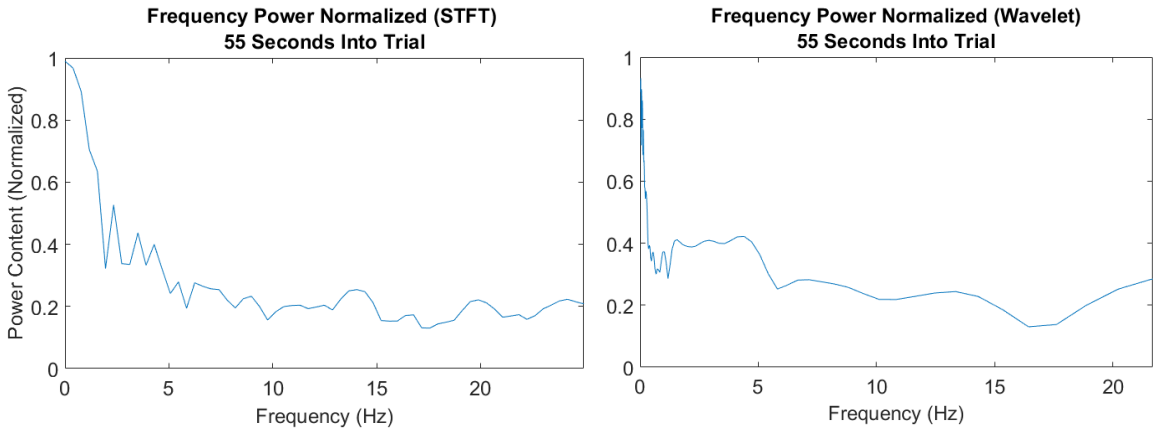

Figure 49. Standing No Tremor, Frequency Spectrum at 55 Seconds (P1, 60% DBS)

Running the detection algorithm on accelerometry data collected from P2 while DBS is set to 40%, and focusing on a period where P2 is standing we see similar results as P1 as shown in Figure 50. Observing the characteristics of the CWT scalogram and accelerometry while P2 is standing over a 7 second period we see the accelerometry time plots appear relatively flat and the CWT scalogram does not show significant power concentration in the 4-7Hz range, boxed in red. Again, on the accelerometry time plots, blue indicates that no tremor was detected. We also know from the provided clinical trial logs that tremor was not present during this session.

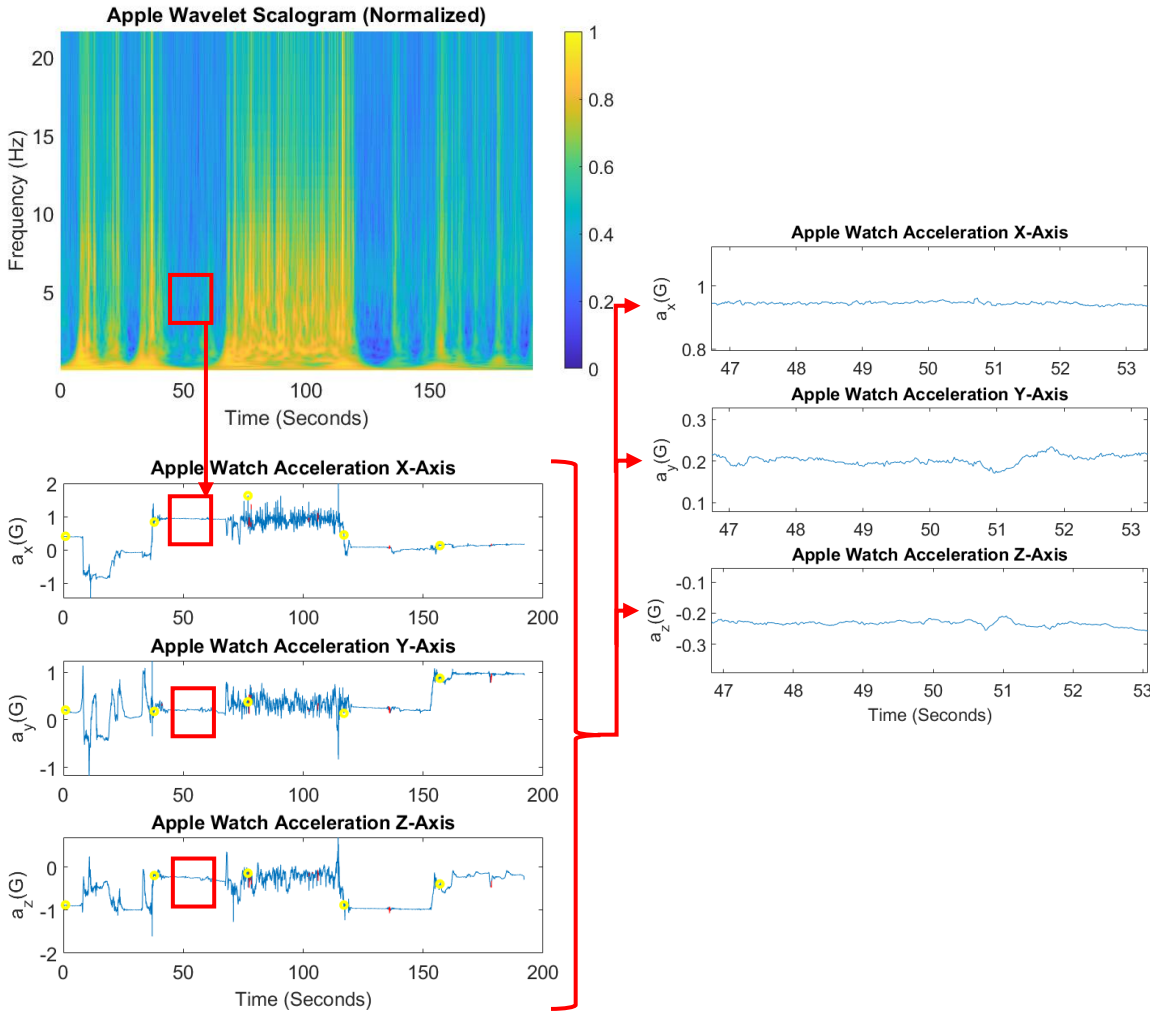

Figure 50. Standing With No Tremor (P2, 40% DBS)

A closer look at the spectral composition of the accelerometry while P2 is standing with no tremor at 50 seconds, shown in Figure 51, it shows similarities to P1. There is little power spread throughout all the frequency bands and generally stays under a normalized value of 0.5.

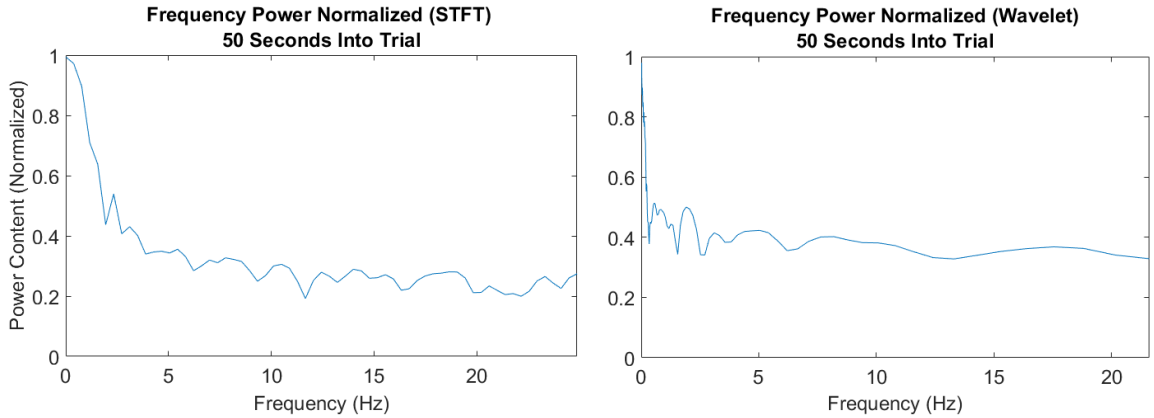

Figure 51. Standing No Tremor, Frequency Spectrum at 50 Seconds (P2, 40% DBS)

Now that a baseline has been established of what standing with no tremor should look like in the time and frequency domain, we can analyze how well the detection algorithm performed while P4 is standing with tremor. The data set we will look at had P4 with DBS settings at 60% of their prescribed DBS intensity, Figure 52. Again, data plotted in red indicate moments where tremor was detected by the algorithm and blue indicated moments where there is no tremor. Analyzing the CWT scalogram period outlined in red, there are is a clear concentration of power in the 4-7 Hz range. Also, taking a closer look at a 7 second segment on the accelerometry time plot between 57 and 64 seconds, we can see that in fact there were oscillations in the Apple Watch accelerometry data in the 4-7 Hz. From the clinical trial logs we also know in fact there was tremor during this period. A visual inspection suggests that the detection algorithm effectively identifies tremor while standing, as it correctly marks this portion in red.

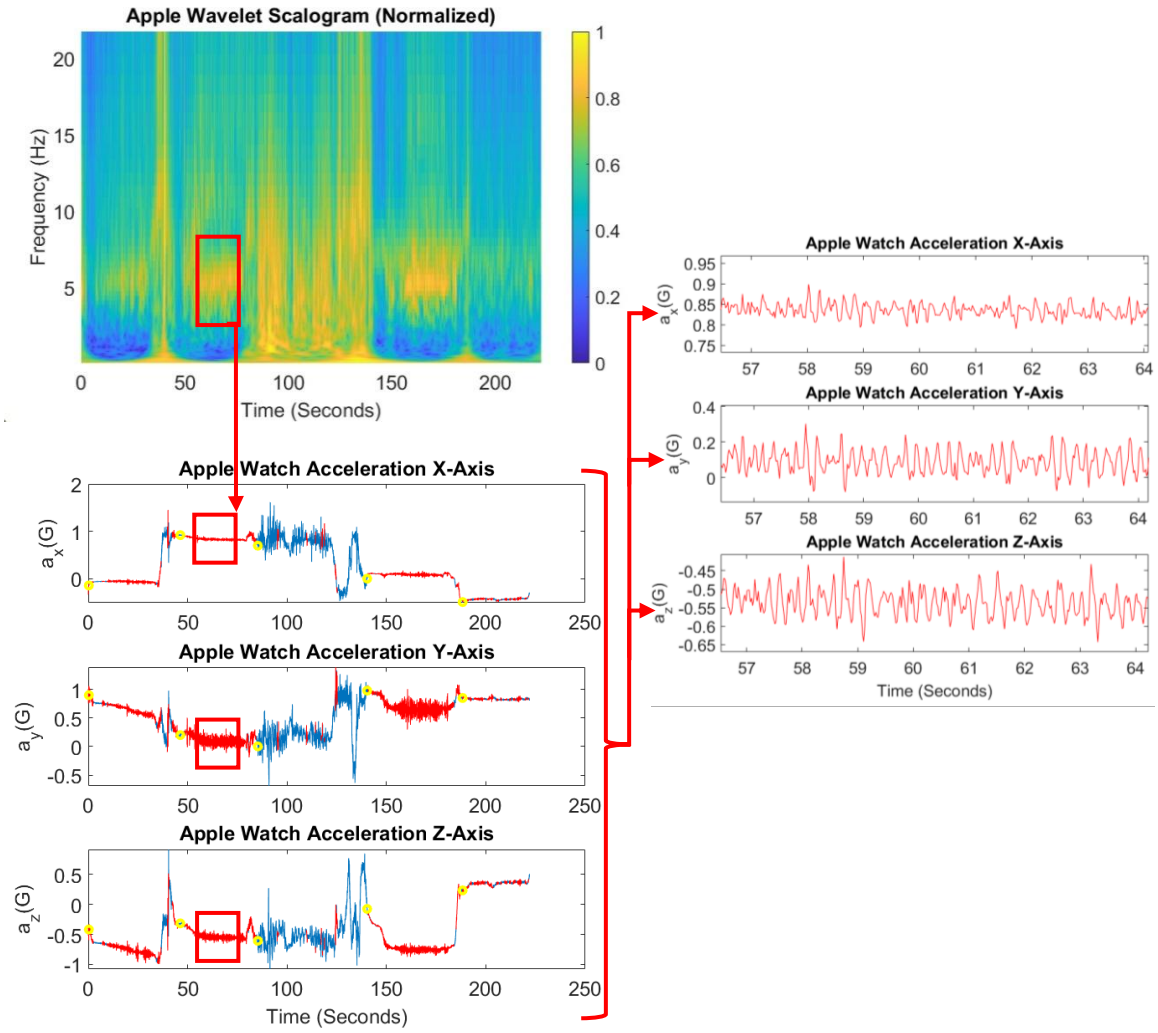

Figure 52. Standing With Tremor (P4, 60% DBS)

Comparing the CWT used for peak detection in our detection algorithm to the STFT at 60 seconds into the trial, Figure 53 shows that frequency peaks within the 4-7 Hz band are better defined and more pronounced with the CWT. We can also see that the normalized power content of the frequency spectrum sits at a higher level when compared to the baseline from P1 and P2.

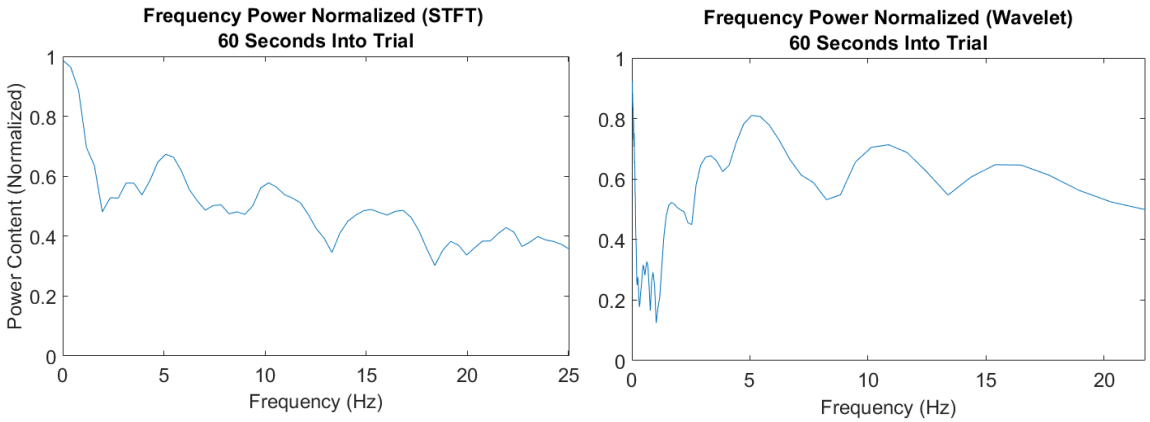

Figure 53. Standing With Tremor, Frequency Spectrum at 60 Seconds (P4, 60% DBS)

Using the detection algorithm on a set of data where P3 exhibited a slight tremor while standing, as indicated by the trial logs, produces the results shown in Figure 54. Similar to P4, P3's accelerometry also demonstrates a power concentration in the CWT scalogram's 4-7 Hz frequency band when a slight tremor present while standing. In Figure 54 tremor that was detected by the algorithm is plotted in red and no tremor is plotted in blue on the accelerometry time plot. Making a visual inspection at a finer scale, we can see in Figure 54 that the selected 7 second period does present oscillations in the accelerometry between 4-7 Hz while standing. This gives us confidence that the thresholds set for peak detection are accurate, even for detecting slight tremors. We can also confirm that there was tremor during this period by consulting with the clinical logs provided by the study.

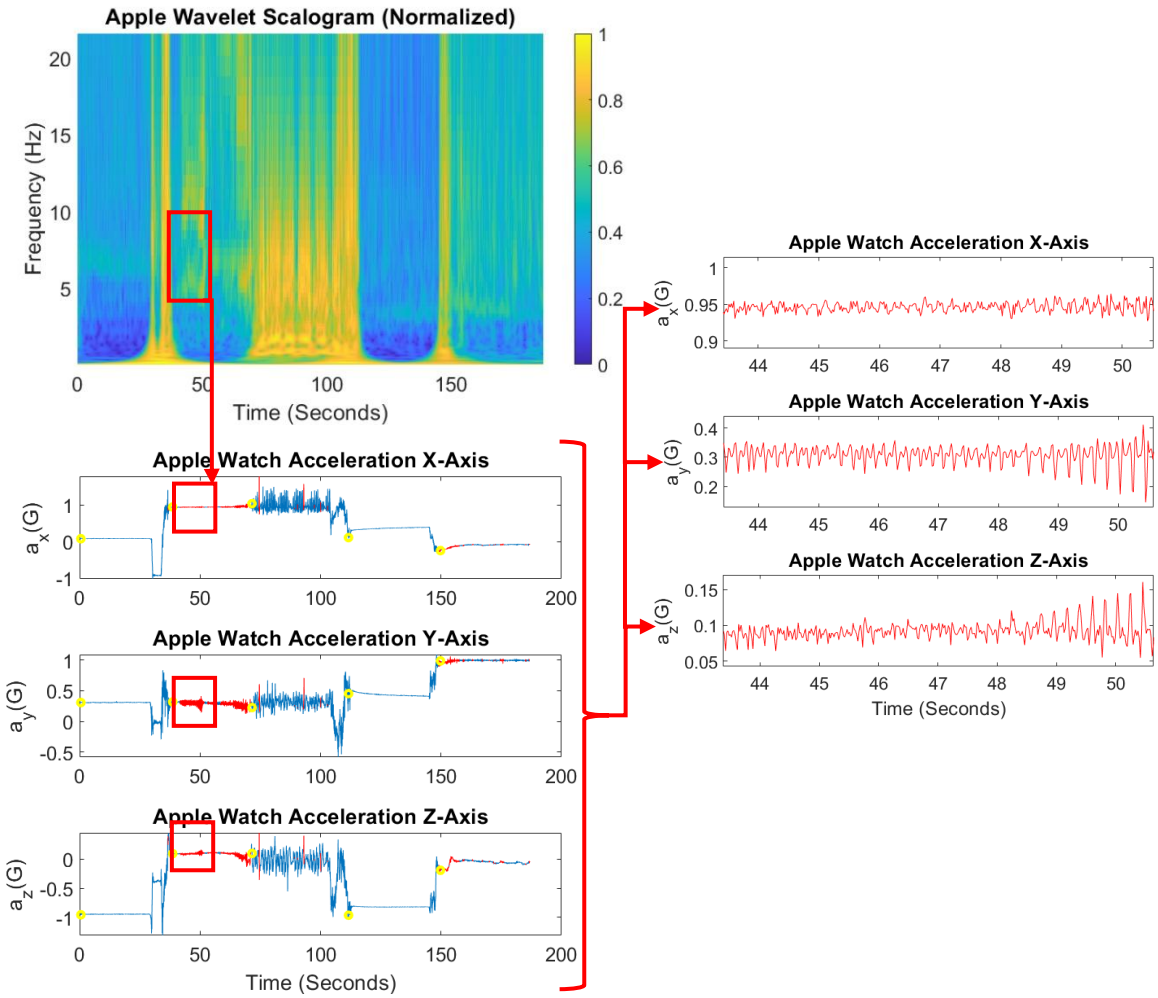

Figure 54. Standing With Tremor (P3, 60% DBS)

Conducting an analysis of the frequency spectrum at 45 seconds into the trial, Figure 55, we again see that by using a CWT we get better defined and a more pronounced peak in the 4-7Hz range while tremor is present. If we look at the STFT of the signal at the same period we see that while the frequency spectrum does present a peak within the 4-7Hz range its width, prominence, and height is smaller when compared to the peak present in in the CWT.

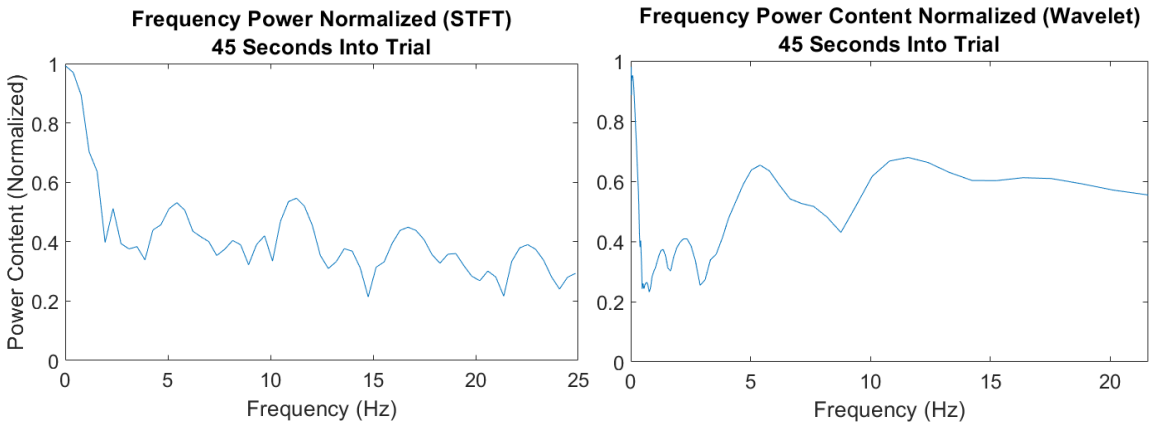

Figure 55. Standing With Tremor, Frequency Spectrum at 45 Seconds (P3, 60% DBS)

4.4 Tremor During Physical Activity

To establish a baseline for the Apple Watch accelerometry during physical activity P2's trial data will be used while they are walking. Figure 56 shows the results after passing the data through the tremor detection algorithm. While we do see a large power concentration within the 4-7Hz tremor frequency band and in the 8-14Hz frequency band there was no tremor detected while P2 was walking. As noted in the 7 second period presented in Figure 56 the signal is plotted in blue, no tremor detected. While we do see there is a lot of activity on the time plot when P2 is walking if we analyze the CWT scalogram there doesn't appear to be any spectral peaks in the 4-7Hz tremor frequency band or in the 8-14Hz harmonics frequency band. We also know from the clinical logs that P2 did not experience tremor while walking.

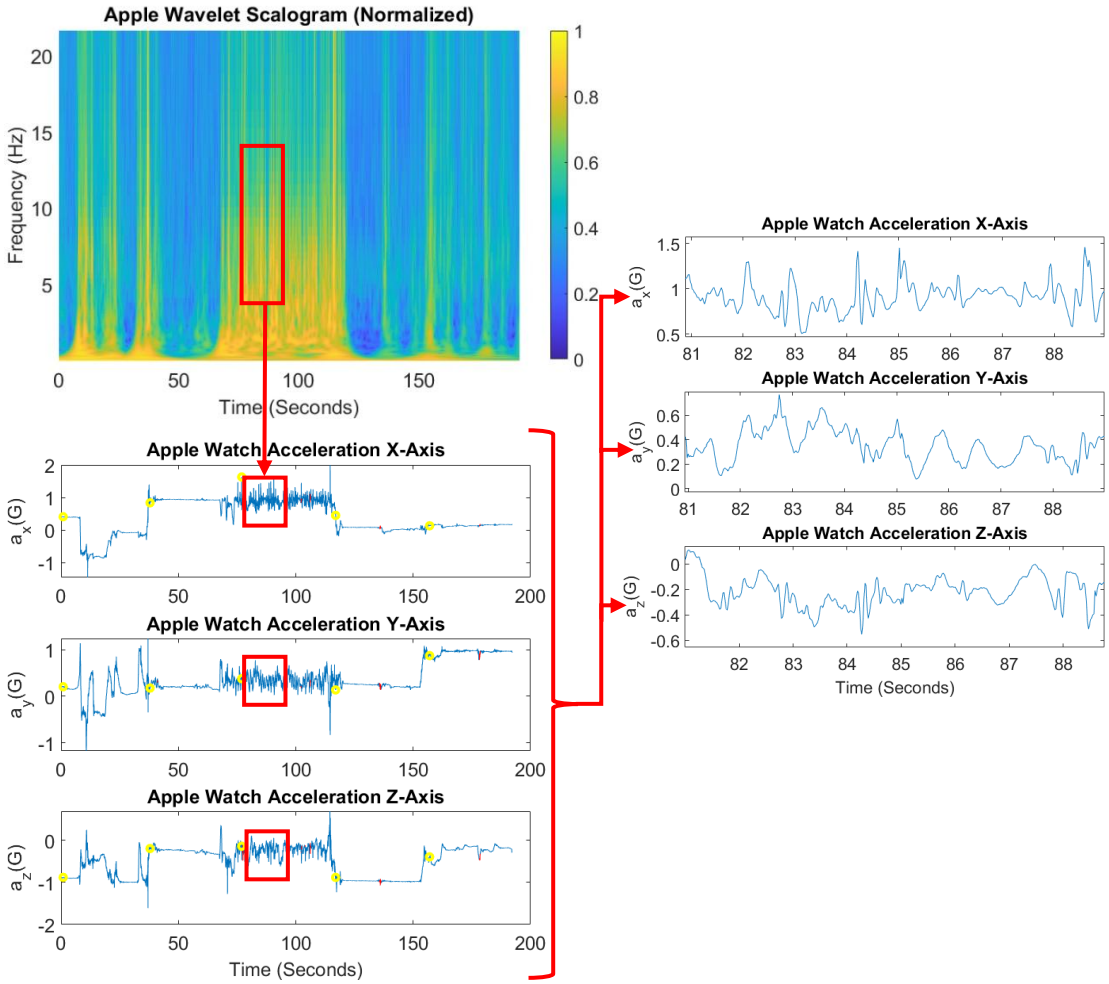

Figure 56. Walking With No Tremor (P2, 40% DBS)

Taking a time slice from the CWT scalogram at 85 seconds when P2 is walking, Figure 57, we see that walking appears as elevated power levels across all frequencies in the frequency spectrum but no peaks in the 4-7Hz band. Something else to note is that between the 0-3Hz band we also see much more power present when compared to the baselines established for sitting and standing without tremor. This large power concentration we observe in the 0-3Hz range is again a key marker used to determine whether a participant is currently in an active physical state [41].

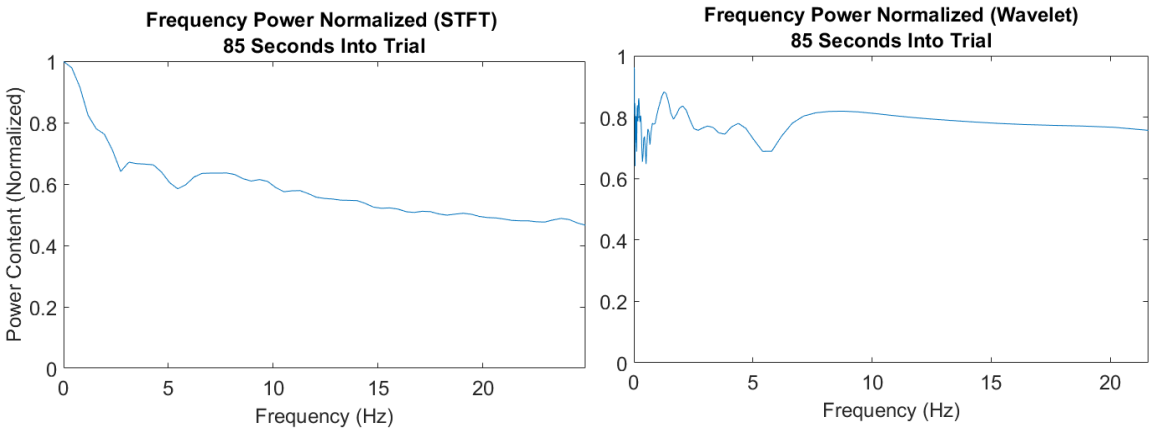

Figure 57. Walking With No Tremor, Frequency Spectrum at 85 Seconds (P2, 40% DBS)

After establishing the frequency and time characteristics of the Apple Watch accelerometry while walking, the detection algorithm could be tested on a set of data where tremor is present while walking. Figure 58 presents the performance of the detection algorithm using accelerometry from P3 while their DBS settings are set to 60% and focuses on a portion while P3 was walking. One key result to note is that half of the walking period in the trial has no tremor in the first half and tremor in the second half. If we focus on the time plot between 90-94 seconds, we notice that the detection algorithm plotted the accelerometry in blue, no tremor, and very closely resembles the baseline previously established by P2. Alternatively, if we observe the results between 94-98 seconds on the time plot, we see that the accelerometry was plotted in red, tremor. If we visually inspect the CWT scalogram during this period, we see that there appears to be peaks in both the 4-7Hz band and its harmonic between 8-14Hz.

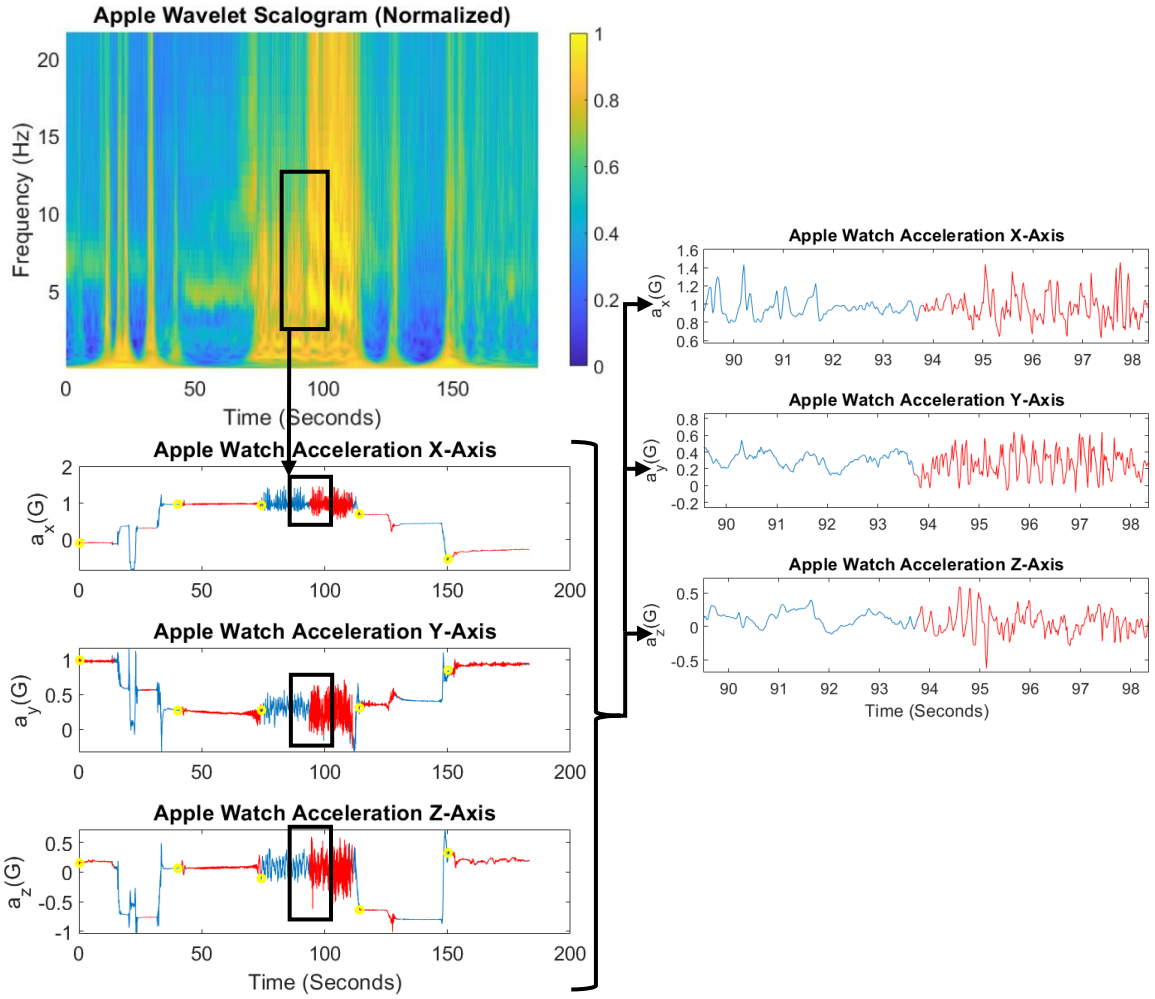

Figure 58. Walking With and Without Tremor (P3, 60% DBS)

Taking a close look at a time slice of the CWT spectrogram at 90 seconds, walking with no tremor, it resembles the baseline provided by P2. Figure 59 shows an elevated power concentration across all frequency bands which again is a characteristic of the participant in an activity state without tremor.

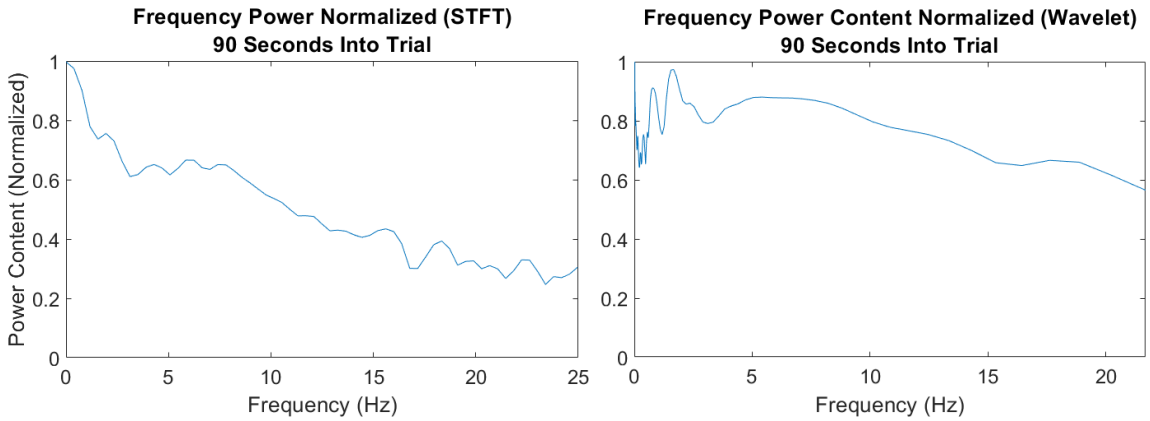

Figure 59. Walking With No Tremor, Frequency Spectrum at 90 Seconds (P3, 60% DBS)

Contrary to Figure 59, if we inspect Figure 60 which shows both frequency spectrums using STFT and CWT, we see that both present peaks in the 4-7Hz frequency band and its harmonics. However, again we see the CWT provides frequency spectrum peaks with a greater height, width, and prominence. Due to this characteristic, using a CWT proved to be more effective for peak detection during physical activity states as well.

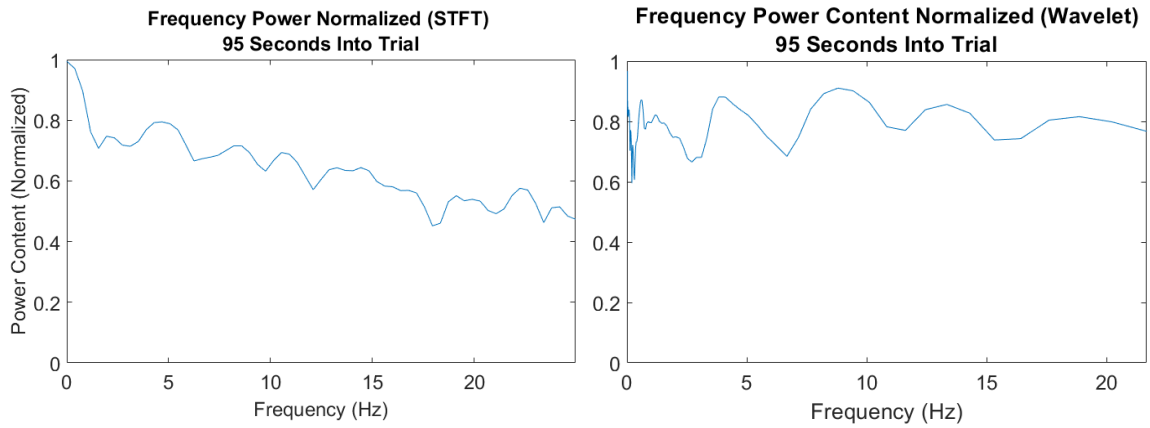

Figure 60. Walking With Tremor, Frequency Spectrum at 95 Seconds (P3, 60% DBS)

Running the detection algorithm on the Apple Watch accelerometry on another set of data from a participant who also experienced tremor while walking provides the results shown in Figure 61. Observing the CWT scalogram we see that there appears to be a concentration of power within the 4-7Hz band and its associated harmonic frequencies in

the portion boxed in black. The detection algorithm plotted periods with tremor in red and periods with no tremor in blue. Tremor was verified to be present using the trial logs provided. Also, if we zoom into a 7 second period where tremor was detected while walking, outline in black on Figure 61, we again see there appears to be 4-7Hz oscillations in the accelerometry data and it looks similar to the previous results given for P3.

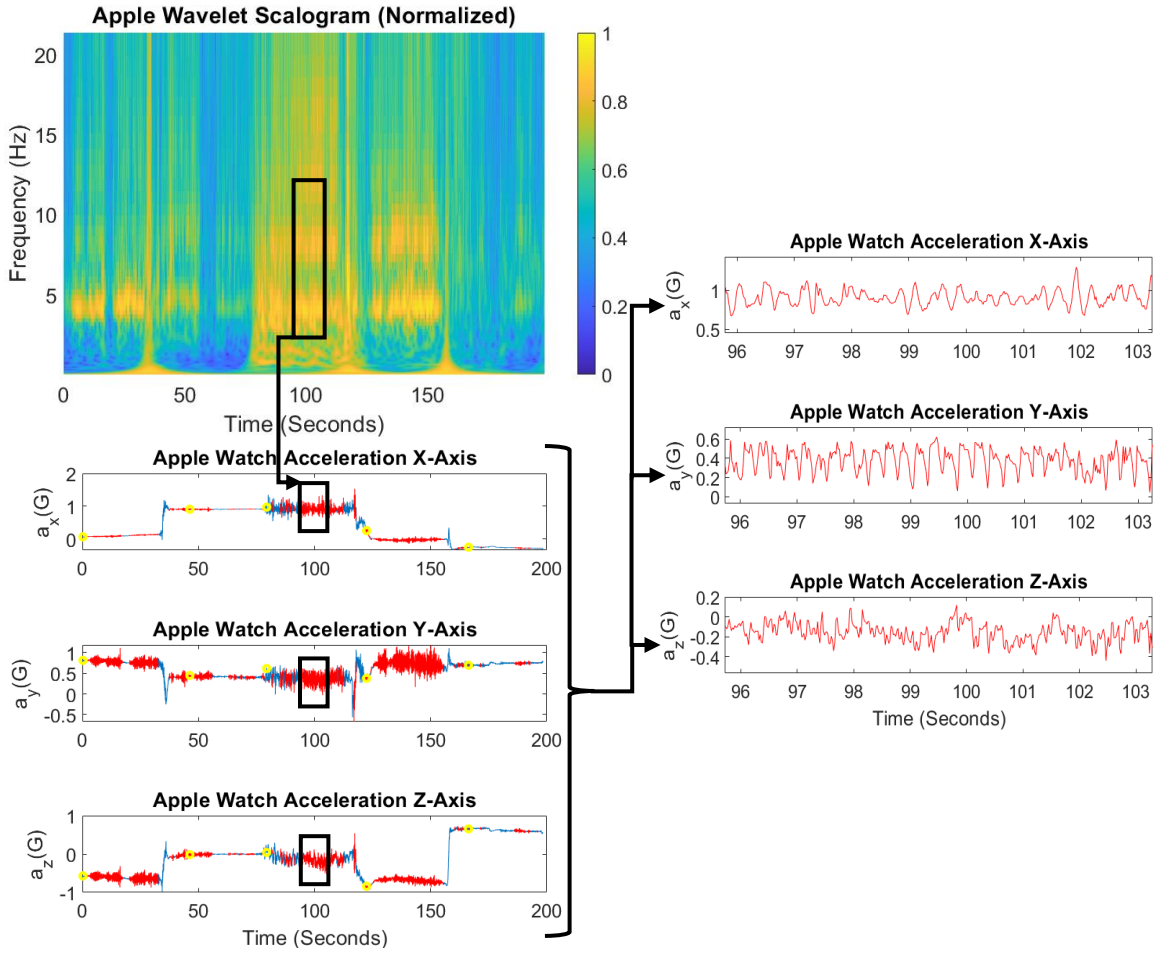

Figure 61. Walking With Tremor (P5, 60% DBS)

Making a closer observation of the STFT and CWT at 100 seconds into the trial, Figure 62, we see peaks in the 4-7Hz range and also in the harmonics while there is tremor when walking. But we also notice that the peaks that are present are better defined when using the CWT.

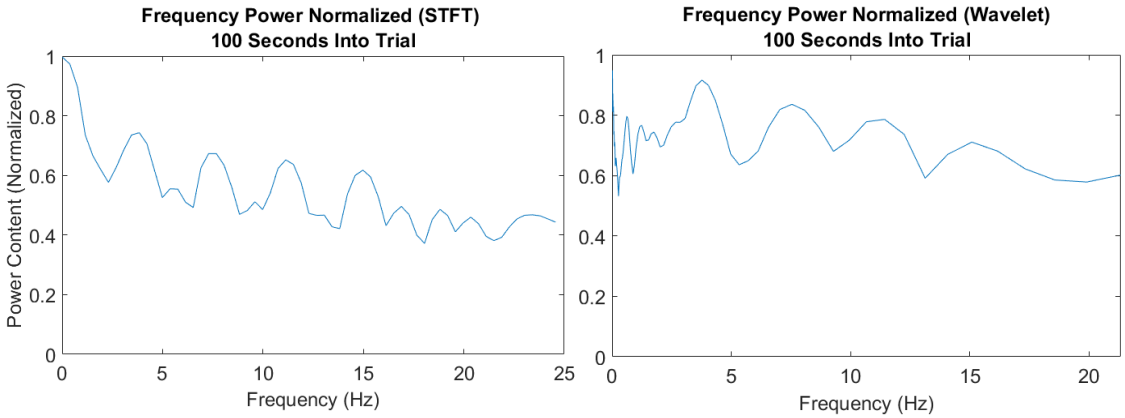

Figure 62. Walking With Tremor, Frequency Spectrum at 100 Seconds (P5, 60% DBS)

4.5 Tremor During Fine Motor Movement

To demonstrate the typical frequency and time accelerometry characteristics we will show a baseline recording of P1 while they are texting and their DBS settings are at 60%. Figure 63 shows the detection results and CWT scalogram for P1. A time period which effectively characterizes the accelerometry in the time and frequency domain during fine motor movement is shown in Figure 63 and boxed in red. If we zoom into the boxed portion between 162-169 seconds into the trial, we see the accelerometry has oscillatory movement, but it does not appear to oscillate at 4-7 Hz as we see with tremor. It should also be noted that the detection algorithm plotted the texting period as blue, no tremor. The trial logs also agree with these results considering no tremor was noted throughout the trial.

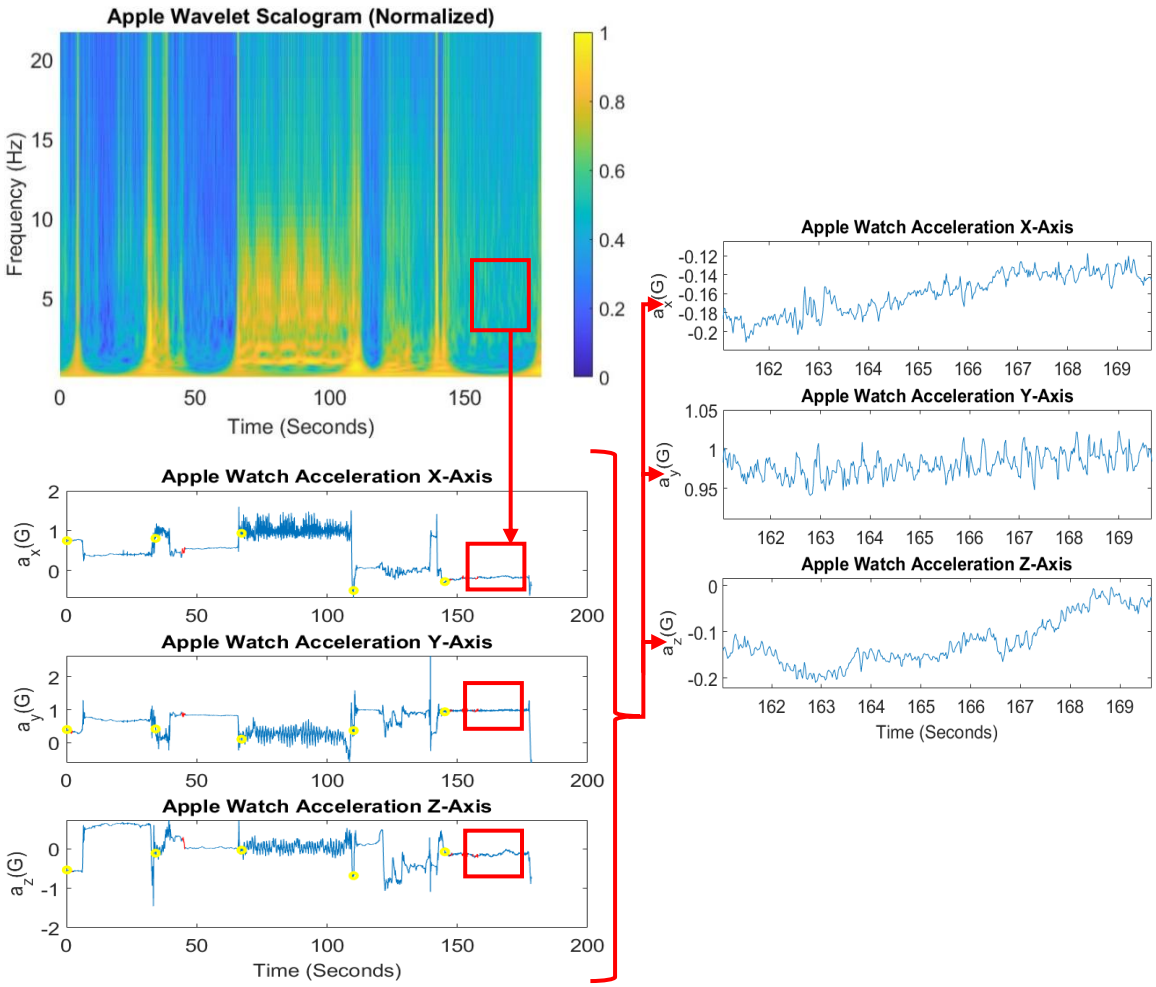

Figure 63. Texting With No Tremor (P1, 60% DBS)

The frequency spectrum shown in Figure 64 shows that the normalized power in the frequency spectrum sits around and slightly above 0.4 during periods of fine motor movement. This is a bit higher than the level we saw when the participants were sitting or standing but still uniform. However, the power level exhibited while P1 was texting is still lower than what was observed when they were walking. In general, this uniform distribution in the frequency spectrum suggests that fine motor movements such as texting come across as just random noise in the accelerometry since there are no distinct peaks present.

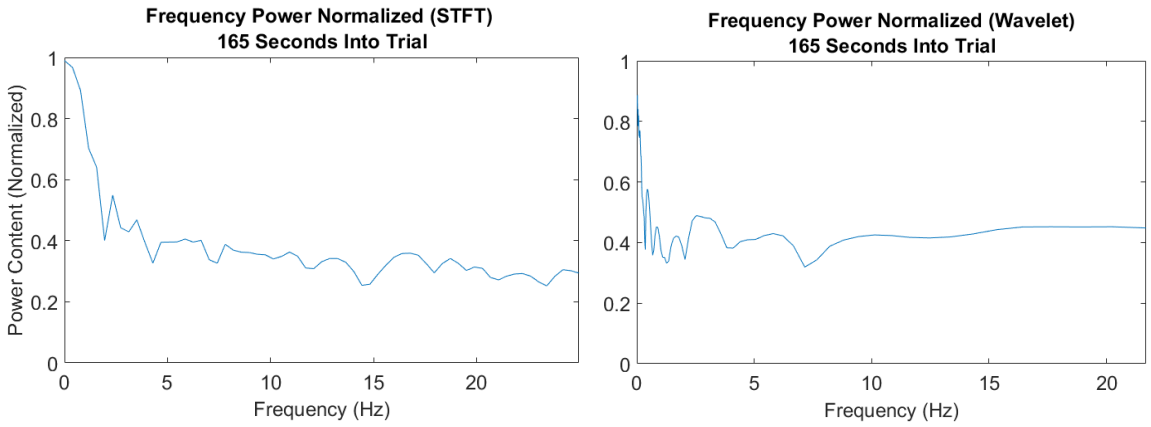

Figure 64. Texting No Tremor, Frequency Spectrum at 165 Seconds (P1, 60% DBS)

Another period of fine motor movement is demonstrated in Figure 65 and is boxed in red. Here we once again see that while P2 is texting the Apple Watch accelerometry manifests itself in the time plots as a noisy signal and in the CWT spectrogram we don't see any significant concentrations of power in the 4-7Hz frequency band. It should also be noted that detection algorithm did not detect periods of tremor during texting and agrees with the trial logs provided.

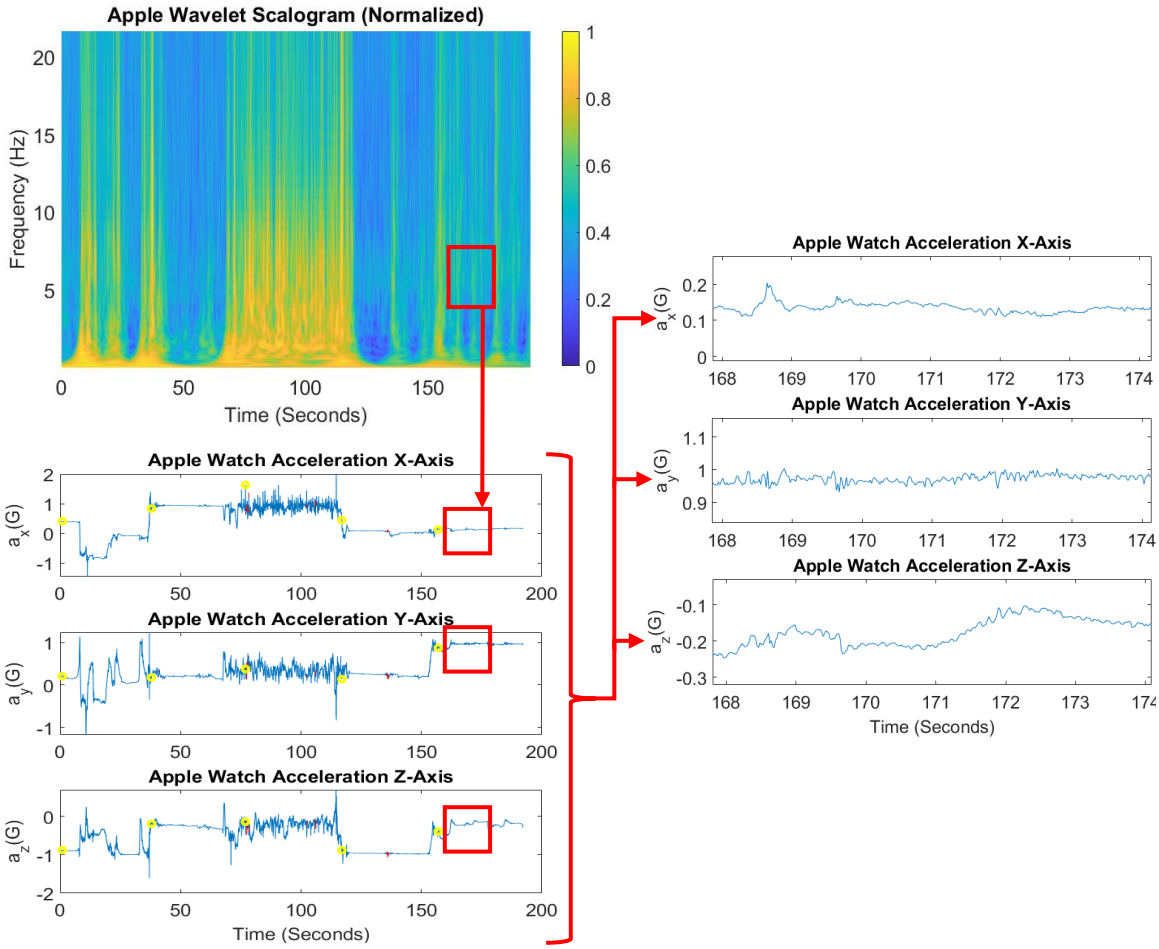

Figure 65. Texting With No Tremor (P2, 40% DBS)

During the 168-174 second period of the trial we can analyze the frequency spectrum at 171 seconds taken from both the signals STFT and CWT and find that again there appears to be a near constant power level present across the entire frequency spectrum, Figure 66. This leads us to believe that as long as fine motor movement is random like texting we can expect not to have peaks present that could lead to missed or inaccurate tremor detection.

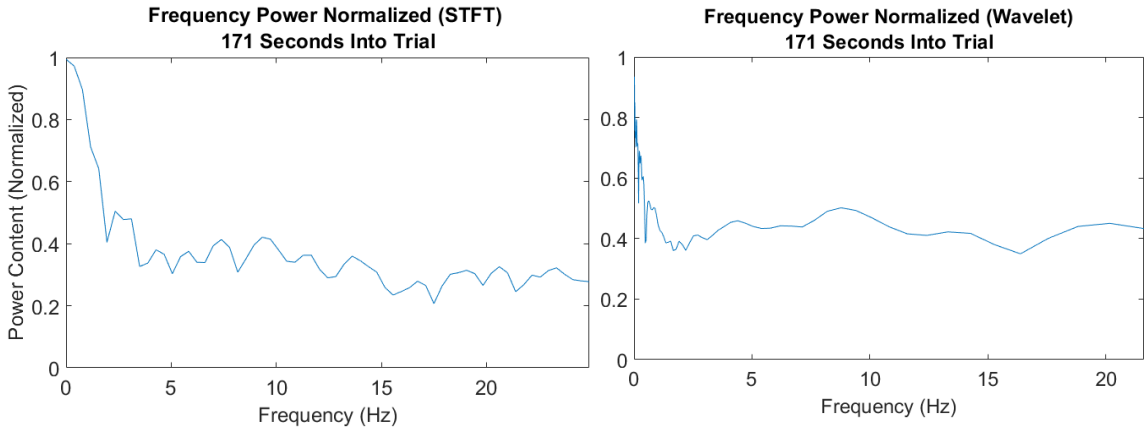

Figure 66. Texting No Tremor, Frequency Spectrum at 171 Seconds (P2, 40% DBS)

By passing another set of the Apple Watch accelerometry through our detection algorithm for a trial in which we know P3 experienced tremor during the fine motor movement we can generate the results shown on Figure 67. Instances without tremor are plotted in blue and instances with tremor are in red. Focusing on the results boxed in black on Figure 67 we see that there are regular oscillations in the 4-7Hz range unlike the baseline previously established using P1 and P2 accelerometry. It should also be noted that the algorithm detected tremor, as indicated by the red plotting in the accelerometry time plot, while the 4-7 Hz oscillations were present. The trial logs also agree with these results and indicate there was slight tremor present.

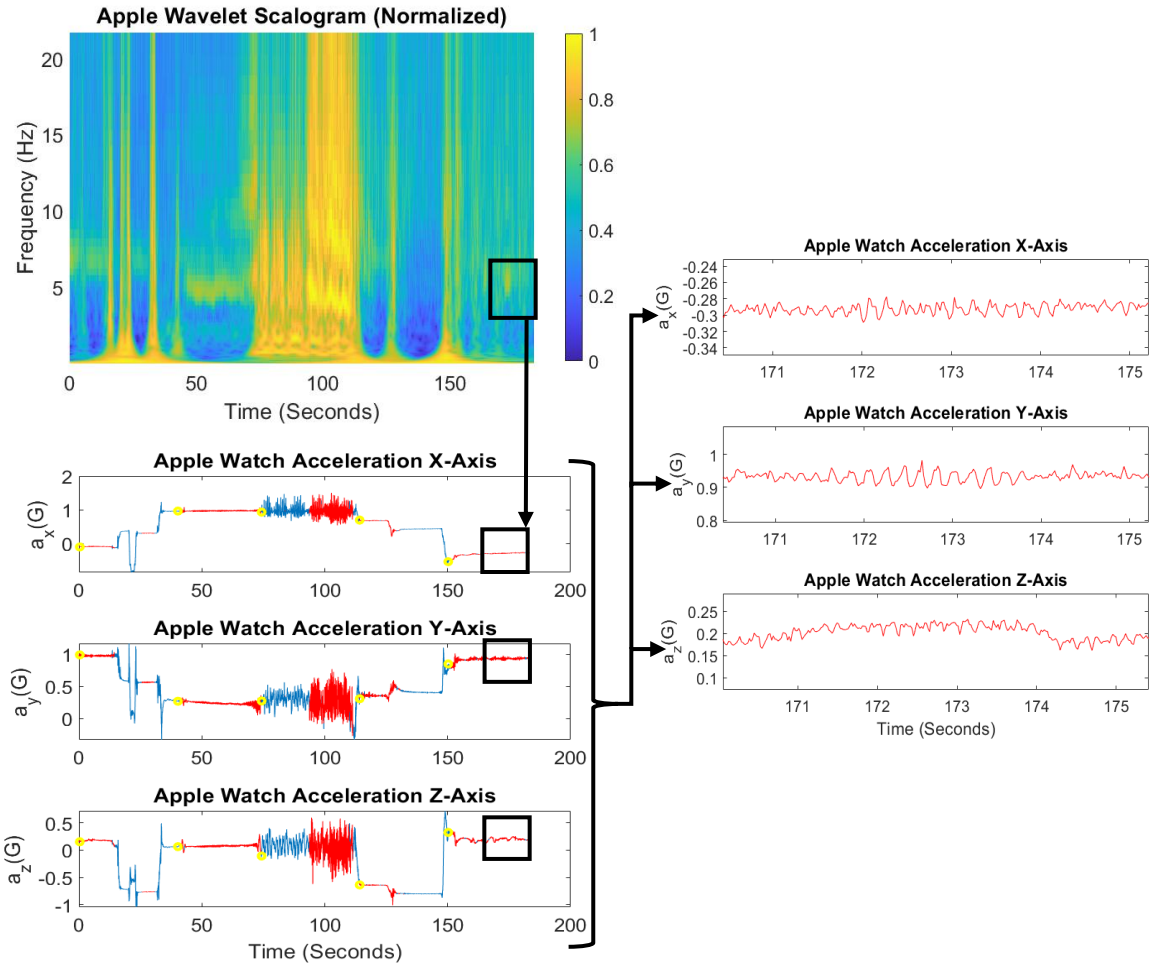

Figure 67. Texting With Tremor (P3, 60% DBS)

Taking a closer look at the frequency spectrum at 173 seconds we see that in fact there is a peak present in the 4-7Hz range in both the CWT and STFT, Figure 68. However, we again see how using a CWT produces a wider and more prominent peak in the 4-7Hz range. The increased peak definition when using CWT facilitates and improves tremor detection when compared to using an STFT.

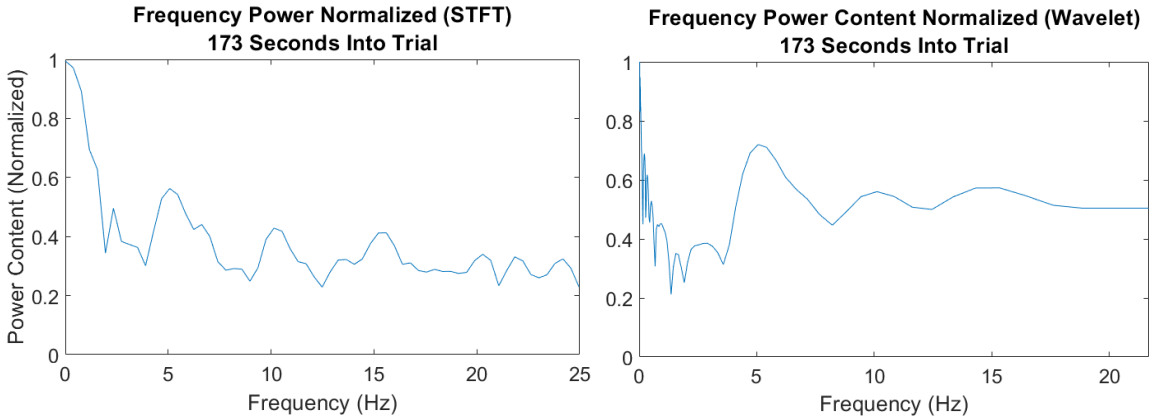

Figure 68. Texting With Tremor, Frequency Spectrum at 173 Seconds (P3, 60% DBS)

Testing the detection algorithm on another set of accelerometry data from P3 and with further decreased DBS settings of 40% we get the results presented in Figure 69. In this trial we also notice that the tremor is more prominent than when DBS was set to 60% in the previous example. Periods in Figure 69 which are plotted in red correspond to tremor detection and data plotted in blue is no tremor detected. Making a closer observation of the accelerometry time plot, the period boxed in black in Figure 69 shows that 4-7 Hz oscillations are indeed present in the accelerometry data. These oscillations also appear to have a larger amplitude than the accelerometry when P3 had their DBS set to 60%. We also know from the clinical logs that tremor was in fact present during texting for this trial.

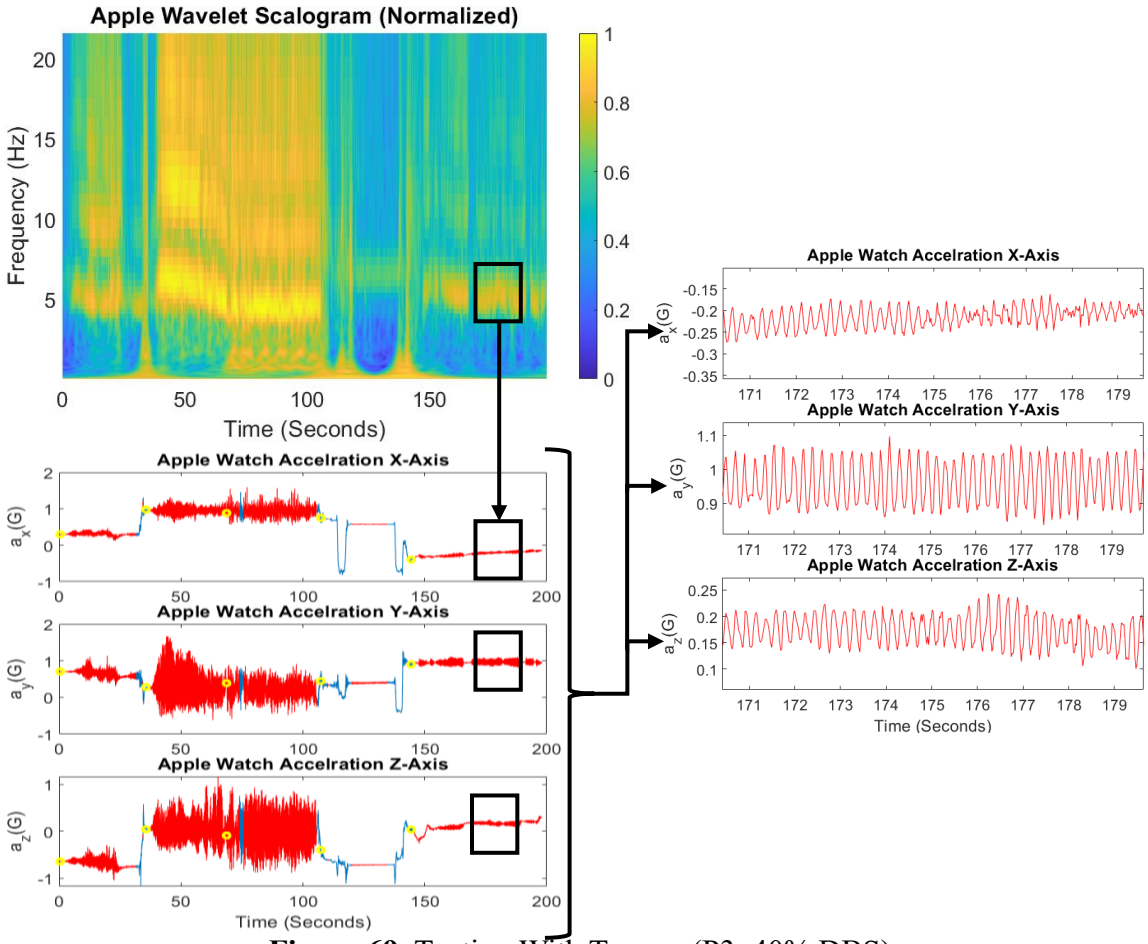

Figure 69. Texting With Tremor (P3, 40% DBS)

Making an observation of the frequency spectrum at 175 seconds, Figure 70, we find there are peaks present in the 4-7Hz range for both the STFT and CWT at this time point. However, we do see that in the CWT we get a better defined peak in the 4-7Hz range which help us with peak detection in our detection algorithm.

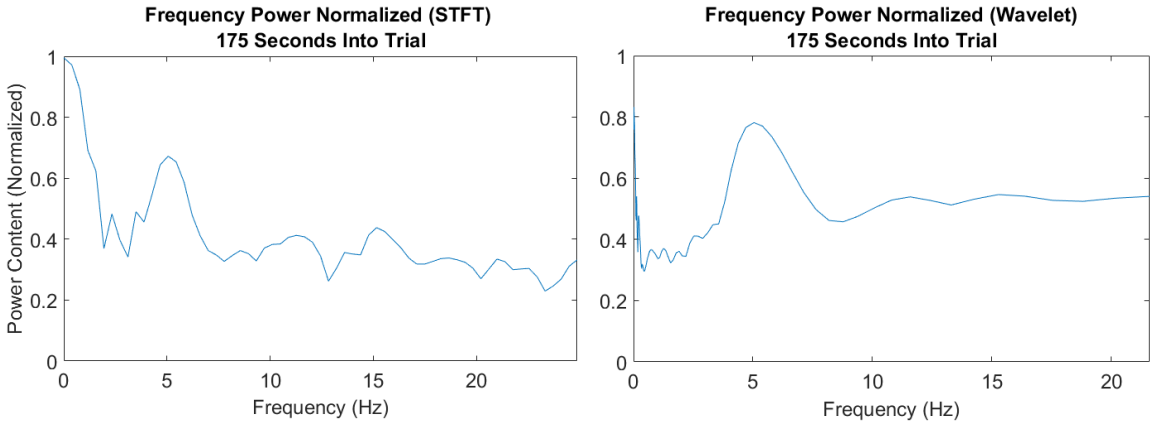

Figure 70. Texting With Tremor, Frequency Spectrum at 175 Seconds (P3, 40% DBS)

4.6 Posture Transitions

For posture transitions we look for high concentrations of power across all frequency bands which indicate the participant is either standing up or sitting down. In Figure 71 we see in the boxed areas what transitions should look like in the spectrogram and scalogram. We also know from the trial logs the boxed areas are where the participant transitioned from sitting to standing or from standing to sitting.

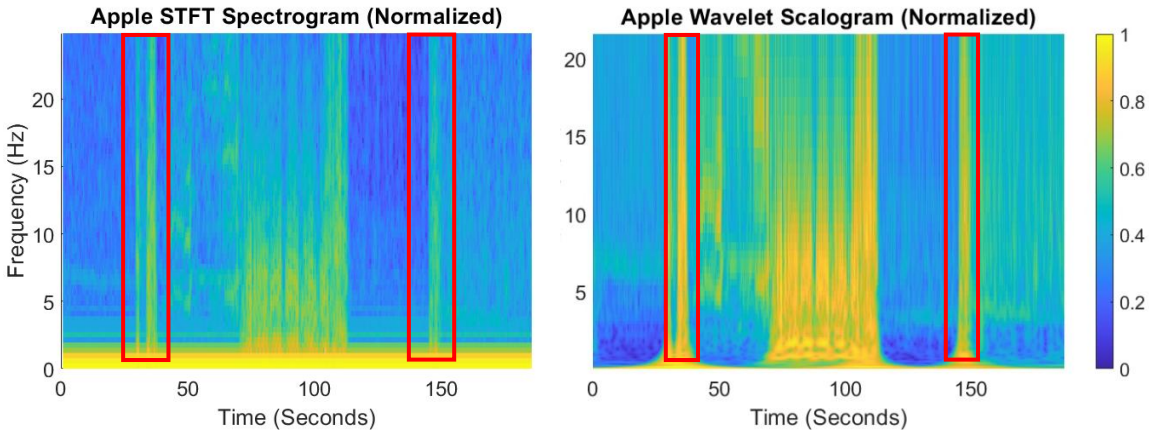

Figure 71. STFT and Wavelet Transform Posture Transition (P3, 60% DBS)

Making a closer observation of the frequency spectrum shown in Figure 72 we can see that there are elevated power levels across all frequency bands during transition periods. For transitions we can see there are no peaks in the 4-7Hz frequency band but by integrating and thresholding the power content across all frequency bands at each

individual time interval we can determine when the participant is transitioning from sitting to standing or standing to sitting.

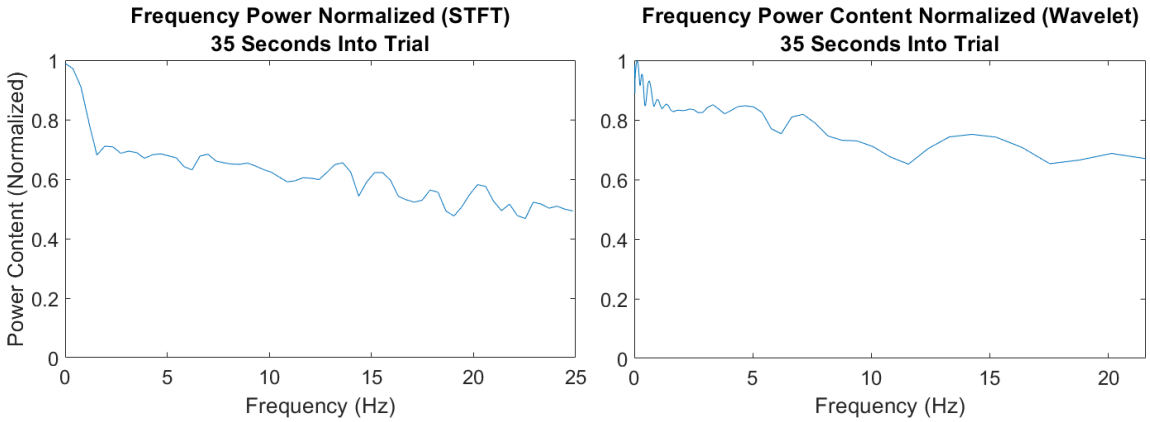

Figure 72. Posture Transition, Frequency Spectrum at 35 Seconds (P3, 60% DBS)

CHAPTER 5

Tremor Detection Algorithm Versus StrivePD

We can now compare our results to the tremor metric developed by RuneLabs and implemented through the StrivePD application. It should also be mentioned StrivePD is currently the industry standard for tremor detection and tremor severity ratings. To retrieve the tremor metric data from StrivePD to compare our results it must be downloaded using the RuneLabs API, the code used to download and access the tremor severity metric data could be found in Appendix B for reference.

While the StrivePD metric provides valuable insight into the participant's tremor severity, it does so over a 60 second period and as post processing measuring the percentage of time tremor was present within that window. Even though this is valuable information for physicians seeking to evaluate the participants tremor progression and severity while undergoing DBS it does not help with tremor detection in real time with a high temporal resolution so it could be implemented in a closed loop system. With a CWT we would be able to provide both, tremor detection in real time and with a much

finer temporal resolution when compared to the 60 second window provided by StrivePD.

If we want to effectively compare the accuracy of our tremor detection algorithm to the one available from StrivePD, a similar metric had to be created. To do this, the total time tremor was detected using our detection algorithm was calculated as a percentage over the same 60 second window as StrivePD. For comparing results, the detection algorithm's sensitivity was set to detect tremor as small as “slight” tremor, with a threshold of -40 dB at the peaks in the tremor frequency band.

Figure 73 shows one trial that was compared using the StrivePD metric in which there was no tremor present. In this trial P3 had their DBS settings at 100% of the prescribed intensity. Also, in Figure 73 we can see that the StrivePD tremor metric data is plotted in blue and the tremor that was detected using our detection algorithm is plotted in a dashed red line. As expected from the clinical logs there was no tremor present in this trial. This also helps verify that our detection algorithm thresholds are correctly set and will not detect small erroneous peaks as tremor.

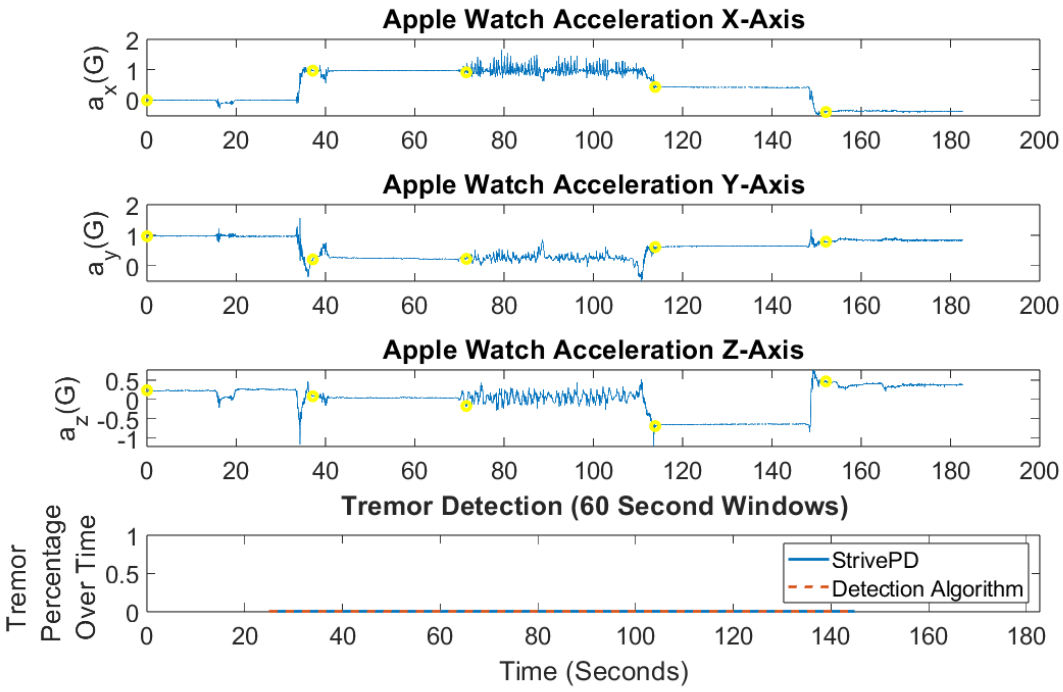

Figure 73. Tremor Detection Comparison (P3, 100% DBS)

By comparing the results for another trial in which there was a slight tremor present as indicated by trial logs we get the results shown in Figure 74 and the exact percentages along with the percent difference are presented in Table 3. The “Interval Number” column in Table 3 indicates which 60 second window is being compared moving from left to right.

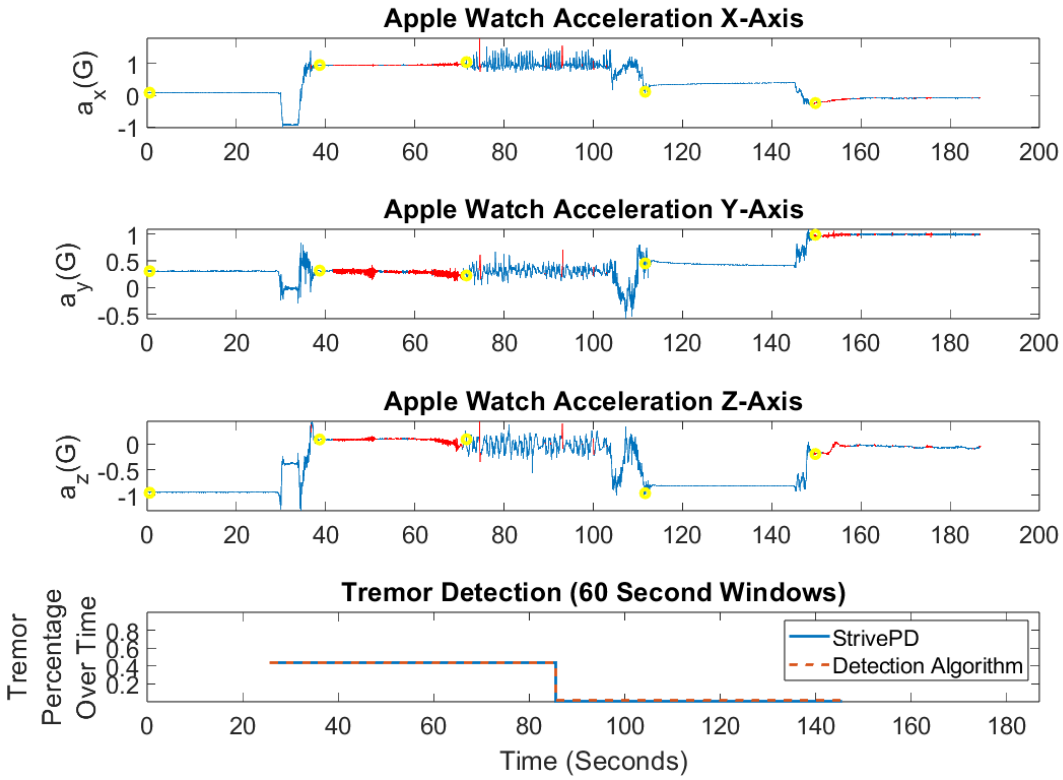

Figure 74. Tremor Detection Comparison, (P3, 60% DBS)

Percent difference for Interval 2 on Table 3 was noted as N/A due to the typical equation used to calculate percentage difference is undefined when the theoretical percentage is 0%:

Table 3. Tremor Metric Comparison (P3, 60% DBS)

Interval Number	StrivePD Tremor Percentage	Detection Algorithm Tremor Percentage	Percent Difference
Interval 1	43.3%	43.4%	0.23%
Interval 2	0%	1.2%	N/A

In this trial we see that the thresholds for detecting tremor worked well and gave us only 0.23% difference from the StrivePD percentage of tremor over the same 60 second time interval. While percent difference could not be calculated for Interval 2 due to the previously mentioned issue, we can note that there was only a 1.2% absolute difference from the StrivePD value of 0%. This proves that our developed algorithm

provides a high degree of accuracy when slight tremor is present, comparable to the industry standard given by StrivePD.

Figure 75 shows the same tremor detection for this trial, but now includes tremor detection at a higher temporal resolution, which was used to compare our results over 60 second intervals. The main advantage to using our detection algorithm is the possibility of using it to detect tremor in real time and apply DBS to individuals with Parkinsons as it is needed instead of always having DBS on.

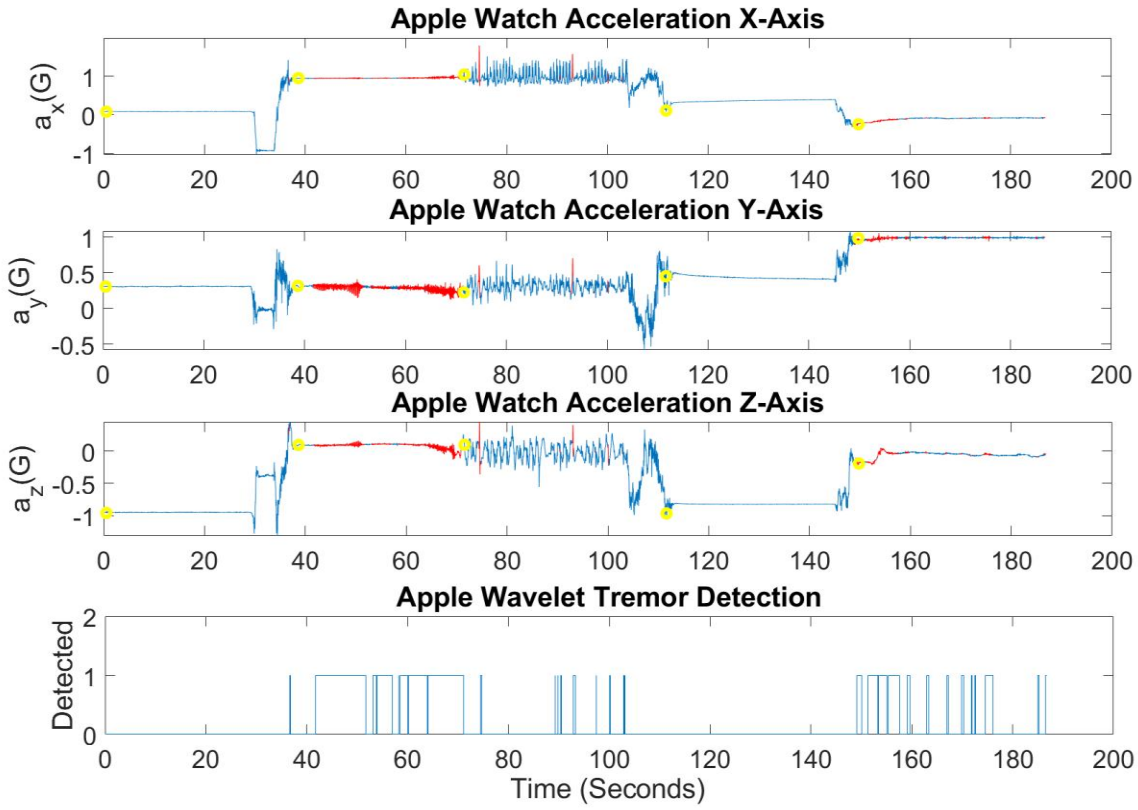

Figure 75. Tremor Detection With Higher Temporal Resolution (P3, 60% DBS)

Comparing the results of another trial in which P3 had DBS settings at 60% of their prescribed levels we get the results shown in Figure 76. Again, we see in the first 60 second interval provided by StrivePD our detection algorithm provides nearly the same tremor percentage and has a percent difference of 2.9% as shown in Table 4. However,

we do see during the second interval the percent difference jumps to 220.8% when we compare our tremor percentage within the second time interval versus the StrivePD tremor percentage.

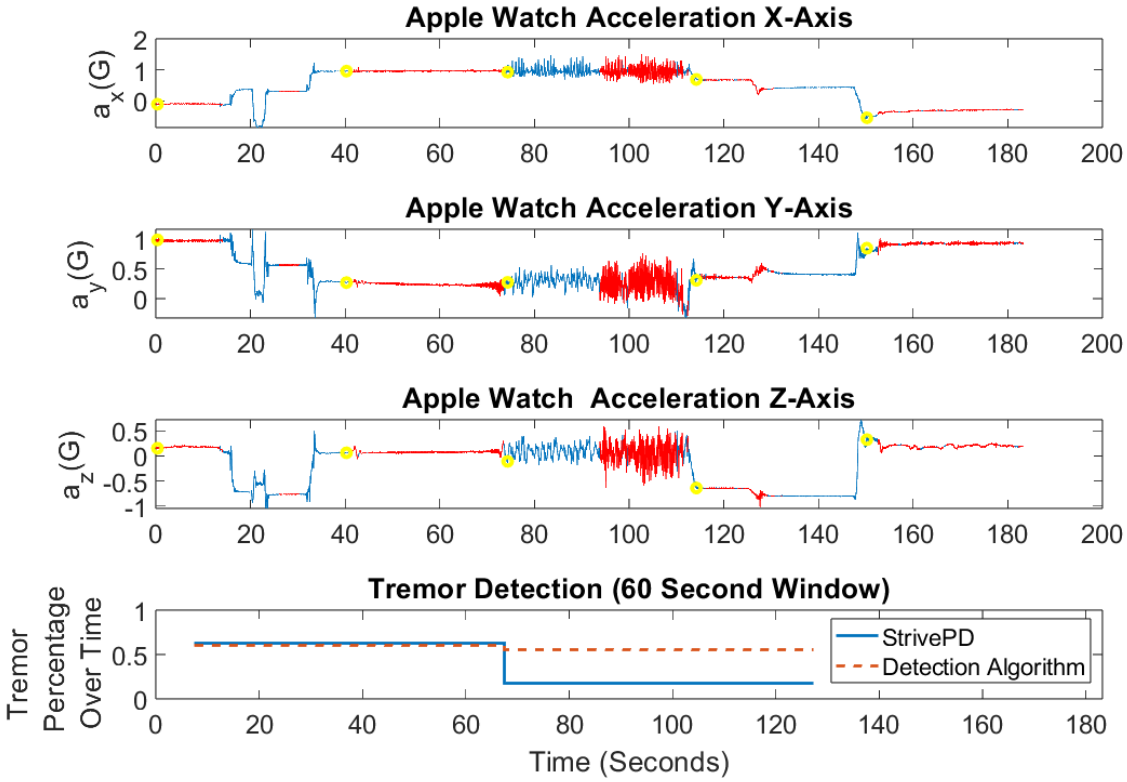

Figure 76. Improved Tremor Detection During High Activity (P3, 60% DBS)

Table 4. Improved Tremor Detection During High Activity (P3, 60% DBS)

Interval Number	StrivePD Tremor Percentage	Detection Algorithm Tremor Percentage	Percent Difference
Interval 1	62.5%	60.7%	2.9%
Interval 2	17.3%	55.5%	220.8%

While this may appear high, if we make a closer observation on Figure 76 we can find the section which is causing our detection to disagree with the tremor metric data provided by StrivePD. As it turns out StrivePD classifies periods with high activity as "unknown" and does not detect tremor within these periods. In short, activities such as walking, posture changes, or any other periods where there is a significant acceleration in

the data cause StivePD to disregarded the data no matter if tremor is present or not. This issue with StrivePD, the industry standard for Parkinson's tremor detection, is one of its major shortcomings. It would also appear that in Figure 76 at around the 94 second mark the general shape of the accelerometry changes during walking from slow rhythmic oscillatory movement resembling a walking gait to 4-7Hz oscillatory acceleration when tremor became present. After further discussion with the lead researcher and consulting with the trial logs it was concluded that in fact our developed detection algorithm correctly classified tremor while P3 was walking and provides valuable detection functionality to existing methods.

To further verify if walking with tremor was classified as "unknow" by StrivePD and ultimately removed from the total tremor percentage we can remove the detected tremor while the participant was walking and compare the results. Figure 77 shows our results with the detected tremor while walking removed. Table 5 shows the percent difference of our tremor detection results versus the StrivePD tremor severity metric data.

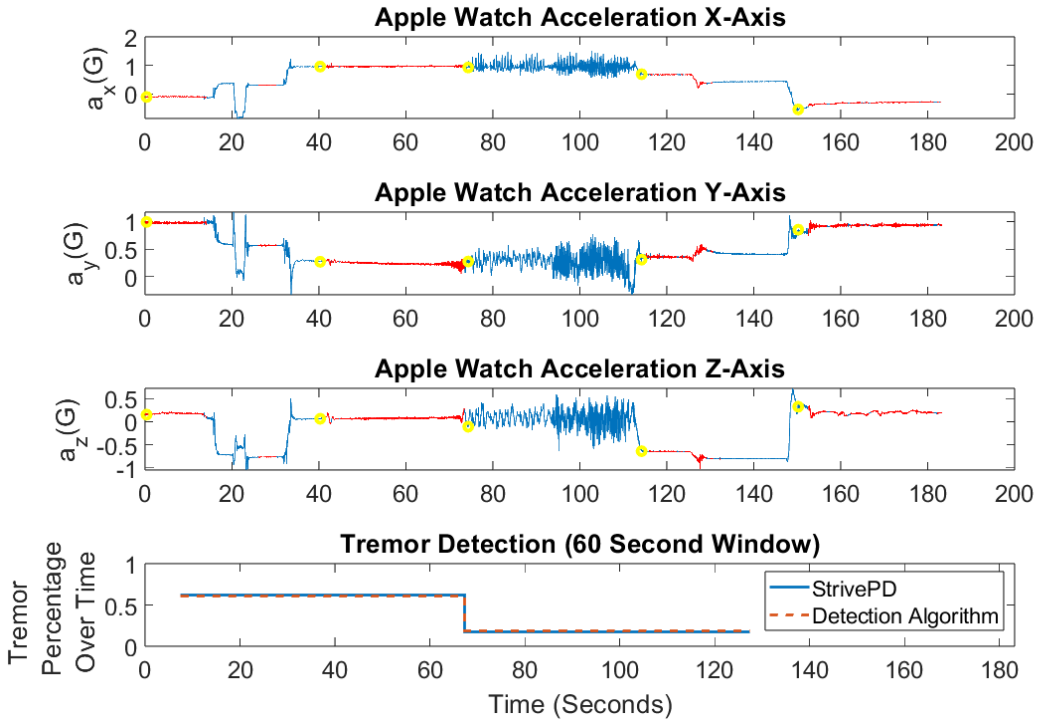

Figure 77. Removing High Activity Tremor Detection (P3, 60% DBS)

Table 5. Tremor Detection Comparison, No High Activity Tremor (P3, 60% DBS)

Interval Number	StrivePD Tremor Percentage	Detection Algorithm Tremor Percentage	Percent Difference
Interval 1	62.5%	60.7%	2.9%
Interval 2	17.3%	18.9%	9.2%

We can see by removing the tremor detection while walking, Interval 2 resembles the StrivePD tremor metric data much closer and has a 9.2% difference. While 9.2% difference may seem high if we compare the time tremor was present, StrivePD indicates there was 10.38 seconds of tremor present in Interval 2 and our detection algorithm detected 11.34 seconds of tremor leading to a difference of only 0.96 seconds over a 60 second period. But with these results we can confirm that StrivePD does in fact disregard periods with a high level of activity and does not account for tremor during high activity periods. Figure 78 shows tremor detection at a higher temporal resolution for this trial.

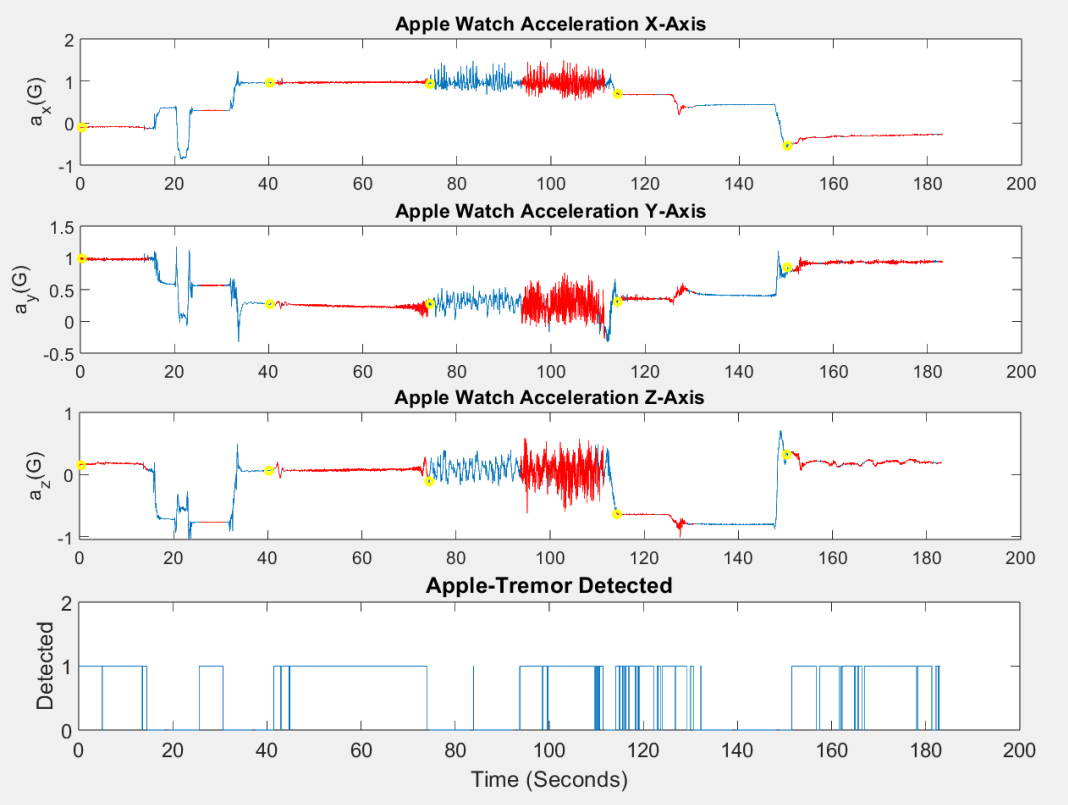

Figure 78. Tremor Detection With Higher Temporal Resolution (P3, 60% DBS)

Comparing the detection results for another trial where there was a high degree of tremor present throughout the trial, we get the results shown in Figure 79 and the tremor percentage metric on Table 6.

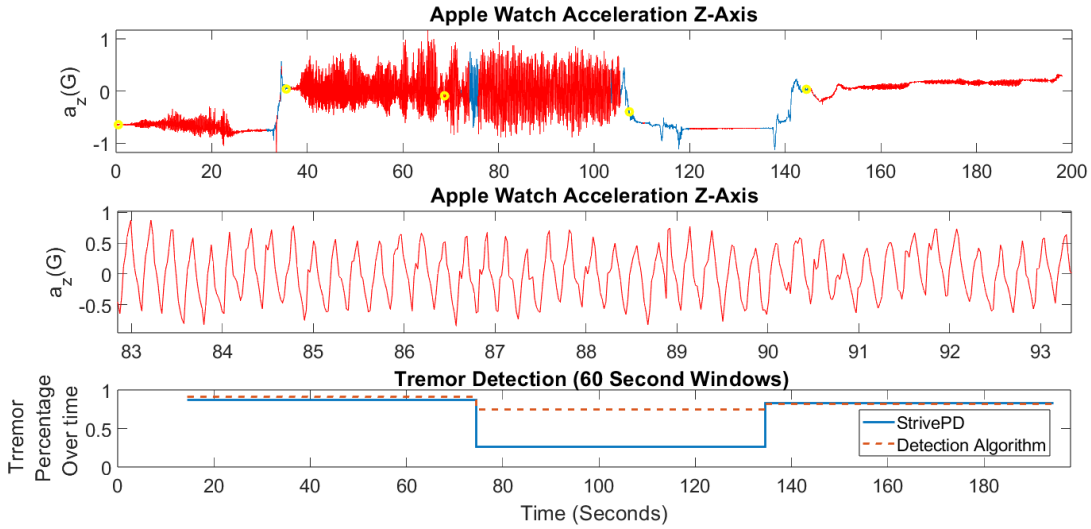

Figure 79. Improved Tremor Detection During High Activity (P3, 40% DBS)

Table 6. Improved Tremor Detection During High Activity (P3, 40% DBS)

Interval Number	StrivePD Tremor Percentage	Detection Algorithm Tremor Percentage	Percent Difference
Interval 1	87.5%	90.9%	3.8%
Interval 2	26.0%	74.6%	186.9%
Interval 3	82.6%	82.3%	0.3%

We once again notice that the tremor percentage detected by our algorithm is very high when compared to the StrivePD value during Interval 2. Interval 2 is also the same interval in which P3 was walking. So, it could be inferred that StrivePD actually marked this entire period as “unknown” since it is a high activity state. Making a closer observation on Figure 79 over a 10 second period between 83-93 seconds into the trial we see that along the accelerometry z-axis the accelerometry does not look like the typical accelerometry that would be expected as established by our baseline shown in Figure 56. We see 4-7Hz oscillatory acceleration, which is an indicator of tremor. Furthermore, after consulting with the clinical logs and the lead investigator of the project we confirmed that in fact there was tremor present during walking.

To verify whether or not this period with high activity was classified as “unknown” by StrivePD we can remove it from our tremor percentage calculation and we get the results shown in Figure 80 and Table 7.

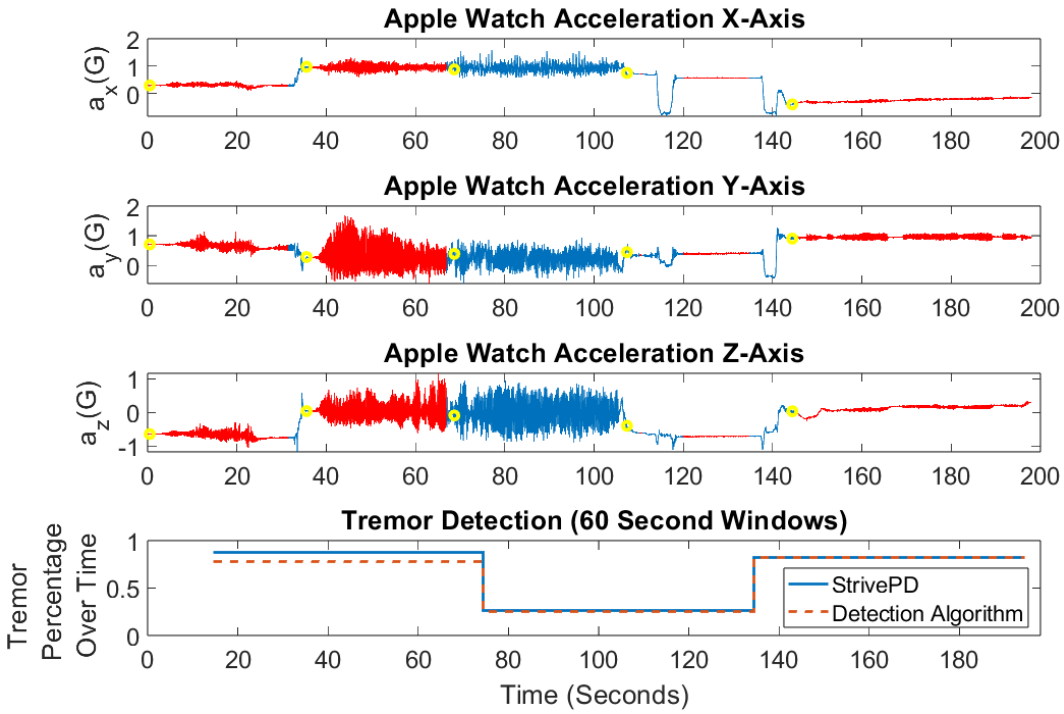

Figure 80. Removing High Activity Tremor Detection (P3, 40% DBS)

Table 7. Tremor Detection Comparison, No High Activity Tremor (P3, 40% DBS)

Interval Number	StrivePD Tremor Percentage	Detection Algorithm Tremor Percentage	Percent Difference
Interval 1	87.5%	78.4%	2.9%
Interval 2	26.0%	25.0%	3.8%
Interval 3	82.6%	82.3%	0.3%

Over all, if we look at the tremor percentage over time shown in Figure 80 we can see that all three 60 second intervals match up well to what was given by StrivePD. Checking the percent difference of our detection algorithm versus StrivePD we see on Table 7 that Interval 1 improves from 3.8% to 2.9%. Interval 2 sees the largest improvement going from a percent difference of 186.9% down to 3.8%. This once again demonstrated that the StrivePD detection method does in fact remove high activity states from its tremor percentage calculation. Interval 3 does not see any change in its percent difference and stays at 0.3%. We do believe this added functionality provides meaningful

improvement to the industry standard detection methods. Figure 81shows the developed algorithms tremor detection at a higher temporal resolution.

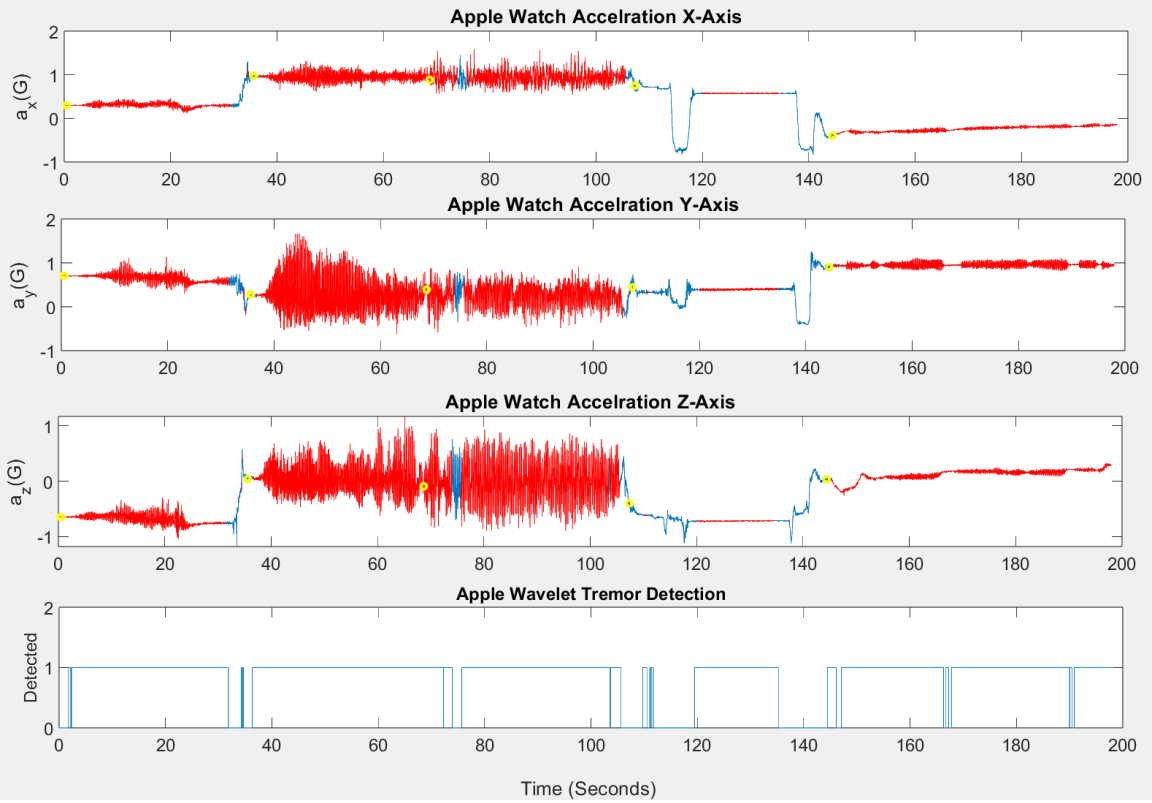

Figure 81. Tremor Detection With Higher Temporal Resolution (P3, 40% DBS)

CHAPTER 6

Summary of Results

After comparing our results against the tremor metric provided by StrivePD we can determine that with our developed detection algorithm we are able to detect tremor during high activity states such as walking, and maintaining accurate tremor detection during low activity states like sitting, standing, and fine motor movements. Also, considering our established thresholds are based on the physical displacement caused by tremor we can be more confident that these thresholds will not change much if at all in the future even if more participant were added to the study. Through our analysis of the accelerometry data we were also able to show that using Apple Watch accelerometry

enables us to detect tremor more effectively when compared to the IPG since tremor is usually more notable in the extremities. Furthermore, we also showed the added frequency localization capabilities offered by using a Wavelet Transform over using a Short Time Fourier Transform. Using a Continuous Wavelet Transform allowed us to more effectively distinguish peaks not only in the tremor frequency band but harmonics that are present when an individual is experiencing tremor while walking. Throughout the development of this tremor detection algorithm methods for how to retrieve, process, and sync the IPG and Apple Watch accelerometry were also established as it is relevant to this ongoing investigation and will continue to be used.

CHAPTER 7

Further Applications of Detection Algorithm

While the developed tremor detection algorithm has proven it provides improvements to existing methods originally developed by StrivePD, these improvements are planned to be used in combination with IPG accelerometry data. The Apple Watch accelerometry is able to provide us with valuable information when an individual is experiencing tremor and if they are in a high or low activity state but it is not able to provide insight of whether an individual is sitting, standing, or walking. For this reason, different methods have been explored for how IPG accelerometry data could be used to better detect postural states. Initial results and analysis have indicated that IPG accelerometry does in fact provide better postural state detection. With further development of the IPG algorithm we should be able to provide a complete and accurate detection of six distinguishable states: sitting with tremor, sitting without tremor, walking with tremor, walking without tremor, standing with tremor, and standing without tremor.

With development of the IPG postural state algorithm and integration with the already developed tremor detection algorithm we will be able to fully implement a closed loop system able to detect and manage tremor symptoms in real time.

REFERENCES

[1] Halpern, C., Hurtig, H., Jaggi, J., Grossman, M., Won, M., & Baltuch, G. (2007). Deep brain stimulation in neurologic disorders. *Parkinsonism & Related Disorders*, *13*(1), 1–16. https://doi.org/10.1016/j.parkreldis.2006.03.001

[2] Hariz, M., & Blomstedt, P. (2022). Deep brain stimulation for Parkinson's disease. *Journal of Internal Medicine*, *292*(5), 764–778. https://doi.org/10.1111/joim.13541

[3] Johnson, M. D., Zhang, J., Ghosh, D., McIntyre, C. C., & Vitek, J. L. (2012). Neural targets for relieving parkinsonian rigidity and bradykinesia with pallidal deep brain stimulation. *Journal of Neurophysiology*, *108*(2), 567–577. https://doi.org/10.1152/jn.00039.2012

[4] Obeso, J. A., Marin, C., Rodriguez-Oroz, C., Blesa, J., Benitez-Temiño, B., Mena-Segovia, J., Rodríguez, M., & Olanow, C. W. (2009). The basal ganglia in Parkinson's disease: Current concepts and unexplained observations. *Annals of Neurology*, *64*(S2), S30–S46. https://doi.org/10.1002/ana.21481

[5] Krack, P., Batir, A., Van Blercom, N., Chabardes, S., Fraix, V., Ardouin, C., Koudsie, A., Limousin, P. D., Benazzouz, A., LeBas, J. F., Benabid, A.-L., & Pollak, P. (2003). Five-year follow-up of bilateral stimulation of the subthalamic nucleus in advanced parkinson's disease. *New England Journal of Medicine*, *349*(20), 1925–1934. https://doi.org/10.1056/NEJMoa035275

[6] Sethi, K. (2008). Levodopa unresponsive symptoms in Parkinson disease: L -Dopa Unresponsive Symptoms in PD. *Movement Disorders*, *23*(S3), S521–S533. https://doi.org/10.1002/mds.22049

[7] Jenner, P., & Katzenschlager, R. (2016). Apomorphine—Pharmacological properties and clinical trials in Parkinson's disease. *Parkinsonism & Related Disorders*, *33*, S13–S21. https://doi.org/10.1016/j.parkreldis.2016.12.003

[8] Antonini, A., & Nitu, B. (2018). Apomorphine and levodopa infusion for motor fluctuations and dyskinesia in advanced Parkinson disease. *Journal of Neural Transmission*, *125*, 1131-1135. https://doi.org/10.1007/s00702-018-1906-0

[9] Poewe, W., & Wenning, G. K. (2000). Apomorphine: An underutilized therapy for Parkinson's disease. *Movement Disorders*, *15*(5), 789–794. https://doi.org/10.1002/1531-8257(200009)15:5<789::AID-MDS1005>3.0.CO;2-H

[10] Antonini, A., Martinez-Martin, P., Chaudhuri, R. K., Merello, M., Hauser, R., Katzenschlager, R., Odin, P., Stacy, M., Stocchi, F., Poewe, W., Rascol, O., Sampaio, C., Schrag, A., Stebbins, G. T., & Goetz, C. G. (2011). Wearing-off scales in Parkinson's disease: Critique and recommendations. *Movement Disorders*, *26*(12), 2169–2175. https://doi.org/10.1002/mds.23875

[11] Weiss, D., Volkmann, J., Fasano, A., Kühn, A., Krack, P., & Deuschl, G. (2021). Changing Gears – DBS For Dopaminergic Desensitization in Parkinson's Disease? *Annals of Neurology*, *90*(5), 699–710. https://doi.org/10.1002/ana.26164

[12] Sarica, C., Iorio-Morin, C., Aguirre-Padilla, D. H., Najjar, A., Paff, M., Fomenko, A., Yamamoto, K., Zemmar, A., Lipsman, N., Ibrahim, G. M., Hamani, C., Hodaie, M., Lozano, A. M., Munhoz, R. P., Fasano, A., & Kalia, S. K. (2021). Implantable pulse generators for deep brain stimulation: Challenges, complications, and strategies for practicality and longevity. *Frontiers in Human Neuroscience*, *15*, 708481. https://doi.org/10.3389/fnhum.2021.708481

[13] Dostrovsky, J. O., & Lozano, A. M. (2002). Mechanisms of deep brain stimulation. *Movement disorders: official journal of the Movement Disorder Society*, *17*(S3), S63-S68. https://doi.org/10.1002/mds.10143

[14] Herrington, T. M., Cheng, J. J., & Eskandar, E. N. (2016). Mechanisms of deep brain stimulation. *Journal of neurophysiology*, *115*(1), 19-38. https://doi.org/10.1152/jn.00281.2015

[15] Mishima, T., Fujioka, S., Morishita, T., Inoue, T., & Tsuboi, Y. (2021). Personalized medicine in Parkinson's disease: new options for advanced treatments. *Journal of personalized medicine*, *11*(7), 650. https://doi.org/10.3390/jpm11070650

[16] Savica, R., Stead, M., Mack, K. J., Lee, K. H., & Klassen, B. T. (2012, January). Deep brain stimulation in tourette syndrome: a description of 3 patients with excellent outcome. In *Mayo Clinic Proceedings* (Vol. 87, No. 1, pp. 59-62). Elsevier. https://doi.org/10.1016/j.mayocp.2011.08.005

[17] Kennedy, S. H., Giacobbe, P., Rizvi, S. J., Placenza, F. M., Nishikawa, Y., Mayberg, H. S., & Lozano, A. M. (2011). Deep brain stimulation for treatment-resistant depression: follow-up after 3 to 6 years. *American Journal of Psychiatry*, *168*(5), 502-510. https://doi.org/10.1176/appi.ajp.2010.10081187

[18] Schmidt, S. L., Chowdhury, A. H., Mitchell, K. T., Peters, J. J., Gao, Q., Lee, H. J., ... & Turner, D. A. (2024). At home adaptive dual target deep brain stimulation in Parkinson's disease with proportional control. *Brain*, *147*(3), 911-922. https://doi.org/10.1093/brain/awad429

[19] Medtronic. (2019). Research Development Kit 4NR013 (Rev B). Medtronic.

[20] Scheiner, A., Mortimer, J. T., & Roessmann, U. (1990). Imbalanced biphasic electrical stimulation: muscle tissue damage. *Annals of biomedical engineering*, *18*(4), 407-425.

[21] Turner, D., Perry, B., Peters, J. (2020). *An Integrated Biomarker Approach to Personalized, Adaptive Deep Brain Stimulation in Parkinson Disease* (Institutional Review Board application, Duke University Health System)

[22] Smaga, S. (2003). Tremor. *American family physician*, *68*(8), 1545-1552.

[23] Pasquini, J., Ceravolo, R., Qamhawi, Z., Lee, J. Y., Deuschl, G., Brooks, D. J., ... & Pavese, N. (2018). Progression of tremor in early stages of Parkinson's disease: a clinical and neuroimaging study. *Brain*, *141*(3), 811-821. https://doi.org/10.1093/brain/awx376

[24] Apple. (2024). Apple Platform Security (December 2024). Apple.

[25] *Unix time stamp—Epoch converter*. (n.d.). Retrieved March 19, 2025, from https://www.unixtimestamp.com/?ref=ad-tech-explained

[26] rune-labs (2020), examples, Github repository: https://github.com/rune-labs/runeq-python/tree/main/examples

[27] *Getting raw accelerometer events*. (n.d.). Apple Developer Documentation. Retrieved March 19, 2025, from https://docs.developer.apple.com/documentation/coremotion/getting-raw-accelerometer-events

[28] Rune Labs, Inc. (2024). Privacy Notice. Retrieved March 19, 2025, from https://runelabs.io/privacy-notice

[29] Su, D., Yang, S., Hu, W., Wang, D., Kou, W., Liu, Z., Wang, X., Wang, Y., Ma, H., Sui, Y., Zhou, J., Pan, H., & Feng, T. (2020). The characteristics of tremor motion help identify parkinson's disease and multiple system atrophy. *Frontiers in Neurology*, *11*, 540. https://doi.org/10.3389/fneur.2020.00540

[30] Deuschl, G., Raethjen, J., Baron, R., Lindemann, M., Wilms, H., & Krack, P. (2000). The pathophysiology of parkinsonian tremor: A review. *Journal of Neurology*, *247*(S5), V33–V48. https://doi.org/10.1007/PL00007781

[31] Zhou, Y., Jenkins, M. E., Naish, M. D., & Trejos, A. L. (2016). The measurement and analysis of Parkinsonian hand tremor. *2016 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*, 414–417. https://doi.org/10.1109/BHI.2016.7455922

[32] Taheri, B., Case, D., & Richer, E. (2014). Robust controller for tremor suppression at musculoskeletal level in human wrist. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *22*(2), 379–388. https://doi.org/10.1109/TNSRE.2013.2295034

[33] The MathWorks Inc. (2024). stft, Natick, Massachusetts: The MathWorks Inc. https://www.mathworks.com/help/signal/ref/stft.html

[34] Semmlow, J., & Griffel, B. (2014). *Bio Signal and Medical Image Processing* (Third Edition). CRC Press, 193-195

[35] Semmlow, J., & Griffel, B. (2014). *Bio Signal and Medical Image Processing* (Third Edition). CRC Press, 217-222

[36] The MathWorks Inc. (2024). cwt, Natick, Massachusetts: The MathWorks Inc. https://www.mathworks.com/help/wavelet/ref/cwt.html

[37] Akansu, A. N., Serdijn, W. A., & Selesnick, I. W. (2010). Emerging applications of wavelets: A review. *Physical Communication*, *3*(1), 1–18. https://doi.org/10.1016/j.phycom.2009.07.001

[38] Huang, G., Meng, J., Zhang, D., & Zhu, X. (2011). Window function for eeg power density estimation and its application in ssvep based bcis. In S. Jeschke, H. Liu, & D. Schilberg (Eds.), *Intelligent Robotics and Applications* (Vol. 7102, pp. 135–144). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-25489-5_14

[39] The MathWorks Inc. (2024). hamming, Natick, Massachusetts: The MathWorks Inc. https://www.mathworks.com/help/signal/ref/hamming.html

[40] The MathWorks Inc. (2024). Morse Wavelets, Natick, Massachusetts: The MathWorks Inc. https://www.mathworks.com/help/wavelet/ug/morse-wavelets.html

[41] Bouten, C. V. C., Koekkoek, K. T. M., Verduin, M., Kodde, R., & Janssen, J. D. (1997). A triaxial accelerometer and portable data processing unit for the assessment of daily physical activity. *IEEE Transactions on Biomedical Engineering*, *44*(3), 136–147. https://doi.org/10.1109/10.554760

APPENDIX A

The following Python script was written to retrieve the Apple Watch accelerometry data from the RuneLabs API.

Initializing libraries and access token

```
from runeq import initialize
from runeq.resources.stream_metadata import get_patient_stream_metadata
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from dateutil import parser
initialize(access_token_id='__________', access_token_secret='_____________')
```

Verify system login

```
from runeq.resources.user import get_current_user

my_user = get_current_user()
print(my_user)
print('Active Org:', my_user.active_org_name)
```

Extract Apple Watch data

```
# Extracting and storing apple data

# using begining and ending IPG time stamps from box.com data, divide by 1000
start_time = (_____________/1000)-100  # Use UNIX time from IPG trial start time
end_time = (_____________/1000)+100  # Use UNIX time from IPG trial end time

#Pull x axis
tremor_metadata_accel_x = get_patient_stream_metadata(
    patient_id='_______________',
    algorithm='ingest-strive-applewatch-motion.0',
    device_id='__________',
    measurement='user',
    axis='x',
)
tremor_accel_x = tremor_metadata_accel_x.get_stream_dataframe(
    start_time=start_time,
    end_time=end_time,
)
tremor_accel_x=tremor_accel_x['acceleration'].values
```

```
tremor_metadata_grav_x = get_patient_stream_metadata(
    patient_id='____________________',
    algorithm='ingest-strive-applewatch-motion.0',
    device_id='_____________',
    measurement='gravity',
    axis='x',
)
tremor_grav_x = tremor_metadata_grav_x.get_stream_dataframe(
    start_time=start_time,
    end_time=end_time,
)
tremor_grav_x=tremor_grav_x['acceleration'].values
apple_recording_x=tremor_grav_x+tremor_accel_x

#Pull y axis
tremor_metadata_accel_y = get_patient_stream_metadata(
    patient_id='____________________',
    algorithm='ingest-strive-applewatch-motion.0',
    device_id='___________',
    measurement='user',
    axis='y',
)
tremor_accel_y = tremor_metadata_accel_y.get_stream_dataframe(
    start_time=start_time,
    end_time=end_time,
)
tremor_accel_y=tremor_accel_y['acceleration'].values

tremor_metadata_grav_y = get_patient_stream_metadata(
    patient_id='____________________',
    algorithm='ingest-strive-applewatch-motion.0',
    device_id='_____________',
    measurement='gravity',
    axis='y',
)
tremor_grav_y = tremor_metadata_grav_y.get_stream_dataframe(
    start_time=start_time,
    end_time=end_time,
)
tremor_grav_y=tremor_grav_y['acceleration'].values
apple_recording_y=tremor_grav_y+tremor_accel_y
```

```
#Pull z axis
tremor_metadata_accel_z = get_patient_stream_metadata(
    patient_id='____________________',
    algorithm='ingest-strive-applewatch-motion.0',
    device_id='___________',
    measurement='user',
    axis='z',
)
tremor_accel_z = tremor_metadata_accel_z.get_stream_dataframe(
    start_time=start_time,
    end_time=end_time,
)
tremor_accel_z=tremor_accel_z['acceleration'].values

tremor_metadata_grav_z = get_patient_stream_metadata(
    patient_id='____________________',
    algorithm='ingest-strive-applewatch-motion.0',
    device_id='_____________',
    measurement='gravity',
    axis='z',
)
tremor_grav_z = tremor_metadata_grav_z.get_stream_dataframe(
    start_time=start_time,
    end_time=end_time,
)
tremor_grav_z=tremor_grav_z['acceleration'].values
apple_recording_z=tremor_grav_z+tremor_accel_z

# Get unix times
tremor_accel_time = tremor_metadata_accel_x.get_stream_dataframe(
    start_time=start_time,
    end_time=end_time,
)
tremor_accel_time=tremor_accel_time['time'].values

unix_times = np.array([parser.isoparse(date).timestamp() for date in tremor_accel_time])
```

Plot extracted data to verify

```
time=np.arange(0,len(tremor_accel_x))
time=time*0.02

plt.figure(figsize=[12,6])
plt.title('Accel')
plt.subplot(3,1,1)
plt.plot(time,apple_recording_x)
plt.subplot(3,1,2)
plt.plot(time,apple_recording_y)
plt.subplot(3,1,3)
plt.plot(time,apple_recording_z)
```

Store extracted data

```
apple_data = {
    "time": unix_times,
    "x": apple_recording_x,
    "y": apple_recording_y,
    "z": apple_recording_z
}
apple_df = pd.DataFrame(apple_data)

# Save in format Patient-RZCH_Apple-Watch_2023_08_25
apple_df.to_csv("Patient-AC27_Apple-Watch_2024_06_17.csv", index=False)
```

APPENDIX B

The following Python script was written to retrieve the Apple Watch tremor metric data from the RuneLabs API.

Initializing libraries and access token

```
from runeq import initialize
from runeq.resources.stream_metadata import get_patient_stream_metadata
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from dateutil import parser
initialize(access_token_id='__________', access_token_secret='_____________')
```

Verify system login

```
from runeq.resources.user import get_current_user

my_user = get_current_user()
print(my_user)
print('Active Org:', my_user.active_org_name)
```

Extract tremor severity data and plot

```
tremor_validation_metadata = get_patient_stream_metadata(
    patient_id='_______________',
    algorithm='ingest-strive-applewatch-md.0',
    device_id='__________',
    measurement='tremor',
    severity='all'
)
tremor_validation_df = tremor_validation_metadata.to_dataframe()

print(tremor_validation_df.stream_type.unique)

start_time = _____________  # Apple Watch starting Unix time
end_time = _____________ # Apple Watch end Unix time

# Pull the actual data for the stream(s) in the stream metadata set.
tremor_validation = tremor_validation_metadata.get_stream_dataframe(
    start_time=start_time,
    end_time=end_time,
)

plt.figure
plt.step(tremor_validation['time'], tremor_validation['percentage'], where='post')
```

Eliminate errors, NaN's, and convert time stamps to Unix time

```
tremor_validation_time=tremor_validation['time'].values
tremor_validation_percentage=tremor_validation['percentage'].values
tremor_validation_percentage = tremor_validation_percentage.astype(float)
tremor_validation_percentage = pd.to_numeric(tremor_validation_percentage, errors='coerce')

tremor_validation_time = tremor_validation_time[~np.isnan(tremor_validation_percentage)]
tremor_validation_percentage = tremor_validation_percentage[~np.isnan(tremor_validation_percentage)]

unix_times = np.array([parser.isoparse(date).timestamp() for date in tremor_validation_time])
```

Store tremor metric data

```
validation_data = {
    "time": unix_times,
    "validation": tremor_validation_percentage
}
validation_df = pd.DataFrame(validation_data)
# Save in format Validation-RZCH_Apple-Watch_5_17_2024_Accel_1.csv
validation_df.to_csv("Validation-RZCH_Apple-Watch_5_17_2024_Accel_1.csv", index=False)
```

APPENDIX C

The following MATLAB script was developed to perform the frequency spectrum analysis and detect tremor present in the IPG and Apple Watch accelerometry data.

Load IPG data

```
clear all;
clc;
close all;

rng('default')

[filename_IPG, pathname] = uigetfile('*.xlsx', 'Time-Stamp data');
path(path,pathname);
filename_IPG
time_stamp_data = readtable([pathname filename_IPG]);

state_data = string(time_stamp_data.State);
date_data = string(time_stamp_data.Time);

idx_accel = find(contains(date_data,"Accel_") == 1);
fn = date_data(idx_accel)';
[indx,tf] = listdlg('PromptString',{'Select an Accel_# that correlate to .json
file'},'SelectionMode','single','ListString',fn);
fn(indx)
[filename_IPG, pathname] = uigetfile('*.json', 'Json File correlated to your
Time Stamp File',pathname);

data=jsondecode(fileread([pathname filename_IPG]));
```

Plot raw IPG data

```
x_samples_struct=cell2mat({data.XSamples});
x_samples=reshape(x_samples_struct,1,[]);
y_samples_struct=cell2mat({data.YSamples});
y_samples=reshape(y_samples_struct,1,[]);
z_samples_struct=cell2mat({data.ZSamples});
z_samples=reshape(z_samples_struct,1,[]);
[rows,columns]=size(x_samples_struct);

% create time vector for accelerometer data
time_struct=cell2mat({data.PacketGenTime});

magnitude=sqrt(x_samples.^2+y_samples.^2+z_samples.^2);

total_time_ipg=(time_struct(end)-time_struct(1))/1000;

IPG_fs=length(time_struct)/total_time_ipg;

time=[0:length(magnitude)-1]/(IPG_fs*rows);

figure
subplot(3,1,1);
plot(time, x_samples);
set(gca,'FontSize',18)
```

```
title('IPG Acceleration X-Axis','FontSize',18)
ylabel('a_x(G)','FontSize',18)
hold on

subplot(3,1,2);
plot(time, y_samples);
set(gca,'FontSize',18)
title('IPG Acceleration Y-Axis','FontSize',18)
ylabel('a_y(G)','FontSize',18)
hold on

subplot(3,1,3);
plot(time, z_samples);
set(gca,'FontSize',18)
title('IPG Acceleration Z-Axis','FontSize',18)
ylabel('a_z(G)','FontSize',18)
hold on


accel3axes_IPG = [x_samples; y_samples; z_samples];
```

Find motor task times

```
searching_motor = true;
imotor = 0;
idx_motor = idx_accel(indx) + 1;
motor_find = 0;

while searching_motor == true
    str_int = string(imotor);
    mot_str = "Motor#" + str_int;
    motor_find = find(mot_str == state_data);
    if null(motor_find) == 1
        searching_motor = false;
    else
        motor_task_start(imotor+1) = idx_motor;
    end
    imotor = imotor + 1;
    idx_motor = idx_motor + 2;
end
```

Define the dates and times we want to label

```
time_stamps={};

for idx = 1:length(motor_task_start)
    date = datetime(date_data(motor_task_start(idx)), 'InputFormat', 'dd-MMM-
yyyy HH:mm:ss','TimeZone','America/New_York');
    % Convert timesptamp to unix format
    unix_time = posixtime(date);
    unix_time_1 = unix_time*1000;
    [num,index]=min(abs(time_struct-unix_time_1));
    for i = 1:3
        subplot(3,1,i)
        plot(time((index)*rows), accel3axes_IPG(i,
(index)*rows),'r*',LineWidth=1.5)
        time_stamp(1,i)= time((index)*rows);
```

```
        time_stamp(2,i)= accel3axes_IPG(i, (index)*rows);
    end
    time_stamps{idx}=time_stamp;
end
```

IPG STFT

```
IPGaccel=accel3axes_IPG';
IPGfs=IPG_fs*rows;
tConvIPG = time;
channels = 3;
IPGaccel(isnan(IPGaccel))=0;
twin = 1.5;
e = nextpow2(twin*IPGfs);
Nfft = 2^e;
LwinSTFT = round(twin*IPGfs);
hwin = hanning(LwinSTFT);
Noverlap = round(0.7*LwinSTFT);

IPG_results = [];
IPG_tremor_abs = [];
IPG_all_abs= [];

figure
for j = 1:channels
    subplot(channels,1,j)
    [IPG_stft_synced, FIPG_synced, IPG_time] = stft(IPGaccel(:,j),IPGfs,
'Window', hwin, 'FFTLength', Nfft, 'OverlapLength', Noverlap);
    df = mean(diff(FIPG_synced));
    start = sum(FIPG_synced<2);
    imagesc((1:(length(IPGaccel(:,j))/IPGfs)-1),
FIPG_synced(start:end),abs(IPG_stft_synced(start:end,:)));
    set(gca,'FontSize',18)
    title('-Axis','FontSize',18)
    xlabel('Time (Seconds)','FontSize',18)
    ylabel('Frequency (Hz)','FontSize',18);
    colorbar;
    cb = colorbar;
    cb.Label.String = 'Magnitude';
    cb.Label.FontSize = 18;
    set(gca, 'YDir', 'normal')
    IPG.stft{j} = IPG_stft_synced;

    tremor_band=find(FIPG_synced >= 4 & FIPG_synced <= 7);

IPG_tremor_abs=[IPG_tremor_abs;abs(IPG_stft_synced(tremor_band(1):tremor_band(e
nd),:))/LwinSTFT];
    IPG_all_abs=[IPG_all_abs;abs(IPG_stft_synced)];
end

[s_1,n]=size(FIPG_synced);
[n,s_2]=size(IPG_all_abs);

IPG_all_mag=[];
for j=1:s_1
    for k=1:s_2
    IPG_all_mag(j,k) = sqrt((IPG_all_abs(j,k)).^2+(IPG_all_abs(s_1+j,k)).^2 +
(IPG_all_abs(2*s_1+j,k)).^2 );
    end
```

```
end
```

Plot IPG accelerometry with time stamps

```
figure
subplot(3,1,1);
plot(time, x_samples);
set(gca,'FontSize',18)
title('IPG Acceleration X-Axis','FontSize',18)
ylabel('a_x(G)','FontSize',18)
hold on

subplot(3,1,2);
plot(time, y_samples);
set(gca,'FontSize',18)
title('IPG Acceleration Y-Axis','FontSize',18)
ylabel('a_y(G)','FontSize',18)
hold on

subplot(3,1,3);
plot(time, z_samples);
set(gca,'FontSize',18)
title('IPG Acceleration Z-Axis','FontSize',18)
ylabel('a_z(G)','FontSize',18)
hold on

for k=1:m
    for i=1:3
        subplot(3,1,i)
        plot(time_stamps{k}(1,i),time_stamps{k}(2,i),'r*',LineWidth= 3);
    end
end
```

Load Apple Watch data

```
[filename_apple, pathname] = uigetfile('*.csv', 'Select Apple Data');
apple_data = readmatrix([pathname filename_apple]);

times=apple_data(:,1);
a_x_samples=apple_data(:,2);
a_y_samples=apple_data(:,3);
a_z_samples=apple_data(:,4);
```

Sync Apple Watch data with IPG data

```
ipg_begining=time_struct(1);
ipg_end=time_struct(end);
i=0;
detected_begin=1;
detected_end=1;

while detected_begin>=0
    i=i+1;
    apple_index_1=i;
    detected_begin=ipg_begining-times(i)*1000;
```

```
end
while detected_end>=0
    i=i+1;
    apple_index_2=i;
    detected_end=ipg_end-times(i)*1000;
end
apple_unix=times(apple_index_1:apple_index_2);
```

Plot raw Apple Watch data

```
fs=1/mean(diff(apple_unix));

synced_time=[0:length(apple_unix)-1]/fs;

figure
subplot(3,1,1);
plot(synced_time, a_x_samples(apple_index_1:apple_index_2));
set(gca,'FontSize',18)
title('Apple Watch Acceleration X-Axis','FontSize',18)
ylabel('a_x(G)','FontSize',18)
hold on

subplot(3,1,2);
plot(synced_time, a_y_samples(apple_index_1:apple_index_2));
set(gca,'FontSize',18)
title('Apple Watch Acceleration Y-Axis','FontSize',18)
ylabel('a_y(G)','FontSize',18)
hold on

subplot(3,1,3);
plot(synced_time, a_z_samples(apple_index_1:apple_index_2));
set(gca,'FontSize',18)
title('Apple Watch Acceleration Z-Axis','FontSize',18)
ylabel('a_z(G)','FontSize',18)
hold on

accel3axes_apple = [a_x_samples(apple_index_1:apple_index_2)';
a_y_samples(apple_index_1:apple_index_2)';
a_z_samples(apple_index_1:apple_index_2)'];

time_stamps_apple={};

for idx = 1:length(motor_task_start)
    date = datetime(date_data(motor_task_start(idx)), 'InputFormat', 'dd-MMM-
yyyy HH:mm:ss','TimeZone','America/New_York');
    % Convert timesptamp to unix format
    unix_time_1 = posixtime(date);
    [num,index]=min(abs(apple_unix-unix_time_1));
    for i = 1:3
        subplot(3,1,i)
        plot(synced_time(index), accel3axes_apple(i,index),'r*',LineWidth=1.5)
        time_stamp_apple(1,i)= synced_time(index);
        time_stamp_apple(2,i)= accel3axes_apple(i, index);
    end
    time_stamps_apple{idx}=time_stamp_apple;
end
```

Perform Apple Watch STFT and store results

```
Appleaccel=accel3axes_apple';
tConvApple = synced_time;
channels = size(Appleaccel,2);
Appleaccel(isnan(Appleaccel))=0;


twin = 1.5;
e = nextpow2(twin*fs); % changes nfft value acccording to time window
Nfft = 2^e;
LwinSTFT = round(twin*fs);
hwin = hanning(LwinSTFT);
Noverlap = round(0.7*LwinSTFT);

Apple_results = [];
Apple_tremor_abs = [];
Apple_all_abs= [];

figure
for j = 1:channels
    subplot(channels,1,j)
    [Apple_stft_synced, Apple_synced, Apple_time] = stft(Appleaccel(:,j),fs,
'Window', hwin, 'FFTLength', Nfft, 'OverlapLength', Noverlap);
    df = mean(diff(Apple_synced));
    start = sum(Apple_synced<2);
    imagesc((1:(length(Appleaccel(:,j))/fs)-1), Apple_synced(start:end),
abs(Apple_stft_synced(start:end,:)));
    set(gca,'FontSize',18)
    title('-Axis','FontSize',18)
    xlabel('Time (Seconds)','FontSize',18)
    ylabel('Frequency (Hz)','FontSize',18);
    colorbar;
    cb = colorbar;
    cb.Label.String = 'Magnitude';
    cb.Label.FontSize = 18;
    set(gca, 'YDir', 'normal')

    tremor_band=find(Apple_synced >= 4 & Apple_synced <= 8);

Apple_tremor_abs=[Apple_tremor_abs;abs(Apple_stft_synced(tremor_band(1):tremor_
band(end),:))];
    Apple_all_abs=[Apple_all_abs;abs(Apple_stft_synced)];
end

[s_1,n]=size(Apple_synced);
[n,s_2]=size(Apple_all_abs);
Apple_all_mag=[];
for j=1:s_1
    for k=1:s_2
    Apple_all_mag(j,k) =
sqrt((Apple_all_abs(j,k)).^2+(Apple_all_abs(s_1+j,k)).^2 +
(Apple_all_abs(2*s_1+j,k)).^2 );
    end
end
```

Plot hamming window in time domain

```
win_t=(1:length(hwin))/fs;


figure
plot(win_t,hwin)
set(gca,'FontSize',18)
xlabel('Time (Seconds)','FontSize',18);
ylabel('Amplitude','FontSize',18);
title('Hamming Window Used in Short Time Fourier Transform ','FontSize',18);
```

Plot frequency response of hamming window

```
[H, f_win] = freqz(hwin, 1, 1024,fs);


figure
plot(f_win, abs(H));
set(gca,'FontSize',18)
xlabel('Frequency (\times\pi rad/sample)','FontSize',18);
ylabel('Magnitude','FontSize',18);
set(gca,'YScale','log')
title('Frequency Response of Hamming Window');
grid on;
```

Plot tremor band and first two harmonics for IPG and Apple Watch accelerometry

```
power_stamp_apple=[];


power_stamp_ipg=[];


tremor_band_titles={'3-7 Hz', '7-11 Hz', '13-18 Hz'};
tremor_bands=[3 7 7 11 13 18];

for k= 1:m
    [number,apple_stamp]=min(abs(Apple_time-time_stamps_apple{k}(1,1)));
    power_stamp_apple=[power_stamp_apple;apple_stamp];
    [number,ipg_stamp]=min(abs(IPG_time-time_stamps{k}(1,1)));
    power_stamp_ipg=[power_stamp_ipg;ipg_stamp];
end
harmonic_mag_results_apple=[];
harmonic_mag_results=[];

% IPG
figure
for k=1:length(tremor_bands)/2
    harmonic_tremor_abs=[];
    harmonic_tremor_band=find(FIPG_synced >=tremor_bands(k*2-1) & FIPG_synced
<=tremor_bands(k*2));

harmonic_tremor_abs=IPG_all_mag(harmonic_tremor_band(1):harmonic_tremor_band(en
d),:);
    dist = trapz(harmonic_tremor_abs);
    df = mean(diff(FIPG_synced));
    BW = FIPG_synced(harmonic_tremor_band(end))-
FIPG_synced(harmonic_tremor_band(1));
```

```
    harmonic_mag_results= [harmonic_mag_results;10*log10(dist.*df/BW)];
    subplot(length(tremor_bands)/2,1,k)
    plot(IPG_time,harmonic_mag_results(k,:))
    title(['IPG ' tremor_band_titles{k}])
    ylabel('Integral Power (dB)')
    xlabel('Time (seconds)')
    hold on
    for j=1:m
        subplot(length(tremor_bands)/2,1,k)

plot(IPG_time(power_stamp_ipg(j)),harmonic_mag_results(k,power_stamp_ipg(j)),'r
*',LineWidth=1.5)
    end
end
% IPG harmonics envelopes
figure
for k=1:length(tremor_bands)/2
    w=round(5*length(IPG_time)/IPG_time(end));
    ones_m=ones(1,w);

harmonic_envelope=conv(harmonic_mag_results(k,:),ones_m,'same')/length(ones_m);
    subplot(length(tremor_bands)/2,1,k)
    plot(IPG_time,harmonic_envelope)
    title(['IPG Power Envelope' tremor_band_titles{k}])
    ylabel('Integral Power (dB)')
    xlabel('Time (seconds)')
    hold on
        for j=1:m
            subplot(length(tremor_bands)/2,1,k)

plot(IPG_time(power_stamp_ipg(j)),harmonic_envelope(power_stamp_ipg(j)),'r*',Li
neWidth=1.5)
        end
end

% Apple
figure
for k=1:length(tremor_bands)/2
    harmonic_tremor_abs_apple=[];
    harmonic_tremor_band_apple=find(Apple_synced >=tremor_bands(k*2-1) &
Apple_synced <=tremor_bands(k*2));

harmonic_tremor_abs_apple=Apple_all_mag(harmonic_tremor_band_apple(1):harmonic_
tremor_band_apple(end),:);
    dist = trapz(harmonic_tremor_abs_apple);
    df = mean(diff(Apple_synced));
    BW = Apple_synced(harmonic_tremor_band_apple(end))-
Apple_synced(harmonic_tremor_band_apple(1));
    harmonic_mag_results_apple=
[harmonic_mag_results_apple;10*log10(dist.*df/BW)];
    subplot(length(tremor_bands)/2,1,k)
    plot(Apple_time,harmonic_mag_results_apple(k,:))
    title(['Apple ' tremor_band_titles{k}])
    ylabel('Integral Power (dB)')
    xlabel('Time (seconds)')
    hold on
    for j=1:m
        subplot(length(tremor_bands)/2,1,k)

plot(Apple_time(power_stamp_apple(j)),harmonic_mag_results_apple(k,power_stamp_
apple(j)),'r*',LineWidth=1.5)
    end
end
```

```
% Apple harmonics envelopes
figure
for k=1:length(tremor_bands)/2
    w=round(5*length(Apple_time)/Apple_time(end));
    ones_m=ones(1,w);

harmonic_envelope_apple=conv(harmonic_mag_results_apple(k,:),ones_m,'same')/len
gth(ones_m);
    subplot(length(tremor_bands)/2,1,k)
    plot(Apple_time,harmonic_envelope_apple)
    title(['Apple Power Envelope ' tremor_band_titles{k}])
    ylabel('Integral Power (dB)')
    xlabel('Time (seconds)')
    hold on
        for j=1:m
            subplot(length(tremor_bands)/2,1,k)

plot(Apple_time(power_stamp_apple(j)),harmonic_envelope_apple(power_stamp_apple
(j)),'r*',LineWidth=1.5)
        end
end
```

Apple STFT spectrogram

```
A=20*log10(Apple_all_mag(end/2:end,:));


A_norm=(A-
min(min(20*log10(Apple_all_mag)))./(max(max(20*log10(Apple_all_mag)))-
min(min(20*log10(Apple_all_mag))));


figure
h=surf(Apple_time,Apple_synced(end/2:end),A_norm);
set(gca,'FontSize',18)
title('Apple STFT Spectrogram (Normalized)','FontSize',18)
xlabel('Time (Seconds)','FontSize',18)
ylabel('Frequency (Hz)','FontSize',18)
zlabel('Magnitude dB','FontSize',18)
ylim([0,max(Apple_synced)])
xlim([0,max(Apple_time)])
set(h, 'EdgeColor', 'none');
colorbar
grid off
```

IPG STFT spectrogram

```
A_IPG=20*log10(IPG_all_mag(end/2:end,:));


A_IPG_norm=(A_IPG-
min(min(20*log10(IPG_all_mag)))./(max(max(20*log10(IPG_all_mag)))-
min(min(20*log10(IPG_all_mag))));


figure
h=surf(IPG_time,FIPG_synced(end/2:end),A_IPG_norm);
set(gca,'FontSize',18)
title('IPG STFT Spectrogram (Normalized)','FontSize',18)
```

```
xlabel('Time (Seconds)','FontSize',18)
ylabel('Frequency (Hz)','FontSize',18)
zlabel('Magnitude dB','FontSize',18)
ylim([0,max(FIPG_synced)])
xlim([0,max(IPG_time)])
set(h, 'EdgeColor', 'none');
colorbar
grid off
```

Tremor detection using Apple Watch STFT

```
apple_detecting=harmonic_mag_results_apple(1,:);
tremor_detected=[];
peak=[];
detected=[];
t_sync=Apple_synced(end/2:end);

for k = 1:length(apple_detecting)
    detected=0;
    if apple_detecting(k)>=-1

[TF,P]=islocalmax(20*log10(Apple_all_mag(end/2:end,k)),'MinProminence',12);
        amp=20*log10(Apple_all_mag(end/2:end,k));
        peaks=find(TF);
        if isempty(peaks)
            detected = 0;
        elseif any((t_sync(peaks) >= 3 & t_sync(peaks) <= 7) | (t_sync(peaks)
>= 7 & t_sync(peaks) <= 10))
            if any(amp(peaks)>=-25)
                detected = 1;
            end
        else
            detected=0;
        end
    end
    tremor_detected=[tremor_detected;detected];
end

figure
stairs(Apple_time,tremor_detected)
title("Apple STFT Tremor Detection")
ylim([0 2])
```

Perform CWT on IPG accelerometry and plot results

```
IPG_wave_abs=[];

figure

for j = 1:channels % acceleration axis - x, y, or z
    [IPG_wavelet, FIPG_wavelet] = cwt(IPGaccel(:,j),IPGfs);
    %cwt(IPGaccel(:,j),IPGfs);
    IPG_wave_abs=[IPG_wave_abs;abs(IPG_wavelet)];
    subplot(3,1,j)
    h=surf(time,FIPG_wavelet,abs(IPG_wavelet));
```

```
    set(gca,'FontSize',18)
    title('-Axis','FontSize',18)
    xlabel('Time (Seconds)','FontSize',18)
    ylabel('Frequency (Hz)','FontSize',18)
    ylim([0.18,max(FIPG_wavelet)])
    xlim([0,max(time)])
    %set(gca, 'YScale', 'log');
    set(h, 'EdgeColor', 'none');
    cb = colorbar;
    cb.Label.String ='Magnitude';
    cb.Label.FontSize = 18;
    grid off
end

[s_1,n]=size(FIPG_wavelet);
[n,s_2]=size(IPG_wave_abs);

IPG_wave_mag=[];
for j=1:s_1
    for k=1:s_2
    IPG_wave_mag(j,k) = sqrt((IPG_wave_abs(j,k)).^2+(IPG_wave_abs(s_1+j,k)).^2
+ (IPG_wave_abs(2*s_1+j,k)).^2 );
    end
end

IPG_wave_db=flipud(20*log10(IPG_wave_mag));
wavelet_freq=flipud(FIPG_wavelet);

A_IPG_wave_norm=(IPG_wave_db-min(min(IPG_wave_db)))./(max(max(IPG_wave_db))-
min(min(IPG_wave_db)));

figure('Name', [filename_IPG '_' char(fn(indx))])
h=surf(time,wavelet_freq,A_IPG_wave_norm);
set(gca,'FontSize',18)
title('IPG Wavelet Scalogram (Normalized)','FontSize',18)
xlabel('Time (Seconds)','FontSize',18)
ylabel('Frequency (Hz)','FontSize',18)
ylim([0.18,max(wavelet_freq)])
xlim([0,max(time)])
%set(gca, 'YScale', 'log');
set(h, 'EdgeColor', 'none');
colorbar
grid off
```

Calculate IPG activity band average power

```
plotting_av=[];


for k=1:length(time)
    IPG_wave_amp=IPG_wave_db(:,k);
    activity_band_i=wavelet_freq>=0.5 & wavelet_freq<=3;
    IPG_wave_amp_st=10.^(IPG_wave_amp/20);

average_power=trapz(wavelet_freq(activity_band_i),IPG_wave_amp_st(activity_band
_i))/(max(wavelet_freq(activity_band_i))-min(wavelet_freq(activity_band_i)));
    plotting_av=[plotting_av,average_power];
end
w=round(3*length(time)/time(end));
ones_m=ones(1,w);
activity_envelope=conv(plotting_av,ones_m,'same')/length(ones_m);
```

```
figure('Name', [filename_IPG '_' char(fn(indx))])
subplot(2,1,1)
plot(time,plotting_av)
xlabel('Time (seconds)')
ylabel('Average power')
title('IPG 0.5-3 Hz Average Power')
hold on

subplot(2,1,2)
plot(time,activity_envelope)
xlabel('Time (seconds)')
ylabel('Power Moving Average')
title('IPG 0.5-3 Hz Moving Average')
hold on
```

IPG CWT tremor detection

```
IPG_resting_tremor_detection=[];
IPG_act_trem_detection=[];
IPG_freq_amp_4_7=[];
IPG_freq_4_7_i=[];
transition_amp=[];
harmonic_freq=[];
harmonic_amp=[];
harmonic_index_1=[];
harmonic_index_2=[];
margin=0.01;



for k=1:length(time)
    % Initialize variables
    activity= activity_envelope(k)>=1;
    detected_resting_tremor=0;
    detected_act_tremor=0;
    IPG_wave_amp=IPG_wave_db(:,k);

    % Find peaks to compare
    [TF,P]=islocalmax(IPG_wave_amp);
    peaks=find(TF);
    [IPG_freq_amp_4_7,f_idx] = max(IPG_wave_amp(peaks(wavelet_freq(TF)>=3.5 &
wavelet_freq(TF)<= 7.5)));
    IPG_freq_4_7_i=peaks(wavelet_freq(TF)>=3.5 & wavelet_freq(TF)<= 7.5);
    transition_amp=max(IPG_wave_amp(peaks(wavelet_freq(TF)>0.1 &
wavelet_freq(TF)<= 0.5)));

    % find harmonics and extract value if present
    if ~isempty(IPG_freq_4_7_i)
        harmonic_freq=2*wavelet_freq(IPG_freq_4_7_i(f_idx));
        harmonic_index_1=find(abs(wavelet_freq-harmonic_freq)<=margin);
        if ~isempty(harmonic_index_1)

harmonic_index_2=find(wavelet_freq==wavelet_freq(harmonic_index_1));
            harmonic_amp=IPG_wave_amp(harmonic_index_2);
        end
    end

    % Detect and distinguish resting or activity tremor
```

```
    switch activity
        case false
            if (~isempty(IPG_freq_4_7_i) && ~isempty(IPG_freq_amp_4_7)) &&
(IPG_freq_amp_4_7 > 4)
                detected_resting_tremor=1;
            end
        case true
            if ((~isempty(harmonic_amp) && ~isempty(IPG_freq_amp_4_7)) &&
transition_amp < 15) && (IPG_freq_amp_4_7 > 20 && harmonic_amp > 15)
                detected_act_tremor=1;
            end
    end

    IPG_act_trem_detection=[IPG_act_trem_detection;detected_act_tremor];

IPG_resting_tremor_detection=[IPG_resting_tremor_detection;detected_resting_tre
mor];
end

IPG_combined_tremor=IPG_resting_tremor_detection|IPG_act_trem_detection;
```

Plot IPG resting tremor, activity tremor, and combined tremor detection

```
figure('Name', [filename_IPG '_' char(fn(indx))])
subplot(3,1,1)
stairs(time,IPG_resting_tremor_detection)
title('IPG Resting Wavelet Detection')
ylabel('Detection')
xlabel('Time (seconds)')
ylim([0,2])

subplot(3,1,2)
stairs(time,IPG_act_trem_detection)
title('IPG Activity Wavelet Detection')
ylabel('Detection')
xlabel('Time (seconds)')
ylim([0,2])

subplot(3,1,3)
stairs(time,IPG_combined_tremor)
title('Combined IPG Wavelet Detection')
ylabel('Detection')
xlabel('Time (seconds)')
ylim([0,2])

% Plot tremor detection over original acceleration plots
trem_i=find(IPG_combined_tremor==1);

x_IPG_trem=NaN(size(x_samples));
x_IPG_trem(trem_i)=x_samples(trem_i);

y_IPG_trem=NaN(size(y_samples));
y_IPG_trem(trem_i)=y_samples(trem_i);

z_IPG_trem=NaN(size(z_samples));
z_IPG_trem(trem_i)=z_samples(trem_i);

figure('Name', [filename_IPG '_' char(fn(indx))])
subplot(3,1,1)
plot(time, x_samples, time, x_IPG_trem, 'r-')
```

```
set(gca,'FontSize',18)
title('IPG Acceleration X-Axis','FontSize',18)
ylabel('a_x(G)','FontSize',18)
hold on

subplot(3,1,2)
plot(time, y_samples, time,y_IPG_trem, 'r-')
set(gca,'FontSize',18)
title('IPG Acceleration Y-Axis','FontSize',18)
ylabel('a_y(G)','FontSize',18)
hold on

subplot(3,1,3)
plot(time, z_samples,time, z_IPG_trem, 'r-')
set(gca,'FontSize',18)
title('IPG Acceleration Z-Axis','FontSize',18)
xlabel('Time (Seconds)','FontSize',18)
ylabel('a_z(G)')
hold on

for k=1:m
    for i=1:channels
        subplot(3,1,i)
        plot(time_stamps{k}(1,i),time_stamps{k}(2,i),'yo',LineWidth= 3);
    end
end
```

Perform CWT on Apple Watch accelerometry and plot results

```
Apple_wave_abs=[];

fb =
cwtfilterbank(SignalLength=numel(synced_time),SamplingFrequency=fs,TimeBandwidt
h=60);

figure

for j = 1:channels
    [Apple_wavelet, FApple_wavelet,coi,scaling] =
cwt(Appleaccel(:,j),FilterBank=fb);
    Apple_wave_abs=[Apple_wave_abs;abs(Apple_wavelet)];
    subplot(3,1,j)
    h=surf(synced_time,FApple_wavelet,abs(Apple_wavelet));
    set(gca,'FontSize',18)
    title('-Axis','FontSize',18)
    xlabel('Time (Seconds)','FontSize',18)
    ylabel('Frequency (Hz)','FontSize',18)
    ylim([0.018,FApple_wavelet(1)])
    xlim([0,max(synced_time)])
    set(h, 'EdgeColor', 'none');
    colorbar
    grid off
    cb = colorbar;
    cb.Label.String ='Magnitude';
    cb.Label.FontSize = 18;
    grid off
end

[s_1,n]=size(FApple_wavelet);
```

```
[n,s_2]=size(Apple_wave_abs);

Apple_wave_mag=[];
for j=1:s_1
    for k=1:s_2
    Apple_wave_mag(j,k) =
sqrt((Apple_wave_abs(j,k)).^2+(Apple_wave_abs(s_1+j,k)).^2 +
(Apple_wave_abs(2*s_1+j,k)).^2 );
    end
end

dif = diff(FApple_wavelet);
dif = [dif(1); dif];
power_per_hz_w = Apple_wave_mag ./ df;
power_per_hz_w=flipud(20*log10(power_per_hz_w));

Apple_wave_db=flipud(20*log10(Apple_wave_mag));
Apple_wavelet_freq=flipud(FApple_wavelet);

Apple_wave_db_norm=(power_per_hz_w-
min(min(power_per_hz_w)))./(max(max(power_per_hz_w))-min(min(power_per_hz_w)));
figure('Name', [filename_IPG '_' char(fn(indx))])
h=surf(synced_time,Apple_wavelet_freq,Apple_wave_db_norm);
set(gca,'FontSize',18)
title('Apple Wavelet Scalogram (Normalized)','FontSize',18)
xlabel('Time (Seconds)','FontSize',18)
ylabel('Frequency (Hz)','FontSize',18)
ylim([0.018,Apple_wavelet_freq(end)])
xlim([0,max(synced_time)])
%set(gca, 'YScale', 'log');
set(h, 'EdgeColor', 'none');
colorbar
grid off
```

Plot CWT filter bank

```
signalLength = length(Appleaccel(:,1));
fb = cwtfilterbank(SamplingFrequency=fs,TimeBandwidth=60);
figure
freqz(fb);
[psi,t] = wavelets(fb);


figure
plot(t,abs(psi(1,:)))
hold on
plot(t,abs(psi(end,:)))
grid on
title('Time-domain Wavelets')
```

Calculate Apple Watch activity band average power

```
plotting_av=[];


for k=1:length(synced_time)
    A_wave_amp=Apple_wave_db(:,k);
    activity_band_i=Apple_wavelet_freq>=0.01 & Apple_wavelet_freq<=3;
    A_wave_amp_st=10.^(A_wave_amp/20);

average_power=trapz(Apple_wavelet_freq(activity_band_i),A_wave_amp_st(activity_
```

```
band_i))/(max(Apple_wavelet_freq(activity_band_i))-
min(Apple_wavelet_freq(activity_band_i)));
    plotting_av=[plotting_av,average_power];
end
w=round(3*length(synced_time)/synced_time(end));
ones_m=ones(1,w);
activity_envelope=conv(plotting_av,ones_m,'same')/length(ones_m);

figure
subplot(2,1,1)
plot(synced_time,plotting_av)
set(gca,'FontSize',18)
xlabel('Time (seconds)','FontSize',18)
ylabel('Average power','FontSize',18)
title('Apple 0.01-3 Hz Average Power','FontSize',18)
hold on

subplot(2,1,2)
plot(synced_time,activity_envelope)
set(gca,'FontSize',18)
xlabel('Time (seconds)','FontSize',18)
ylabel('Power Moving Average','FontSize',18)
title('Apple 0.01-3 Hz Moving Average','FontSize',18)
hold on
```

Apple Watch CWT tremor detection

```
resting_tremor_detection=[];
act_trem_detection=[];
A_freq_amp_4_7=[];
A_freq_4_7_i=[];
transition_amp=[];
harmonic_freq=[];
harmonic_amp=[];
harmonic_index_1=[];
harmonic_index_2=[];
margin=0.01;


for k=1:length(synced_time)
    % Initialize variables
    activity= activity_envelope(k)>=0.04;
    detected_resting_tremor=0;
    detected_act_tremor=0;
    A_wave_amp=Apple_wave_db(:,k);

    % Find peaks to compare
    [TF,P]=islocalmax(A_wave_amp);
    peaks=find(TF);
    [A_freq_amp_4_7,f_idx] = max(A_wave_amp(peaks(Apple_wavelet_freq(TF)>=3.5 &
Apple_wavelet_freq(TF)<= 7.5)));
    A_freq_4_7_i=peaks(Apple_wavelet_freq(TF)>=3.5 & Apple_wavelet_freq(TF)<=
7.5);
    transition_amp=max(A_wave_amp(peaks(Apple_wavelet_freq(TF)>0.1 &
Apple_wavelet_freq(TF)<= 0.5)));

    % find harmonics and extract value if present
    if ~isempty(A_freq_4_7_i)
        harmonic_freq=2*Apple_wavelet_freq(A_freq_4_7_i(f_idx));
        harmonic_index_1=find(abs(Apple_wavelet_freq-harmonic_freq)<=margin);
        if ~isempty(harmonic_index_1)
```

```
harmonic_index_2=find(Apple_wavelet_freq==Apple_wavelet_freq(harmonic_index_1))
;
            harmonic_amp=A_wave_amp(harmonic_index_2);
        end
    end

    % Detect and distinguish resting or activity tremor
    switch activity
        case false
            if ~isempty(A_freq_amp_4_7) && A_freq_amp_4_7>-40
                detected_resting_tremor=1;
            end
        case true
            if (~isempty(harmonic_amp) && ~isempty(A_freq_amp_4_7)) &&
(A_freq_amp_4_7>-20 && harmonic_amp > -25 && activity_envelope(k)<=0.08)
                detected_act_tremor=1;
            end
    end

    act_trem_detection=[act_trem_detection;detected_act_tremor];

resting_tremor_detection=[resting_tremor_detection;detected_resting_tremor];
end

Apple_combined_tremor=resting_tremor_detection|act_trem_detection;
```

Plot resting tremor, activity tremor, and combined tremor detection

```
figure


subplot(3,1,1)


stairs(synced_time,resting_tremor_detection)


title('Resting-Apple Wavelet Detection')
ylabel('Detection')
xlabel('Time (Seconds)')
ylim([0,2])

subplot(3,1,2)
stairs(synced_time,act_trem_detection)
title('Activity-Apple Wavelet Detection')
ylabel('Detection')
xlabel('Time (Seconds)')
ylim([0,2])

subplot(3,1,3)
stairs(synced_time,Apple_combined_tremor)
title('Combined-Apple Wavelet Detection')
ylabel('Detection')
xlabel('Time (Seconds)')
ylim([0,2])

% Plot tremor detection over original acceleration plots
trem_i=find(Apple_combined_tremor==1);
x_apple=a_x_samples(apple_index_1:apple_index_2);
y_apple=a_y_samples(apple_index_1:apple_index_2);
z_apple=a_z_samples(apple_index_1:apple_index_2);
```

```
x_apple_trem=NaN(size(x_apple));
x_apple_trem(trem_i)=x_apple(trem_i);

y_apple_trem=NaN(size(y_apple));
y_apple_trem(trem_i)=y_apple(trem_i);

z_apple_trem=NaN(size(z_apple));
z_apple_trem(trem_i)=z_apple(trem_i);

figure
subplot(3,1,1)
plot(synced_time, x_apple, synced_time, x_apple_trem, 'r-')
set(gca,'FontSize',18)
title('Apple Watch Acceleration X-Axis','FontSize',18)
ylabel('a_x(G)','FontSize',18)
hold on

subplot(3,1,2)
plot(synced_time, y_apple, synced_time,y_apple_trem, 'r-')
set(gca,'FontSize',18)
title('Apple Watch Acceleration Y-Axis','FontSize',18)
ylabel('a_y(G)','FontSize',18)
hold on

subplot(3,1,3)
plot(synced_time, z_apple,synced_time, z_apple_trem, 'r-')
set(gca,'FontSize',18)
title('Apple Watch Acceleration Z-Axis','FontSize',18)
xlabel('Time (Seconds)','FontSize',18)
ylabel('a_z(G)','FontSize',18)
hold on

for k=1:m
    for i=1:channels
        subplot(3,1,i)

plot(time_stamps_apple{k}(1,i),time_stamps_apple{k}(2,i),'yo',LineWidth= 3);
    end
end
```

Compare Apple Watch and IPG CWT tremor detection

```
int_time=min(min(synced_time),min(time)):0.01:max(max(synced_time),max(time));


Apple_interp=interp1(synced_time,double(Apple_combined_tremor),int_time,'neares
t');
IPG_interp=interp1(time,double(IPG_combined_tremor),int_time,'nearest');

trem_overlap=(Apple_interp==1)&(IPG_interp==1);

trem_interp_values=int_time(trem_overlap);

% compute pairwise time differences
distances_to_Apple = synced_time'-trem_interp_values;
distances_to_IPG = time'-trem_interp_values;

% Find the closest indices
[~, closest_Apple] = min(abs(distances_to_Apple));
[~, closest_IPG] = min(abs(distances_to_IPG));
```

```
% Retrieve the original times
original_times_Apple = unique(synced_time(closest_Apple));
original_times_IPG = unique(time(closest_IPG));

%logic array for when both IPG and Apple detect tremor
Apple_overlap=ismember(synced_time,original_times_Apple);
IPG_overlap=ismember(time,original_times_IPG);

figure ('Name', [filename_IPG '_' char(fn(indx))])
subplot(3,1,1)
stairs(time,IPG_combined_tremor)
set(gca,'FontSize',18)
ylim([0,2])
title('IPG-Tremor Detected','FontSize',18)
ylabel('Detected','FontSize',18)
subplot(3,1,2)
stairs(synced_time,Apple_combined_tremor);
set(gca,'FontSize',18)
ylim([0,2])
title('Apple-Tremor Detected','FontSize',18)
ylabel('Detected','FontSize',18)
subplot(3,1,3)
stairs(int_time,trem_overlap);
set(gca,'FontSize',18)
ylim([0,2])
title('Apple and IPG Overlap','FontSize',18)
xlabel('Time (Seconds)','FontSize',18)
ylabel('Detected','FontSize',18)

% Compare Apple STFT vs Wavelet Detection
int_time=min(min(synced_time),min(Apple_time)):0.01:max(max(synced_time),max(Ap
ple_time));

wavelet_interp=interp1(synced_time,double(Apple_combined_tremor),int_time,'line
ar', 'extrap');
stft_interp=interp1(Apple_time,tremor_detected,int_time,'linear', 'extrap');

apple_overlap=(wavelet_interp==1)&(stft_interp==1);

apple_interp_values=int_time(apple_overlap);

% compute pairwise time differences
distances_to_wavelet = synced_time'-apple_interp_values;
distances_to_stft = Apple_time-apple_interp_values;

% Find the closest indices
[~, closest_wavelet] = min(abs(distances_to_wavelet));
[~, closest_stft] = min(abs(distances_to_stft));

% Retrieve the original times
original_times_wavelet = unique(synced_time(closest_wavelet));
original_times_stft = unique(Apple_time(closest_stft));

% logic array for when both IPG and Apple detect tremor
wavelet_overlap=ismember(synced_time,original_times_wavelet);
stft_overlap=ismember(time,original_times_stft);

figure ('Name', [filename_apple '_' char(fn(indx))])
subplot(3,1,1)
stairs(synced_time,Apple_combined_tremor);
set(gca,'FontSize',18)
ylim([0,2])
```

```
title('Apple Wavelet Tremor Detection','FontSize', 18)
ylabel('Detected','FontSize', 18)
subplot(3,1,2)
stairs(Apple_time,tremor_detected)
set(gca,'FontSize',18)
ylim([0,2])
title('Apple STFT Tremor Detection','FontSize', 18)
ylabel('Detected','FontSize', 18)
subplot(3,1,3)
stairs(int_time,apple_overlap);
set(gca,'FontSize',18)
ylim([0,2])
title('Apple Wavelet vs STFT Overlap','FontSize', 18)
xlabel('Time (Seconds)','FontSize', 18)
ylabel('Detected','FontSize', 18)
```

Plot frequencies at individual time points Apple Watch CWT

```
find_this_time=177;
time_index=find(synced_time>find_this_time, 1);
y_wavelet=Apple_wave_db_norm(:,time_index);
[TF,P]= islocalmax(y_wavelet);


figure
plot(Apple_wavelet_freq,y_wavelet)
set(gca, 'FontSize', 18)
title(sprintf('Frequency Power Normalized (Wavelet)\n%d Seconds Into Trial',
find_this_time),'FontSize', 18);
xlabel('Frequency (Hz)','FontSize', 18)
ylabel('Power Content (Normalized)','FontSize', 18)
xlim([0 max(Apple_wavelet_freq)])
ylim([0 1])
```

Plot frequencies at individual time points Apple Watch STFT

```
find_this_time=175;
time_index=find(Apple_time>find_this_time, 1);
A=20*log10(Apple_all_mag(end/2:end,time_index)/(t_sync(2)-t_sync(1)));
A_norm=(A-
min(min(20*log10(Apple_all_mag)))./(max(max(20*log10(Apple_all_mag)))-
min(min(20*log10(Apple_all_mag))));
t_sync=Apple_synced(end/2:end);


figure
plot(t_sync,A_norm)
set(gca, 'FontSize', 18)
title(sprintf('Frequency Power Normalized (STFT) \n%d Seconds Into Trial',
find_this_time),'FontSize', 18);
xlabel('Frequency (Hz)','FontSize', 18)
ylabel('Power Content (Normalized)','FontSize', 18)
xlim([0 max(t_sync)])
ylim([0 1])
```