# The Efficacy of Using Color in Sub-Threshold Vibration to Improve Older Adult Balance

Zaccur Nkrumah

Submitted to the graduate degree program in Bioengineering and the Graduate Faculty of the University of Kansas in partial fulfillment of the requirements for the degree of Master of Science.

_____

Chairperson: Dr. Carl Luchies

_____

Dr. Sara Wilson

_____

Dr. Huazhen Fang

Date Defended: May 4, 2022

The Thesis Committee for Zaccur Yenyi Nkrumah certifies that this is the approved version of the following thesis:

**The Efficacy of Using Color in Sub-threshold Vibration to Improve Older Adult Balance**

_____

Chairperson: Dr. Carl Luchies

Date Approved: May 4,2022

## Abstract

The purpose of this study is to investigate (1)the variability of the initial state balance measures within each participant, across different days and within each day, across the participants. (2) the effect of different noise color on the magnitude and the direction of change in the state. (3) if the initial state affects the magnitude and direction in which each treatment color pushes the change in state. We hypothesized that (1) the sway measures before stimulation will differ across subjects and vary within a subject across the sessions. (2). Each vibration color would cause a change in the state different from the placebo. (3)The efficacy of utilizing subthreshold vibration to change postural sway magnitude, predictability, and complexity depends on their initial values and the color used. The long-term goal is to demonstrate that the state of balance can be changed to the desired direction using a specific noise color.

Nine healthy older adults (HO: three male and six females; age: 62.56 ± 1.42 years; height: 169.48 ± 9.32 cm) with no significant history of musculoskeletal, neurologic, vestibular, or mobility deficiencies participated in the study. The participants visited the lab 4 times, and each time their sway before vibration, during vibration and soon after vibration was measured using a force plate. For each visit, different vibration color (white, pink, brown, and placebo) was used, and the order was random for each participant. All sway trials were conducted with people's heels 17 cm apart, eyes closed, hands by their sides, and toes pointing forward. The force plate kinetic data were collected at 2500 Hz using a 16-bit A/D CED Power mkII and Spike2 recorder (Cambridge Electronic Design, UK). All the data analysis was done in MATLAB R2020b (MathWorks, MA, USA). The force and moment data were filtered to remove any motor vibration sway time series with a low pass filter at 20 Hz cut-off frequency. The center of pressure in the anteroposterior(AP) and Mediolateral(ML) directions was calculated from each sway trial . The COP data ($COP_{AP}$, $COP_{ML}$) were downsampled to 50 Hz and measures of magnitude(RMS), predictability(Sample Entropy), and

complexity(DFA) were extracted for each of the sway trials. The statistical analysis was done using Python 3. First, ANOVA was used to determine the statistical significance of the baseline measures (entropy, DFA, and RMS) across the participants. Then a repeated measure of ANOVA was done between baseline measures (entropy, DFA, and RMS) across the four visits. Another repeated measure of ANOVA was performed to determine the statistical significance of the change in measure (measure $_{vibration}$ − measure $_{baseline}$) in the ML direction for the different treatments (white, pink, brown and placebo). The statistical significance level was set to <0.05.

Also, a linear regression model was fitted between the change in measure (measure $_{vibration}$ − measure $_{baseline}$) and the baseline for each treatment (white, pink, brown, and placebo) in the ML direction

There was a significant difference in the baseline entropy, DFA and RMS value across subjects. Also, within-subjects, baseline measures show different means and variability across subjects. Some had low standard deviations, while others had high standard deviations. However, there was no significant difference in entropy, DFA and RMS baseline across the four visit days.

There was a general increase in predictability (Sample Entropy ) when white, pink, and brown were administered but not the placebo, and this change was found to be statistically significant. The change in RMS and DFA were found not to be significant across the treatments and the placebo. All regression showed negative slopes except the white regression line for the RMS. Most of the regression lines have very low R2 and non-significant slope values. This was expected due to the low sample size and high variability.

The study shows evidence that subthreshold vibration under the feet has the tendency to increase predictability and reduce sway magnitude. The effect of subthreshold vibration on sway could depend on the initial sway and the type of vibration used.

## Acknowledgements

I would like to express my sincere gratitude to my advisor Dr. Carl Luchies for the continuous support , guidance, motivation and immense knowledge. It's been a pleasure working with you. I'd also like to thank my committee members, Dr. Sara Wilson and Dr. Hauzen Fang, for their cooperation and feedback

My heartfelt gratitude also goes to Camilo Giraldo, Eryn Gerber, Scott Ring, Victoria Blackwood, Jessica Kirchner, Di Bin, and Alex Wilson from my lab. This would not have been possible without your support, comments, suggestions, and feedback. I'm grateful

Last but not least, I'd like to express my gratitude to my family and friends, particularly Marilyn Owusu, Isaac Koduah, and Ashley Washington. Thank you for your encouragement and love.

**Table of Contents**

**List of Figures**

**List of Tables**

# Chapter 1: Introduction

## 1.1 Background and Motivation

Somatosensory deficiency resulting from disease and/or aging reduces the ability to effectively and accurately process input received from the skin [1,2]. This deficiency negatively affects a person's postural control, resulting in increased fall risk and foot injury [3]. Falls become very prevalent as one gets older and significantly worsens after 75 years. Injury due to a fall includes fractures, frequent hospitalizations, increased morbidity, and mortality [4]. Falls are the leading cause of death in older adults in the US (CDC, 2016). Fall-related incidents accounted for about $50.8 billion in health care costs in 2020 (CDC, 2020). These expenses are estimated to rise as this population increases in size.

The body's balance system works through constant position detection, feedback, and adjustment using coordination between the proprioceptive, vestibular, and visual sensory systems and the brain. Proprioception provides a sense of body position and movement of different body parts relative to each other, while the somatosensory provides a sense of physical contact with objects in our external environment through physical contact with our skin (e.g., support surface) [5]. The vestibular system provides information about the motion of the head in space [6]. The visual system provides information about our position and movement concerning surrounding objects. The cerebellum then utilizes all of this sensory feedback information to make appropriate limb and trunk adjustments and balance opposing forces to maintain balance [7]. Aging and many diseases that affect the sensory system, motor control, or the CNS can impair balance, increasing the risk of falls resulting in injuries [8,9] Given the high prevalence of falls in older adults and the adverse effects on the quality of life, there is an urgent need to develop ways to improve balance and prevent falls. A novel intervention to improve sensory deficit and enhance the function of the sensory system utilizes white noise in a process called stochastic resonance (SR)

[4,10,11]. SR enhances the detection of low tactile stimuli putting the system in a better condition to maintain balance and prevent falls. The efficacy of applying white vibration noise on the plantar surface of the foot is well established [1,2,8,12]. No study has analyzed the effects of color on the efficacy of this treatment. All previous studies used either constant or white vibration, which undermines the importance of complexity and predictability of postural sway (PS) in humans. We believe that the treatment efficacy depends on an interplay between the system's initial state and the color of the vibration used to change that state, suggesting a personalized medicine approach. Understanding the relationship between the system's initial state, the color of the subthreshold vibration used as treatment, and the treatment's effect on the system will enable us to tailor vibration parameters to meet individual needs and maximize benefits.

## 1.2 Specific Aims

The specific aims of this study are to determine: (1) the variability of the initial state measures within each participant, across the visits and within each day, across the participants, (2) the effect of different noise color on the magnitude and the direction of change in the state, and (3) if the initial state affects the magnitude and direction in which each treatment color pushes the change in state. We hypothesized that: (1) the sway measures before stimulation will differ across subjects and vary within a subject across the sessions, (2) each vibration color would cause a change in the state different from the placebo, and (3) the efficacy of utilizing subthreshold vibration to change postural sway magnitude, predictability, and complexity depends on their initial values and the color used.

The long-term goal is to demonstrate that the state of balance can be changed in the desired direction using a specific noise color.

## 1.3 Thesis Content

This document is divided into four chapters. The first chapter provides a brief overview of the factors that influence balance and a new intervention that can help improve it. Chapter 2 provides background information on the physiology of balance, factors that contribute to a decline in postural sway, measures of balance, and a promising intervention to improve balance. Chapter 3 contains a manuscript describing the study's background, motivation, methods, results, and discussion of the efficacy of using color in sub-threshold vibration to improve older adult balance. The fourth chapter summarizes the study and makes recommendations for future research.

## References


[1] A. Aboutorabi, M. Arazpour, F. Farahmand, M. Bahramizadeh, R. Fadayevatan, and E. Abdollahi, “Design and evaluation of vibratory shoe on balance control for elderly subjects: technical note,” *Disabil. Rehabil. Assist. Technol.*, vol. 13, no. 2, pp. 173–177, Feb. 2018, doi: 10.1080/17483107.2017.1300346.

[2] M. Costa *et al.*, “Noise and poise: Enhancement of postural complexity in the elderly with a stochastic-resonance–based therapy,” *Europhys. Lett.*, vol. 77, no. 6, p. 68008, Mar. 2007, doi: 10.1209/0295-5075/77/68008.

[3] R. J. Peterka, “Sensorimotor Integration in Human Postural Control,” *J. Neurophysiol.*, vol. 88, no. 3, pp. 1097–1118, Sep. 2002, doi: 10.1152/jn.2002.88.3.1097.

[4] O. White, J. Babič, C. Trenado, L. Johannsen, and N. Goswami, “The Promise of Stochastic Resonance in Falls Prevention,” *Front. Physiol.*, vol. 9, Jan. 2019, doi: 10.3389/fphys.2018.01865.

[5] U. Proske and S. C. Gandevia, “The Proprioceptive Senses: Their Roles in Signaling Body Shape, Body Position and Movement, and Muscle Force,” *Physiol. Rev.*, vol. 92, no. 4, pp. 1651–1697, Oct. 2012, doi: 10.1152/physrev.00048.2011.

[6] A. . Hall, J.E. & Guyton, *Textbook of Medical Physiology (12 ed.). Philadelphia, PA: Saunders Elsevier.(2011)*. 1369.

[7] A. F. Crişan, C. Oancea, B. Timar, O. Fira-Mladinescu, and V. Tudorache, “Balance impairment in patients with COPD,” *PLoS One*, vol. 10, no. 3, pp. 1–11, 2015, doi: 10.1371/journal.pone.0120573.

[8] A. A. Priplata *et al.*, “Noise-enhanced balance control in patients with diabetes and patients with stroke,” *Ann. Neurol.*, vol. 59, no. 1, pp. 4–12, Jan. 2006, doi: 10.1002/ana.20670.

[9] S. C. Regueme *et al.*, “A Therapeutic Insole Device for Postural Stability in Older People With Type 2 Diabetes. A Feasibility Study (SENSOLE Part I),” *Front. Med.*, vol. 6, Jun. 2019, doi: 10.3389/fmed.2019.00127.

[10] A. Priplata, J. Niemi, M. Salen, J. Harry, L. A. Lipsitz, and J. J. Collins, “Noise-Enhanced Human Balance Control,” *Phys. Rev. Lett.*, vol. 89, no. 23, p. 238101, Nov. 2002, doi: 10.1103/PhysRevLett.89.238101.

[11] M. Dettmer, A. Pourmoghaddam, B.-C. Lee, and C. S. Layne, “Effects of aging and tactile stochastic resonance on postural performance and postural control in a sensory conflict task,” *Somatosens. Mot. Res.*, vol. 32, no. 2, pp. 128–135, Apr. 2015, doi: 10.3109/08990220.2015.1004045.

[12] A. M. Galica *et al.*, “Subsensory vibrations to the feet reduce gait variability in elderly fallers,” *Gait Posture*, vol. 30, no. 3, pp. 383–387, Oct. 2009, doi: 10.1016/j.gaitpost.2009.07.005.

# Chapter 2 : Background

## 2.1 Physiology of Balance

Balance is necessary to maintain posture and perform everyday tasks. The body’s center of mass must stay within the base of support, ’limit of stability”, to maintain balance. The "limit of stability" depends on many factors, including the individual's sensory system, the nature of the task at hand, and the type of surface the individual is standing on [1,2]. A working balance system allows the individual to identify the orientation of the body with respect to gravity, determine direction and speed of movements and generate appropriate motor responses to maintain posture and stability in various conditions and activities. The body's balance system works through a constant process of position detection, feedback, and adjustment using coordination between the different sensory systems and the brain. The somatosensory system provides information from the skin and joints (pressure and vibration sensitivity) about body position and movement of different body parts relative to each other and the support surface, the vestibular system provides information about the head and trunk orientation in space, and visual system provides information about our position and movement concerning surrounding objects [3]. The cerebellum then utilizes this information by making limb and trunk adjustments and balancing opposing forces to maintain balance [4]. Also, prior experience and situational clues modify these inputs and help maintain balance. Aging and any disease that affect the sensory system, motor control or the CNS can impair balance increasing the risk of falls and injuries.

### 2.1.1 Vestibular System

The vestibular system provides sensory information about motion, equilibrium, and spatial orientation. It lies within the inner ear and is made of three semicircular canals and two pockets called the otolith organs: the utricle and saccule. The three semicircular canals are positioned at right angles to each other to detect rotational acceleration in any direction. When the head rotates, endolymph fluid in the semicircular canals flows through the canal into an enlarged chamber called the ampulla. The fluid force causes deflection of the cupula, a gelatinous tissue in each of the ampullae, leading to depolarization of the hair cells. The depolarization generates an electrical signal that is transmitted through the vestibular nerve to the cerebellum to be processed[5].

The otolith organs operate similarly to the semicircular canals, but they are designed to detect linear accelerations, including translation (backward/forwards or upwards/downwards) and tilts of the head relative to gravity. The otolith organs contain a sensory tissue called macula with thousands of receptor hair cells. The hair cells extend into a gelatinous membrane with a mass of minute calcium carbonate crystals called otoconia embedded in it. These heavy crystals cause the otolithic membrane to shift relative to the macula when the gravitational field changes leading to the displacement of hair cells. The hair cells become polarized and generate nerve signals that is transmitted through the vestibular nerve to the CNS.

The utricular maculae are oriented in the horizontal plane to help determine the position of the head and its side-to-side tilts during standing, while the saccular macula is positioned vertically to help determine orientation laying down [5].

### 2.1.2 Somatosensory System

The somatosensory system allows the sensation of touch, body position, temperature, and pain using sensory receptors located all over the skin, bones, muscles, and specific internal organs. The somatosensory system includes various somatosensory neurons such as thermoreceptors (detect temperature changes), mechanoreceptors (which detect mechanical stimuli), chemoreceptors (detects changes and is responsive to chemicals), and nociceptors (detects pain). When a physical stimulus triggers a sensory neuron, the neuron creates a nerve impulse that is carried to the spinal cord and finally to the brain using the spinal tracts. Mechanoreceptors are classified into cutaneous and proprioceptive. The cutaneous mechanoreceptors are located close to the skin surface and allow for the sensation of touch and the detection of pressure and vibrations. Four different types of cutaneous mechanoreceptors are specialized to provide information about these stimuli to the brain. Meissner's corpuscles, Pacinian corpuscles, Merkel's disks, and Ruffini's corpuscles

The Merkel's disks are slow adapting nerve fibers with a small receptive field that are sensitive to touch. They provide information about pressure and texture. Ruffini corpuscles are also slow adapting spindle-shaped receptors that are sensitive to skin stretch and contribute to finger position and movement. Meissner corpuscles are fast-adapting receptors that are sensitive to low-frequency vibration (30-50Hz) when objects move across the skin. Pacinian corpuscles are fast adapting and detect high-frequency vibration. All the four mechanoreceptor types can be found in all areas of the skin, but they are more sensitive at the non-hairy(glabrous) parts [6]. At the feet, the boundary between the body and support surface, the mechanoreceptors play an essential role in balance. They detect the pressure distribution of the feet relative to the support surface and provide the CNS with accurate information to make the necessary adjustments to remain balanced.

Proprioception enables one to have conscious awareness of posture and movement of one's own body, particularly the limbs and head. The purpose of proprioceptors is primarily to give detailed and continuous information about the limb's position and other body parts in space. The proprioceptive system contributes to joint position, joint motion sense, and kinesthesia using information from the Golgi tendon organs and muscle spindle. The Golgi tendon organs lie at the interface of muscles and tendons and are sensitive to muscle tension.

The muscle spindles are found in all but a few striated muscles and are made of intrafusal muscle fibers. They provide information about changes in muscle length. The muscle spindle detects changing tensions as the muscle expands and contracts to allow the perception of body velocity [5].

Some cutaneous mechanoreceptors also contribute to proprioception. Pacinian corpuscle and Ruffini's endings are responsible for sensing joint angle and position (where the joint is at a specific location). Using this information from the proprioceptive system, the CNS can monitor our position in space and make the necessary corrections to maintain balance.

### 2.1.3 Visual System

The visual system provides information about the body in relation to the surroundings. It provides cues to identify how a person is oriented relative to other objects, avoid obstacles, and maneuver through the environment. It works by converting light into electrical signals, which are then sent to the brain to be interpreted as images. When electromagnetic waves enter the cornea in the form of light, it is focused by the cornea to pass through the pupil. The light then passes through the lens, which works together with the cornea to refract the light to focus on the retina at the back of the eye. The retina contains photoreceptors called cones and rods, which turn the

light into electrical signals. The cones are responsible for daytime vision and colors, and the rods are responsible for night vision. The electrical signals are then transmitted through the optic nerve to the visual cortex of the brain to be processed. The CNS then uses this information to process a lot of tasks, including balancing.

## 2.2 Conditions Causing Balance Deficits

### 2.2.1 Aging

Aging is associated with a progressive decline in balance. Maintaining postural balance during standing is dependent on descending commands from the central nervous system (CNS) and also on the availability, accuracy, and reliability of somatosensory inputs (from muscle, joint, skin, and pressure receptors) and visual and vestibular inputs[7]. Unfortunately, most functioning of these systems declines with age. For example, vision continuously worsens after the age of 50[8], with a lot of physiological changes taking place in the eye. This results in the deterioration of many visual processes such as contrast sensitivity[9,10], visual acuity[11,12], glare sensitivity, dark adaptation, accommodation, and depth perception[13]. Older people are also susceptible to developing reduce visual function from common eye pathologies, including cataracts, macular degeneration, and glaucoma [14]. With reduced visual function, balance control and ability to navigate the environment become impaired due to misinterpretation of spatial information.

The vestibular function also reduces with normal aging [15]. Previous studies have shown that the number of labyrinthine hair receptors gradually starts decreasing as early as age 40 [16]. This is followed by a sharp decline in the number of vestibular receptors by ages 55 to 63, and only 60% of the nerve cells of the vestibular system remain by age 70 [17]. The decline of the nerve cells in the vestibular system affects balance.

There is also structural and functional declination of the somatosensory system as one becomes older. Aging causes the muscle spindle to change morphologically. There are changes such as increased capsular thickness and a decreased number of intrafusal fibers [18], which may account for the reduced static and dynamic sensitivity of the muscle fibers [19]. Also, aging results in a reduction in the number of Pacinian [20] and Meissner's receptors [21] leading to a decrease in vibration perception [22,23] and touch threshold [24].

In addition to the reduced function of the various sensory system in older adults, there is also a decline in the function of the CNS, which may affect the ability to integrate sensory information and also to compensate for discordant sensory input. This all affects their postural control process and causes a decline in balance performance resulting in falls and injuries.

Fall becomes very prevalent as one gets older and progressively deteriorates after 75 years. The consequences include fractures, frequent hospitalizations, increased morbidity, and mortality [25]. Falls are the leading cause of death in older adults in the US (CDC, 2016). Falls may negatively affect the individual physically, psychologically, emotionally, and financially. Economically, fall-related incidents accounted for about $50.8 billion in health care costs in 2020 (CDC, 2020). These expenses are estimated to rise each year.

### 2.2.2 Type II Diabetes

Diabetes mellitus is a chronic metabolic condition characterized by high blood glucose levels as a result of the body's inability to produce insulin, resist insulin action, or both [26].In 2020, about 34.2 million people in the US were estimated to have diabetes, with approximately 90-95 of them having type 2 diabetes (Center for Disease Control 2021). Type 2 diabetes can often cause changes that affect the sensory systems [27,28]. Prolonged hyperglycemia (high blood glucose)

associated with type 2 diabetes can lead to complications that affect peripheral nerves (resulting in diabetic peripheral neuropathy [DPN]), the retina (resulting in diabetic retinopathy), and the vestibular system [29,30]. DPN often damages the nerves in the legs and feet. The deterioration of the nerves causes a reduction in sensation and control of the lower limb, making it unable to detect balance changes and make the necessary adjustments to avoid falls. This phenomenon is widely associated with an increased risk of falling in people who have type 2 diabetes [31]. Diabetic retinopathy is caused by damage to the blood vessels in the retina. Blood vessel damage and the accumulative tissue scar in the retina can result in blurred central vision and reduced contrast sensitivity. Diabetic retinopathy can also lead to other eye conditions such as cataracts and glaucoma that may result in blindness. Also, diabetes can reduce sensitivity in the vestibular system, which can affect perception about motion equilibrium and spatial orientation required to maintain balance [29]. People with diabetes suffer complications causing loss of touch, visual acuity, contrast sensitivity, and depth perception that affect their balance and increase the risk of falls.

### 2.2.3 Multiple sclerosis

Multiple sclerosis is a chronic disease that affects the CNS. It is one of the most common progressive neurological diseases and the leading cause of non-traumatic neurological disability in young adults worldwide. It affects more women than men. The causes of MS are not yet well understood, but it is believed to be a combination of environmental, infectious, immunologic, or genetic. Its symptoms differ from patient to patient and may include limb weakness, numbness, vision loss, fatigue, cognitive difficulties, unsteady gait, and balance problems [32]. The severity of each attack may also be different. It may be severe or mild, short term or long term. In MS,

the immune system attacks the myelin and causes the breakdown of the myelin sheaths that covers the axons, long threadlike part of the nerve cell that transmit electrical nerve impulses, leaving scars called lesions or sclerosis. This can eventually cause the nerve themselves to deteriorate or damage permanently. Myelin sheaths protect the axons and facilitate the conduction of nerve impulses, so their breakdown causes slower conduction or complete blockages when the exposed axons are deteriorated [33].

MS can also cause dizziness and vertigo [34], which affects balance and increases the risk of falls.

### 2.2.4 Parkinson Disease

Parkinson's disease is a neurogenerative disorder caused by the degeneration of dopaminergic neurons in the brain. It is the 2$^{nd}$ most common neurodegenerative disorder and affects about 1 to 2% of older adults above 60 [35]. In older adults of 85 years and above, the number increases to 3 to 5% [36]. Men are 1.5 times likely to get PD than women [37].In PD, there is a loss of dopamine cells in the basal ganglia, the part of the brain responsible for processing sensory, motor, and cognitive information [38]. The loss of these cells results in the reduced activity in the motor cortex. PD affects people differently, and the symptoms worsen as the disease progresses. PD symptoms are mostly motor-based and include loss of automatic movements(akinesia), slowed movement(bradykinesia), increased muscle rigidity, and resting tremor [39]. This makes it difficult to walk and also maintain balance and coordination. There are also nonmotor symptoms of PD, which include cognitive abnormalities, sleep disorders, autonomic dysfunction, dementia, and pain [40].

PD patients have also been shown to have a significantly increased tactile and thermal threshold, reduced mechanical perception of pain, and a significant loss of Meissner corpuscle [41]. Apart from the sensory deficit stated above, other studies have also shown that PD patients have problems integrating proprioceptive information [42]. PD impact on the CNS results in a delay of a person's reaction time, postural reflexes, and movement speed, making it challenging to respond swiftly to perturbations.

## 2.3 Postural Sway

During upright stand, there is a constant need to make small adjustments to prevent loss of balance. These movements are known as Postural sway, and they reflect the time course of a central point of the body (center of mass, gravity, or pressure) during quiet standing [43]. It represents the interplay between the stabilizing force acting on the body and the adjustment by the postural control system to remain balanced. Postural sway allows for the observation of the combined effect of the visual, somatosensory, and vestibular systems and can be analyzed by various parameters independent of individual body characteristics providing vital information for clinical and research purposes.

## 2.4 Center of Pressure

Center of Pressure (COP) is the point of application of the ground reaction force vector and its time series is often used to assess postural sway. It represents the weighted average of all the pressures over the body in contact with the ground. It is calculated using the force and moment measurements obtained from one or more force plates attached to the ground. When a person

stands on the force plate, it detects changes in the ground reaction force as the body sways. The COP is determined by dividing the sum of moments in each axis by the overall vertical force [44]. The COP position is calculated using the following equations.

$$COP_{AP} = \frac{M_{ML} + F_{AP} * \mathrm{dz}}{F_Z} \qquad COP_{ML} = \frac{M_{AP} + F_{ML} * \mathrm{dz}}{F_Z}$$

The F and M represent the force and moment, respectively. AP represents the anterior-posterior axis, ML represents the medial-lateral axis, and Z represents the vertical axis. Lastly, dz represents the distance between the force plate surface and the origin of the coordinate system provided by the force plate manufacturer.

The COP values for the ML and AP are calculated for each step in the time series. This results in a two-dimensional path depicting where the COP travels during postural sway. The complex fluctuation of COP can be represented as a two-dimensional stochastic process [45].

Characterizing the trajectory of COP is of crucial importance for understanding neural mechanisms of postural control [46,47] and also for diagnosing the severity of neurological diseases with postural instability better [48,49,50]. The COP time series can be characterized by using parameters (linear and nonlinear) which help assess changes in postural control.

## 2.5 Linear Measures

Several linear measures have been used to quantify COP and to assess postural instability. Some of these measures allow assessment in the AP and ML direction and some by the magnitude of the COP time series. The magnitude is determined by taking the square root of the sum of

squares of the AP and ML values. Some of the COP linear measures include the maximum displacement, path length, mean displacement, velocity, and the root mean square. Maximum Amplitude is the absolute maximum displacement from the mean. It allows assessment in both the AP and ML direction. An increase in the maximum Amplitude corresponds to a decrease in postural stability [51]. The COP velocity represents the total displacement traveled by the COP over time. An increase in COP velocity corresponds to a decreased ability to control posture. The COP velocity was significantly higher in the MS group than in healthy subjects [52]. The path length has also been shown to be higher for MS patients in the AP and ML direction than for healthy subjects [52]. Fallers had a significantly higher maximum displacement than non-fallers in the AP and ML direction [51]. The RMS was significantly greater in the DPN group in the ML and AP direction than in the healthy subjects [28].

## 2.6 Nonlinear Analysis

In addition to the linear measures, the COP can be characterized using nonlinear measures. It is assumed that the COP dynamics are generated by nonlinear high dimensional dynamical systems, so using nonlinear measures seems appropriate [53]. Nonlinear measures consider the temporal characteristics of sway and effectively describe the qualitative changes in postural control dynamics. They take into account the irregular and nonstationarity of the COP time series. The nonlinear analysis offers a way to extract information relating to complexity, stability, and variability, which can be used for clinical assessment. Nonlinear measures that have been successfully used to characterize COP include Entropy [54,55,56,57], DFA [54,58,59,60], and Lyapunov exponent [58,61].

### 2.6.1 Entropy

The concept of entropy was first introduced by Rudolph Clausius (1867) in classical thermodynamics. It was developed to help understand Carnot's work about the limits of mechanical work that could be produced by steam engines. It quantifies the energy in the system that is unavailable for doing work. In 1896, Ludwig Boltzmann provided more insight into the concept of entropy by using probability concepts to describe entropy (estimate the variability of positions and velocities) on a molecular scale. Statistical thermodynamics views entropy as the amount of microscopic variability that a system has for a given microscopically state. Boltzmann entropy conceptualizes the measure of randomness of the system and can be quantified using the following equation.

$$S = -k \sum_{i}^{N} p_i \, In \, p_i$$

In 1949, Claude Shannon used the concept of entropy in information theory. He was interested in determining the information content of a signal and how much it could be compressed [62]. The Shannon entropy "Hs" is calculated using the following equation:

$$H_s = -\sum_{i}^{N} p_i \; l \, og \, p_i$$

The Shannon entropy equation looks similar to Boltzmann's but no Boltzmann constant to convert the units since the units are bits of information. If a signal has low entropy, it means it is more predictable with repeating patterns. On the other hand, a high entropy signal means a less predictable signal with highly random outcomes [63].

The concept of entropy has various interpretations depending on the context in which it is used. The entropy of a time series measures how difficult to compress the time series by finding recurring patterns in it. In biological time series, entropy provides us a way to quantify the unpredictability or irregularity of the times series. It provides us with the probability that similar behavioral patterns will not be followed by similar patterns.

### 2.6.2 Entropy for Data Analysis

Various algorithms have been used to estimate entropy. One of the earliest to be derived from Shannon entropy was the Kolmogorov entropy. It was ideal for quantifying chaos in a dynamic system and was exclusive for long time series [64]. Later other variations of Kolmogorov entropy were developed for specific data types, including biological signals. It has been used to analyze normal heart rate and ECG data of dementia and Parkinson's patients [65,66,67]. Many derivations of Kolmogorov entropy have also been developed. One of them is the Eckmann-Ruelle entropy [68].

There is also permutation entropy which is appropriate for chaotic time series. It uses the differences between the data points to find the predictability of a signal's derivative [69]. Spectral entropy, a derivation of Shannon entropy, uses the spectral power density of the time series. The most popular entropy method is approximate entropy, and other entropies such as sample entropy and multiscale entropy are a buildup of the approximate entropy.

### 2.6.2.1 Approximate Entropy

Approximate entropy (AppEn) was developed by Steve Pincus in 1991 to analyze experimental data generated by biological processes. The method for estimation of entropy of a time series at that time was not suited for the analysis of short and noisy signals produced by biological systems. To find the actual entropy of a system, one requires infinite time series. Therefore, the name approximate was used to emphasize that the measure is an approximate of the actual entropy based on a time series of finite length. To calculate the AppEn of a time series, the algorithm takes as input two parameters: tolerance radius r and vector length m (sometimes called the pattern length or the segment length). For a time series of N data points,

$u(i): 1 \leq i \leq N$ forms a vector sequence x(1) ,x(2).....x(N-m+1)

$X_i$ = {u(i), u(i+1),.....u(i+m-1)} for $1 \leq i \leq N - m + 1$

The vector sequence is duplicated into identical sequences $X_i$ and $X_j$. The distance between the two vectors is defined as

$$d[x(i), x(j)] = max_{k=1,2....m}(|u(i + k - 1) - u(j + k - 1)|)$$

$$C_i^m(r) = \left(number\ of\ j \leq N - m + 1\ such\ that \frac{d[x(i),x(j) \leq r)}{N-m+1}\right)$$

$C_i^m\ (r)$ is the probability that any vector $x_m\ (j)$ is within r of $x_m\ (i)$

$$C_r^m(i) = (N - m + 1)^{-1} \sum_{j=1}^{N-m+1} \theta(d_{ij}^m - r)$$

Where θ is the Heaviside function, $\theta(z) = \begin{cases} 1\ if\ z\ \leq 0 \\ 0\ if\ z\ > 0 \end{cases}$

$$\varphi^m(r) = (N - m + 1)^{-1} \sum_{i=1}^{N-m+1} In\ [C_r^m(i)]$$

$$AppEn(m, r) = \lim_{N \to \infty} [\varphi^m(r) - \varphi^{m+1}(r)]$$

The equation takes the limit of N to infinity. Therefore, an approximation is made to work on finite time series by getting rid of the limit altogether.

$$AppEn(m,r) = [\varphi^m(r) - \varphi^{m+1}(r)$$

AppEn is approximately equal to the negative average natural logarithm of the conditional probability that two subseries of length m that are similar (within a tolerance r) remain similar for subseries of length m+1. AppEn values are unitless and range from 0 to 2. An AppEn value of 0 corresponds to a perfectly predictable signal (e.g. Periodic signal), whereas an AppEn value of 2 represents a perfectly random signal (i.e. white noise). Thus, the higher the probability, the smaller the AppEn value indicating less complexity.

One limitation of AppEn is that it creates a bias towards regularity. It counts each vector similar to itself, therefore, no matter how random the signal is, every vector will have at least one match. The bias influence becomes greater as the length of the time series decreases [70]. In addition, ApEn also lacks relative consistency. AppEn values may flip if the input parameters are changed. A white noise may result in a higher AppEn value than a sine wave for some value of r, and this may flip (that is, the AppEn value of the sine becomes bigger than the random noise) for a different value of r [65,71].

### 2.6.2.2 Sample Entropy

SampEn was introduced as an improved version of AppEn entropy. Richman and Moorman developed it to overcome the regularity bias observed in AppEn. SampleEn takes the logarithm of the sum of all the conditional probabilities rather than taking the logarithm of individual conditional probabilities and adding them together as encountered in AppEn. This reduces the risk of getting log(0) during SampleEn calculations. To calculate the SampleEn of a time series, the algorithm requires the same two input parameters as AppEn: tolerance radius r and vector length m. The SampleEn algorithm is as follows

$$B_i^m(r) = (N-m-1)^{-1} * \sum_{j=1, j\neq i}^{N-m} \theta(d_{ij}^m - r)$$

$$A_i^m(r) = (N-m-1)^{-1} * \sum_{j=1, j\neq i}^{N-m} \theta(d_{ij}^m - r)$$

$$B^m(r) = (N-m-1)^{-1} * \sum_{j=1, j\neq i}^{N-m} B_i^m(r)$$

$$A^m(r) = (N-m-1)^{-1} * \sum_{j=1, j\neq i}^{N-m} A_i^m(r)$$

$$SampEn(m,r) = -In\frac{A^m(r)}{B^m(r)}$$

$B^m(r)$ is then the probability that two sequences will match for m points, whereas $A^m(r)$ is the probability that two sequences will match for m + 1 points.

For a perfectly repeatable time series, the SampEn calculates to zero but converges towards infinity the closer the signal is to random noise [65]. In addition to eliminating regularity bias

SampEn has been shown to be independent of data length and also maintains relative consistency. Yentes et al. [72] found that SampEn was less sensitive to changing parameters r or m than AppEn when analyzing gait data

### 2.6.3 Fractal Analysis

Fractal analysis is another measure for assessing the complexity of a dynamic system. It is used to describe patterns occurring in signals known to exhibit self-similar patterns [73].

These self-similar patterns are an indication of the repeating trends in the data series. Many physiological processes, including heart rate, blood pressure, EEG potentials, stride interval, and center of pressure displacement, have been shown to exhibit fractal-like behavior [74]. Fractal analysis produces a qualitative measure known as fractal dimension and has been proven to successfully differentiate healthy from disease function. Thus, changes in the fractal dimension can be used to represent changes in health.

There are two types of fractal processes: fractional Gaussian noise(fGn) based on white noise and fractional Brownian motion(fBm) based on Brownian motion. fGn is stationary with constant mean value and variance over time, while fBm is nonstationary with time-dependent variance [60]. Though the two processes have fundamentally different properties, they are related signals. The cumulative sum of fGn will exhibit fBm signal properties. The type of process the data represent affects how to go about calculating the fractal scaling.

Mandelbrot and van Ness proposed a continuum of processes between fGn and fBm [75]. A scaling law characterizes this continuum, in the frequency domain, relating power to frequency: $S(f) \propto f^{-\beta}$ (1) This scaling law can be revealed by the log-log plot of the power spectrum, which is characterized by a linear shape of slope $-\beta$. The value of the scaling exponent indicates where the empirical data lies on the "noise" continuum. This method is known as the power spectral density. Other methods have also been proposed for estimating the exponent that governs the scaling of fluctuations in experimental time series. One of these methods is the detrended fluctuation analysis, which produces a scaling exponent that is linearly related to the power exponent( $B= 2\alpha-1$ ). The scaling exponents $\alpha$ and $\beta$ can both be used to determine the Hurst component H, which is used to characterize fractal signals. The Hurst component (H) ranges from 0 to 1. $H<0.5$ represents anti-persistent series which means an increasing trend in the past is likely to be followed by a decreasing trend in the future. Conversely, $H> 0.5$ indicates a persistent series. This means an increasing trend in the past is likely to be followed by an increasing trend in the future. A Hurst component equal to 0.5 indicates a random walk, which implies that it is not possible to predict future data based on past data.

### 2.6.3.1 Detrended Fluctuation Analysis

Detrended Fluctuation Analysis is a fractal scaling method introduced by Peng et al. it was designed to detect long-range correlation in nonstationary time series [76]. DFA has the advantage of avoiding spurious detection of apparent self-similarity, which may be artifacts of nonstationarity [77]. It has been applied in different areas of interest, including DNA [78,79], human gait [80,81] cloud structure [82,83] neuron spiking [84,85] and economic time series

[86,87]. The main output of the DFA analysis is the scaling exponent α which is related to the Hurst exponent[88]

The DFA can be calculated using the following steps [60]

1. The signal is integrated using the equation.
2. The integrated time series is then partitioned into non-overlapping windows of equal length(n)
3. The data in each box is fitted with a first-order least square line. The line represents the trend inside each window. The y coordinate of each trend can be expressed as $y_n(k)$
4. The integrated time series, y(k) is then detrended by subtracting the local trend,$y_n(k)$, for each window.
5. Next, the RMS fluctuation of the integrated and detrended time series can be calculated using the equation.

This computation is repeated for all time scales (box sizes)

6. A log-log graph of F(n) versus n is constructed.

   If there is a clear relationship between them, the slope of the straight-line fit to the log-log plot of F(n) versus n is the scaling exponent α. The correlation properties of the time series are characterized based on the scaling exponent $\alpha$. If $\alpha = 0.5$, then the time series is uncorrelated (random process). In the case when $\alpha < 0.5$, the time series is anticorrelated, and for $\alpha > 0.5$, the time series contains positive correlations.

## 2.7 Stochastic Resonance

Stochastic resonance is a phenomenon where the response of a nonlinear system to a weak signal is better with noise than when noise is absent [89]. That is, the addition of a particular level of noise enhances the detection of a weak stimulus or makes the information content of the signal better. The stimuli on their own are below the threshold, making them unable to be detected. When noise is added to the stimulus, there is a threshold crossing making the stimuli be detected and characterized. To ensure stochastic resonance, an optimal amount of noise needs to be added. A large intensity of added noise only degrades the information content of the signal. Stochastic resonance does not occur in linear systems. In linear systems addition of noise to the system or stimuli results in degradation of the signal quality. For stochastic resonance to occur, there has to be a threshold, subthreshold stimulus, and an optimal amount of noise.

The sensitivity of biological sensors (cutaneous mechanoreceptors in hands and feet, hair cells in the inner ear, proprioceptors in the muscle and joints) to a stimulus can be expressed as absolute thresholds. The Amplitude of the stimuli must be above the absolute threshold to elicit a response to the CNS.

When the mechanoreceptors are damaged, their ability to elicit responses above the sensing threshold declines, affecting feedback loops. Superficial random pattern noise, when combined with the signal, can provide a sense of "helpful randomness [90]" that allows the system to cross the threshold.

While the exact neurophysiological process of how the noise augments weakened sensorimotor pathways is still unknown, the leading theory explains that deficient afferent neurons exist in a hyperpolarized state away from the firing threshold. When the noise is applied, it partially

depolarizes the deficient afferent neurons back to a healthy polarization level close to the firing threshold.

The unhealthy afferent acquires enhanced (healthy/normal) sensitivity as it comes back to the normal resting potential allowing the sensorimotor pathways to resume necessary communication for maintaining balance. Different systems will require a different set of noise characteristics (i.e., frequency content, magnitude) to be effective. The noise must be perceived as physiologically "relevant" by the system for the noise to facilitate optimum system performance. Stochastic resonance can make the signal detectable again, putting the system in a better condition to maintain balance and prevent falls [91]. Stochastic resonance can be applied to different physiological systems and has been shown to enhance the detection of low tactile stimuli. It can be achieved by electrical, magnetic, or mechanical(vibratory) simulation. In addition, the input magnitude can be below the human's sensing threshold (subthreshold) or above their sensing threshold (suprathreshold).

## 2.8 Specific Aims

The specific aims of this study are to determine: (1) the variability of the initial state measures within each participant, across the visits and within each day, across the participants, (2) the effect of different noise color on the magnitude and the direction of change in the state**,** and (3) if the initial state affects the magnitude and direction in which each treatment color pushes the change in state. We hypothesized that: (1) the sway measures before stimulation will differ across subjects and vary within a subject across the sessions, (2) each vibration color would cause a change in the state different from the placebo, and (3) the efficacy of utilizing

subthreshold vibration to change postural sway magnitude, predictability, and complexity depends on their initial values and the color used.

The long-term goal is to demonstrate that the state of balance can be changed in the desired direction using a specific noise color.

## References


[1] A. L. Hof, M. G. J. Gazendam, and W. E. Sinke, “The condition for dynamic stability,” *J. Biomech.*, vol. 38, no. 1, pp. 1–8, 2005, doi: 10.1016/j.jbiomech.2004.03.025.

[2] M. D. McDonnell and L. M. Ward, “The benefits of noise in neural systems: Bridging theory and experiment,” *Nat. Rev. Neurosci.*, vol. 12, no. 7, pp. 415–425, 2011, doi: 10.1038/nrn3061.

[3] U. Proske and S. C. Gandevia, “The proprioceptive senses: Their roles in signaling body shape, body position and movement, and muscle force,” *Physiol. Rev.*, vol. 92, no. 4, pp. 1651–1697, 2012, doi: 10.1152/physrev.00048.2011.

[4] A. F. Crişan, C. Oancea, B. Timar, O. Fira-Mladinescu, and V. Tudorache, “Balance impairment in patients with COPD,” *PLoS One*, vol. 10, no. 3, pp. 1–11, 2015, doi: 10.1371/journal.pone.0120573.

[5] A. . Hall, J.E. & Guyton, *Textbook of Medical Physiology (12 ed.). Philadelphia, PA: Saunders Elsevier.(2011)*. 1369.

[6] A. Zimmerman, L. Bai, and D. D. Ginty, “of Mammalian Skin,” *Science (80-. ).*, vol. 346, no. 6212, pp. 950–954, 2014.

[7] A. Kavounoudias, C. A. R. Roll, R. Roll, and J.-P. Roll, “1998-plantar.sole.balance.control-NeuroReport.pdf,” *Neuroreport*, vol. 9, no. 14, pp. 3247–3252, 1998, [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/9831459%0Ahttps://insights.ovid.com/crossref?an=00001756-199810050-00021.

[8] N. S. Gittings and J. L. Fozard, “Age related changes in visual acuity,” *Exp. Gerontol.*, vol. 21, no. 4–5, pp. 423–433, Jan. 1986, doi: 10.1016/0531-5565(86)90047-1.

[9] S. R. Lord, R. D. Clark, and I. W. Webster, “Postural Stability and Associated Physiological Factors in a Population of Aged Persons,” *J. Gerontol.*, vol. 46, no. 3, pp. M69–M76, May 1991, doi: 10.1093/geronj/46.3.M69.

[10] S. R. Lord, D. McLean, and G. Stathers, “Physiological Factors Associated with Injurious Falls in Older People Living in the Community,” *Gerontology*, vol. 38, no. 6, pp. 338–346, 1992, doi: 10.1159/000213351.

[11] S. R. Lord, R. D. Clark, and I. W. Webster, “Physiological Factors Associated with Falls in an Elderly Population,” *J. Am. Geriatr. Soc.*, vol. 39, no. 12, pp. 1194–1200, Dec. 1991, doi: 10.1111/j.1532-5415.1991.tb03574.x.

[12] C. I. A. Jack, T. Smith, C. Neoh, M. Lye, and J. N. McGalliard, “Prevalence of Low Vision in Elderly Patients Admitted to an Acute Geriatric Unit in Liverpool: Elderly People Who Fall Are More Likely to Have Low Vision,” *Gerontology*, vol. 41, no. 5, pp. 280–285, 1995, doi: 10.1159/000213695.

[13] M. C. Nevitt, S. R. Cummings, S. Kidd, and D. Black, “Risk factors for recurrent nonsyncopal falls. A prospective study.,” *JAMA*, vol. 261, no. 18, pp. 2663–8, May 1989, [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/2709546.

[14] H. A. Kahn *et al.*, “The Framingham Eye Study. I. Outline and major prevalence findings.,” *Am. J. Epidemiol.*, vol. 106, no. 1, pp. 17–32, Jul. 1977, doi: 10.1093/oxfordjournals.aje.a112428.

[15] T. D. Fife and R. W. Baloh, “Disequilibrium of unknown cause in older people,” *Ann. Neurol.*, vol. 34, no. 5, pp. 694–702, Nov. 1993, doi: 10.1002/ana.410340512.

[16] P. F. Smith, “Age-Related Neurochemical Changes in the Vestibular Nuclei,” *Front. Neurol.*, vol. 7, Mar. 2016, doi: 10.3389/fneur.2016.00020.

[17] J. C. Alvarez *et al.*, “Neuronal loss in human medial vestibular nucleus,” *Anat. Rec.*, vol. 251, no. 4, pp. 431–438, 1998, doi: 10.1002/(SICI)1097-0185(199808)251:4<431::AID-AR2>3.0.CO;2-V.

[18] M. Swash and K. P. Fox, “The effect of age on human skeletal muscle studies of the morphology and innervation of muscle spindles,” *J. Neurol. Sci.*, vol. 16, no. 4, pp. 417–432, Aug. 1972, doi: 10.1016/0022-510X(72)90048-2.

[19] T. Miwa, Y. Miwa, and K. Kanda, “Dynamic and static sensitivities of muscle spindle primary endings in aged rats to ramp stretch,” *Neurosci. Lett.*, vol. 201, no. 2, pp. 179–182, Dec. 1995, doi: 10.1016/0304-3940(95)12165-X.

[20] N. CAUNA and G. MANNAN, “The structure of human digital pacinian corpuscles (corpus cula lamellosa) and its functional significance.,” *J. Anat.*, vol. 92, no. 1, pp. 1–20, Jan. 1958, [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/13513492.

[21] C. F. Bolton, R. K. Winkelmann, and P. J. Dyck, “A quantitative study of Meissner’s corpuscles in man,” *Neurology*, vol. 16, no. 1, pp. 1–1, Jan. 1966, doi: 10.1212/WNL.16.1.1.

[22] J. Pearson and V. Sensibility, “Supply Spinal,” 2015.

[23] R. T. Verrillo, S. J. Bolanowski, and G. A. Gescheider, “Effect of aging on the subjective magnitude of vibration,” *Somatosens. Mot. Res.*, vol. 19, no. 3, pp. 238–244, Jan. 2002, doi: 10.1080/0899022021000009161.

[24] S. D. Perry, “Evaluation of age-related plantar-surface insensitivity and onset age of advanced insensitivity in older adults using vibratory and touch sensation tests,” *Neurosci. Lett.*, vol. 392, no. 1–2, pp. 62–67, Jan. 2006, doi: 10.1016/j.neulet.2005.08.060.

[25] O. White, J. Babic, C. Trenado, L. Johannsen, and N. Goswami, “The promise of

stochastic resonance in falls prevention," *Frontiers in Physiology*, vol. 10, no. JAN. 2019, doi: 10.3389/fphys.2018.01865.

[26] "WHO_NCD_NCS_99.2.pdf." .

[27] K. M. Sibley, M. K. Beauchamp, K. Van Ooteghem, S. E. Straus, and S. B. Jaglal, "Using the Systems Framework for Postural Control to Analyze the Components of Balance Evaluated in Standardized Balance Measures: A Scoping Review," *Arch. Phys. Med. Rehabil.*, vol. 96, no. 1, pp. 122-132.e29, Jan. 2015, doi: 10.1016/j.apmr.2014.06.021.

[28] H. Corriveau *et al.*, "Evaluation of postural stability in elderly with diabetic neuropathy," *Diabetes Care*, vol. 23, no. 8, pp. 1187–1191, Aug. 2000, doi: 10.2337/diacare.23.8.1187.

[29] P. Hewston and N. Deshpande, "Falls and Balance Impairments in Older Adults with Type 2 Diabetes: Thinking Beyond Diabetic Peripheral Neuropathy," *Can. J. Diabetes*, vol. 40, no. 1, pp. 6–9, Feb. 2016, doi: 10.1016/j.jcjd.2015.08.005.

[30] N. E. Cameron, S. E. M. Eaton, M. A. Cotter, and S. Tesfaye, "Vascular factors and metabolic interactions in the pathogenesis of diabetic neuropathy," *Diabetologia*, vol. 44, no. 11, pp. 1973–1988, Nov. 2001, doi: 10.1007/s001250100001.

[31] H. Salsabili, F. Bahrpeyma, B. Forogh, and S. Rajabali, "Dynamic stability training improves standing balance control in neuropathic patients with type 2 diabetes," *J. Rehabil. Res. Dev.*, vol. 48, no. 7, pp. 775–786, 2011, doi: 10.1682/JRRD.2010.08.0160.

[32] M. H. Cameron and S. Lord, "Postural Control in Multiple Sclerosis: Implications for Fall Prevention," *Curr. Neurol. Neurosci. Rep.*, vol. 10, no. 5, pp. 407–412, Sep. 2010, doi: 10.1007/s11910-010-0128-0.

[33] E. M. Frohman, M. K. Racke, and C. S. Raine, "Multiple Sclerosis — The Plaque and Its Pathogenesis," *N. Engl. J. Med.*, vol. 354, no. 9, pp. 942–955, Mar. 2006, doi:

10.1056/NEJMra052130.

[34] J. H. Pula, D. E. Newman-Toker, and J. C. Kattah, “Multiple sclerosis as a cause of the acute vestibular syndrome,” *J. Neurol.*, vol. 260, no. 6, pp. 1649–1654, Jun. 2013, doi: 10.1007/s00415-013-6850-1.

[35] C. W. Olanow, M. B. Stern, and K. Sethi, “The scientific and clinical basis for the treatment of Parkinson disease (2009),” *Neurology*, vol. 72, no. Issue 21, Supplement 4, pp. S1–S136, May 2009, doi: 10.1212/WNL.0b013e3181a1d44c.

[36] G. Alves, E. B. Forsaa, K. F. Pedersen, M. Dreetz Gjerstad, and J. P. Larsen, “Epidemiology of Parkinson’s disease,” *J. Neurol.*, vol. 255, no. S5, pp. 18–32, Sep. 2008, doi: 10.1007/s00415-008-5004-3.

[37] A. J. Lees, J. Hardy, and T. Revesz, “Parkinson’s disease,” *Lancet*, vol. 373, no. 9680, pp. 2055–2066, Jun. 2009, doi: 10.1016/S0140-6736(09)60492-X.

[38] D. Centonze, “Neurophysiology of Parkinson’s disease: from basic research to clinical correlates,” *Clin. Neurophysiol.*, vol. 110, no. 12, pp. 2006–2013, Dec. 1999, doi: 10.1016/S1388-2457(99)00173-X.

[39] M. W. Anne Shumway-Cook, *Motor control : translating research into clinical practice*, 4th ed. Philadelphia : Wolters Kluwer Health/Lippincott Williams & Wilkins, c2012., 2012.

[40] K. R. Chaudhuri, D. G. Healy, and A. H. Schapira, “Non-motor symptoms of Parkinson’s disease: diagnosis and management,” *Lancet Neurol.*, vol. 5, no. 3, pp. 235–245, Mar. 2006, doi: 10.1016/S1474-4422(06)70373-8.

[41] M. Nolano *et al.*, “Sensory deficit in Parkinson’s disease: evidence of a cutaneous denervation,” *Brain*, vol. 131, no. 7, pp. 1903–1911, Jul. 2008, doi:

10.1093/brain/awn102.

[42] N. Patel, J. Jankovic, and M. Hallett, “Sensory aspects of movement disorders,” *Lancet Neurol.*, vol. 13, no. 1, pp. 100–112, Jan. 2014, doi: 10.1016/S1474-4422(13)70213-8.

[43] D. Winter, “Human balance and posture control during standing and walking,” *Gait Posture*, vol. 3, no. 4, pp. 193–214, Dec. 1995, doi: 10.1016/0966-6362(96)82849-9.

[44] D. A. Winter, *Biomechanics and Motor Control of Human Movement*. Toronto: Wiley, 1990.

[45] P. J. Loughlin, M. S. Redfern, and J. M. Furman, “Time-varying characteristics of visually induced postural sway,” *IEEE Trans. Rehabil. Eng.*, vol. 4, no. 4, pp. 416–424, Dec. 1996, doi: 10.1109/86.547944.

[46] D. A. Winter, “Biomechanics of Normal and Pathological Gait,” *J. Mot. Behav.*, vol. 21, no. 4, pp. 337–355, Dec. 1989, doi: 10.1080/00222895.1989.10735488.

[47] R. J. Peterka, “Sensorimotor Integration in Human Postural Control,” *J. Neurophysiol.*, vol. 88, no. 3, pp. 1097–1118, Sep. 2002, doi: 10.1152/jn.2002.88.3.1097.

[48] F. B. Horak, J. G. Nutt, and L. M. Nashner, “Postural inflexibility in parkinsonian subjects,” *J. Neurol. Sci.*, vol. 111, no. 1, pp. 46–58, Aug. 1992, doi: 10.1016/0022-510X(92)90111-W.

[49] L. Rocchi, “Effects of deep brain stimulation and levodopa on postural sway in Parkinson’s disease,” *J. Neurol. Neurosurg. Psychiatry*, vol. 73, no. 3, pp. 267–274, Sep. 2002, doi: 10.1136/jnnp.73.3.267.

[50] J. E. Visser, M. G. Carpenter, H. van der Kooij, and B. R. Bloem, “The clinical utility of posturography,” *Clin. Neurophysiol.*, vol. 119, no. 11, pp. 2424–2436, Nov. 2008, doi: 10.1016/j.clinph.2008.07.220.

[51] M. Matinolli, J. T. Korpelainen, R. Korpelainen, K. A. Sotaniemi, M. Virranniemi, and V. V. Myllylä, “Postural sway and falls in Parkinson’s disease: A regression approach,” *Mov. Disord.*, vol. 22, no. 13, pp. 1927–1935, Oct. 2007, doi: 10.1002/mds.21633.

[52] A. Porosińska, K. Pierzchała, M. Mentel, and J. Karpe, “Evaluation of postural balance control in patients with multiple sclerosis – effect of different sensory conditions and arithmetic task execution. A pilot study,” *Neurol. Neurochir. Pol.*, vol. 44, no. 1, pp. 35–42, 2010, doi: 10.1016/S0028-3843(14)60405-9.

[53] T. Yamamoto *et al.*, “Universal and individual characteristics of postural sway during quiet standing in healthy young adults,” *Physiol. Rep.*, vol. 3, no. 3, p. e12329, Mar. 2015, doi: 10.14814/phy2.12329.

[54] C. J. C. Lamoth, R. C. van Lummel, and P. J. Beek, “Athletic skill level is reflected in body sway: A test case for accelometry in combination with stochastic dynamics,” *Gait Posture*, vol. 29, no. 4, pp. 546–551, Jun. 2009, doi: 10.1016/j.gaitpost.2008.12.006.

[55] M. Costa *et al.*, “Noise and poise: Enhancement of postural complexity in the elderly with a stochastic-resonance–based therapy,” *Europhys. Lett.*, vol. 77, no. 6, p. 68008, Mar. 2007, doi: 10.1209/0295-5075/77/68008.

[56] H. G. Kang *et al.*, “Frailty and the Degradation of Complex Balance Dynamics During a Dual-Task Protocol,” *Journals Gerontol. Ser. A Biol. Sci. Med. Sci.*, vol. 64A, no. 12, pp. 1304–1311, Dec. 2009, doi: 10.1093/gerona/glp113.

[57] B. Manor *et al.*, “Physiological complexity and system adaptability: evidence from postural control dynamics of older adults,” *J. Appl. Physiol.*, vol. 109, no. 6, pp. 1786–1791, Dec. 2010, doi: 10.1152/japplphysiol.00390.2010.

[58] J. J. Collins and C. J. De Luca, “Random Walking during Quiet Standing,” *Phys. Rev.*

*Lett.*, vol. 73, no. 5, pp. 764–767, Aug. 1994, doi: 10.1103/PhysRevLett.73.764.

[59] M. Roerdink, M. De Haart, A. Daffertshofer, S. F. Donker, A. C. H. Geurts, and P. J. Beek, "Dynamical structure of center-of-pressure trajectories in patients recovering from stroke," *Exp. Brain Res.*, vol. 174, no. 2, pp. 256–269, Sep. 2006, doi: 10.1007/s00221-006-0441-7.

[60] D. Delignières, K. Torre, and P.-L. Bernard, "Transition from Persistent to Anti-Persistent Correlations in Postural Sway Indicates Velocity-Based Control," *PLoS Comput. Biol.*, vol. 7, no. 2, p. e1001089, Feb. 2011, doi: 10.1371/journal.pcbi.1001089.

[61] C. J. C. Lamoth and M. J. G. van Heuvelen, "Sports activities are reflected in the local stability and regularity of body sway: Older ice-skaters have better postural control than inactive elderly," *Gait Posture*, vol. 35, no. 3, pp. 489–493, Mar. 2012, doi: 10.1016/j.gaitpost.2011.11.014.

[62] & Shannon, C. E. and W. Weaver, *The Mathematical Theory of Communication*. Urbana, IL: The University of Illinois Press., 1949.

[63] C. E. Shannon, "A Mathematical Theory of Communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, Jul. 1948, doi: 10.1002/j.1538-7305.1948.tb01338.x.

[64] A. Kolmogorov, *On the Properties and Functions of the Concentrations of M. P*, Levy (1 ed. Annales de l'Institut Henri Poincaré., 1958.

[65] J. Stergiou, N., & Yentes, *Nonlinear Analysis for Human Movement Variability*, Chapter 6: Boca Raton, FL: CRC Press., 2016.

[66] A. Babloyants, A., & Destexte, *Is the Normal Heart a Periodic Oscillator*, Vol 58. BIological Cybernetics., 1988.

[67] C. J. Stam, B. Jelles, H. A. M. Achtereekte, S. A. R. B. Rombouts, J. P. J. Slaets, and R.

W. M. Keunen, “Investigation of EEG non-linearity in dementia and Parkinson’s disease,” *Electroencephalogr. Clin. Neurophysiol.*, vol. 95, no. 5, pp. 309–317, Nov. 1995, doi: 10.1016/0013-4694(95)00147-Q.

[68] J.-P. Eckmann and D. Ruelle, “Ergodic theory of chaos and strange attractors,” in *The Theory of Chaotic Attractors*, New York, NY: Springer New York, 1985, pp. 273–312.

[69] C. Bandt and B. Pompe, “Permutation Entropy: A Natural Complexity Measure for Time Series,” *Phys. Rev. Lett.*, vol. 88, no. 17, p. 174102, Apr. 2002, doi: 10.1103/PhysRevLett.88.174102.

[70] J. S. Richman and J. R. Moorman, “Physiological time-series analysis using approximate entropy and sample entropy,” *Am. J. Physiol. Circ. Physiol.*, vol. 278, no. 6, pp. H2039–H2049, Jun. 2000, doi: 10.1152/ajpheart.2000.278.6.H2039.

[71] S. Pincus, “Approximate entropy (ApEn) as a complexity measure,” *Chaos An Interdiscip. J. Nonlinear Sci.*, vol. 5, no. 1, pp. 110–117, Mar. 1995, doi: 10.1063/1.166092.

[72] J. M. Yentes, N. Hunt, K. K. Schmid, J. P. Kaipust, D. McGrath, and N. Stergiou, “The Appropriate Use of Approximate Entropy and Sample Entropy with Short Data Sets,” *Ann. Biomed. Eng.*, vol. 41, no. 2, pp. 349–365, Feb. 2013, doi: 10.1007/s10439-012-0668-3.

[73] M. J. Kirkby, “The fractal geometry of nature. Benoit B. Mandelbrot. W. H. Freeman and co., San Francisco, 1982. No. of pages: 460. Price: £22.75 (hardback),” *Earth Surf. Process. Landforms*, vol. 8, no. 4, pp. 406–406, Jul. 1983, doi: 10.1002/esp.3290080415.

[74] L. A. Lipsitz, “Dynamics of Stability: The Physiologic Basis of Functional Health and Frailty,” *Journals Gerontol. Ser. A Biol. Sci. Med. Sci.*, vol. 57, no. 3, pp. B115–B125, Mar. 2002, doi: 10.1093/gerona/57.3.B115.

[75] B. B. Mandelbrot and J. W. Van Ness, “Fractional Brownian Motions, Fractional Noises and Applications,” *SIAM Rev.*, vol. 10, no. 4, pp. 422–437, Oct. 1968, doi: 10.1137/1010093.

[76] J. W. Kantelhardt, E. Koscielny-Bunde, H. H. . Rego, S. Havlin, and A. Bunde, “Detecting long-range correlations with detrended fluctuation analysis,” *Phys. A Stat. Mech. its Appl.*, vol. 295, no. 3–4, pp. 441–454, Jun. 2001, doi: 10.1016/S0378-4371(01)00144-3.

[77] C. -K. Peng, S. Havlin, H. E. Stanley, and A. L. Goldberger, “Quantification of scaling exponents and crossover phenomena in nonstationary heartbeat time series,” *Chaos An Interdiscip. J. Nonlinear Sci.*, vol. 5, no. 1, pp. 82–87, Mar. 1995, doi: 10.1063/1.166141.

[78] S. V. Buldyrev *et al.*, “Analysis of DNA sequences using methods of statistical physics,” *Phys. A Stat. Mech. its Appl.*, vol. 249, no. 1–4, pp. 430–438, 1998, doi: 10.1016/S0378-4371(97)00503-7.

[79] C.-K. Peng *et al.*, “Statistical properties of DNA sequences,” *Phys. A Stat. Mech. its Appl.*, vol. 221, no. 1–3, pp. 180–192, Nov. 1995, doi: 10.1016/0378-4371(95)00247-5.

[80] C.-K. Peng, J. M. Hausdorff, S. Havlin, J. E. Mietus, H. E. Stanley, and A. L. Goldberger, “Multiple-time scales analysis of physiological time series under neural control,” *Phys. A Stat. Mech. its Appl.*, vol. 249, no. 1–4, pp. 491–500, Jan. 1998, doi: 10.1016/S0378-4371(97)00508-6.

[81] J. M. Hausdorff *et al.*, “Altered fractal dynamics of gait: reduced stride-interval correlations with aging and Huntington’s disease,” *J. Appl. Physiol.*, vol. 82, no. 1, pp. 262–269, Jan. 1997, doi: 10.1152/jappl.1997.82.1.262.

[82] E. Koscielny–Bunde, H. Eduardo Roman, A. Bunde, S. Havlin, and H. Schellnhuber,

"Long-range power-law correlations in local daily temperature fluctuations," *Philos. Mag. B*, vol. 77, no. 5, pp. 1331–1340, May 1998, doi: 10.1080/13642819808205026.

[83] E. Koscielny-Bunde, A. Bunde, S. Havlin, H. E. Roman, Y. Goldreich, and H.-J. Schellnhuber, "Indication of a Universal Persistence Law Governing Atmospheric Variability," *Phys. Rev. Lett.*, vol. 81, no. 3, pp. 729–732, Jul. 1998, doi: 10.1103/PhysRevLett.81.729.

[84] S. Blesić, S. Milošević, D. Stratimirović, and M. Ljubisavljević, "Detrended fluctuation analysis of time series of a firing fusimotor neuron," *Phys. A Stat. Mech. its Appl.*, vol. 268, no. 3–4, pp. 275–282, Jun. 1999, doi: 10.1016/S0378-4371(99)00110-7.

[85] S. Bahar *et al.*, "Long-range temporal anti-correlations in paddlefish electroreceptors," *Europhys. Lett.*, vol. 56, no. 3, pp. 454–460, 2001, doi: 10.1209/epl/i2001-00540-7.

[86] Y. Liu, P. Cizeau, M. Meyer, C.-K. Peng, and H. Eugene Stanley, "Correlations in economic time series," *Phys. A Stat. Mech. its Appl.*, vol. 245, no. 3–4, pp. 437–440, Nov. 1997, doi: 10.1016/S0378-4371(97)00368-3.

[87] P. Cizeau, Y. Liu, M. Meyer, C.-K. Peng, and H. Eugene Stanley, "Volatility distribution in the S&P500 stock index," *Phys. A Stat. Mech. its Appl.*, vol. 245, no. 3–4, pp. 441–445, Nov. 1997, doi: 10.1016/S0378-4371(97)00417-2.

[88] H. E. Hurst, "THE PROBLEM OF LONG-TERM STORAGE IN RESERVOIRS," *Int. Assoc. Sci. Hydrol. Bull.*, vol. 1, no. 3, pp. 13–27, Jan. 1956, doi: 10.1080/02626665609493644.

[89] M. D. McDonnell and L. M. Ward, "The benefits of noise in neural systems: bridging theory and experiment," *Nat. Rev. Neurosci.*, vol. 12, no. 7, pp. 415–425, Jul. 2011, doi: 10.1038/nrn3061.

[90] M. D. McDonnell and D. Abbott, “What Is Stochastic Resonance? Definitions, Misconceptions, Debates, and Its Relevance to Biology,” *PLoS Comput. Biol.*, vol. 5, no. 5, p. e1000348, May 2009, doi: 10.1371/journal.pcbi.1000348.

[91] D. R. Toledo, J. A. Barela, and A. F. Kohn, “Improved proprioceptive function by application of subsensory electrical noise: Effects of aging and task-demand,” *Neuroscience*, vol. 358, pp. 103–114, Sep. 2017, doi: 10.1016/j.neuroscience.2017.06.045.

# Chapter 3: The Efficacy of Using Color In Sub-threshold Vibration to Improve Older Adult Balance

## 3.1 Abstract

**Introduction**: Somatosensory deficiency resulting from disease and/or aging reduces the ability to effectively and accurately process input received from the skin resulting in increased fall risk and foot injury. Stochastic resonance using white noise can improve sensory deficit and enhance balance. This study aims to explore the effect of color on the efficacy of sub-threshold vibration to improve older adults' balance.

**Methods**: Nine healthy older adults(age: 62.56 ± 1.42 years) participated in the study. A force plate was used to measure the participant's sway before, during, and shortly after vibration. For each of the visit, different vibration color (white, pink, brown, and placebo) was used, and the order was random for each participant. The center of pressure(COP) in the anteroposterior(AP) and Mediolateral(ML) directions were calculated from each sway trial. The COP data were downsampled to 50 Hz, and measures of magnitude(RMS), predictability(Sample Entropy), and complexity(DFA) were extracted.

**Results**: A repeated measure ANOVA for the change in the measure in the ML direction showed significant differences across the treatments for sample entropy but not DFA and RMS. Also, a linear regression model fitted between the change in measure and the baseline for each treatment in the ML direction showed negative slopes except for the white regression line for the RMS.

**Discussion**: There was a general increase in predictability (Sample Entropy) for the different vibration colors but not the placebo. As the initial value becomes larger, the amount of predictability added becomes small. Also, DFA regression for white, pink and brown had

breakpoints implying subthreshold vibration can increase or decrease complexity depending on the initial value.

**Conclusion**: The study shows evidence that subthreshold vibration under the feet has the tendency to increase predictability and reduce sway magnitude, and this depends on the initial sway and the color of vibration used.

**Word Count : 298**

## 3.2 Introduction

The somatosensory system plays a critical role in the balance control process. As people get older, there is a decline in the functions of the somatosensory system leading to a considerable deficit in the perception of cutaneous and proprioceptive stimuli [1,2]. The changes to the somatosensory system increase the mechanoreceptor detection thresholds and low somatosensory feedback, which impairs balance and increases the risk of falls [3]. These age-related declines in sensory functions become worse in the presence of neuromuscular diseases such as diabetes [4,5], Parkinson and stroke [4]. Fall becomes very prevalent as one gets older and is significantly worse after 75 years. Fall consequences include fractures, frequent hospitalizations, reduced quality of life, fear of falls, increased morbidity, and mortality [6]. Falls are the leading cause of death in older adults in the US (CDC,2016). Fall-related incidents accounted for about $50.8 billion in health care costs in 2020 (CDC, 2020). These expenses are estimated to rise as this population increases in size. Considering the high prevalence of falls in older adults and their negative effects on the quality of life, there is an urgent need to develop interventions to help improve balance and prevent falls.

A promising intervention to improve sensory deficit and enhance the function of the sensory system utilizes noise in a process called stochastic resonance (SR) [6,7]. Stochastic resonance is where the response of a nonlinear system to a weak signal is better with noise than when noise is absent [8]. The stimuli on their own are below the threshold, making them unable to be detected. When noise is added to the stimulus, there is a threshold crossing making the stimuli detectable and characterized [9]. To ensure stochastic resonance, an optimal amount of noise needs to be added. Stochastic resonance can be applied to different physiological systems and can be achieved by electrical, magnetic, or mechanical (vibratory) simulation. In addition, the input magnitude can be below the human sensing threshold (subthreshold) or above their sensing threshold (suprathreshold).

The effect of SR has been explored in both postural sway [1,2,10] and gait [11,12]. SR through subthreshold vibration under the feet has been found to enhance balance during [2,9,13] or soon after

vibration [14,15]. In these studies, the center of pressure (COP) times series was characterized using the magnitude, predictability, and complexity in the anterior-posterior, Medial-lateral and/or spatial directions. The results from these studies showed that the participants' sway reduced in magnitude and increased in both predictability and complexity after subthreshold vibration was applied to their feet. Moreover, SR through subthreshold vibration (70-85% of sensory threshold) has also been found to improve dynamic postural tasks such as timed-up and go [16]. While the precise neurophysiological process by which noise augments weakened sensorimotor pathways remains unknown, the improved balance could be attributed to altered muscle activity as a result of SR vibrations [17].

The effect of SR through subthreshold vibration on a person's postural sway (PS) could be dependent on the participant's initial PS before vibration [4,18]. Results from previous studies support this idea [12,19]. Stephen et al. showed that the change in gait parameter was dependent on the gait parameter before vibration. It found that participants with relatively low initial gait variability reported an increase while there was a substantial variability decrease for those that initially had high variability. Also, other studies have suggested that using an individualized level of SR vibration can further enhance balance performance [20,21,22,23] than using the same predetermined intensity level of SR for all participants. Moreover, after reanalyzing the original data by Priplata et al., Kelty-Stephen et al found out that the effect of subthreshold vibration on body sway dynamics is regulated by the extent of autocorrelations that exist in the body sway, thus explaining the difference in responses between individuals [18]. These all show that individuals will benefit more from a personalized treatment instead of a single treatment for all. It is therefore critical to customize vibration parameters to individual requirements.

No study has analyzed the effects of different vibration types on COP time series. All previous studies used either constant or white vibration, which undermines the importance of variability, complexity, and predictability of PS in humans. The relationship between the system's initial state, the color of the subthreshold vibration used as a treatment, and the treatment's effect on the system are still not clear. This study is designed to explore the effect of subthreshold vibration in postural sway when different types of

vibrations (white, pink, and brown) are administered at the feet of healthy older adults. It is hypothesized that (1) the sway measures prior to stimulation will be different across subjects and also vary within a subject across the sessions (2). Each vibration color would cause a change in state different from the placebo (3)The efficacy of utilizing subthreshold vibration to change postural sway magnitude, predictability, and complexity depends on their initial values and the color used.

## 3.3 Methods

### 3.3.1 Vibratory Mat

The vibratory mat consists of four sets of eccentric mass motors (307-105, 310-003, 306-10H and 307-103 from Precision Microdrives, UK) embedded in a 10-mm thick Shore A50 silicone. The motors are placed under the heel, first and fifth metatarsal of each foot and each offers varying force and frequency ranges. Each set of motors on the mat is positioned according to foot placement recommendations: the heels are 17cm apart, and the angle between the lines produced by the big toe and each of the foot's heels is 14 [24]. The mat is designed to accommodate different shoe sizes and can output different vibration colors (white, pink and brown) at various power levels. A custom-built circuit and Arduino UNO (Arduino, MA) are used to control the power level of the mat. Finally, the vibratory mat uses an enhanced 421 [25,26] vibration perception threshold (VPT) detection method to find the participants' sensing threshold.

### 3.3.2 Participants

Nine healthy older adults (HO: three male and six females; age: 62.56 ± 1.42 years; height: 169.48 ± 9.32 cm), with no significant history of musculoskeletal, neurologic, vestibular, or mobility deficiencies participated in the study. All the participants could also stand for 5 minutes without any support while keeping their eyes closed. All participants gave informed consent as approved by the University of Kansas' Institutional Review Board.

### 3.3.3 Task and Data Collection

Each participant attended four sessions in the Biodynamics Laboratory, each session involved one of four different vibration treatments. The sessions were separated by at least one night. All visits followed the same protocol. For each visit different vibration color (white, pink, brown, and placebo) was used and the order was random for each participant. All sway trials were conducted with the participant's heels 17 cm apart [24] and their toes pointing forward. In all sway trials, the participant was instructed to close their eyes, stand barefoot, keep their hands at their sides, and stand naturally without speaking. The vibratory mat was placed on top of a force plate (AMTI, Watertown, USA). The participant remained standing for 90 seconds for each sway trial. The force plate kinetic data were collected at 2500 Hz using a 16-bit A/D CED Power mkII and Spike2 recorder (Cambridge Electronic Design, UK). The experimental protocol was as follows. First, the participant's sway is recorded while standing on the vibratory mat without vibration. This serves as the mat's baseline trial. The sensing threshold of each participant was determined using the modified 421 protocol [25,26], and 307-103 motors. After a 2-minute break, the participant stood on the vibratory mat and received the vibration treatment at 90% of their sensing threshold [1,9,10,13,17] . The vibration lasted 90 seconds and was produced using the 307-103 motors. Lastly, the participant's sway was recorded while standing on the vibratory mat without any vibration immediately following the vibration treatment.

### 3.3.4 Data Analysis

All the data analysis was done in MATLAB R2020b (MathWorks, MA, USA).The force and moment data were filtered to remove any motor vibration sway time series with a low pass filter at 20 Hz cut-off frequency [28,29].The center of pressure in the anteroposterior (AP) and Mediolateral (ML) directions was calculated from each sway trial using the force and moment measurements obtained from the force plates attached to the ground.

$$COP_{AP} = \frac{M_{ML} + F_{AP} * \mathrm{dz}}{F_Z} \qquad COP_{ML} = \frac{M_{AP} + F_{ML} * \mathrm{dz}}{F_Z}$$

The COP data ($COP_{AP}$, $COP_{ML}$) were downsampled to 50 Hz following best practices for variability measurements like entropy and fractality. The downsampled COP data were then used to calculate the spatial COP using the formula $COP_{spatial} = \sqrt{COP_{AP}^2 + COP_{ML}^2}$. Measures of magnitude, predictability, and complexity were extracted for each of the sway trials ([BLMat, STIM, T0]) for all the directions ([$COP_{AP}$, $COP_{ML}$, $COP_{spatial}$]). The RMS of the time series was used to represent the magnitude of the $COP_{AP}$ and $COP_{ML}$, while the 95 percent ellipse was used to show the magnitude of the $COP_{spatial}$. Sample entropy [30] defined as $SampEn(m,r) = -In\frac{A^m(r)}{B^m(r)}$ was used to assess the predictability in the time series. The sample entropy calculation was done using m=2 and R= 0.1 (i.e. r = .1*SD). Finally, a scaling exponent α, derived from a Detrended Fluctuation Analysis (DFA) [31,32], with tmin = 0.5 s and tmax = 15 s (or, nmin = 25, nmax = N/6), was used to indicate the complexity in all the time series. The ground response forces were only employed in the non-placebo STIM sway trials to determine the rectified force that the motors exerted during the sway experiment. This was achieved by band-pass filtering the ground reaction forces between 20 and 400 Hz to remove any sway from the data [28,29]. Following that, the filtered data was rectified sequentially over 0.1 seconds time window. The rectified and filtered motor forces were used in a DFA (tmin = 0.5 s and tmax = 15 s, or nmin = 5, nmax = N/6) to evaluate the color accuracy of the output.

### 3.3.5 Statistical Analysis

The statistical analysis was done using Python 3. First, ANOVA was used to determine the statistical significance of the baseline measures (entropy, DFA, and RMS) across the participants. Then a repeated measure of ANOVA was done between baseline measures (entropy, DFA, and RMS) across the four visits. Another repeated measure of ANOVA was performed to determine the statistical significance of the change in measure (measure $_{vibration}$ − measure $_{baseline}$) in the ML direction for the different treatments (white, pink, brown and placebo). The statistical significance level was set to $p<0.05$

Also, a linear regression model was fitted between the change in measure (measure $_{vibration}$ − measure $_{baseline}$) and the baseline for each treatment (white, pink, brown, and placebo) in the ML direction. For each linear model, the statistical significance of the coefficients was set to $p<0.05$, and the 95 percent confidence interval was calculated. Finally, a power analysis to estimate the sample size required to find statistical difference was done for each measure in the ML direction.

## 3.4 Results

### 3.4.1 Entropy, DFA and RMS Baseline Measures

The Entropy, DFA and RMS baseline measures show significant differences (p-value <0.05) across the participants(Table 1). The baseline sway measures were different across subjects in all the directions(AP and ML).Within participants revealed different baseline values and variability for all the measured parameters. This is represented by the mean and standard deviation, as shown in the Table 2. Repeated measures of ANOVA of the baseline measures showed no significant difference across any four sessions in any direction. The one closer to significance was the sample entropy in the ML direction.

### 3.4.2 Change in Entropy, DFA and RMS Measures.

Figure 7 shows the difference between the baseline and stimulation in the ML direction across the treatment. There was a general increased in Sample Entropy for Brown, Pink and white but not in the placebo. There was also an increase in complexity (DFA) for white, Pink and placebo and a decrease in complexity when Brown was administered. RMS also decreased for brown and white and increased for the pink and placebo. A repeated measure ANOVA for the change in measure (measure $_{vibration}$ − measure $_{baseline}$) in the ML direction showed significant difference

across the treatments (white, pink, brown and placebo) for sample entropy ( p-value<0.01) but not for DFA  and RMS.

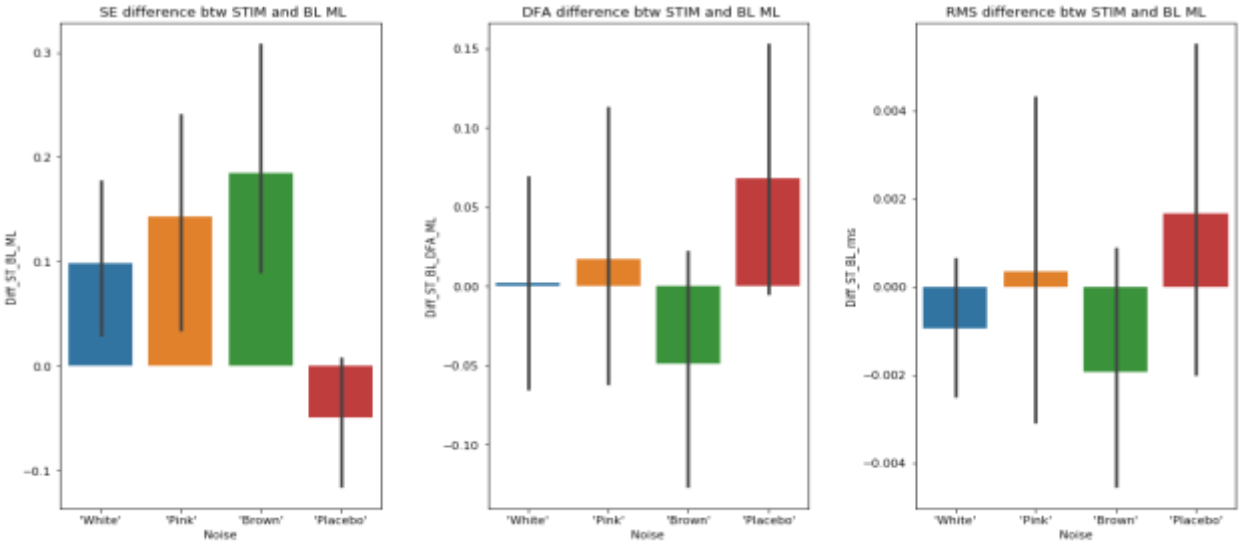


*Figure 1: Difference between baseline and stimulation measures for the vibration treatment in ML*

### 3.4.3 Linear Regression

Figures 11,12 and 13 show the effect of subthreshold vibration on the sway while stimulation was administered with respect to the baseline sway in the ML direction. All regression showed negative slopes except the white regression line for the RMS. Most of the regression lines have very low $R^2$ and non-significant slope values.

Entropy: All the treatments showed a negative slope. Pink, white and brown do not cross zero. This indicates that as the baseline value becomes bigger, the change in predictability becomes small. Pink has the least slope, while the placebo has the highest slope and the highest $R^2$. Pink, white and brown appear to cause a greater change compared to the placebo.

DFA: All the treatments showed a downward trend. White has the highest slope and $R^2$ (0.507). Pink, Brown and placebo have non-significant slope differences and low $R^2$ values (Pink: $R^2$=0.158 ; Brown: $R^2$=0.250; Placebo: $R^2$=0.145)

RMS: All the treatments, except for white, showed a downward trend. Pink has the highest slope and intersects with all the other lines. Brown has a very small negative slope that almost looks like a horizontal trend and has the lowest $R^2$.

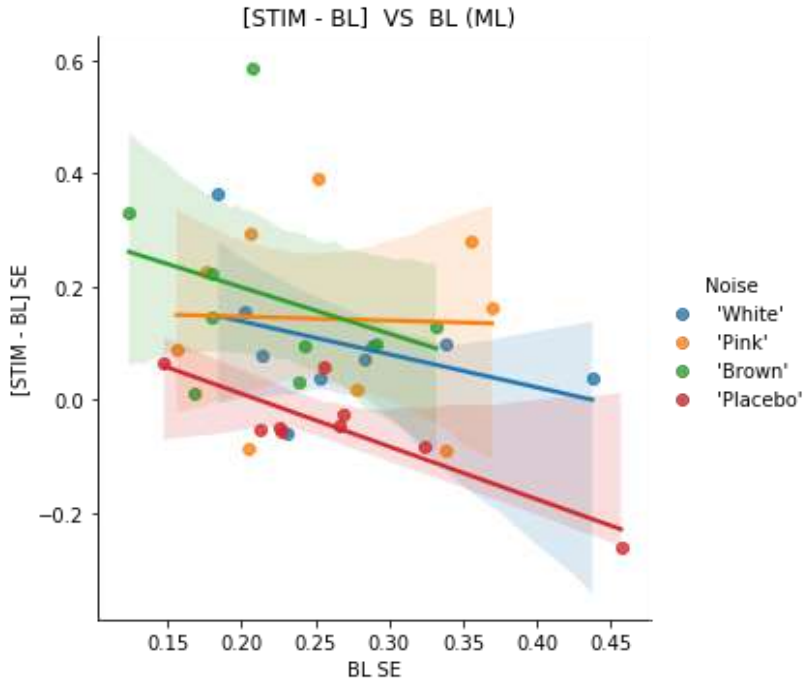


*Figure 2: A linear regression of change in sample entropy(Stimulation – baseline) vs baseline in ML*

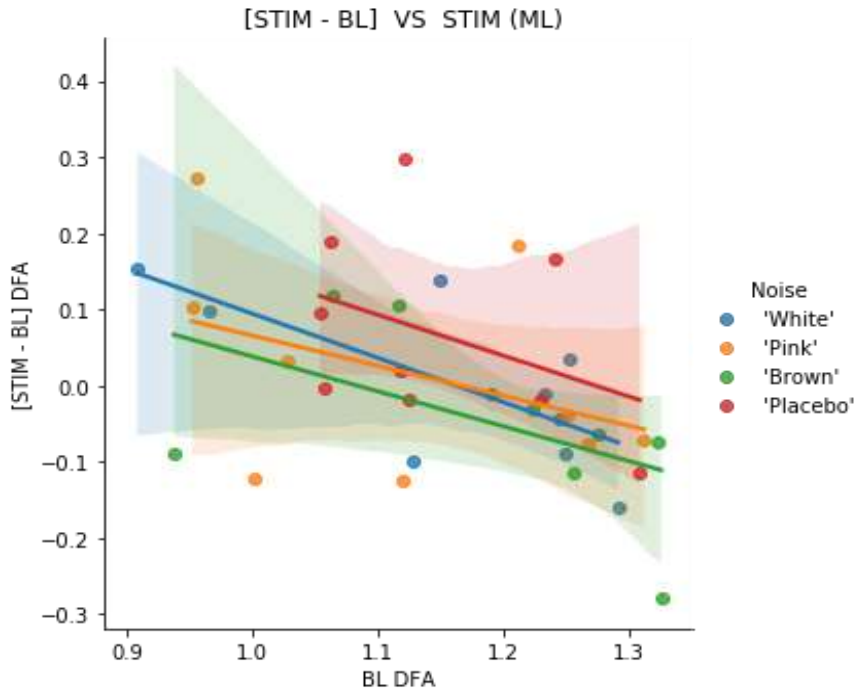


*Figure 3:A linear regression of change in DFA(Stimulation-baseline) vs baseline in ML*

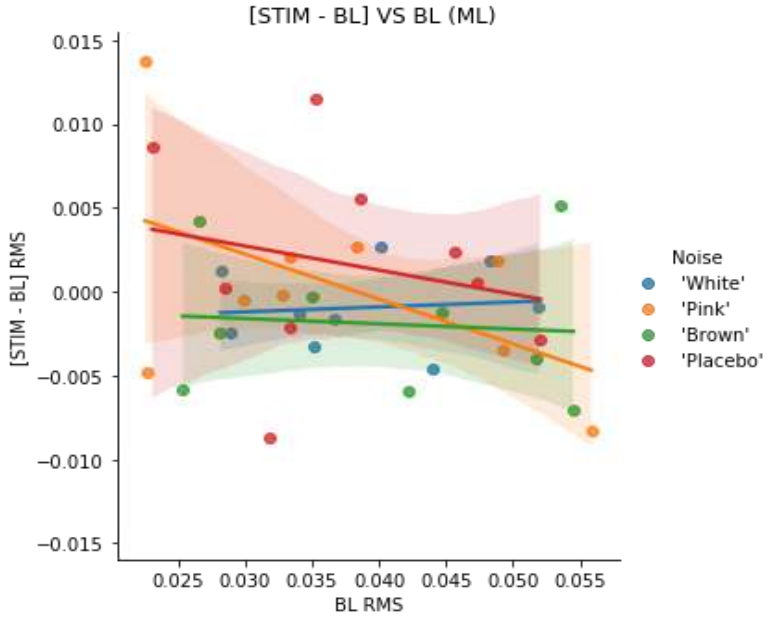


*Figure 4:A linear regression of change in RMS(Stimulation-baseline) vs baseline in ML*

## 3.5 Discussion

Specific aim 1 was to determine the variability of the initial state measures within each participant, across the four visits and within each day, across the nine participants. We hypothesized that the sway measures before stimulation will differ across subjects and vary within a subject across the sessions. Results provided support for H1. There was a significant difference in the baseline entropy, DFA and rms value across subjects. Also, within-subjects, baseline measures show different means and variability across subjects. Some had low standard deviations, while others had high standard deviations. This can be explained by the different factors that affect balance. The body maintains balance using information from the somatosensory, visual and vestibular systems. The cerebellum then utilizes this information by making limb and trunk adjustments and balancing opposing forces to maintain balance. Also, prior experience and situational clues modify these inputs and help maintain balance. These factors that affect balance will vary between people and might also change for a person from one day to the next. Therefore, measures used to characterize balance will have different values across people of the same age group and different for a person at different times. This also implies that for a personalized treatment design, the "state" of the system needs to be determined prior to the treatment, which is then designed based on the direction of desired change in the state.

However, there was no significant difference in entropy, DFA and RMS baseline across the four visit days. This suggests that within a patient being treated, it is possible that the initial state need not be determine immediately prior to each treatment. Further study is needed to determine the appropriate time between state determination, which would be affected by the rate of change in the state due to sequential treatments,

The effect of the different treatments on sway was analyzed by finding the change in entropy, DFA and RMS values in the ML direction. It was determined that subthreshold vibration using white noise increased predictability (Sample Entropy) and complexity(DFA) and reduced sway magnitude. The results agreed with previous studies that found a reduction in sway magnitude during subthreshold vibration in healthy old [9,13] and the diabetic population[33]. And also, an increase in predictability in healthy young[27]. Though this is a general conclusion made, individual analysis suggests that this was not the same for all the subjects. (i.e., subthreshold vibration increased sway and reduced predictability.)

Hypothesis 2 stated that each vibration color would cause a change in state different from the placebo. This was found to be valid for sample entropy but not for DFA and RMS. There was a general increase in predictability (Sample Entropy) when white, pink and brown were administered but not the placebo, and this change was found to be statistically significant. In terms of numbers, all the 9 participants increased in predictability when Brown was administered, 8 for white and 7 for pink (Table 6). This was different for the placebo(control) where only 2 participants increased in predictability. There was also a significant difference in the treatment when the magnitude of the state  change was also considered. In general, there was an increase in complexity (DFA) for white, Pink and placebo and a decrease in complexity when Brown was administered. This change was found not to be significant across the treatments. There was also a general decrease in RMS when white and brown were administered and an increase in RMS for pink and placebo. It was impossible to statistically differentiate the change in RMS across the different vibration treatments. Though the overall change in RMS was insignificant across vibration colors, most of the participants decreased in RMS when pink,

brown and white was administered compared to the placebo (Table 6).These results suggest that one type of vibration treatment might be better in reducing sway magnitude while a different vibration type will be better in increasing predictability or complexity.

Hypothesis 3 stated that the efficacy of utilizing subthreshold vibration to change postural sway magnitude, predictability, and complexity depends on their initial values and the color used. The impact was analyzed by using baseline value vs change linear regression. It was determined that subthreshold vibration increase predictability(Sample Entropy). This is consistent with a previous study that reported increased predictability in healthy young when subthreshold vibration was administered [27]. Also, all the vibration treatments have different negative slopes. The negative slopes obtained mean that as the initial value becomes larger, the amount of predictability added becomes small. This agrees with our hypothesis that the change in postural sway predictability is dependent on the initial value and the color used.

Figure 12. shows all the vibration treatments have negative slopes and that white, Pink and brown cross zero. These points are the breakpoints. That means that when small initial values of DFA were present, complexity was added. This agrees with previous SR studies [4,34].Also, the "breakpoints" and negative slopes indicate that the current findings are consistent with another SR study that discovered a reduction in complexity due to subthreshold vibration[35]. Thus, we can conclude that subthreshold vibration can increase [4,34]  or decrease [35] complexity depending on the initial value instead of choosing one of the two.

Figure 13. shows that the RMS generally decreased when white and green vibration color was administered. This agrees with previous studies that reported a reduction in sway magnitude during subthreshold vibration in healthy old [9,13,34] and the diabetic population [33]. Most of

the regression yielded high p-values implying non-significant and low R2 values signifying relatively poor fit, limiting the implications of these models. This was expected due to the low sample size and high variability.

Postural sway magnitude, predictability and complexity will differ across people of the same age group and vary for a person at different times. For effective personalized treatment, the "state" of the system needs to be determined prior to the treatment. Pink, White and brown noise can increase predictability(Sample Entropy ), and as the initial value becomes larger, the amount of predictability added becomes small. They also tend to increase complexity for low initial values and decrease complexity for high initial values. Also, one type of vibration treatment might be better in reducing sway magnitude, while a different vibration type will be better in increasing predictability or complexity for a person.

## Limitations

There are several limitations to this study that affect the findings' implications. First is the small sample size. Power analysis done shows that achievable and justifiable sample size to determine significant difference between the effect of the vibration types in the ML direction should be 25-30. The study was conducted during the COVID-19 pandemic. Our target group (older adults 60-65 years) was part of the COVID-19 vulnerable population, which affected the testing efforts and made it difficult to recruit participants. Some participants had to discontinue the study because of exposure to COVID-19. The low sample size made it difficult to assess and draw valid conclusions on the direction of change in the state caused by the type of vibration (white, pink or brown). Also, in assessing how the postural sway magnitude, predictability and complexity changed by the different vibration treatment colors relative to their initial states, the resulting

regression resulted in low fit and non-significant p-values due to the low sample size making it unable to draw solid conclusions.

Second, the inability to control the initial value makes it challenging to compare the change in postural sway magnitude, predictability and complexity caused by the type of vibration (white, pink or brown). Previous studies have shown that the effect of SR through subthreshold vibration on a person's PS could be dependent on the participant's initial PS before vibration [4,18]. Participants' initial PS were different for the other visits, and the variability across the four visits was also different across the participants. This makes it difficult to only assess the change in postural sway magnitude, predictability and complexity across the type of vibration (white, pink or brown) as it is also affected by the initial values.

## Future work

Future work should investigate how the type of vibration (white, pink or brown) affects postural sway magnitude, predictability and complexity while controlling for the initial values. It may also explore how a combination of vibration types affects postural sway. This study suggests that one type of vibration treatment will be better in increasing predictability or complexity, while a different one will be better in reducing sway magnitude. Therefore, a combination of vibration types might be better in achieving the desired outcome. Researchers should also continue this study with a different patient population, such as Parkinson's or diabetic neuropathy, to further understand its benefits. In addition to the above, future work may use different parameters to characterize the postural sway magnitude, predictability and complexity. Further studies can also focus on the retention capabilities of the types of vibration.

## 3.6 Conclusion

The study shows evidence that subthreshold vibration under the feet has the tendency to increase predictability and reduce sway magnitude. The effect of subthreshold vibration on sway could be dependent on the initial sway and the type of vibration used. SR through subthreshold vibration offers a promising intervention to enhance somatosensory sensitivity and improve postural stability. Understanding the relationship between the system's initial state, the color of the subthreshold vibration used as treatment, and the treatment's effect on the system will enable us to customize vibration parameters to meet individual requirements and achieve optimum benefits. We believe individuals will benefit more from a personalized treatment than a one-size-fits-all treatment design. This will further pave the way for clinical application to help reduce the risk of falls and ensure good quality of life for people.

# References


[1] A. Aboutorabi, M. Arazpour, F. Farahmand, M. Bahramizadeh, R. Fadayevatan, and E. Abdollahi, "Design and evaluation of vibratory shoe on balance control for elderly subjects: technical note," *Disabil. Rehabil. Assist. Technol.*, vol. 13, no. 2, pp. 173–177, Feb. 2018, doi: 10.1080/17483107.2017.1300346.

[2] M. Costa *et al.*, "Noise and poise: Enhancement of postural complexity in the elderly with a stochastic-resonance–based therapy," *Europhys. Lett.*, vol. 77, no. 6, p. 68008, Mar. 2007, doi: 10.1209/0295-5075/77/68008.

[3] R. J. Peterka, "Sensorimotor Integration in Human Postural Control," *J. Neurophysiol.*, vol. 88, no. 3, pp. 1097–1118, Sep. 2002, doi: 10.1152/jn.2002.88.3.1097.

[4] A. A. Priplata *et al.*, "Noise-enhanced balance control in patients with diabetes and patients with stroke," *Ann. Neurol.*, vol. 59, no. 1, pp. 4–12, Jan. 2006, doi: 10.1002/ana.20670.

[5] S. C. Regueme *et al.*, "A Therapeutic Insole Device for Postural Stability in Older People With Type 2 Diabetes. A Feasibility Study (SENSOLE Part I)," *Front. Med.*, vol. 6, Jun. 2019, doi: 10.3389/fmed.2019.00127.

[6] O. White, J. Babič, C. Trenado, L. Johannsen, and N. Goswami, "The Promise of Stochastic Resonance in Falls Prevention," *Front. Physiol.*, vol. 9, Jan. 2019, doi: 10.3389/fphys.2018.01865.

[7] C. Z.-H. Ma, W.-K. Lam, B.-C. Chang, and W. C.-C. Lee, "Can Insoles Be Used to Improve Static and Dynamic Balance of Community-Dwelling Older Adults? A Systematic Review on Recent Advances and Future Perspectives," *J. Aging Phys. Act.*, vol. 28, no. 6, pp. 971–986, Dec. 2020, doi: 10.1123/japa.2019-0293.

[8] M. D. McDonnell and L. M. Ward, "The benefits of noise in neural systems: bridging theory and experiment," *Nat. Rev. Neurosci.*, vol. 12, no. 7, pp. 415–425, Jul. 2011, doi: 10.1038/nrn3061.

[9] A. Priplata, J. Niemi, M. Salen, J. Harry, L. A. Lipsitz, and J. J. Collins, "Noise-Enhanced Human Balance Control," *Phys. Rev. Lett.*, vol. 89, no. 23, p. 238101, Nov. 2002, doi: 10.1103/PhysRevLett.89.238101.

[10] M. Dettmer, A. Pourmoghaddam, B.-C. Lee, and C. S. Layne, "Associations between Tactile Sensory Threshold and Postural Performance and Effects of Healthy Aging and Subthreshold Vibrotactile Stimulation on Postural Outcomes in a Simple Dual Task," *Curr. Gerontol. Geriatr. Res.*, vol. 2016, pp. 1–11, 2016, doi: 10.1155/2016/9797369.

[11] A. M. Galica *et al.*, "Subsensory vibrations to the feet reduce gait variability in elderly fallers," *Gait Posture*, vol. 30, no. 3, pp. 383–387, Oct. 2009, doi: 10.1016/j.gaitpost.2009.07.005.

[12] D. G. Stephen, B. J. Wilcox, J. B. Niemi, J. Franz, D. Casey Kerrigan, and S. E. D'Andrea, "Baseline-dependent effect of noise-enhanced insoles on gait variability in healthy elderly walkers," *Gait Posture*, vol. 36, no. 3, pp. 537–540, Jul. 2012, doi: 10.1016/j.gaitpost.2012.05.014.

[13] M. Dettmer, A. Pourmoghaddam, B.-C. Lee, and C. S. Layne, "Effects of aging and tactile stochastic resonance on postural performance and postural control in a sensory conflict task," *Somatosens. Mot. Res.*, vol. 32, no. 2, pp. 128–135, Apr. 2015, doi: 10.3109/08990220.2015.1004045.

[14] C.-C. Wang and W.-H. Yang, "Using detrended fluctuation analysis (DFA) to analyze whether vibratory insoles enhance balance stability for elderly fallers," *Arch. Gerontol. Geriatr.*, vol. 55, no. 3, pp. 673–676, Nov. 2012, doi: 10.1016/j.archger.2011.11.008.

[15] Q. Wei *et al.*, "Multivariate Multiscale Entropy Applied to Center of Pressure Signals Analysis: An Effect of Vibration Stimulation of Shoes," *Entropy*, vol. 14, no. 11, pp. 2157–2172, Nov. 2012, doi: 10.3390/e14112157.

[16] D. Podsiadlo and S. Richardson, “The Timed ‘Up & Go’: A Test of Basic Functional Mobility for Frail Elderly Persons,” *J. Am. Geriatr. Soc.*, vol. 39, no. 2, pp. 142–148, Feb. 1991, doi: 10.1111/j.1532-5415.1991.tb01616.x.

[17] W.-M. Chen, J.-W. Li, X. Geng, C. Wang, L. Chen, and X. Ma, “The potential influence of stochastic resonance vibrations on neuromuscular strategies and center of pressure sway during single-leg stance,” *Clin. Biomech.*, vol. 77, p. 105069, Jul. 2020, doi: 10.1016/j.clinbiomech.2020.105069.

[18] D. G. Kelty-Stephen and J. A. Dixon, “Temporal correlations in postural sway moderate effects of stochastic resonance on postural stability,” *Hum. Mov. Sci.*, vol. 32, no. 1, pp. 91–105, Feb. 2013, doi: 10.1016/j.humov.2012.08.006.

[19] H. Toda, Y. Hashimoto, and M. Tada, “Vibrotactile Stimulation of Nail of Hallux during Walking: Effect on Center-of-Mass Movement in Healthy Young Adults,” *Appl. Sci.*, vol. 10, no. 13, p. 4562, Jun. 2020, doi: 10.3390/app10134562.

[20] A. P. Mulavara *et al.*, “Improving balance function using vestibular stochastic resonance: optimizing stimulus characteristics,” *Exp. Brain Res.*, vol. 210, no. 2, pp. 303–312, Apr. 2011, doi: 10.1007/s00221-011-2633-z.

[21] R. Goel *et al.*, “Using Low Levels of Stochastic Vestibular Stimulation to Improve Balance Function,” *PLoS One*, vol. 10, no. 8, p. e0136335, Aug. 2015, doi: 10.1371/journal.pone.0136335.

[22] S. E. Ross, S. W. Linens, C. J. Wright, and B. L. Arnold, “Customized Noise-Stimulation Intensity for Bipedal Stability and Unipedal Balance Deficits Associated With Functional Ankle Instability,” *J. Athl. Train.*, vol. 48, no. 4, pp. 463–470, Jul. 2013, doi: 10.4085/1062-6050-48.3.12.

[23] F. Moss, “Stochastic resonance and sensory information processing: a tutorial and review of

application," *Clin. Neurophysiol.*, vol. 115, no. 2, pp. 267–281, Feb. 2004, doi: 10.1016/j.clinph.2003.09.014.

[24] W. McIlroy and B. Maki, "Preferred placement of the feet during quiet stance: development of a standardized foot placement for balance testing," *Clin. Biomech.*, vol. 12, no. 1, pp. 66–70, Jan. 1997, doi: 10.1016/S0268-0033(96)00040-X.

[25] P. J. Dyck, P. C. O'Brien, J. L. Kosanke, D. A. Gillen, and J. L. Karnes, "A 4, 2, and 1 stepping algorithm for quick and accurate estimation of cutaneous sensation threshold," *Neurology*, vol. 43, no. 8, pp. 1508–1508, Aug. 1993, doi: 10.1212/WNL.43.8.1508.

[26] E. F. B. Whorley, "Physiology-Based Approach for Detecting Vibration Perception Threshold in the Plantar Foot," *M.S. Thesis, Univ. Kansas, Lawrence, KS*, 2020.

[27] E. A. Keshner, J. C. Slaboda, L. Day, and K. Darvish, "Visual conflict and cognitive load modify postural responses to vibrotactile noise," *J. Neuroeng. Rehabil.*, vol. 11, no. 1, p. 6, 2014, doi: 10.1186/1743-0003-11-6.

[28] J. M. Hijmans, J. H. B. Geertzen, W. Zijlstra, A. L. Hof, and K. Postema, "Effects of vibrating insoles on standing balance in diabetic neuropathy," *J. Rehabil. Res. Dev.*, vol. 45, no. 9, pp. 1441–1450, 2008, doi: 10.1682/JRRD.2008.02.0023.

[29] J. Moon *et al.*, "Shoes with active insoles mitigate declines in balance after fatigue," *Sci. Rep.*, vol. 10, no. 1, p. 1951, Dec. 2020, doi: 10.1038/s41598-020-58815-9.

[30] J. S. Richman and J. R. Moorman, "Physiological time-series analysis using approximate entropy and sample entropy," *Am. J. Physiol. Circ. Physiol.*, vol. 278, no. 6, pp. H2039–H2049, Jun. 2000, doi: 10.1152/ajpheart.2000.278.6.H2039.

[31] C.-K. Peng, S. V. Buldyrev, S. Havlin, M. Simons, H. E. Stanley, and A. L. Goldberger, "Mosaic organization of DNA nucleotides," *Phys. Rev. E*, vol. 49, no. 2, pp. 1685–1689, Feb. 1994, doi:

10.1103/PhysRevE.49.1685.

[32] C. -K. Peng, S. Havlin, H. E. Stanley, and A. L. Goldberger, “Quantification of scaling exponents and crossover phenomena in nonstationary heartbeat time series,” *Chaos An Interdiscip. J. Nonlinear Sci.*, vol. 5, no. 1, pp. 82–87, Mar. 1995, doi: 10.1063/1.166141.

[33] G. J. Postema K, Hijmans J, Zijlstra W, Hof A, “The effects of vibrating insoles on standing balance in diabetic neuropathy,” *Gait Posture*, vol. 30, pp. S38--S39, 2009.

[34] A. A. Priplata, J. B. Niemi, J. D. Harry, L. A. Lipsitz, and J. J. Collins, “Vibrating insoles and balance control in elderly people,” *Lancet*, vol. 362, no. 9390, pp. 1123–1124, Oct. 2003, doi: 10.1016/S0140-6736(03)14470-4.

[35] P. Simeonov, H. Hsiao, J. Powers, D. Ammons, T. Kau, and A. Amendola, “Postural stability effects of random vibration at the feet of construction workers in simulated elevation,” *Appl. Ergon.*, vol. 42, no. 5, pp. 672–681, Jul. 2011, doi: 10.1016/j.apergo.2010.10.002.

# Chapter 4: Summary

## 4.1 Summary of Study

The study aims to explore the effect of color on the efficacy of sub-threshold vibration to improve balance in older adults. Nine healthy older adults (three male and six females; age: 62.56 ± 1.42 years; height: 169.48 ± 9.32 cm) with no significant history of musculoskeletal, neurologic, vestibular, or mobility deficiencies participated in the study. The participants visited the lab 4 times, and each time their sway before vibration, during vibration and soon after vibration was measured using a force plate. For each visit, different vibration color (white, pink, brown, and placebo) was used, and the order was random for each participant. All sway trials were conducted with people's heels 17 cm apart, eyes closed, hands by their sides, and toes pointing forward. Measures of magnitude (RMS), predictability (Sample Entropy), and complexity (DFA) were extracted for each of the sway trials. The Entropy, DFA and RMS baseline measures show significant differences across the participants. All of the measured parameters revealed different baseline values and variability within participants. There was a general increase in predictability (Sample Entropy) when white, pink and brown were administered but not the placebo. There was also a general decrease in RMS when white and brown were administered and an increase in RMS for pink and placebo. All regression analysis showed negative slopes except the white regression line for the RMS. Most of the regression lines have very low R2 and non-significant slope values.

## 4.2 Conclusions and Recommendations

The study shows evidence that subthreshold vibration under the feet has the tendency to increase predictability and reduce sway magnitude. The effect of subthreshold vibration on sway could depend on the initial sway and the type of vibration used. SR through subthreshold vibration offers a promising intervention to enhance somatosensory sensitivity and improve postural

stability. Understanding the relationship between the system's initial state, the color of the subthreshold vibration used as treatment, and the treatment's effect on the system will enable us to tailor vibration parameters to meet individual needs and maximize benefits. We believe individuals will benefit more from a personalized treatment than a one-size-fits-all treatment design. This will pave the way for clinical applications to help reduce the risk of falls and ensure good quality of life for people.

## 4.3 Limitations and Future Work

There are several limitations to this study that affect the implications of the findings. The first issue is the small sample size. The second limitation is the inability to control the initial values to compare the change in postural sway measures caused solely by the type of vibration (white, pink, or brown). Future research should look into how the type of vibration (white, pink, or brown) affects the magnitude, predictability, and complexity of postural sway while controlling for the initial values. It may also be advantageous to investigate how repetitive vibration treatment and a combination of vibration types within a treatment affects postural sway. Researchers should also continue this study with a different patient population, such as those with Parkinson's disease or diabetic neuropathy, to further understand its benefits. In addition to the above, future work may use different parameters to characterize the postural sway magnitude, predictability, and complexity. Further studies can also focus on the retention capabilities of the types of vibration.

# Appendix A : Supplementary Materials

| | Sample Entropy | DFA | RMS |
|---|---|---|---|
| AP | <0.0001 | <0.0001 | <0.0001 |
| ML | 0.013026 | 0.000152 | <0.0001 |

*Table 1:P-values for baseline sway measures across participants*

AP

| Subjects | Sample Entropy | DFA | RMS |
|---|---|---|---|
| S001 | 0.281321±0.05650 | 1.310376±0.043352 | 0.009564±0.003361 |
| S002 | 0.367428±0.067293 | 1.173270±0.048557 | 0.009264±0.004433 |
| S003 | 0.390949±0.023714 | 1.078883±0.036532 | 0.010490±0.004792 |
| S004 | 0.310557±0.055543 | 1.106624±0.053554 | 0.012273±0.004746 |
| S005 | 0.555841±0.041033 | 1.015461±0.070333 | 0.015426±0.004683 |
| S006 | 0.292658±0.101308 | 1.160926±0.096646 | 0.022150±0.008441 |
| S007 | 0.506556±0.038794 | 1.038104±0.061292 | 0.015581±0.008276 |
| S008 | 0.201242±0.034616 | 1.395357±0.081926 | 0.009903±0.005017 |
| S009 | 0.386106±0.033615 | 1.091864±0.036426 | 0.018108±0.015268 |

*Table 2:Mean and standard deviation of the sample entropy, DFA and RMS of the participants in the AP direction.*

ML

| Subjects | Sample Entropy | DFA | RMS |
|---|---|---|---|
| S001 | 0.266285±0.077270 | 1.163035±0.107911 | 0.048800±0.004390 |
| S002 | 0.213654±0.054182 | 1.229724±0.028706 | 0.039614±0.003938 |
| S003 | 0.251289±0.059459 | 1.091675±0.112571 | 0.049107±0.003281 |
| S004 | 0.300727±0.109120 | 1.215706±0.096628 | 0.026635±0.003007 |
| S005 | 0.361880±0.051975 | 0.989070±0.050234 | 0.034398±0.005715 |
| S006 | 0.179230±0.027398 | 1.091893±0.117170 | 0.035538±0.002215 |
| S008 | 0.277162±0.015238 | 1.296842±0.016674 | 0.030166±0.006202 |
| S009 | 0.204119±0.026223 | 1.256933±0.047660 | 0.027529±0.003852 |
| S1010 | 0.223407±0.088574 | 1.050157±0.098962 | 0.052886±0.001973 |

*Table 3:Mean and standard deviation of the sample entropy, DFA and RMS of the participants in the ML direction*

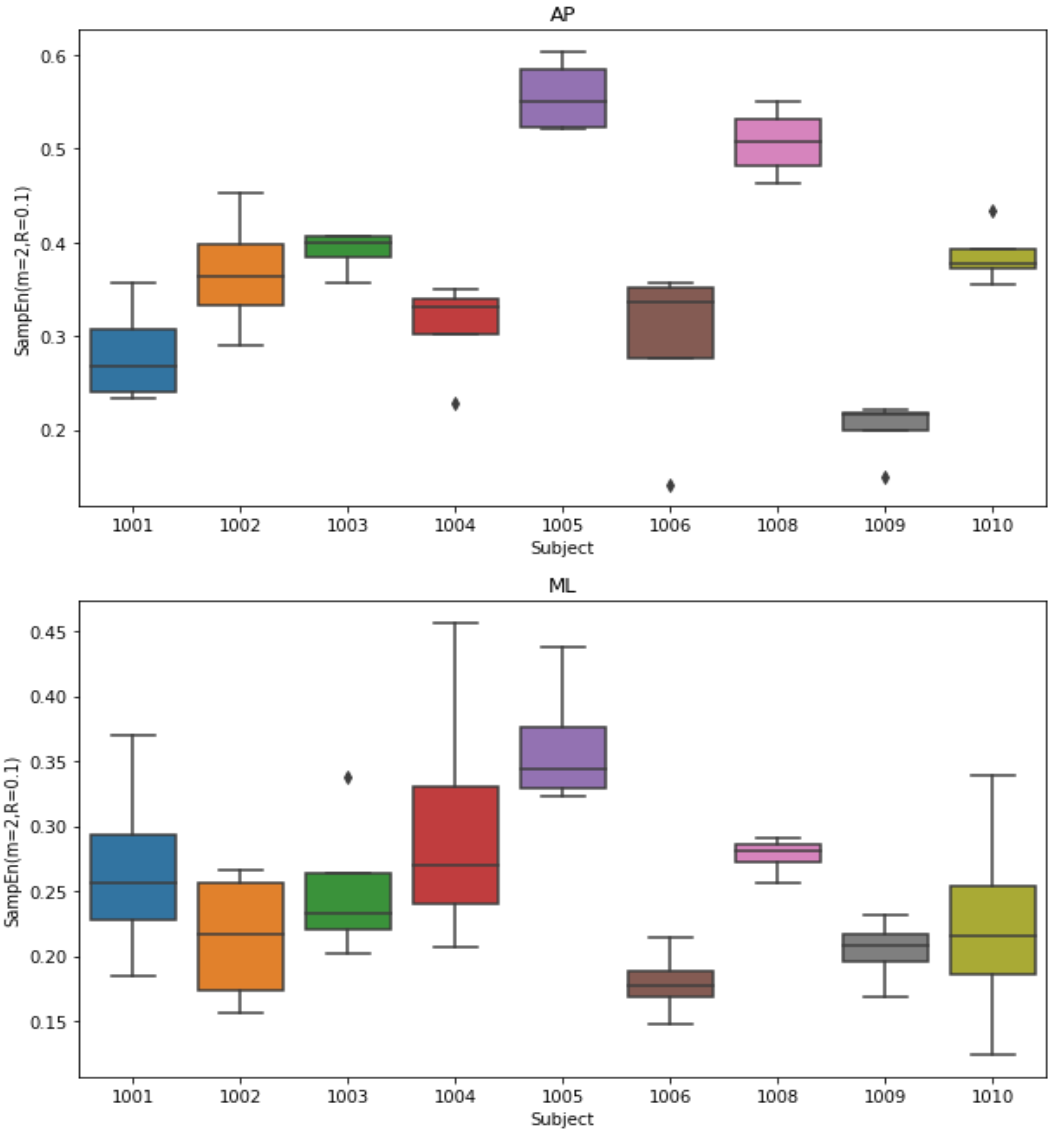


*Figure 5: Sample entropy of the participants in the AP and ML*

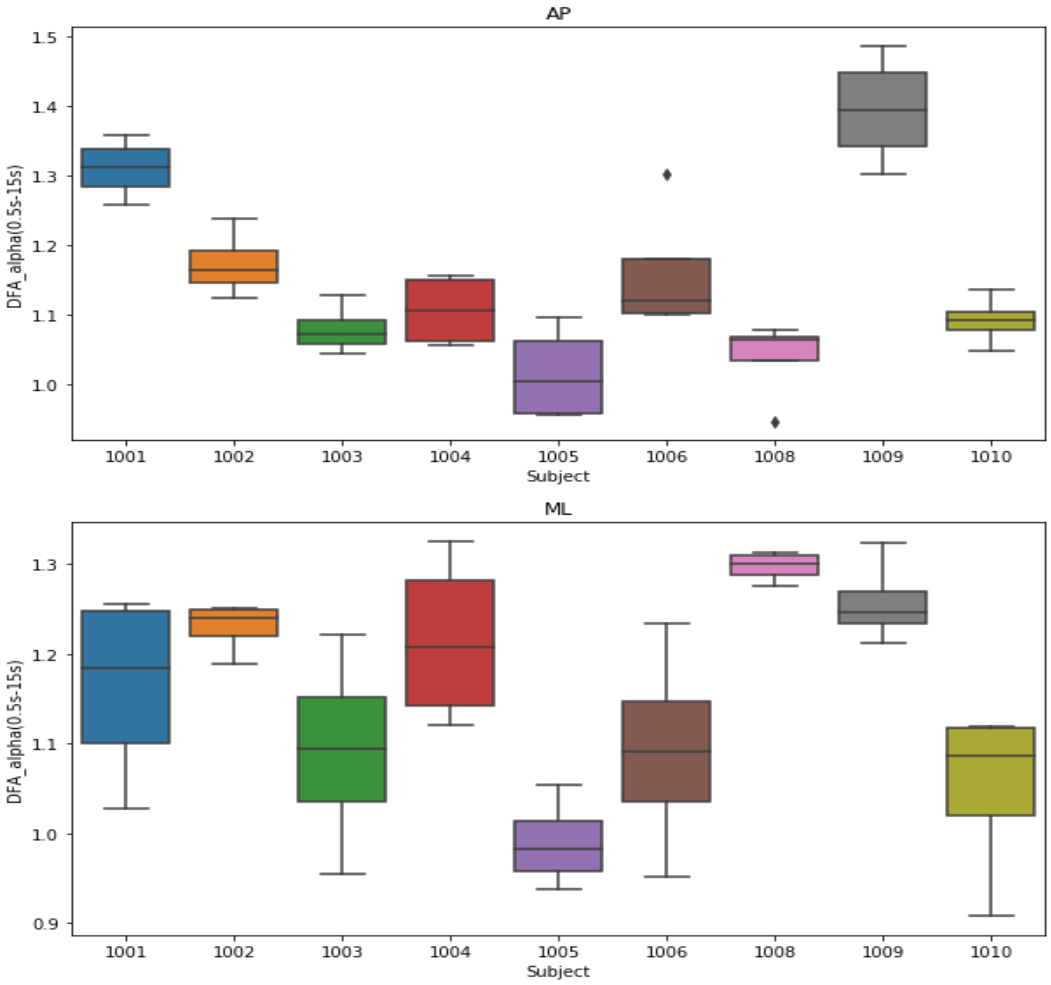


*Figure 6:DFA of the participants in the AP and ML*

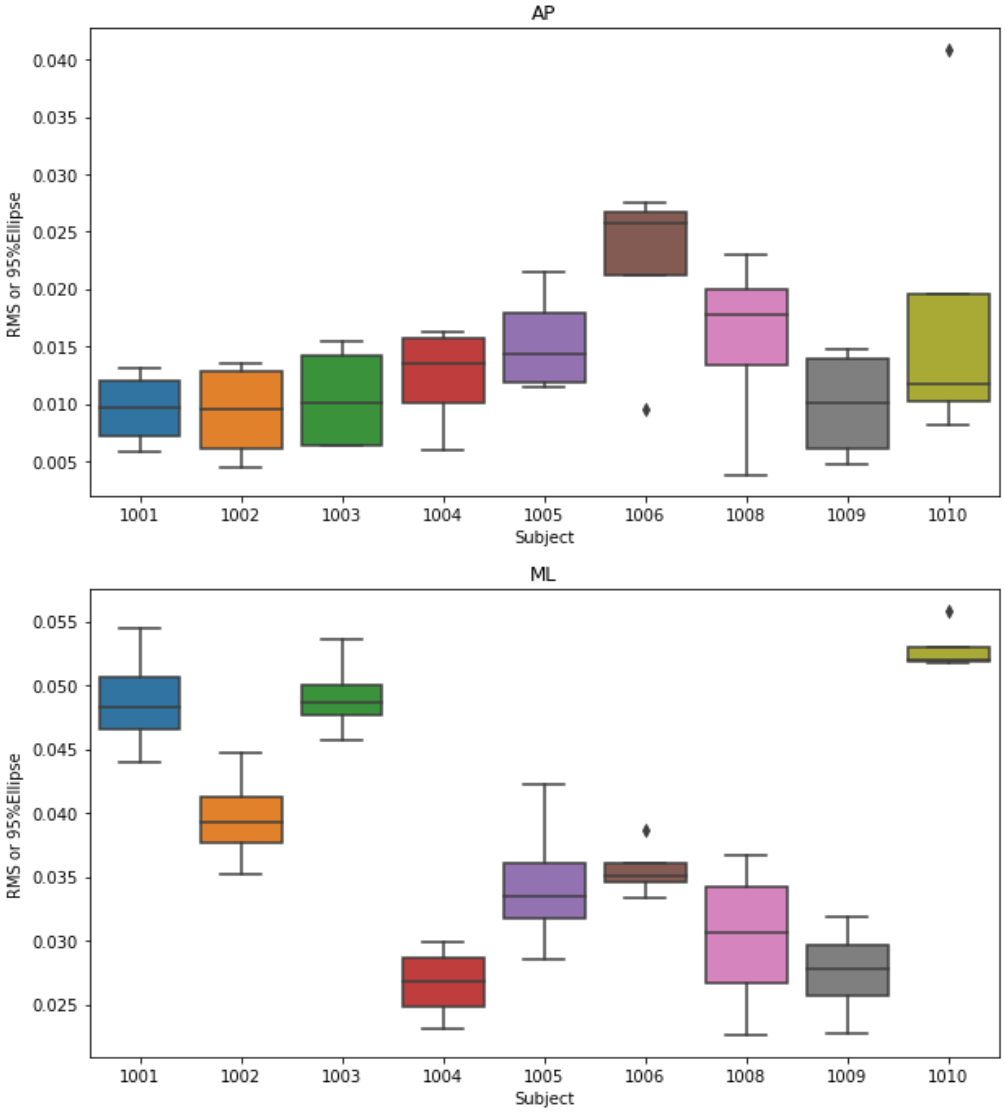


*Figure 7: RMS of the participants in the AP and ML*

| | Sample Entropy | DFA | RMS |
|---|---|---|---|
| AP | 0.2437 | 0.6958 | 0.2535 |
| ML | 0.1660 | 0.5085 | 0.5432 |

*Table 4: P-values for measure comparisons across sessions*

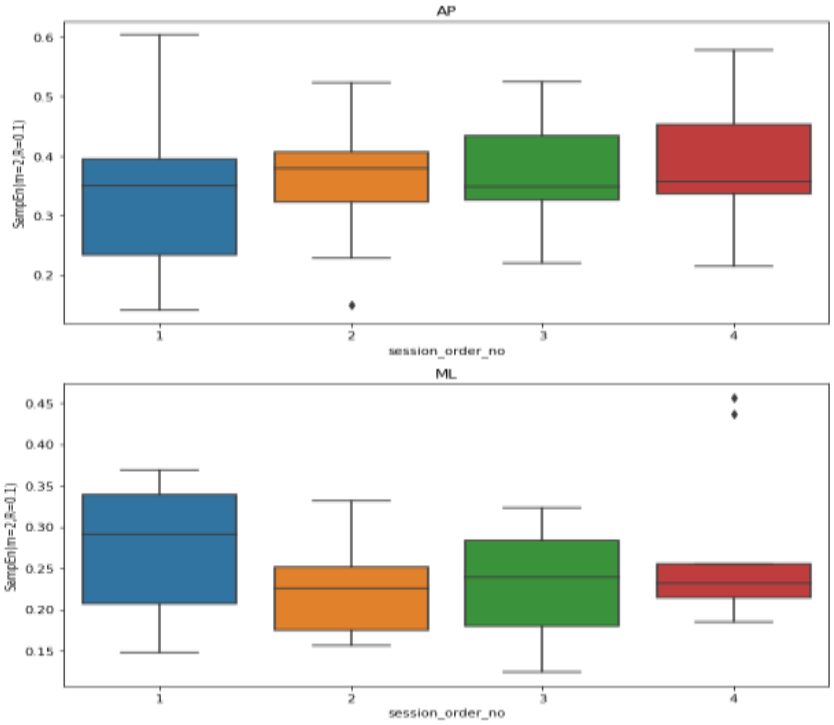


*Figure 8:Sample Entropy across sessions*

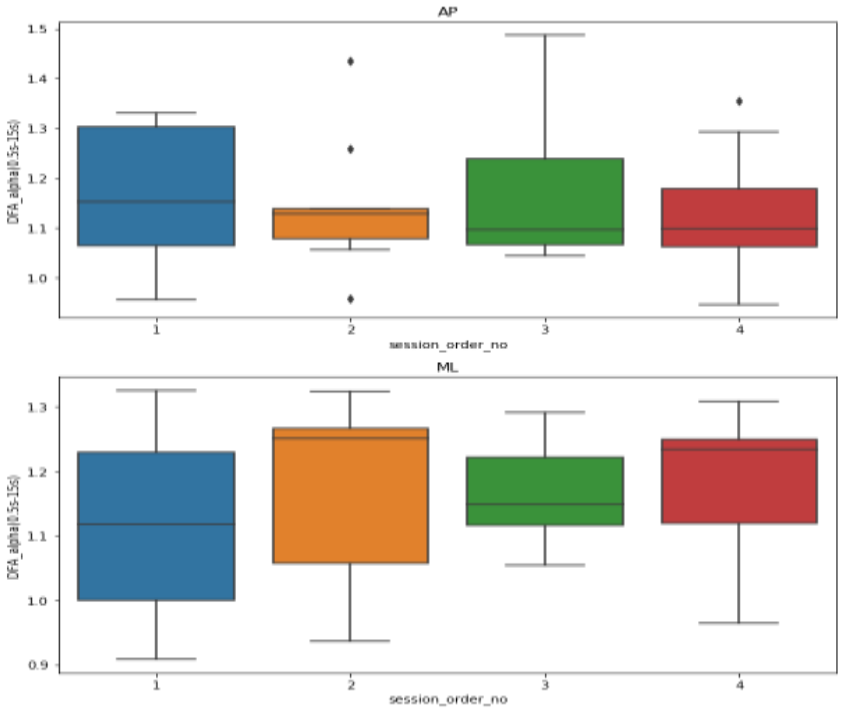


*Figure 9:DFA values across sessions*

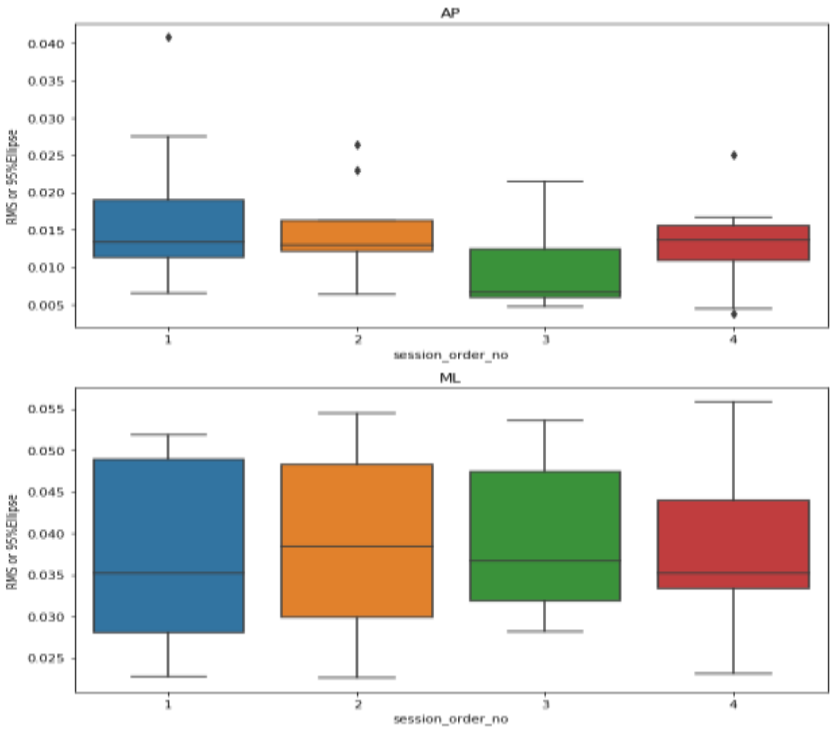


*Figure 10:RMS values across sessions*

| | Sample Entropy | DFA | RMS |
|---|---|---|---|
| measure $_{vibration}$ − measure $_{baseline}$ | 0.0087 | 0.2952 | 0.4182 |
| \|measure $_{vibration}$ − measure $_{baseline}$\| | 0.0937 | 0.9271 | 0.4733 |

*Table 5: P-values for change in measures across vibration treatment*

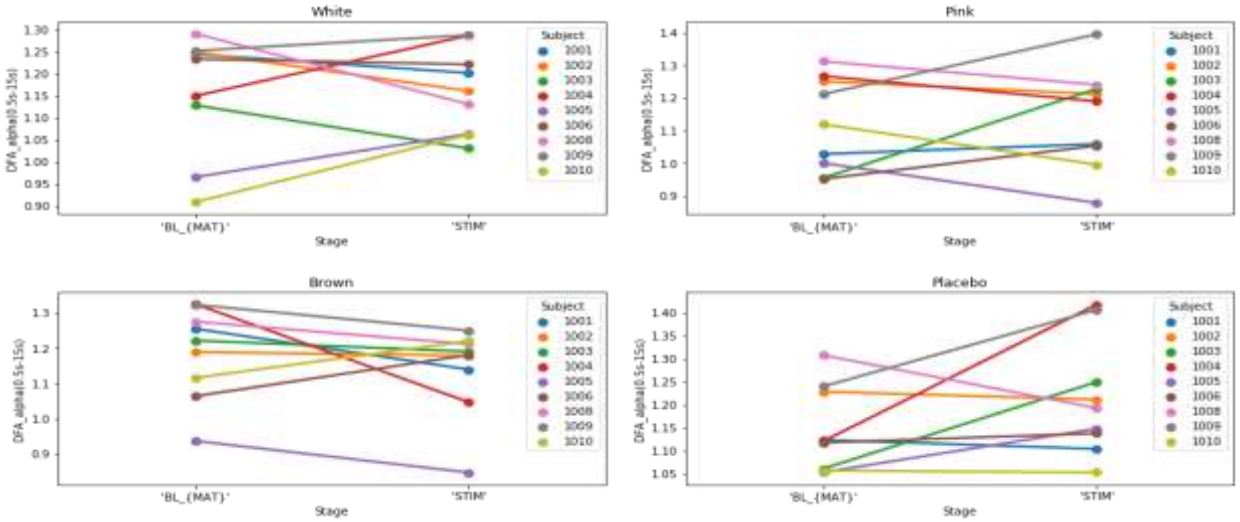


*Figure 11: Baseline and Stimulation DFA measures across participants in ML*

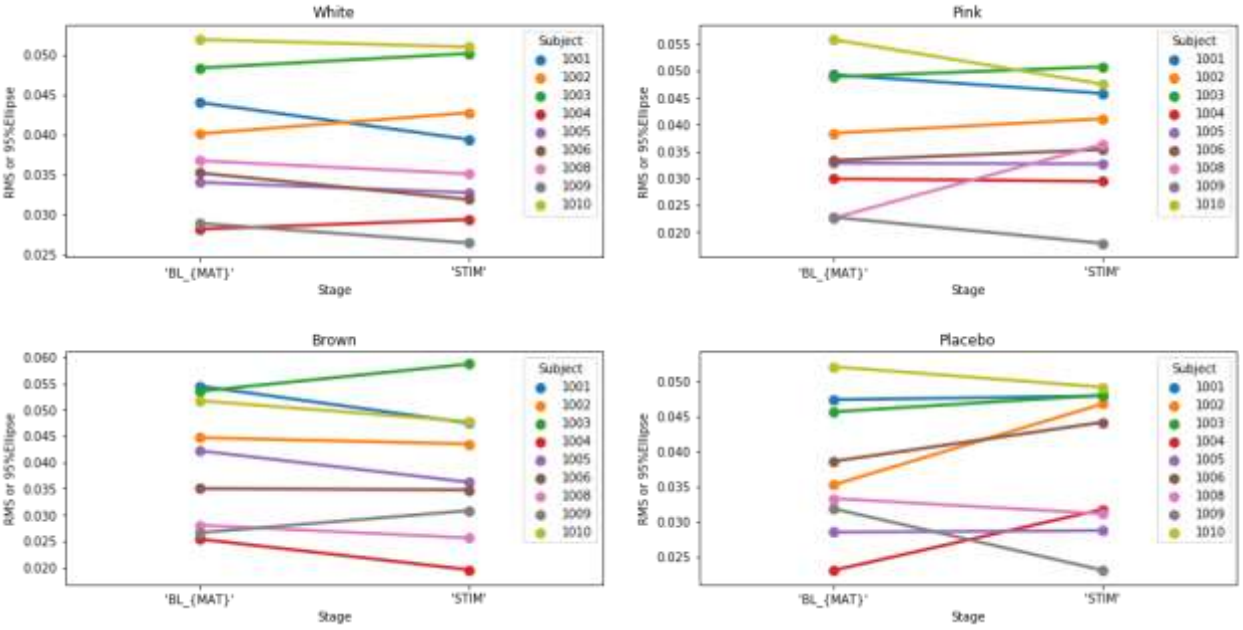


*Figure 12: Baseline and Stimulation RMS values across participants in ML*

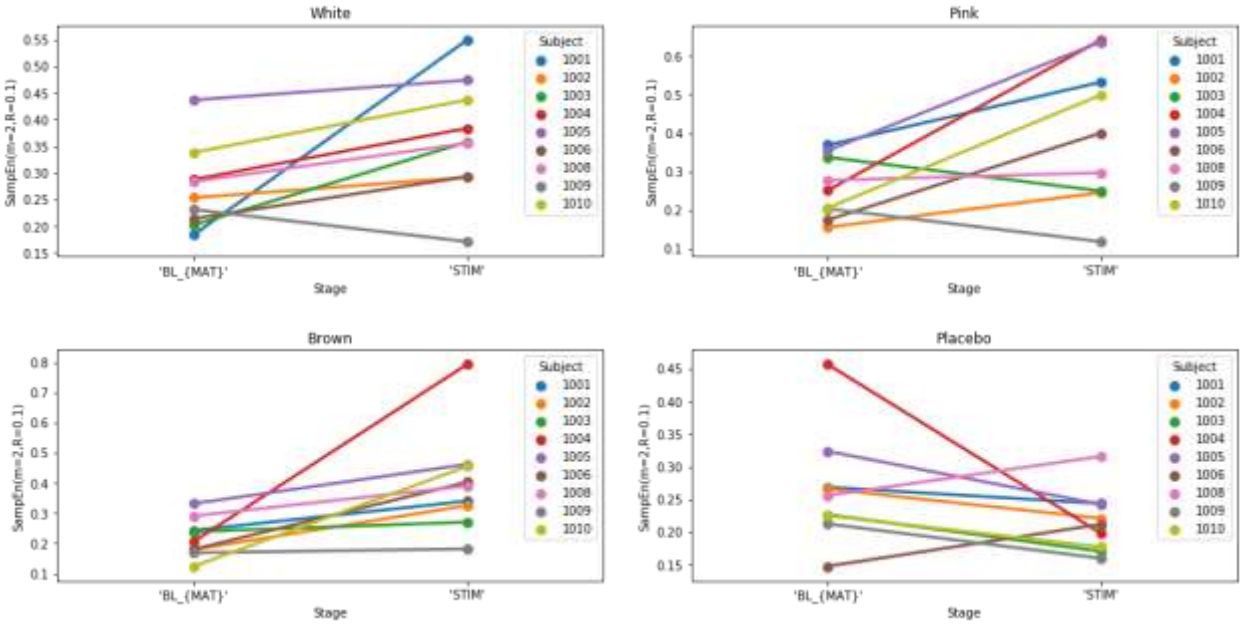


*Figure 13:Baseline and Stimulation sample entropy values across participants in ML*

| | | White | Pink | Brown | Placebo |
|---|---|---|---|---|---|
| Sample Entropy | slope | -0.5893 | -0.0699 | -0.8201 | -0.9320 |
| | p-value | 0.290 | 0.933 | 0.438 | 0.003 |
| | $R^2$ | 0.157 | 0.001 | 0.088 | 0.740 |
| DFA | slope | -0.5799 | -0.3944 | -0.4576 | -0.5361 |
| | p-value | 0.031 | 0.289 | 0.171 | 0.311 |

| | | | | | |
|---|---|---|---|---|---|
| | $R^2$ | 0.507 | 0.158 | 0.250 | 0.145 |
| RMS | slope | 0.0293 | -0.2677 | -0.0312 | -0.1436 |
| | p-value | 0.799 | 0.155 | 0.829 | 0.571 |
| | $R^2$ | 0.010 | 0.266 | 0.007 | 0.048 |

*Table 6: Linear regression parameters for the vibration colors for the different sway measures in the ML*

## Power Analysis

Power analysis was performed to determine the number of participants required to find statistically significant results using the study's data. Using 80% power, the needed sample size to find a significant difference between the effect of the vibration types in the ML direction was performed for each measure (RMS, SampEn, and DFA α). Table 8 summarize the needed sample size for each case. Instances where the needed sample size is more than 100 were marked as N/A.

| | White/Brown | White/Pink | Placebo/White | Placebo/Brown | Placebo/Pink |
|---|---|---|---|---|---|
| Sample Entropy | 17 | 54 | 6 | 4 | 5 |
| DFA | 39 | N/A | 25 | 11 | 64 |
| RMS | 30 | 52 | 9 | 14 | N/A |

*Table 7. Needed Sample Size at 80% power*

| | Vibration type | Increase | Decrease |
|---|---|---|---|
| Sample Entropy | White | 8 | 1 |
| | Pink | 7 | 2 |
| | Brown | 9 | 0 |
| | Placebo | 2 | 7 |
| DFA | White | 4 | 5 |
| | Pink | 4 | 5 |
| | Brown | 2 | 7 |
| | Placebo | 5 | 4 |
| RMS | White | 3 | 6 |
| | Pink | 4 | 5 |
| | Brown | 2 | 7 |
| | Placebo | 7 | 2 |

*Table 8:The direction in the change in state of sway measures for the different vibration treatments*

# Appendix B : MATLAB Codes

```
%Camilo Giraldo - Healthy Vibration Sway Study v2 - Data Visualization
%University of Kansas - Biodynamics Lab
%Last Update: April 16, 2021
%Edited by Zaccur Nkrumah , March 5, 2022

clear; close all; clc;

%% General
%File paths
path_raw='C:\Users\17856\Desktop\Vibration_Study\Healthy Vibration Sway v2\';
path_res='C:\Users\17856\Desktop\Vibration_Study\Results\';
path_pro='C:\Users\17856\Desktop\Vibration_Study\Processed Data\';

%Subject numbers
sub_no=[1001 1002 1003 1004 1005 1006 1008 1009 1010];

%Color names
protocol_colors_abv={'WH' 'PK' 'BR' 'PB'};
protocol_colors={'White' 'Pink' 'Brown' 'Placebo'};

%Protocol order
protocol_order_abv={'BL_EC_GND' 'BL_EC_MAT' 'THR_EC' 'THR' 'STIM_EC' 'T_0_EC'};

%Names of tabs for figures
fig_tabs={'BL: Time' 'BL: Spatial' 'BL-MAT: Time' 'BL-MAT: Spatial' 'THR: Time' 'THR:
Spatial' 'THR: Calculation' ...
    'STIM: Time' 'STIM: Spatial' 'T0: Time' 'T0: Spatial'};

%Names of the figures to be saved
fig_names={'BL_TimeSeries' 'BL_Spatial' 'BL-Mat_TimeSeries' 'BL-Mat_Spatial' ...
    'THR_TimeSeries' 'THR_Spatial' 'THR_Calculation' 'STIM_TimeSeries' 'STIM_Spatial'
...
    'T0_TimeSeries' 'T0_Spatial'};

%Names of the data to be saved
var_names={'Zeros' 'BL_GND' 'BL_MAT' 'THR_Sway' 'THR' 'STIM_Sway' ...
    'T_0_Sway' 'THR_Sway Mot-Butt' 'STIM_Sway Mot-Butt' 'T_0_Sway Mot-Butt'};


%% Generation Visualization of Data
%Going over all the subjects
for ii = 1:length(sub_no)

    %Preallocating space for subject raw data
    data_visual=cell(length(var_names),5);
    for jj = 1:length(var_names)
        data_visual{jj,1}=var_names{jj};
    end

    %Command window message
    fprintf('Subject: s%d\n',sub_no(ii));

    %Going over all the sessions
    for jj = 1:length(protocol_colors)

        %Reseting counter for data_visual
        count_data=1;

        %Command window message
```

```
        fprintf('\tSession: %s\n',protocol_colors{jj});
        fprintf('\t\tZeros: ');

        %Reading zeros file
        zeross=readmatrix([path_raw 's' num2str(sub_no(ii)) '\s' num2str(sub_no(ii))
...
            '_' protocol_colors_abv{jj} '_zeros3.txt']);
        zeross_check=0;
        for kk = 1:size(zeross,1)
            for LL = 2:size(zeross,2)
                if kk == 1 && isnan(zeross(kk,LL))
                    zeross(kk,LL)=zeross(kk+1,LL);
                elseif isnan(zeross(kk,LL))
                    fprintf('\n\t\t\tNaN at Row = %d, Column = %d',kk,LL);
                    zeross_check=1;
                end
            end
        end
        if zeross_check == 0
            fprintf(' Good!\n');
        else
            fprintf('\n');
        end
        zeross_mean=mean(zeross,1);
        zeross_stdv=std(zeross,0,1);

        %Saving zeros data
        data_visual{count_data,jj+1}=[zeross_mean(2:end); zeross_stdv(2:end)];
        count_data=count_data+1;

        %Opening figure for subject and session
        fig=figure('Name',['s' num2str(sub_no(ii)) ' - Session: '
protocol_colors{jj}],...
            'Units','Normalized','Outerposition',[0 0 1 1]);
        tabgp=uitabgroup('Parent',fig);
        tab=zeros(length(fig_tabs),1);
        for kk = 1:length(fig_tabs)
            tab(kk)=uitab('Parent',tabgp,'Title',fig_tabs{kk});
        end

        %Declaring limits for axes
        axis_ground_time_AP=[inf -inf]; axis_ground_time_ML=[inf -inf];
        axis_mat_time_AP=[inf -inf];    axis_mat_time_ML=[inf -inf];

        %%%% Going over the protocol order
        for kk = 1:length(protocol_order_abv)

            %Baseline on ground
            if kk == 1

                %Command window message ----- Baseline on Ground
                fprintf('\t\t%s-%s: ',protocol_order_abv{kk});

                %Reading current data
                data=readmatrix([path_raw 's' num2str(sub_no(ii)) '\s'
num2str(sub_no(ii)) ...
                    '_' protocol_colors_abv{jj} '_' protocol_order_abv{kk} '.txt']);
                data_check=0;
                for mm = 1:size(data,1)
                    for nn = 2:size(data,2)
                        if mm == 1 && isnan(data(mm,nn))
                            data(mm,nn)=data(mm+1,nn);
                        elseif isnan(data(mm,nn))
```

```
                            fprintf('\n\t\t\tNaN at Row = %d, Column = %d',mm,nn);
                            data_check=1;
                        end
                    end
                end
                if data_check == 0
                    fprintf(' Good!\n');
                else
                    fprintf('\n');
                end

                %Converting current data to N and N-m
                [data(:,2:7),dz_3364]=V2f_fp3364(data(:,2:7),zeross_mean(2:7),1000);
%Right foot on 3364

[data(:,8:13),dz_3477]=V2f_fp3477(data(:,8:13),zeross_mean(8:13),1000);     %Left foot
on 3477
                dz=mean([dz_3364 dz_3477]);

                %Rotating data around z-axis -90 degrees
                data=[data(:,1) ...
%Time
                    -data(:,3) data(:,2) data(:,4) -data(:,6) data(:,5) data(:,7) ...
%Right 3364
                    -data(:,9) data(:,8) data(:,10) -data(:,12) data(:,11)
data(:,13)]; %Left 3477

                %Combining force plates
                data_comb=Comb_fp3477_fp3364(data(:,8:13),data(:,2:7));
                data_comb=[data(:,1) data_comb];

                %Calculating COP_AP: +x facing forward
                COP(:,1)=-(data_comb(:,6)+data_comb(:,2)*dz)./data_comb(:,4);

                %Calculating COP_ML: +y right hand
                COP(:,2)= (data_comb(:,5)-data_comb(:,3)*dz)./data_comb(:,4);

                %Updating limits for time series and spatial plots
                if max(COP(:,1)) > axis_ground_time_AP(2)
                    axis_ground_time_AP(2)=max(COP(:,1));
                end
                if min(COP(:,1)) < axis_ground_time_AP(1)
                    axis_ground_time_AP(1)=min(COP(:,1));
                end
                if max(COP(:,2)) > axis_ground_time_ML(2)
                    axis_ground_time_ML(2)=max(COP(:,2));
                end
                if min(COP(:,2)) < axis_ground_time_ML(1)
                    axis_ground_time_ML(1)=min(COP(:,2));
                end

                %Plotting time series: COP_AP
                axes('Parent',tab(1));
                subplot(2,1,1); plot(data_comb(:,1),1000*COP(:,1),'-k');
                grid; xlabel('Time [s]'); ylabel('Back \leftarrow COP [mm] \rightarrow
Face');
                title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-BL_{GND}'
...
                    '-COP_{AP}-EC']); xlim([data_comb(1,1) data_comb(end,1)]);

                %Plotting time series: COP_ML
                subplot(2,1,2); plot(data_comb(:,1),1000*COP(:,2),'-k');
```

```
                grid; xlabel('Time [s]'); ylabel('Left \leftarrow COP [mm] \rightarrow
Right');
                title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-BL_{GND}'
...
                    '-COP_{ML}-EC']); xlim([data_comb(1,1) data_comb(end,1)]);

                %Plotting spatial: COP_AP vs. COP_ML
                axes('Parent',tab(2)); subplot(1,2,1);

phase_time(1000*COP(:,2),1000*COP(:,1),[],data_comb(:,1),1,'horizontal','southoutside'
)
                grid; xlabel('Left \leftarrow COP_{ML} [mm] \rightarrow Right');
                ylabel('Back \leftarrow COP_{AP} [mm] \rightarrow Face');
                title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-BL_{GND}'
...
                    '-COP-EC']); axis square;

                %Recording COP data
                data_visual{count_data,jj+1}=COP;
                count_data=count_data+1;

                %Clearing COP data
                clear COP

            elseif kk == 2 || kk == 3 || kk == 5 || kk == 6  %Sway on MAT

                %Command window message
                fprintf('\t\t%s Sway: ',protocol_order_abv{kk});

                %Reading current data
                data=readmatrix([path_raw 's' num2str(sub_no(ii)) '\s'
num2str(sub_no(ii)) ...
                    '_' protocol_colors_abv{jj} '_' protocol_order_abv{kk}
'_sway.txt']);
                data_check=0;
                for mm = 1:size(data,1)
                    for nn = 2:size(data,2)
                        if mm == 1 && isnan(data(mm,nn))
                            data(mm,nn)=data(mm+1,nn);
                        elseif isnan(data(mm,nn))
                            fprintf('\n\t\t\tNaN at Row = %d, Column = %d',mm,nn);
                            data_check=1;
                        end
                    end
                end
                if data_check == 0
                    fprintf(' Good!\n');
                else
                    fprintf('\n');
                end

                %Converting current data to N and N-m (4033)
                [data(:,2:7),dz]=V2f_fp4033(data(:,2:7),zeross_mean(14:19),1000);

                %Rotating data around z-axis 180 degrees
                data_comb=[data(:,1) ...
%Time
                    -data(:,2) -data(:,3) data(:,4) -data(:,5) -data(:,6) data(:,7)];
%4033

                %Calculating COP_AP: +x facing forward
                COP(:,1)=-(data_comb(:,6)+data_comb(:,2)*dz)./data_comb(:,4);
```

```
                %Calculating COP_ML: +y right hand
                COP(:,2)= (data_comb(:,5)-data_comb(:,3)*dz)./data_comb(:,4);

                %Updating limits for time series and spatial plots
                if max(COP(:,1)) > axis_mat_time_AP(2)
                    axis_mat_time_AP(2)=max(COP(:,1));
                end
                if min(COP(:,1)) < axis_mat_time_AP(1)
                    axis_mat_time_AP(1)=min(COP(:,1));
                end
                if max(COP(:,2)) > axis_mat_time_ML(2)
                    axis_mat_time_ML(2)=max(COP(:,2));
                end
                if min(COP(:,2)) < axis_mat_time_ML(1)
                    axis_mat_time_ML(1)=min(COP(:,2));
                end

                %Selecting tab for time series
                if kk == 2          %Baseline on mat
                    axes('Parent',tab(3));
                elseif kk == 3      %Threshold sway
                    axes('Parent',tab(5));
                elseif kk == 5      %Stimulus
                    axes('Parent',tab(8));
                elseif kk == 6      %T0
                    axes('Parent',tab(10));
                end

                %Plotting time series: COP_AP
                subplot(2,1,1); plot(data_comb(:,1),1000*COP(:,1),'-k');
                grid; xlabel('Time [s]'); ylabel('Back \leftarrow COP [mm] \rightarrow
Face');
                if kk == 2          %Baseline on mat
                    title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-' ...
                        protocol_order_abv{kk}(1:2) '_{' protocol_order_abv{kk}(7:end)
'}' ...
                        '-COP_{AP}-EC']); xlim([data_comb(1,1) data_comb(end,1)]);
                elseif kk == 3      %Threshold sway
                    title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-' ...
                        protocol_order_abv{kk}(1:3) ...
                        '-COP_{AP}-EC']); xlim([data_comb(1,1) data_comb(end,1)]);
                elseif kk == 5      %Stimulus
                    title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-' ...
                        protocol_order_abv{kk}(1:4) ...
                        '-COP_{AP}-EC']); xlim([data_comb(1,1) data_comb(end,1)]);
                elseif kk == 6      %T0
                    title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-' ...
                        protocol_order_abv{kk}(1:3) ...
                        '-COP_{AP}-EC']); xlim([data_comb(1,1) data_comb(end,1)]);
                end
                hold on;

                %Plotting time series: COP_ML
                subplot(2,1,2); plot(data_comb(:,1),1000*COP(:,2),'-k');
                grid; xlabel('Time [s]'); ylabel('Left \leftarrow COP [mm] \rightarrow
Right');
                if kk == 2          %Baseline on mat
                    title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-' ...
                        protocol_order_abv{kk}(1:2) '_{' protocol_order_abv{kk}(7:end)
'}' ...
                        '-COP_{ML}-EC']); xlim([data_comb(1,1) data_comb(end,1)]);
                elseif kk == 3      %Threshold sway
                    title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-' ...
```

```
                        protocol_order_abv{kk}(1:3) ...
                        '-COP_{ML}-EC']); xlim([data_comb(1,1) data_comb(end,1)]);
                elseif kk == 5      %Stimulus
                    title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-' ...
                        protocol_order_abv{kk}(1:4) ...
                        '-COP_{ML}-EC']); xlim([data_comb(1,1) data_comb(end,1)]);
                elseif kk == 6      %T0
                    title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-' ...
                        protocol_order_abv{kk}(1:3) ...
                        '-COP_{ML}-EC']); xlim([data_comb(1,1) data_comb(end,1)]);
                end
                hold on;

                %Selecting tab for spatial plots
                if kk == 2          %Baseline on mat
                    axes('Parent',tab(4));
                elseif kk == 3      %Threshold sway
                    axes('Parent',tab(6));
                elseif kk == 5      %Stimulus
                    axes('Parent',tab(9));
                elseif kk == 6      %T0
                    axes('Parent',tab(11));
                end

                %Plotting spatial plots: COP_AP vs. COP_ML
                subplot(1,2,1);

phase_time(1000*COP(:,2),1000*COP(:,1),[],data_comb(:,1),1,'horizontal','southoutside'
)
                grid; xlabel('Left \leftarrow COP_{ML} [mm] \rightarrow Right');
                ylabel('Back \leftarrow COP_{AP} [mm] \rightarrow Face');
                if kk == 2          %Baseline on mat
                    title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-' ...
                        protocol_order_abv{kk}(1:2) '_{' protocol_order_abv{kk}(7:end)
'}' ...
                        '-COP-EC']);
                elseif kk == 3      %Threshold sway
                    title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-' ...
                        protocol_order_abv{kk}(1:3) ...
                        '-COP-EC']);
                elseif kk == 5      %Stimulus
                    title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-' ...
                        protocol_order_abv{kk}(1:4) ...
                        '-COP-EC']);
                elseif kk == 6      %T0
                    title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-' ...
                        protocol_order_abv{kk}(1:3) ...
                        '-COP-EC']);
                end
                 axis square; hold on;

                %Recording COP data
                data_visual{count_data,jj+1}=COP;
                count_data=count_data+1;

                %Clearing COP data
                clear COP

            else                        %EC 421 Threshold calculator

                %Command window message
                fprintf('\t\t%s: ',protocol_order_abv{kk});
```

```
                %Reading current data
                data=readmatrix([path_raw 's' num2str(sub_no(ii)) '\s'
num2str(sub_no(ii)) ...
                    '_' protocol_colors_abv{jj} '_' protocol_order_abv{kk} '.txt'],...
                    'OutputType','string','Range',1);

                %Selecting tab for time series
                axes('Parent',tab(7));

                %421 Threshold calculation and plot
                [THR_data,THR_value,THR_value_n,motor]=THR_421_Calculator(data);
                title(['s' num2str(sub_no(ii)) '-421' protocol_colors_abv{jj} ...
                    '-' protocol_order_abv{kk}(1:3) ' | Motor ' motor ' | THR ' ...
                    num2str(THR_value) '/127 = ' num2str(round(THR_value/127*100)) '%
| 90% THR = ' ...
                    num2str(round(0.9*THR_value/127*100)) '%']);

                %Command window message
                if ~isnan(THR_value)
                    fprintf(' Good!\n');
                end

                %Recording 421 information
                %Recording COP data
                data_visual{count_data,jj+1}={THR_data,THR_value,THR_value_n,motor};
                count_data=count_data+1;

            end

        end

        %Adjusting the axis limits of the ground force plate data
        for kk = 1:length(protocol_order_abv)

            %Baseline on ground
            if kk == 1

                %Applying axis limits to COP_AP and COP_ML
                axes('Parent',tab(1));
                subplot(2,1,1); ylim(1000*axis_ground_time_AP);
                subplot(2,1,2); ylim(1000*axis_ground_time_ML);

                %Applying axis limits to COP_AP vs. COP_ML
                axes('Parent',tab(2)); subplot(1,2,1);
                axis(1000*[axis_ground_time_ML axis_ground_time_AP]);

            elseif kk == 2 || kk == 3 || kk == 5 || kk == 6  %Sway on MAT

                %Selecting tab for time series
                if kk == 2          %Baseline on mat
                    axes('Parent',tab(3));
                elseif kk == 3      %Threshold sway
                    axes('Parent',tab(5));
                elseif kk == 5      %Stimulus
                    axes('Parent',tab(8));
                elseif kk == 6      %T0
                    axes('Parent',tab(10));
                end

                %Applying axis limits to COP_AP and COP_ML
                subplot(2,1,1); ylim(1000*axis_mat_time_AP);
                subplot(2,1,2); ylim(1000*axis_mat_time_ML);
```

```
                %Motbutt stuff
                if kk >= 3

                    %Command window message
                    fprintf('\t\t%s Mot-Butt: ',protocol_order_abv{kk});

                    %Reading mot_butt data
                    data=readmatrix([path_raw 's' num2str(sub_no(ii)) '\s'
num2str(sub_no(ii)) ...
                        '_' protocol_colors_abv{jj} '_' protocol_order_abv{kk}
'_motbutt.txt']);
                    if kk == 3
                        idx_nan=[];
                        for mm = 1:size(data,1)
                            if sum(isnan(data(mm,2:end))) == 7
                                idx_nan=[idx_nan; mm];
                            end
                        end
                        data(idx_nan,:)=[];
                    end
                    data_check=0;
                    for mm = 1:size(data,1)
                        for nn = 2:size(data,2)
                            if mm == 1 && isnan(data(mm,nn))
                                data(mm,nn)=data(mm+1,nn);
                            elseif isnan(data(mm,nn))
                                fprintf('\n\t\t\tNaN at Row = %d, Column = %d',mm,nn);
                                data_check=1;
                            end
                        end
                    end
                    if data_check == 0
                        fprintf(' Good!\n');
                    else
                        fprintf('\n');
                    end

                    %Plotting when motors were ON
                    Motor_OFF_ON=zeros(size(data,1),2);
                    for LL = 1:size(data,1)

                        %Checking if more motors were ON, and plotting based on it
                        if sum(data(LL,2:7) > 3) >= 1            %Motor is ON
                            Motor_OFF_ON(LL,1)=NaN; Motor_OFF_ON(LL,2)=1000;
                        else                                     %Motor is OFF
                            Motor_OFF_ON(LL,1)=1000; Motor_OFF_ON(LL,2)=NaN;
                        end

                    end
                    subplot(2,1,1);
plot(data(:,1),Motor_OFF_ON(:,1)*axis_mat_time_AP(2),...
                        '.r','MarkerSize',4); hold on;
                    subplot(2,1,1);
plot(data(:,1),Motor_OFF_ON(:,2)*axis_mat_time_AP(2),...
                        '.g','MarkerSize',4); hold on;

                    %Plotting when button was pressed
                    Button_OFF_ON=zeros(size(data,1),2);
                    for LL = 1:size(data,1)

                        %Checking if all motors were OFF, and plotting based on it
                        if data(LL,8) > 4        %Button is not pressed
                            Button_OFF_ON(LL,1)=1000; Button_OFF_ON(LL,2)=NaN;
```

```
                        else                                %Motor is ON
                            Button_OFF_ON(LL,1)=NaN; Button_OFF_ON(LL,2)=1000;
                        end

                    end
                    %Correcting error for s1002, WH, T0
                    if ii == 2 && jj == 1 && kk == 6
                        Button_OFF_ON(:,1)=1000; Button_OFF_ON(:,2)=NaN;
                    end
                    subplot(2,1,1);
plot(data(:,1),Button_OFF_ON(:,1)*axis_mat_time_AP(1),...
                        '.r','MarkerSize',4); hold on;
                    subplot(2,1,1);
plot(data(:,1),Button_OFF_ON(:,2)*axis_mat_time_AP(1),...
                        '.g','MarkerSize',4); hold on;

                end

                %Selecting tab for spatial plots
                if kk == 2          %Baseline on mat
                    axes('Parent',tab(4));
                elseif kk == 3      %Threshold sway
                    axes('Parent',tab(6));
                elseif kk == 5      %Stimulus
                    axes('Parent',tab(9));
                elseif kk == 6      %T0
                    axes('Parent',tab(11));
                end

                %Applying axis limits to COP_AP vs. COP_ML
                subplot(1,2,1); axis(1000*[axis_mat_time_ML axis_mat_time_AP]);

                %Recording motor and button information
                if kk >= 3

data_visual{count_data,jj+1}={Motor_OFF_ON/1000,Button_OFF_ON/1000};
                    count_data=count_data+1;
                end

            end

        end

        %Saving each tab as a figure
        for kk = 1:length(fig_tabs)
            tabgp.SelectedTab = tab(kk);
            saveas(fig,[path_res '1 Data Visualization\s' num2str(sub_no(ii)) '\s'
num2str(sub_no(ii)) ...
                '_' protocol_colors_abv{jj} '_' num2str(kk) '_' fig_names{kk}
'.jpeg']);
        end
        close all;

    end

    %Saving data_visual for subject
    save([path_pro 's' num2str(sub_no(ii)) '_DataVisual.mat'],'data_visual');

end
```

```
%Camilo Giraldo - Healthy Vibratio Sway Study v2 - Data Check (PSD)
%University of Kansas - Biodynamics Lab
%Last Update: April 18, 2021

clear; close all; clc;

%% General
%File paths
path_raw='C:\Users\17856\Desktop\Final_Vibration_study\Vibrotactile Study (2021)\';
path_res='C:\Users\17856\Desktop\Final_Vibration_study\Results\';

%Subject numbers
sub_no=[1001 1002 1003 1004 1005 1006 1008 1009 1010 2001:2003];

%Color names
protocol_colors_abv={'WH' 'PK' 'BR' 'PB'};
protocol_colors={'White' 'Pink' 'Brown' 'Placebo'};

%Protocol order
protocol_order_abv={'BL_EC_GND' 'BL_EC_MAT' 'THR_EC' 'THR' 'STIM_EC' 'T_0_EC'};

%PSD frequency [Ground and Mat]
freq_PSD_ground=(0:0.1:50)';
freq_PSD_mat=(0:0.1:450)';

%Surfaces where the subjects stood
protocol_surfaces={'Ground' 'Mat'};

%PSD plot titles
plot_titles={'My' 'Fx'; 'Mx' 'Fy'; 'Mz' 'Fz'};

%Indexes to grab variables in PSD plots
idx=[5 1; 4 2; 6 3];

%CED frequencies [Ground and Mat]
freq_CED=[100 2500];

%X-tick labels for color plots
ytick_array_ground=0:1:(length(sub_no)-1);
yticklabels_string_ground=cell(1,length(ytick_array_ground));
ytick_array_mat=0:1:(length(sub_no)*4-1);
yticklabels_string_mat=cell(1,length(ytick_array_mat));

%% PSD on All Subjects' Data
%Preallocating space
data_all_PSD_ground=cell(length(protocol_colors),length(sub_no));
data_all_PSD_mat=cell(length(protocol_colors),length(sub_no)*4);

%Going over all the subjects
for ii = 1:length(sub_no)

    %Command window message
    fprintf('Subject: s%d\n',sub_no(ii));

    %Going over all the sessions
    for jj = 1:length(protocol_colors)

        %Reseting counters
        count_mat=1;
```

```
        %Command window message
        fprintf('\tSession: %s\n',protocol_colors{jj});
        fprintf('\t\tZeros: ');

        %Reading zeros file
        zeross=readmatrix([path_raw 's' num2str(sub_no(ii)) '\s' num2str(sub_no(ii))
...
            '_' protocol_colors_abv{jj} '_zeros3.txt']);
        zeross_check=0;
        for kk = 1:size(zeross,1)
            for LL = 2:size(zeross,2)
                if kk == 1 && isnan(zeross(kk,LL))
                    zeross(kk,LL)=zeross(kk+1,LL);
                elseif isnan(zeross(kk,LL))
                    fprintf('\n\t\t\tNaN at Row = %d, Column = %d',kk,LL);
                    zeross_check=1;
                end
            end
        end
        if zeross_check == 0
            fprintf(' Good!\n');
        else
            fprintf('\n');
        end
        zeross_mean=mean(zeross,1);

        %%%% Going over the protocol order
        for kk = 1:length(protocol_order_abv)

            %Baseline on ground
            if kk == 1

                %Command window message ----- Baseline on Ground
                fprintf('\t\t%s-%s: ',protocol_order_abv{kk});

                %Reading current data
                data=readmatrix([path_raw 's' num2str(sub_no(ii)) '\s'
num2str(sub_no(ii)) ...
                    '_' protocol_colors_abv{jj} '_' protocol_order_abv{kk} '.txt']);
                data_check=0;
                for mm = 1:size(data,1)
                    for nn = 2:size(data,2)
                        if mm == 1 && isnan(data(mm,nn))
                            data(mm,nn)=data(mm+1,nn);
                        elseif isnan(data(mm,nn))
                            fprintf('\n\t\t\tNaN at Row = %d, Column = %d',mm,nn);
                            data_check=1;
                        end
                    end
                end
                if data_check == 0
                    fprintf(' Good!\n');
                else
                    fprintf('\n');
                end

                %Converting current data to N and N-m
                [data(:,2:7),dz_3364]=V2f_fp3364(data(:,2:7),zeross_mean(2:7),1000);
%Right foot on 3364

[data(:,8:13),dz_3477]=V2f_fp3477(data(:,8:13),zeross_mean(8:13),1000);     %Left foot
on 3477
                dz=mean([dz_3364 dz_3477]);
```

```
                %Rotating data around z-axis -90 degrees
                data=[data(:,1) ...
%Time
                    -data(:,3) data(:,2) data(:,4) -data(:,6) data(:,5) data(:,7) ...
%Right 3364
                    -data(:,9) data(:,8) data(:,10) -data(:,12) data(:,11)
data(:,13)]; %Left 3477

                %Combining force plates
                data_comb=Comb_fp3477_fp3364(data(:,8:13),data(:,2:7));
                data_comb=[data(:,1) data_comb];

                %PSD on combined force plate data
                data_comb_PSD=zeros(length(freq_PSD_ground),3);
                for mm = 1:size(data_comb,2)-1

data_comb_PSD(:,mm)=periodogram(data_comb(:,mm+1),[],freq_PSD_ground,freq_CED(1));
                end

                %Recording PSD of combined force plate
                data_all_PSD_ground{jj,ii}=data_comb_PSD;

                %Recording ticks labels for x-axis of color plots
                yticklabels_string_ground{ii}=['s' num2str(sub_no(ii))];

            elseif kk == 2 || kk == 3 || kk == 5 || kk == 6  %Sway on MAT

                %Command window message
                fprintf('\t\t%s Sway: ',protocol_order_abv{kk});

                %Reading current data
                data=readmatrix([path_raw 's' num2str(sub_no(ii)) '\s'
num2str(sub_no(ii)) ...
                    '_' protocol_colors_abv{jj} '_' protocol_order_abv{kk}
'_sway.txt']);
                data_check=0;
                for mm = 1:size(data,1)
                    for nn = 2:size(data,2)
                        if mm == 1 && isnan(data(mm,nn))
                            data(mm,nn)=data(mm+1,nn);
                        elseif isnan(data(mm,nn))
                            fprintf('\n\t\t\tNaN at Row = %d, Column = %d',mm,nn);
                            data_check=1;
                        end
                    end
                end
                if data_check == 0
                    fprintf(' Good!\n');
                else
                    fprintf('\n');
                end

                %Converting current data to N and N-m (4033)
                [data(:,2:7),dz]=V2f_fp4033(data(:,2:7),zeross_mean(14:19),1000);

                %Rotating data around z-axis 180 degrees
                data_comb=[data(:,1) ...
%Time
                    -data(:,2) -data(:,3) data(:,4) -data(:,5) -data(:,6) data(:,7)];
%4033

                %PSD on force plate data
```

```
                data_comb_PSD=zeros(length(freq_PSD_mat),3);
                for mm = 1:size(data_comb,2)-1

data_comb_PSD(:,mm)=periodogram(data_comb(:,mm+1),[],freq_PSD_mat,freq_CED(2));
                end

                %Recording PSD of force plate
                data_all_PSD_mat{jj,4*(ii-1)+count_mat}=data_comb_PSD;

                %Recording ticks labels for x-axis of color plots
                if kk == 6
                    yticklabels_string_mat{4*(ii-1)+count_mat}=['s'
num2str(sub_no(ii))];
                end

                %Updating counter
                count_mat=count_mat+1;

            end

        end

    end

end

%% Plotting PSD Plots for All Subjects
%Destroying unnecessaty yticks
idx_empty=[]; ytick_array_ground_new=ytick_array_ground;
for ii = 1:length(ytick_array_ground)
    if isempty(yticklabels_string_ground{ii})
        idx_empty=[idx_empty; ii];
    end
end
ytick_array_ground_new(idx_empty)=[];
yticklabels_string_ground(idx_empty)=[];

idx_empty=[]; ytick_array_mat_new=ytick_array_mat;
for ii = 1:length(ytick_array_mat)
    if isempty(yticklabels_string_mat{ii})
        idx_empty=[idx_empty; ii];
    end
end
ytick_array_mat_new(idx_empty)=[];
yticklabels_string_mat(idx_empty)=[];

%Making x-ticks for mat surfaces
xtick_array_mat=[0.1 0.3 0.5 1 2:2:10 20 40 60 100 200 400];
xticklabels_string_mat=cell(1,length(xtick_array_mat));
for ii = 1:length(xtick_array_mat)
    xticklabels_string_mat{ii}=num2str(xtick_array_mat(ii));
end

%Going both surfaces (Ground and Mat)
for ii = 1:length(protocol_surfaces)

    %Grabbing the correct data set
    if ii == 1      %Ground
        data=data_all_PSD_ground;
    else            %Mat
        data=data_all_PSD_mat;
    end
```

```
    %Making meshgrid for surfaces
    if ii == 1      %Ground
        [X,Y]=meshgrid([freq_PSD_ground(2:end);
freq_PSD_ground(end)+diff(freq_PSD_ground([1 2]))],...
            [ytick_array_ground ytick_array_ground(end)+diff(ytick_array_ground([1
2]))]);
    else            %Mat
        [X,Y]=meshgrid([freq_PSD_mat(2:end); freq_PSD_mat(end)+diff(freq_PSD_ground([1
2]))],...
            [ytick_array_mat ytick_array_mat(end)+diff(ytick_array_mat([1 2]))]);
    end

    %Going over all the sessions
    for jj = 1:length(protocol_colors)

        %Going over all the force plate's axis groups
        for kk = 1:size(plot_titles,1)

            %Setting up figure and getting indexes to use
            fig=figure('Name',['Session ' protocol_colors{jj} ' | Healthy Vibration
Sway v2 | '...
                'Data Check PSD: ' plot_titles{kk,1} ' & ' plot_titles{kk,2} ' | ' ...
                protocol_surfaces{ii}],'Units','Normalized','Outerposition',[0 0 1
1]);

            %Preallocating space for surface Z-axis
            if ii == 1          %Ground

Zm=zeros(length(ytick_array_ground)+1,length(freq_PSD_ground(2:end))+1);
                Zf=Zm;
            else                %Mat
                Zm=zeros(length(ytick_array_mat)+1,length(freq_PSD_mat(2:end))+1);
                Zf=Zm;
            end

            %Going over all frequencies
            for LL = 1:size(X,2)

                %Not using the first frequency (0 Hz)
                if LL == 1
                    continue
                end

                %Going over all subjects' sway data
                for mm = 1:size(X,1)

                    %Creating Z-surface variables
                    if LL == size(X,2) && mm == size(X,1)
                        Zm(mm,LL-1)=data{jj,mm-1}(LL-1,idx(kk,1));
                        Zf(mm,LL-1)=data{jj,mm-1}(LL-1,idx(kk,2));
                    elseif mm == size(X,1)
                        Zm(mm,LL-1)=data{jj,mm-1}(LL,idx(kk,1));
                        Zf(mm,LL-1)=data{jj,mm-1}(LL,idx(kk,2));
                    elseif LL == size(X,2)
                        Zm(mm,LL-1)=data{jj,mm}(LL-1,idx(kk,1));
                        Zf(mm,LL-1)=data{jj,mm}(LL-1,idx(kk,2));
                    else
                        Zm(mm,LL-1)=data{jj,mm}(LL,idx(kk,1));
                        Zf(mm,LL-1)=data{jj,mm}(LL,idx(kk,2));
                    end

                end
```

```
            end

            %Plotting surfaces
            for LL = 1:length(protocol_surfaces)

                %Selecting Z data
                if LL == 1
                    Z=Zm;
                else
                    Z=Zf;
                end

                %Plotting surface
                subplot(1,2,LL); surf(X,Y,Z,'EdgeColor','flat','FaceColor','flat');
                xlim([X(1,1) X(1,end)]); ylim([Y(1,1) Y(end,1)]); zlim([min(Z(:))
max(Z(:))]);
                set(gca,'ZScale','log');
                colormap jet; xlabel('Freq [Hz]'); ylabel(['Subjects '
protocol_surfaces{ii} ' Sway']);
                title([protocol_colors{jj} ' | ' plot_titles{kk,LL} ' | '
protocol_surfaces{ii}]);
                set(gca,'ColorScale','log'); c=colorbar('SouthOutside');
c.Label.String='PSD [dB/Hz]';
                if ii == 1
                    yticks(ytick_array_ground_new);
yticklabels(yticklabels_string_ground);
                    view([52.5 30]);
                else
                    yticks(ytick_array_mat_new); yticklabels(yticklabels_string_mat);
                    xticks(xtick_array_mat); xticklabels(xticklabels_string_mat);
                    set(gca,'XScale','log'); view(2);
                end

            end

            %Saving figure
            saveas(fig,[path_res '2 Data Check - PSD\sAll_' protocol_colors_abv{jj}
'_DataCheck_PSD_' ...
                plot_titles{kk,1} '&' plot_titles{kk,2} '_' protocol_surfaces{ii}
'.jpeg']);
            close all;

        end

    end

end
```

```
%Camilo Giraldo - Healthy Vibratio Sway Study v2 - Data Check (Filt)
%University of Kansas - Biodynamics Lab
%Last Update: April 18, 2021

clear; close all; clc;

%% General
%File paths
path_raw='C:\Users\17856\Desktop\Final_Vibration_study\Vibrotactile Study (2021)\';
```

```
path_res='C:\Users\17856\Desktop\Final_Vibration_study\Results\';
path_pro='C:\Users\17856\Desktop\Final_Vibration_study\Processed Data\';


%Subject numbers
sub_no=[1001 1002 1003 1004 1005 1006 1008 1009 1010 2001:2003];

%Color names
protocol_colors_abv={'WH' 'PK' 'BR' 'PB'};
protocol_colors={'White' 'Pink' 'Brown' 'Placebo'};

%Protocol order
protocol_order_abv={'BL_EC_GND' 'BL_EC_MAT' 'THR_EC' 'THR' 'STIM_EC' 'T_0_EC'};

%Low pass frequency
freq_LP=20;

%Band padd frequency
freq_BP=[20 400];

%CED Frequencies
freq_CED=[100 2500];

%Names of tabs for figures
fig_tabs={'BL: Time' 'BL: Spatial' 'BL-MAT: Time' 'BL-MAT: Spatial' 'THR: Time' 'THR:
Spatial' 'THR: Calculation' ...
    'STIM: Time' 'STIM: Spatial' 'T0: Time' 'T0: Spatial'};

%Names of the figures to be saved
fig_names={'BL_TimeSeries' 'BL_Spatial' 'BL-Mat_TimeSeries' 'BL-Mat_Spatial' ...
    'THR_TimeSeries' 'THR_Spatial' 'THR_Calculation' 'STIM_TimeSeries' 'STIM_Spatial'
...
    'T0_TimeSeries' 'T0_Spatial'};

%Names of the data to be saved
var_names={'Zeros' 'BL_GND' 'BL_MAT' 'THR_Sway' 'THR' 'STIM_Sway' ...
    'T_0_Sway' 'THR_Sway Mot-Butt' 'STIM_Sway Mot-Butt' 'T_0_Sway Mot-Butt'};

%% Generation Visualization of Data
%Going over all the subjects
for ii = 1:length(sub_no)

    %Preallocating space for subject filtered data
    data_filt=cell(length(var_names),5);
    for jj = 1:length(var_names)
        data_filt{jj,1}=var_names{jj};
    end

    %Command window message
    fprintf('Subject: s%d\n',sub_no(ii));

    %Going over all the sessions
    for jj = 1:length(protocol_colors)

        %Reseting counter for data_visual
        count_data=1;

        %Command window message
        fprintf('\tSession: %s\n',protocol_colors{jj});
        fprintf('\t\tZeros: ');

        %Reading zeros file
```

```
        zeross=readmatrix([path_raw 's' num2str(sub_no(ii)) '\s' num2str(sub_no(ii))
...
            '_' protocol_colors_abv{jj} '_zeros3.txt']);
        zeross_check=0;
        for kk = 1:size(zeross,1)
            for LL = 2:size(zeross,2)
                if kk == 1 && isnan(zeross(kk,LL))
                    zeross(kk,LL)=zeross(kk+1,LL);
                elseif isnan(zeross(kk,LL))
                    fprintf('\n\t\t\tNaN at Row = %d, Column = %d',kk,LL);
                    zeross_check=1;
                end
            end
        end
        if zeross_check == 0
            fprintf(' Good!\n');
        else
            fprintf('\n');
        end
        zeross_mean=mean(zeross,1);
        zeross_stdv=std(zeross,0,1);

        %Saving zeros data
        data_filt{count_data,jj+1}=[zeross_mean(2:end); zeross_stdv(2:end)];
        count_data=count_data+1;

        %Opening figure for subject and session
        fig=figure('Name',['s' num2str(sub_no(ii)) ' - Session: ' protocol_colors{jj}
...
            ' - Filtered Data'],'Units','Normalized','Outerposition',[0 0 1 1]);
        tabgp=uitabgroup('Parent',fig);
        tab=zeros(length(fig_tabs),1);
        for kk = 1:length(fig_tabs)
            tab(kk)=uitab('Parent',tabgp,'Title',fig_tabs{kk});
        end

        %Declaring limits for axes
        axis_ground_time_AP=[inf -inf]; axis_ground_time_ML=[inf -inf];
        axis_mat_time_AP=[inf -inf];    axis_mat_time_ML=[inf -inf];

        %Going over the protocol order
        for kk = 1:length(protocol_order_abv)

            %Baseline on ground
            if kk == 1

                %Command window message
                fprintf('\t\t%s-%s: ',protocol_order_abv{kk});

                %Reading current data
                data=readmatrix([path_raw 's' num2str(sub_no(ii)) '\s'
num2str(sub_no(ii)) ...
                    '_' protocol_colors_abv{jj} '_' protocol_order_abv{kk} '.txt']);
                data_check=0;
                for mm = 1:size(data,1)
                    for nn = 2:size(data,2)
                        if mm == 1 && isnan(data(mm,nn))
                            data(mm,nn)=data(mm+1,nn);
                        elseif isnan(data(mm,nn))
                            fprintf('\n\t\t\tNaN at Row = %d, Column = %d',mm,nn);
                            data_check=1;
                        end
                    end
```

```
                end
                if data_check == 0
                    fprintf(' Good!\n');
                else
                    fprintf('\n');
                end

                %Converting current data to N and N-m (Raw)
                [data(:,2:7),dz_3364]=V2f_fp3364(data(:,2:7),zeross_mean(2:7),1000);
%Right foot on 3364

[data(:,8:13),dz_3477]=V2f_fp3477(data(:,8:13),zeross_mean(8:13),1000);     %Left foot
on 3477
                dz=mean([dz_3364 dz_3477]);

                %Rotating data around z-axis -90 degrees (Raw)
                data=[data(:,1) ...
%Time
                    -data(:,3) data(:,2) data(:,4) -data(:,6) data(:,5) data(:,7) ...
%Right 3364
                    -data(:,9) data(:,8) data(:,10) -data(:,12) data(:,11)
data(:,13)]; %Left 3477

                %Combining force plates (Raw)
                data_comb=Comb_fp3477_fp3364(data(:,8:13),data(:,2:7));
                data_comb=[data(:,1) data_comb];

                %Low-passing data with 20 Hz (Adding 10 seconds at the start and end)
                data_comb_LP=zeros(size(data_comb));
                for mm = 1:size(data_comb_LP,2)
                    if mm == 1
                        data_comb_LP(:,mm)=data_comb(:,mm);
                    else
                        data_10_start=data_comb(2:1001,mm); data_10_end=data_comb(end-
999:end,mm);
                        data_LP_temp=lowpass([data_10_start(end:-1:1);
data_comb(:,mm); data_10_end(end:-1:1)],...
                            freq_LP,freq_CED(1));
                        data_comb_LP(:,mm)=data_LP_temp(1001:10101);
                    end
                end
                clear data_10_start data_10_end data_LP_temp

                %Calculating COP_AP: +x facing forward (Raw and Filt)
                COP(:,1)=-(data_comb(:,6)+data_comb(:,2)*dz)./data_comb(:,4);
                COP_LP(:,1)=-
(data_comb_LP(:,6)+data_comb_LP(:,2)*dz)./data_comb_LP(:,4);

                %Calculating COP_ML: +y right hand (Raw and Filt)
                COP(:,2)= (data_comb(:,5)-data_comb(:,3)*dz)./data_comb(:,4);
                COP_LP(:,2)= (data_comb_LP(:,5)-
data_comb_LP(:,3)*dz)./data_comb_LP(:,4);

                %Updating limits for time series and spatial plots
                if max(COP_LP(:,1)) > axis_ground_time_AP(2)
                    axis_ground_time_AP(2)=max(COP_LP(:,1));
                end
                if min(COP_LP(:,1)) < axis_ground_time_AP(1)
                    axis_ground_time_AP(1)=min(COP_LP(:,1));
                end
                if max(COP_LP(:,2)) > axis_ground_time_ML(2)
                    axis_ground_time_ML(2)=max(COP_LP(:,2));
                end
```

```
                if min(COP_LP(:,2)) < axis_ground_time_ML(1)
                    axis_ground_time_ML(1)=min(COP_LP(:,2));
                end

                %Plotting time series: COP_AP
                axes('Parent',tab(1));
                subplot(2,1,1);
plot(data_comb(:,1),1000*COP(:,1),data_comb_LP(:,1),1000*COP_LP(:,1));
                grid; xlabel('Time [s]'); ylabel('Back \leftarrow COP [mm] \rightarrow
Face');
                legend('Raw','Low-Pass','location','best');
                title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-BL_{GND}'
...
                    '-COP_{AP}-EC']); xlim([data_comb(1,1) data_comb(end,1)]);

                %Plotting time series: COP_ML
                subplot(2,1,2);
plot(data_comb(:,1),1000*COP(:,2),data_comb_LP(:,1),1000*COP_LP(:,2));
                grid; xlabel('Time [s]'); ylabel('Left \leftarrow COP [mm] \rightarrow
Right');
                legend('Raw','Low-Pass','location','best');
                title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-BL_{GND}'
...
                    '-COP_{ML}-EC']); xlim([data_comb(1,1) data_comb(end,1)]);

                %Plotting spatial: COP_AP vs. COP_ML
                axes('Parent',tab(2)); subplot(1,2,1);
                plot(1000*COP(:,2),1000*COP(:,1),1000*COP_LP(:,2),1000*COP_LP(:,1));
                grid; xlabel('Left \leftarrow COP_{ML} [mm] \rightarrow Right');
                ylabel('Back \leftarrow COP_{AP} [mm] \rightarrow Face');
                legend('Raw','Low-Pass','location','northwest');
                title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-BL_{GND}'
...
                    '-COP-EC']); axis square;

                %Recording COP data
                data_filt{count_data,jj+1}=COP_LP;
                count_data=count_data+1;

                %Clearing COP data
                clear COP COP_LP

            elseif kk == 2 || kk == 3 || kk == 5 || kk == 6  %Sway on MAT

                %Command window message
                fprintf('\t\t%s Sway: ',protocol_order_abv{kk});

                %Reading current data
                data=readmatrix([path_raw 's' num2str(sub_no(ii)) '\s'
num2str(sub_no(ii)) ...
                    '_' protocol_colors_abv{jj} '_' protocol_order_abv{kk}
'_sway.txt']);
                data_check=0;
                for mm = 1:size(data,1)
                    for nn = 2:size(data,2)
                        if mm == 1 && isnan(data(mm,nn))
                            data(mm,nn)=data(mm+1,nn);
                        elseif isnan(data(mm,nn))
                            fprintf('\n\t\t\tNaN at Row = %d, Column = %d',mm,nn);
                            data_check=1;
                        end
                    end
                end
```

```
                if data_check == 0
                    fprintf(' Good!\n');
                else
                    fprintf('\n');
                end

                %Converting current data to N and N-m (4033) (Raw)
                [data(:,2:7),dz]=V2f_fp4033(data(:,2:7),zeross_mean(14:19),1000);

                %Rotating data around z-axis 180 degrees (Raw)
                data_comb=[data(:,1) -data(:,2) -data(:,3) data(:,4) -data(:,5) -
data(:,6) data(:,7)];

                %Sway Data: Low-passing data with 20 Hz (Adding 10 seconds at the
start and end)
                data_comb_LP=zeros(size(data_comb));
                for mm = 1:size(data_comb_LP,2)
                    if mm == 1
                        data_comb_LP(:,mm)=data_comb(:,mm);
                    else
                        data_10_start=data_comb(2:25001,mm);
data_10_end=data_comb(end-24999:end,mm);
                        data_LP_temp=lowpass([data_10_start(end:-1:1);
data_comb(:,mm); data_10_end(end:-1:1)],...
                            freq_LP,freq_CED(2));
                        data_comb_LP(:,mm)=data_LP_temp(25001:end-25000);
                    end
                end
                clear data_10_start data_10_end data_LP_temp

                %Motor Data: Band-passing data with 20-400 Hz (Adding 10 seconds at
the start and end)
                data_comb_BP=zeros(size(data_comb));
                for mm = 1:size(data_comb_BP,2)
                    if mm == 1
                        data_comb_BP(:,mm)=data_comb(:,mm);
                    else
                        data_10_start=data_comb(2:25001,mm);
data_10_end=data_comb(end-24999:end,mm);
                        data_BP_temp=bandpass([data_10_start(end:-1:1);
data_comb(:,mm); data_10_end(end:-1:1)],...
                            freq_BP,freq_CED(2));
                        data_comb_BP(:,mm)=data_BP_temp(25001:end-25000);
                    end
                end
                clear data_10_start data_10_end data_BP_temp

                %Calculating COP_AP: +x facing forward (Raw and Low-Pass)
                COP(:,1)=-(data_comb(:,6)+data_comb(:,2)*dz)./data_comb(:,4);
                COP_LP(:,1)=-
(data_comb_LP(:,6)+data_comb_LP(:,2)*dz)./data_comb_LP(:,4);

                %Calculating COP_ML: +y right hand (Raw and Low-Pass)
                COP(:,2)= (data_comb(:,5)-data_comb(:,3)*dz)./data_comb(:,4);
                COP_LP(:,2)= (data_comb_LP(:,5)-
data_comb_LP(:,3)*dz)./data_comb_LP(:,4);

                %Calculating resulting force created by the motors (Band-Pass)

F_xyz=sqrt(data_comb_BP(:,2).^2+data_comb_BP(:,3).^2+data_comb_BP(:,4).^2);

                %Updating limits for time series and spatial plots
                if max(COP_LP(:,1)) > axis_mat_time_AP(2)
```

```
                    axis_mat_time_AP(2)=max(COP_LP(:,1));
                end
                if min(COP_LP(:,1)) < axis_mat_time_AP(1)
                    axis_mat_time_AP(1)=min(COP_LP(:,1));
                end
                if max(COP_LP(:,2)) > axis_mat_time_ML(2)
                    axis_mat_time_ML(2)=max(COP_LP(:,2));
                end
                if min(COP_LP(:,2)) < axis_mat_time_ML(1)
                    axis_mat_time_ML(1)=min(COP_LP(:,2));
                end

                %Selecting tab for time series
                if kk == 2          %Baseline on mat
                    axes('Parent',tab(3));
                elseif kk == 3      %Threshold sway
                    axes('Parent',tab(5));
                elseif kk == 5      %Stimulus
                    axes('Parent',tab(8));
                elseif kk == 6      %T0
                    axes('Parent',tab(10));
                end

                %Plotting time series: COP_AP
                subplot(2,2,1);
plot(data_comb(:,1),1000*COP(:,1),data_comb_LP(:,1),1000*COP_LP(:,1));
                grid; xlabel('Time [s]'); ylabel('Back \leftarrow COP [mm] \rightarrow
Face');
                legend('Raw','Low-Pass','location','best');
                if kk == 2          %Baseline on mat
                    title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-' ...
                        protocol_order_abv{kk}(1:2) '_{' protocol_order_abv{kk}(7:end)
'}' ...
                        '-COP_{AP}-EC']); xlim([data_comb(1,1) data_comb(end,1)]);
                elseif kk == 3      %Threshold sway
                    title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-' ...
                        protocol_order_abv{kk}(1:3) ...
                        '-COP_{AP}-EC']); xlim([data_comb(1,1) data_comb(end,1)]);
                elseif kk == 5      %Stimulus
                    title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-' ...
                        protocol_order_abv{kk}(1:4) ...
                        '-COP_{AP}-EC']); xlim([data_comb(1,1) data_comb(end,1)]);
                elseif kk == 6      %T0
                    title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-' ...
                        protocol_order_abv{kk}(1:3) ...
                        '-COP_{AP}-EC']); xlim([data_comb(1,1) data_comb(end,1)]);
                end
                hold on;

                %Plotting time series: COP_ML
                subplot(2,2,3);
plot(data_comb(:,1),1000*COP(:,2),data_comb_LP(:,1),1000*COP_LP(:,2));
                grid; xlabel('Time [s]'); ylabel('Left \leftarrow COP [mm] \rightarrow
Right');
                legend('Raw','Low-Pass','location','best');
                if kk == 2          %Baseline on mat
                    title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-' ...
                        protocol_order_abv{kk}(1:2) '_{' protocol_order_abv{kk}(7:end)
'}' ...
                        '-COP_{ML}-EC']); xlim([data_comb(1,1) data_comb(end,1)]);
                elseif kk == 3      %Threshold sway
                    title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-' ...
                        protocol_order_abv{kk}(1:3) ...
```

```
                    '-COP_{ML}-EC']); xlim([data_comb(1,1) data_comb(end,1)]);
                elseif kk == 5      %Stimulus
                    title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-' ...
                        protocol_order_abv{kk}(1:4) ...
                        '-COP_{ML}-EC']); xlim([data_comb(1,1) data_comb(end,1)]);
                elseif kk == 6      %T0
                    title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-' ...
                        protocol_order_abv{kk}(1:3) ...
                        '-COP_{ML}-EC']); xlim([data_comb(1,1) data_comb(end,1)]);
                end
                hold on;

                %Plotting forces created by the motors
                stackedplot_data=[data_comb_BP(:,2) data_comb_BP(:,3)
data_comb_BP(:,4) F_xyz];
                subplot(2,2,[2 4]);
s=stackedplot(data_comb_BP(:,1),stackedplot_data,...
                    'DisplayLabels',{'Fx' 'Fy' 'Fz' 'Fxyz'},'XLimits',[data_comb(1,1)
data_comb(end,1)],...
                    'GridVisible','on'); xlabel('Time [s]');
                if kk == 2          %Baseline on mat
                    title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-' ...
                        protocol_order_abv{kk}(1:2) '_{' protocol_order_abv{kk}(7:end)
'}' ...
                        '-Motors']); xlim([data_comb(1,1) data_comb(end,1)]);
                elseif kk == 3      %Threshold sway
                    title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-' ...
                        protocol_order_abv{kk}(1:3) ...
                        '-Motors']); xlim([data_comb(1,1) data_comb(end,1)]);
                elseif kk == 5      %Stimulus
                    title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-' ...
                        protocol_order_abv{kk}(1:4) ...
                        '-Motors']); xlim([data_comb(1,1) data_comb(end,1)]);
                elseif kk == 6      %T0
                    title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-' ...
                        protocol_order_abv{kk}(1:3) ...
                        '-Motors']); xlim([data_comb(1,1) data_comb(end,1)]);
                end

                %Selecting tab for spatial plots
                if kk == 2          %Baseline on mat
                    axes('Parent',tab(4));
                elseif kk == 3      %Threshold sway
                    axes('Parent',tab(6));
                elseif kk == 5      %Stimulus
                    axes('Parent',tab(9));
                elseif kk == 6      %T0
                    axes('Parent',tab(11));
                end

                %Plotting spatial plots: COP_AP vs. COP_ML
                subplot(1,2,1);
plot(1000*COP(:,2),1000*COP(:,1),1000*COP_LP(:,2),1000*COP_LP(:,1));
                grid; xlabel('Left \leftarrow COP_{ML} [mm] \rightarrow Right');
                ylabel('Back \leftarrow COP_{AP} [mm] \rightarrow Face');
                legend('Raw','Low-Pass','location','northwest');
                if kk == 2          %Baseline on mat
                    title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-' ...
                        protocol_order_abv{kk}(1:2) '_{' protocol_order_abv{kk}(7:end)
'}' ...
                        '-COP-EC']);
                elseif kk == 3      %Threshold sway
                    title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-' ...
```

```
                        protocol_order_abv{kk}(1:3) ...
                        '-COP-EC']);
                elseif kk == 5      %Stimulus
                    title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-' ...
                        protocol_order_abv{kk}(1:4) ...
                        '-COP-EC']);
                elseif kk == 6      %T0
                    title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-' ...
                        protocol_order_abv{kk}(1:3) ...
                        '-COP-EC']);
                end
                axis square; hold on;

                %Recording COP data
                data_filt{count_data,jj+1}=[COP_LP F_xyz];
                count_data=count_data+1;

                %Clearing COP data
                clear COP COP_LP F_xyz

            else                        %EC 421 Threshold calculator

                %Command window message
                fprintf('\t\t%s: ',protocol_order_abv{kk});

                %Reading current data
                data=readmatrix([path_raw 's' num2str(sub_no(ii)) '\s'
num2str(sub_no(ii)) ...
                    '_' protocol_colors_abv{jj} '_' protocol_order_abv{kk} '.txt'],...
                    'OutputType','string','Range',1);

                %Selecting tab for time series
                axes('Parent',tab(7));

                %421 Threshold calculation and plot
                [THR_data,THR_value,THR_value_n,motor]=THR_421_Calculator(data);
                title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-'
protocol_order_abv{kk}(1:3) ' | ' ...
                    'Motor ' motor ' | THR ' ...
                    num2str(THR_value) '/127 = ' num2str(round(THR_value/127*100)) '%
| 90% THR = ' ...
                    num2str(round(0.9*THR_value/127*100)) '%']);

                %Command window message
                if ~isnan(THR_value)
                    fprintf(' Good!\n');
                end

                %Recording 421 information
                %Recording COP data
                data_filt{count_data,jj+1}={THR_data,THR_value,THR_value_n,motor};
                count_data=count_data+1;

            end

        end

        %Adjusting the axis limits of the ground force plate data
        for kk = 1:length(protocol_order_abv)

            %Baseline on ground
            if kk == 1
```

```
            %Applying axis limits to COP_AP and COP_ML
            axes('Parent',tab(1));
            subplot(2,1,1); ylim(1000*axis_ground_time_AP);
            subplot(2,1,2); ylim(1000*axis_ground_time_ML);

            %Applying axis limits to COP_AP vs. COP_ML
            axes('Parent',tab(2)); subplot(1,2,1);
            axis(1000*[axis_ground_time_ML axis_ground_time_AP]);

        elseif kk == 2 || kk == 3 || kk == 5 || kk == 6  %Sway on MAT

            %Selecting tab for time series
            if kk == 2          %Baseline on mat
                axes('Parent',tab(3));
            elseif kk == 3      %Threshold sway
                axes('Parent',tab(5));
            elseif kk == 5      %Stimulus
                axes('Parent',tab(8));
            elseif kk == 6      %T0
                axes('Parent',tab(10));
            end

            %Applying axis limits to COP_AP and COP_ML
            subplot(2,2,1); ylim(1000*axis_mat_time_AP);
            subplot(2,2,3); ylim(1000*axis_mat_time_ML);

            %Motbutt stuff
            if kk >= 3

                %Command window message
                fprintf('\t\t%s Mot-Butt: ',protocol_order_abv{kk});

                %Reading mot_butt data
                data=readmatrix([path_raw 's' num2str(sub_no(ii)) '\s'
num2str(sub_no(ii)) ...
                    '_' protocol_colors_abv{jj} '_' protocol_order_abv{kk}
'_motbutt.txt']);
                if kk == 3
                    idx_nan=[];
                    for mm = 1:size(data,1)
                        if sum(isnan(data(mm,2:end))) == 7
                            idx_nan=[idx_nan; mm];
                        end
                    end
                    data(idx_nan,:)=[];
                end
                data_check=0;
                for mm = 1:size(data,1)
                    for nn = 2:size(data,2)
                        if mm == 1 && isnan(data(mm,nn))
                            data(mm,nn)=data(mm+1,nn);
                        elseif isnan(data(mm,nn))
                            fprintf('\n\t\t\tNaN at Row = %d, Column = %d',mm,nn);
                            data_check=1;
                        end
                    end
                end
                if data_check == 0
                    fprintf(' Good!\n');
                else
                    fprintf('\n');
                end
```

```
                    %Plotting when motors were ON
                    Motor_OFF_ON=zeros(size(data,1),2);
                    for LL = 1:size(data,1)

                        %Checking if more motors were ON, and plotting based on it
                        if sum(data(LL,2:7) > 3) >= 1            %Motor is ON
                            Motor_OFF_ON(LL,1)=NaN; Motor_OFF_ON(LL,2)=1000;
                        else                                     %Motor is OFF
                            Motor_OFF_ON(LL,1)=1000; Motor_OFF_ON(LL,2)=NaN;
                        end

                    end
                    subplot(2,2,1);
plot(data(:,1),Motor_OFF_ON(:,1)*axis_mat_time_AP(2),'.r','MarkerSize',4,...
                        'HandleVisibility','off'); hold on;
                    subplot(2,2,1);
plot(data(:,1),Motor_OFF_ON(:,2)*axis_mat_time_AP(2),'.g','MarkerSize',4,...
                        'HandleVisibility','off'); hold on;

                    %Plotting when button was pressed
                    Button_OFF_ON=zeros(size(data,1),2);
                    for LL = 1:size(data,1)

                        %Checking if all motors were OFF, and plotting based on it
                        if data(LL,8) > 4        %Button is not pressed
                            Button_OFF_ON(LL,1)=1000; Button_OFF_ON(LL,2)=NaN;
                        else                          %Motor is ON
                            Button_OFF_ON(LL,1)=NaN; Button_OFF_ON(LL,2)=1000;
                        end

                    end
                    %Correcting error for s1002, WH, T0
                    if ii == 2 && jj == 1 && kk == 6
                        Button_OFF_ON(:,1)=1000; Button_OFF_ON(:,2)=NaN;
                    end
                    subplot(2,2,1);
plot(data(:,1),Button_OFF_ON(:,1)*axis_mat_time_AP(1),...
                        '.r','MarkerSize',4,'HandleVisibility','off'); hold on;
                    subplot(2,2,1);
plot(data(:,1),Button_OFF_ON(:,2)*axis_mat_time_AP(1),...
                        '.g','MarkerSize',4,'HandleVisibility','off'); hold on;

                end

                %Selecting tab for spatial plots
                if kk == 2          %Baseline on mat
                    axes('Parent',tab(4));
                elseif kk == 3      %Threshold sway
                    axes('Parent',tab(6));
                elseif kk == 5      %Stimulus
                    axes('Parent',tab(9));
                elseif kk == 6      %T0
                    axes('Parent',tab(11));
                end

                %Applying axis limits to COP_AP vs. COP_ML
                subplot(1,2,1); axis(1000*[axis_mat_time_ML axis_mat_time_AP]);

                %Recording motor and button information
                if kk >= 3
                    data_filt{count_data,jj+1}={Motor_OFF_ON/1000,Button_OFF_ON/1000};
                    count_data=count_data+1;
                end
```

```
            end

        end

        %Saving each tab as a figure
        for kk = 1:length(fig_tabs)
            tabgp.SelectedTab = tab(kk);
            saveas(fig,[path_res '3 Data Check - Filt\s' num2str(sub_no(ii)) '\s'
num2str(sub_no(ii)) ...
                '_' protocol_colors_abv{jj} '_' num2str(kk) '_' fig_names{kk}
'.jpeg']);
        end
        close all;

    end

    %Saving data_visual for subject
    save([path_pro 's' num2str(sub_no(ii)) '_DataFilt.mat'],'data_filt');

end
```

```
%Camilo Giraldo - Healthy Vibratio Sway Study v2 - Data Analysis
%University of Kansas - Biodynamics Lab
%Last Update: April 20, 2021

clear; close all; clc;

%% General
%File paths
path_raw='C:\Users\17856\Desktop\Final_Vibration_study\Vibrotactile Study (2021)\';
path_res='C:\Users\17856\Desktop\Final_Vibration_study\Results\';
path_pro='C:\Users\17856\Desktop\Final_Vibration_study\Processed Data\';


%Subject numbers
sub_no=[1001 1002 1003 1004 1005 1006 1008 1009 1010 2001:2003];

%Color names
protocol_colors_abv={'WH' 'PK' 'BR' 'PB'};
protocol_colors={'White' 'Pink' 'Brown' 'Placebo'};

%Protocol order
protocol_order_abv={'BL_EC_GND' 'BL_EC_MAT' 'THR_EC' 'THR' 'STIM_EC' 'T_0_EC'};

%CED Frequencies [Hz]
freq_CED=[100 2500];

%Arduino frequency [Hz]
freq_ard=10;

%Names of tabs for figures
fig_tabs={'BL: Time' 'BL: Spatial' 'BL-MAT: Time' 'BL-MAT: Spatial' 'THR: Time' 'THR:
Spatial' 'THR: Calculation' ...
    'STIM: Time' 'STIM: Spatial' 'T0: Time' 'T0: Spatial'};

%Names of the figures to be saved
fig_names={'BL_TimeSeries' 'BL_Spatial' 'BL-Mat_TimeSeries' 'BL-Mat_Spatial' ...
    'THR_TimeSeries' 'THR_Spatial' 'THR_Calculation' 'STIM_TimeSeries' 'STIM_Spatial'
...
```

```
    'T0_TimeSeries' 'T0_Spatial'};

%Names of the data to be saved
var_names={'Zeros' 'BL_GND' 'BL_MAT' 'THR_Sway' 'THR' 'STIM_Sway' ...
    'T_0_Sway' 'THR_Sway Butt' 'STIM_Sway Butt' 'T_0_Sway Butt'};

%% Generation Analysis Data
%Going over all the subjects
for ii = 1:length(sub_no)

    %Preallocating space for subject analysis data
    data_analysis=cell(length(var_names),5);
    for jj = 1:length(var_names)
        data_analysis{jj,1}=var_names{jj};
    end

    %Loading filtered data for each subject
    load([path_pro 's' num2str(sub_no(ii)) '_DataFilt.mat'],'data_filt');

    %Command window message
    fprintf('Subject: s%d\n',sub_no(ii));

    %Going over all the sessions
    for jj = 1:length(protocol_colors)

        %Reseting counter for data_visual
        count_data=1;

        %Command window message
        fprintf('\tSession: %s\n',protocol_colors{jj});
        fprintf('\t\tZeros\n');

        %Saving zeros data
        data_analysis{count_data,jj+1}=data_filt{count_data,jj+1};
        count_data=count_data+1;

        %Opening figure for subject and session
        fig=figure('Name',['s' num2str(sub_no(ii)) ' - Session: ' protocol_colors{jj}
...
            ' - Analysis Data'],'Units','Normalized','Outerposition',[0 0 1 1]);
        tabgp=uitabgroup('Parent',fig);
        tab=zeros(length(fig_tabs),1);
        for kk = 1:length(fig_tabs)
            tab(kk)=uitab('Parent',tabgp,'Title',fig_tabs{kk});
        end

        %Going over the protocol order
        for kk = 1:length(protocol_order_abv)

            %Baseline on ground
            if kk == 1

                %Command window message
                fprintf('\t\t%s\n',protocol_order_abv{kk});

                %Grabbing the first 90 seconds of data [AP ML] where +x is forward,
and +y is right hand
                COP=data_filt{count_data,1+jj}(1:1+90*freq_CED(1),:);
                t=(0:1/freq_CED(1):(size(COP,1)-1)/freq_CED(1))';

                %Plotting time series: COP_AP
                axes('Parent',tab(1));
                subplot(2,1,1); plot(t,1000*COP(:,1)); grid; xlabel('Time [s]');
```

```
                ylabel('Back \leftarrow COP [mm] \rightarrow Face');
                title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-BL_{GND}'
...
                    '-COP_{AP}-EC']); xlim([t(1) t(end)]);

                %Plotting time series: COP_ML
                subplot(2,1,2); plot(t,1000*COP(:,2)); grid; xlabel('Time [s]');
                ylabel('Left \leftarrow COP [mm] \rightarrow Right');
                title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-BL_{GND}'
...
                    '-COP_{ML}-EC']); xlim([t(1) t(end)]);

                %Plotting spatial: COP_AP vs. COP_ML
                axes('Parent',tab(2)); subplot(1,2,1);
                plot(1000*COP(:,2),1000*COP(:,1));
                grid; xlabel('Left \leftarrow COP_{ML} [mm] \rightarrow Right');
                ylabel('Back \leftarrow COP_{AP} [mm] \rightarrow Face');
                title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-BL_{GND}'
...
                    '-COP-EC']); axis square;

                %Recording COP data
                data_analysis{count_data,jj+1}=[t COP];
                count_data=count_data+1;

                %Clearing COP data
                clear COP t

            elseif kk == 2                          %Sway on MAT and no motbutt

                %Command window message
                fprintf('\t\t%s\n',protocol_order_abv{kk});

                %Sway: First 90 seconds of data and downsample it to 100 Hz
                COP=data_filt{count_data,1+jj}(1:1+90*freq_CED(2),[1 2]);
                COP=downsample(COP,freq_CED(2)/freq_CED(1));
                t=(0:1/freq_CED(1):(size(COP,1)-1)/freq_CED(1))';

                %Plotting time series: COP_AP
                axes('Parent',tab(3)); subplot(3,1,1); plot(t,1000*COP(:,1));
                grid; xlabel('Time [s]'); ylabel('Back \leftarrow COP [mm] \rightarrow
Face');
                title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-' ...
                    protocol_order_abv{kk}(1:2) '_{' protocol_order_abv{kk}(7:end) '}'
...
                    '-COP_{AP}-EC']); xlim([t(1) t(end)]);

                %Plotting time series: COP_ML
                axes('Parent',tab(3)); subplot(3,1,2); plot(t,1000*COP(:,2));
                grid; xlabel('Time [s]'); ylabel('Left \leftarrow COP [mm] \rightarrow
Right');
                title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-' ...
                    protocol_order_abv{kk}(1:2) '_{' protocol_order_abv{kk}(7:end) '}'
...
                    '-COP_{ML}-EC']); xlim([t(1) t(end)]);

                %Plotting spatial plots: COP_AP vs. COP_ML
                axes('Parent',tab(4)); subplot(1,2,1);
plot(1000*COP(:,2),1000*COP(:,1));
                grid; xlabel('Left \leftarrow COP_{ML} [mm] \rightarrow Right');
                ylabel('Back \leftarrow COP_{AP} [mm] \rightarrow Face');
                title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-' ...
```

```
                    protocol_order_abv{kk}(1:2) '_{' protocol_order_abv{kk}(7:end) '}'
...
                    '-COP-EC']); axis square;

                %Motor: First 90 seconds of data at 2500 Hz
                F_xyz=data_filt{count_data,1+jj}(1:1+90*freq_CED(2),3);
                t_fast=(0:1/freq_CED(2):(size(F_xyz,1)-1)/freq_CED(2))';

                %Rectifying motor data
                t_rect=linspace(min(t_fast),max(t_fast),freq_ard*90+1)';
                freq_rect=1/(t_rect(2)-t_rect(1));
                F_xyz_rect=zeros(length(t_rect)-1,1);
                for mm = 1:length(t_rect)-1

F_xyz_rect(mm)=sqrt(freq_rect*trapz(t_fast(1+floor(freq_CED(2)/freq_rect)*(mm-1):...

floor(freq_CED(2)/freq_rect)*mm),F_xyz(1+floor(freq_CED(2)/freq_rect)*(mm-1):...
                        floor(freq_CED(2)/freq_rect)*mm).^2));
                end
                t_rect=t_rect(1:end-1)+1/freq_rect/2;

                %Plotting motor data
                axes('Parent',tab(3)); subplot(3,1,3);
plot(t_fast,F_xyz,'LineWidth',0.5); hold on;
                subplot(3,1,3); plot(t_rect,F_xyz_rect,'LineWidth',2);
                grid; xlabel('Time [s]'); ylabel('F_{xyz} [N]');
                title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-' ...
                    protocol_order_abv{kk}(1:2) '_{' protocol_order_abv{kk}(7:end) '}'
...
                    '-Motor']); xlim([t_fast(1) t_fast(end)]);
                legend('Raw','Rectified','location','best');

                %Recording COP data
                data_analysis{count_data,jj+1}={[t COP] [t_fast F_xyz] [t_rect
F_xyz_rect]};
                count_data=count_data+1;

                %Clearing COP data
                clear COP t t_fast F_xyz t_rect F_xyz_rect

            elseif kk == 3              %THR sway

                %Command window message
                fprintf('\t\t%s\t',protocol_order_abv{kk});

                %Sway: grabbing all data and downsample it to 100 Hz
                COP=data_filt{count_data,1+jj}(:,[1 2]);
                COP=downsample(COP,freq_CED(2)/freq_CED(1));
                t=(0:1/freq_CED(1):(size(COP,1)-1)/freq_CED(1))';

                %Checking if after downsample COP matches the Mot-Butt's lengths
                if size(COP,1) ~= size(data_filt{8,1+jj}{1,1},1)
                    if size(COP,1)+1 == size(data_filt{8,1+jj}{1,1},1)
                        COP=[COP(1,:); COP];
                        fprintf('+1 Equal lengths, ');
                    elseif size(COP,1)-1 == size(data_filt{8,1+jj}{1,1},1)
                        COP=COP(1:end-1,:);
                        fprintf('-1 Equal lengths, ');
                    else
                        fprintf('Error in lengths');
                        return
                    end
                else
```

```
                    fprintf('Equal lengths, ');
                end

                %Getting index when the 421 starts
                idx=[0 0];
                for LL = 1:size(data_filt{8,1+jj}{1,1},1)-4*freq_CED(1)
                    if sum(isnan(data_filt{8,1+jj}{1,1}(LL:LL+4*freq_CED(1),2))) ==
4*freq_CED(1)+1
                        idx(1)=1;
                    end
                    if idx(1) == 1 &&
sum(isnan(data_filt{8,1+jj}{1,1}(LL:LL+4*freq_CED(1),2))) ~= 4*freq_CED(1)+1
                        idx(1)=LL+4*freq_CED(1);
                        break
                    end
                end
                if idx(1) == 0 || idx(1) == 1
                    fprintf('Error in 421 start');
                    return
                else
                    fprintf('421 start found, ')
                end
                t_start=t(idx(1));

                %Finding when motors were off for 0.5 seconds
                for LL = idx(1):size(data_filt{8,1+jj}{1,1},1)-0.5*freq_CED(1)
                    if sum(isnan(data_filt{8,1+jj}{1,1}(LL:LL+0.5*freq_CED(1),2))) ==
0.5*freq_CED(1)+1
                        idx(2)=LL-1;
                        break
                    end
                end
                if idx(2) == 0
                    fprintf('Error in 421 end');
                    return
                else
                    fprintf('421 end found, ')
                end
                t_end=t(idx(2));

                %Getting button (OFF and ON)
                Button_OFF_ON=data_filt{8,1+jj}{1,2}(idx(1):idx(2),:);

                %Getting COP data when 421 was happening
                COP=COP(idx(1):idx(2),:); t=t(idx(1):idx(2))-t(idx(1));

                %Showing how long 421 was
                fprintf('T = %.2f sec\n',(size(COP,1)-1)/freq_CED(1));

                %Plotting time series: COP_AP
                axes('Parent',tab(5)); subplot(3,1,1); plot(t,1000*COP(:,1));
                grid; xlabel('Time [s]'); ylabel('Back \leftarrow COP [mm] \rightarrow
Face');
                title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-' ...
                    protocol_order_abv{kk}(1:3) '-COP_{AP}-EC']); xlim([t(1) t(end)]);
                hold on;

                %Plotting when button is pressed
                button_val=1000*(min(COP(:,1))-0.1*range(COP(:,1)));
                subplot(3,1,1); plot(t,Button_OFF_ON(:,1)*button_val,...
                    '.r','MarkerSize',4,'HandleVisibility','off'); hold on;
                subplot(3,1,1); plot(t,Button_OFF_ON(:,2)*button_val,...
                    '.g','MarkerSize',4,'HandleVisibility','off'); hold on;
```

```
                %Making Button variable into single column (0/1)
                Button_OFF_ON=Button_OFF_ON(:,2);
                Button_OFF_ON(isnan(Button_OFF_ON))=0;

                %Plotting time series: COP_ML
                axes('Parent',tab(5)); subplot(3,1,2); plot(t,1000*COP(:,2));
                grid; xlabel('Time [s]'); ylabel('Left \leftarrow COP [mm] \rightarrow
Right');
                title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-' ...
                    protocol_order_abv{kk}(1:3) '-COP_{ML}-EC']); xlim([t(1) t(end)]);

                %Plotting spatial plots: COP_AP vs. COP_ML
                axes('Parent',tab(6)); subplot(1,2,1);
plot(1000*COP(:,2),1000*COP(:,1));
                grid; xlabel('Left \leftarrow COP_{ML} [mm] \rightarrow Right');
                ylabel('Back \leftarrow COP_{AP} [mm] \rightarrow Face');
                title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-' ...
                    protocol_order_abv{kk}(1:3) '-COP-EC']); axis square;

                %Motor: Grabbing all data at 2500 Hz
                F_xyz=data_filt{count_data,1+jj}(:,3);
                t_fast=(0:1/freq_CED(2):(size(F_xyz,1)-1)/freq_CED(2))';

                %Getting data when 421 was going
                [~,idx1]=min(abs(t_fast-t_start));
                [~,idx2]=min(abs(t_fast-t_end));
                F_xyz=F_xyz(idx1:idx2); t_fast=t_fast(idx1:idx2)-t_fast(idx1);

                %Rectifying motor data
                t_rect=linspace(min(t_fast),max(t_fast),freq_ard*90+1)';
                freq_rect=1/(t_rect(2)-t_rect(1));
                F_xyz_rect=zeros(length(t_rect)-1,1);
                for mm = 1:length(t_rect)-1

F_xyz_rect(mm)=sqrt(freq_rect*trapz(t_fast(1+floor(freq_CED(2)/freq_rect)*(mm-1):...

floor(freq_CED(2)/freq_rect)*mm),F_xyz(1+floor(freq_CED(2)/freq_rect)*(mm-1):...
                        floor(freq_CED(2)/freq_rect)*mm).^2));
                end
                t_rect=t_rect(1:end-1)+1/freq_rect/2;

                %Plotting motor data
                axes('Parent',tab(5)); subplot(3,1,3);
plot(t_fast,F_xyz,'LineWidth',0.5); hold on;
                subplot(3,1,3); plot(t_rect,F_xyz_rect,'LineWidth',2);
                grid; xlabel('Time [s]'); ylabel('F_{xyz} [N]');
                title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-' ...
                    protocol_order_abv{kk}(1:3) '-Motor']); xlim([t_fast(1)
t_fast(end)]);
                legend('Raw','Rectified','location','best');

                %Recording COP data
                data_analysis{count_data,jj+1}={[t COP] [t_fast F_xyz] [t_rect
F_xyz_rect]};
                count_data=count_data+1;

                %Recording button (0: Not pressed, 1: Pressed)
                data_analysis{8,jj+1}=Button_OFF_ON;

                %Clearing COP data
                clear COP t t_fast F_xyz t_rect F_xyz_rect idx idx1 idx2 t_start t_end
Button_OFF_ON
```

```
            elseif kk == 4                    %EC 421 Threshold calculator

                %Command window message
                fprintf('\t\t%s\t',protocol_order_abv{kk});

                %Reading current data
                data=readmatrix([path_raw 's' num2str(sub_no(ii)) '\s'
num2str(sub_no(ii)) ...
                    '_' protocol_colors_abv{jj} '_' protocol_order_abv{kk} '.txt'],...
                    'OutputType','string','Range',1);

                %Selecting tab for time series
                axes('Parent',tab(7));

                %421 Threshold calculation and plot
                [THR_data,THR_value,THR_value_n,motor]=THR_421_Calculator(data);
                title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-'
protocol_order_abv{kk}(1:3) ' | ' ...
                    'Motor ' motor ' | THR ' num2str(THR_value) '/127 = '
num2str(round(THR_value/127*100)) ...
                    '% | 90% THR = ' num2str(round(0.9*THR_value/127*100)) '%']);

                %Command window message
                if ~isnan(THR_value)
                    fprintf(' Good!\n');
                end

                %Recording 421 information
                %Recording COP data
                data_analysis{count_data,jj+1}={THR_data,THR_value,THR_value_n,motor};
                count_data=count_data+1;

            elseif kk == 5                        %STIM sway

                %Command window message
                fprintf('\t\t%s\t',protocol_order_abv{kk});

                %Only checking for mot-butt stuff for non-PB visits
                if jj ~= 4

                    %All motor data at 2500 Hz
                    F_xyz=data_filt{count_data,1+jj}(:,3);

                    %Sway: All data and downsample it to 100 Hz
                    COP=data_filt{count_data,1+jj}(:,[1 2]);
                    COP=downsample(COP,freq_CED(2)/freq_CED(1));

                    %Checking if after downsample COP matches the Mot-Butt's lengths
                    if size(COP,1) ~= size(data_filt{9,1+jj}{1,1},1)
                        if size(COP,1)+1 == size(data_filt{9,1+jj}{1,1},1)
                            COP=[COP(1,:); COP];
                            fprintf('+1 Equal lengths, ');
                        elseif size(COP,1)-1 == size(data_filt{9,1+jj}{1,1},1)
                            COP=COP(1:end-1,:);
                            fprintf('-1 Equal lengths, ');
                        else
                            fprintf('Error in lengths');
                            return
                        end
                    else
                        fprintf('Equal lengths, ');
                    end
```

```
                    %Finding when motors were off for 0.25 seconds
                    idx=0;
                    for LL = 1:size(data_filt{9,1+jj}{1,1},1)-0.25*freq_CED(1)
                        if
sum(isnan(data_filt{9,1+jj}{1,1}(LL:LL+0.25*freq_CED(1),2))) == 0.25*freq_CED(1)+1
                            idx=LL-1;
                            break
                        end
                    end
                    if idx == 0
                        fprintf('Error in STIM end');
                        return
                    else
                        fprintf('STIM end found, ')
                    end

                    %Getting motor data when motors were ON
                    F_xyz=F_xyz(1:(idx-1)*25+1);
                    t_fast=(0:1/freq_CED(2):(length(F_xyz)-1)/freq_CED(2))';

                    %Rectifying motor data
                    t_rect=linspace(min(t_fast),max(t_fast),freq_ard*90+1)';
                    freq_rect=1/(t_rect(2)-t_rect(1));
                    F_xyz_rect=zeros(length(t_rect)-1,1);
                    for mm = 1:length(t_rect)-1

F_xyz_rect(mm)=sqrt(freq_rect*trapz(t_fast(1+floor(freq_CED(2)/freq_rect)*(mm-1):...

floor(freq_CED(2)/freq_rect)*mm),F_xyz(1+floor(freq_CED(2)/freq_rect)*(mm-1):...
                            floor(freq_CED(2)/freq_rect)*mm).^2));
                    end
                    t_rect=t_rect(1:end-1)+1/freq_rect/2;

                    %Getting COP when motor was ON
                    COP=COP(1:idx,:); t=(0:1/freq_CED(1):(size(COP,1)-
1)/freq_CED(1))';

                    %Getting button (OFF and ON)
                    Button_OFF_ON=data_filt{9,1+jj}{1,2}(1:idx,:);

                    %Clear idx
                    clear idx

                else

                    %Sway: First 90 seconds of data and downsample it to 100 Hz
                    COP=data_filt{count_data,1+jj}(1:1+90*freq_CED(2),[1 2]);
                    COP=downsample(COP,freq_CED(2)/freq_CED(1));
                    t=(0:1/freq_CED(1):(size(COP,1)-1)/freq_CED(1))';

                    %Motor: First 90 seconds of data at 2500 Hz
                    F_xyz=data_filt{count_data,1+jj}(1:1+90*freq_CED(2),3);
                    t_fast=(0:1/freq_CED(2):(size(F_xyz,1)-1)/freq_CED(2))';

                    %Button
                    Button_OFF_ON=data_filt{9,1+jj}{1,2}(1:1+90*freq_CED(1),:);

                    %Rectifying motor data
                    t_rect=linspace(min(t_fast),max(t_fast),freq_ard*90+1)';
                    freq_rect=1/(t_rect(2)-t_rect(1));
                    F_xyz_rect=zeros(length(t_rect)-1,1);
                    for mm = 1:length(t_rect)-1
```

```
F_xyz_rect(mm)=sqrt(freq_rect*trapz(t_fast(1+floor(freq_CED(2)/freq_rect)*(mm-1):...

floor(freq_CED(2)/freq_rect)*mm),F_xyz(1+floor(freq_CED(2)/freq_rect)*(mm-1):...
                            floor(freq_CED(2)/freq_rect)*mm).^2));
                    end
                    t_rect=t_rect(1:end-1)+1/freq_rect/2;

                end

                %Showing how long COP is
                fprintf('T = %.2f sec\n',(size(COP,1)-1)/freq_CED(1));

                %Plotting time series: COP_AP
                axes('Parent',tab(8)); subplot(3,1,1); plot(t,1000*COP(:,1));
                grid; xlabel('Time [s]'); ylabel('Back \leftarrow COP [mm] \rightarrow
Face');
                title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-' ...
                    protocol_order_abv{kk}(1:4) '-COP_{AP}-EC']); xlim([t(1) t(end)]);
hold on;

                %Plotting when button is pressed
                button_val=1000*(min(COP(:,1))-0.1*range(COP(:,1)));
                subplot(3,1,1); plot(t,Button_OFF_ON(:,1)*button_val,...
                    '.r','MarkerSize',4,'HandleVisibility','off'); hold on;
                subplot(3,1,1); plot(t,Button_OFF_ON(:,2)*button_val,...
                    '.g','MarkerSize',4,'HandleVisibility','off'); hold on;

                %Making Button variable into single column (0/1)
                Button_OFF_ON=Button_OFF_ON(:,2);
                Button_OFF_ON(isnan(Button_OFF_ON))=0;

                %Plotting time series: COP_ML
                axes('Parent',tab(8)); subplot(3,1,2); plot(t,1000*COP(:,2));
                grid; xlabel('Time [s]'); ylabel('Left \leftarrow COP [mm] \rightarrow
Right');
                title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-' ...
                    protocol_order_abv{kk}(1:4) '-COP_{AP}-EC']); xlim([t(1) t(end)]);

                %Plotting spatial plots: COP_AP vs. COP_ML
                axes('Parent',tab(9)); subplot(1,2,1);
plot(1000*COP(:,2),1000*COP(:,1));
                grid; xlabel('Left \leftarrow COP_{ML} [mm] \rightarrow Right');
                ylabel('Back \leftarrow COP_{AP} [mm] \rightarrow Face');
                title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-' ...
                    protocol_order_abv{kk}(1:4) '-COP-EC']); axis square;

                %Plotting motor data
                axes('Parent',tab(8)); subplot(3,1,3);
plot(t_fast,F_xyz,'LineWidth',0.5); hold on;
                subplot(3,1,3); plot(t_rect,F_xyz_rect,'LineWidth',2);
                grid; xlabel('Time [s]'); ylabel('F_{xyz} [N]');
                title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-' ...
                    protocol_order_abv{kk}(1:4) '-Motor']); xlim([t_fast(1)
t_fast(end)]);
                legend('Raw','Rectified','location','best');

                %Recording COP data
                data_analysis{count_data,jj+1}={[t COP] [t_fast F_xyz] [t_rect
F_xyz_rect]};
                count_data=count_data+1;

                %Recording button (0: Not pressed, 1: Pressed)
```

```
                data_analysis{9,jj+1}=Button_OFF_ON;

                %Clearing COP data
                clear COP t t_fast F_xyz t_rect F_xyz_rect Button_OFF_ON

            elseif kk == 6                  %Sway on MAT

                %Sway: First 90 seconds of data and downsample it to 100 Hz
                COP=data_filt{count_data,1+jj}(1:1+90*freq_CED(2),[1 2]);
                COP=downsample(COP,freq_CED(2)/freq_CED(1));
                t=(0:1/freq_CED(1):(size(COP,1)-1)/freq_CED(1))';

                %Motor: First 90 seconds of data at 2500 Hz
                F_xyz=data_filt{count_data,1+jj}(1:1+90*freq_CED(2),3);
                t_fast=(0:1/freq_CED(2):(size(F_xyz,1)-1)/freq_CED(2))';

                %Button
                Button_OFF_ON=data_filt{10,1+jj}{1,2}(1:1+90*freq_CED(1),:);

                %Rectifying motor data
                t_rect=linspace(min(t_fast),max(t_fast),freq_ard*90+1)';
                freq_rect=1/(t_rect(2)-t_rect(1));
                F_xyz_rect=zeros(length(t_rect)-1,1);
                for mm = 1:length(t_rect)-1

F_xyz_rect(mm)=sqrt(freq_rect*trapz(t_fast(1+floor(freq_CED(2)/freq_rect)*(mm-1):...

floor(freq_CED(2)/freq_rect)*mm),F_xyz(1+floor(freq_CED(2)/freq_rect)*(mm-1):...
                        floor(freq_CED(2)/freq_rect)*mm).^2));
                end
                t_rect=t_rect(1:end-1)+1/freq_rect/2;

                %Plotting time series: COP_AP
                axes('Parent',tab(10)); subplot(3,1,1); plot(t,1000*COP(:,1));
                grid; xlabel('Time [s]'); ylabel('Back \leftarrow COP [mm] \rightarrow
Face');
                title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-' ...
                    protocol_order_abv{kk}(1:3) '-COP_{AP}-EC']); xlim([t(1) t(end)]);
hold on;

                %Plotting when button is pressed
                button_val=1000*(min(COP(:,1))-0.1*range(COP(:,1)));
                subplot(3,1,1); plot(t,Button_OFF_ON(:,1)*button_val,...
                    '.r','MarkerSize',4,'HandleVisibility','off'); hold on;
                subplot(3,1,1); plot(t,Button_OFF_ON(:,2)*button_val,...
                    '.g','MarkerSize',4,'HandleVisibility','off'); hold on;

                %Making Button variable into single column (0/1)
                Button_OFF_ON=Button_OFF_ON(:,2);
                Button_OFF_ON(isnan(Button_OFF_ON))=0;

                %Plotting time series: COP_ML
                axes('Parent',tab(10)); subplot(3,1,2); plot(t,1000*COP(:,2));
                grid; xlabel('Time [s]'); ylabel('Left \leftarrow COP [mm] \rightarrow
Right');
                title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-' ...
                    protocol_order_abv{kk}(1:3) '-COP_{AP}-EC']); xlim([t(1) t(end)]);

                %Plotting spatial plots: COP_AP vs. COP_ML
                axes('Parent',tab(11)); subplot(1,2,1);
plot(1000*COP(:,2),1000*COP(:,1));
                grid; xlabel('Left \leftarrow COP_{ML} [mm] \rightarrow Right');
                ylabel('Back \leftarrow COP_{AP} [mm] \rightarrow Face');
```

```
                title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-' ...
                    protocol_order_abv{kk}(1:3) '-COP-EC']); axis square;

                %Plotting motor data
                axes('Parent',tab(10)); subplot(3,1,3);
plot(t_fast,F_xyz,'LineWidth',0.5); hold on;
                subplot(3,1,3); plot(t_rect,F_xyz_rect,'LineWidth',2);
                grid; xlabel('Time [s]'); ylabel('F_{xyz} [N]');
                title(['s' num2str(sub_no(ii)) '-' protocol_colors_abv{jj} '-' ...
                    protocol_order_abv{kk}(1:3) '-Motor']); xlim([t_fast(1)
t_fast(end)]);
                legend('Raw','Rectified','location','best');

                %Recording COP data
                data_analysis{count_data,jj+1}={[t COP] [t_fast F_xyz] [t_rect
F_xyz_rect]};
                count_data=count_data+1;

                %Recording button (0: Not pressed, 1: Pressed)
                data_analysis{10,jj+1}=Button_OFF_ON;

                %Clearing COP data
                clear COP t t_fast F_xyz t_rect F_xyz_rect Button_OFF_ON

            end

        end

        %Saving each tab as a figure
        for kk = 1:length(fig_tabs)
            tabgp.SelectedTab = tab(kk);
            saveas(fig,[path_res '4 Data Analysis\s' num2str(sub_no(ii)) '\s'
num2str(sub_no(ii)) ...
                '_' protocol_colors_abv{jj} '_' num2str(kk) '_' fig_names{kk}
'.jpeg']);
        end
        close all;

    end

    %Saving data_visual for subject
    save([path_pro 's' num2str(sub_no(ii)) '_DataAnalysis.mat'],'data_analysis');

end
```

```
%Camilo Giraldo - Healthy Vibratio Sway Study v2 - Linear Measures
%University of Kansas - Biodynamics Lab
%Last Update: April 30, 2021

clear; close all; clc;

%% General
%File paths
path_raw='C:\Users\17856\Desktop\Final_Vibration_study\Vibrotactile Study (2021)\';
path_res='C:\Users\17856\Desktop\Final_Vibration_study\Results\';
path_pro='C:\Users\17856\Desktop\Final_Vibration_study\Processed Data\';
```

```
sub_no=[1001 1002 1003 1004 1005 1006 1008 1009 1010 2001:2003];

%Color names
protocol_colors={'White' 'Pink' 'Brown' 'Placebo'};
protocol_colors_abv={'WH' 'PK' 'BR' 'PB'};

%CED Frequencies
freq_CED=[100 2500];

%Measure extraction frequency [Hz]
freq_nonlinear=50;

%Names of the times series
var_names={'Zeros' 'BL_GND' 'BL_MAT' 'THR_Sway' 'THR' 'STIM_Sway' ...
    'T_0_Sway' 'THR_Sway Butt' 'STIM_Sway Butt' 'T_0_Sway Butt'};

%% Subjects Weights
%Preallocating space for subject weights
sub_weight=zeros(1,length(sub_no));

%Going over all subjects
for ii = 1:length(sub_no)

    %Temp weight
    var_temp=[];

    %Going over all colors
    for jj = 1:length(protocol_colors)

        %Reading zeros file
        zeross=readmatrix([path_raw 's' num2str(sub_no(ii)) '\s' num2str(sub_no(ii))
...
            '_' protocol_colors_abv{jj} '_zeros3.txt']);
        zeross_check=0;
        for kk = 1:size(zeross,1)
            for LL = 2:size(zeross,2)
                if kk == 1 && isnan(zeross(kk,LL))
                    zeross(kk,LL)=zeross(kk+1,LL);
                elseif isnan(zeross(kk,LL))
                    fprintf('\n\t\t\tNaN at Row = %d, Column = %d',kk,LL);
                    zeross_check=1;
                end
            end
        end
        if zeross_check == 0
            fprintf(' Good!\n');
        else
            fprintf('\n');
        end
        zeross_mean=mean(zeross,1);

        %Reading baseline data on ground
        data=readmatrix([path_raw 's' num2str(sub_no(ii)) '\s' num2str(sub_no(ii)) ...
            '_' protocol_colors_abv{jj} '_BL_EC_GND.txt']);
        data_check=0;
        for mm = 1:size(data,1)
            for nn = 2:size(data,2)
                if mm == 1 && isnan(data(mm,nn))
                    data(mm,nn)=data(mm+1,nn);
                elseif isnan(data(mm,nn))
                    fprintf('\n\t\t\tNaN at Row = %d, Column = %d',mm,nn);
                    data_check=1;
```

```
                end
            end
        end
        if data_check == 0
            fprintf(' Good!\n');
        else
            fprintf('\n');
        end

        %Converting current data to N and N-m
        [data(:,2:7),dz_3364]=V2f_fp3364(data(:,2:7),zeross_mean(2:7),1000);
%Right foot on 3364
        [data(:,8:13),dz_3477]=V2f_fp3477(data(:,8:13),zeross_mean(8:13),1000);
%Left foot on 3477
        dz=mean([dz_3364 dz_3477]);

        %Rotating data around z-axis -90 degrees
        data=[data(:,1) ...                                                    %Time
            -data(:,3) data(:,2) data(:,4) -data(:,6) data(:,5) data(:,7) ...   %Right
3364
            -data(:,9) data(:,8) data(:,10) -data(:,12) data(:,11) data(:,13)]; %Left
3477

        %Combining force plates
        data_comb=Comb_fp3477_fp3364(data(:,8:13),data(:,2:7));
        data_comb=[data(:,1) data_comb];

        %Updating temp variable
        var_temp=[var_temp; data_comb(:,4)];

    end

    %Getting subjects' masses [kg]
    sub_weight(ii)=mean(var_temp)/9.81;

end

%Clearing large variables
clear zeross data data_comb var_temp

%% Summary of Threshold Information
%Loading regression equations
load(['StaticGForce_Processed_01.mat'],'fitresult_magnitude_all',...
    'fitresult_frequency_all');
%Columns of table
tab_titles={'Subject' 'Noise' '0.9*THR [%]' 'Favg [N]' 'favg [Hz]' '%Sub-STIM' '%Sub-
T0'};

%Preallocating space for table
tab_cell=cell(1+length(sub_no)*length(protocol_colors),length(tab_titles));
tab_cell(1,:)=tab_titles; tab_count=2;

%Going over all subjects
for ii = 1:length(sub_no)

    %Loading data
    load([path_pro 's' num2str(sub_no(ii)) '_DataAnalysis.mat'],'data_analysis');

    %Going over all noises
    for jj = 1:length(protocol_colors)

        %Updating table: subject, noise, 90
        tab_cell{tab_count,1}=['s' num2str(sub_no(ii))];
```

```
        tab_cell{tab_count,2}=protocol_colors{jj};
        tab_cell{tab_count,3}=round(0.9*data_analysis{5,1+jj}{2}/127*100,2);

        %Calculating average force
        data_predint=round(predint(fitresult_magnitude_all{4,2,2},...
            [0.9*data_analysis{5,1+jj}{2}/127*100 sub_weight(ii)],0.95,...
            'functional','on'),2);
        data=round(feval(fitresult_magnitude_all{4,2,2},...
            [0.9*data_analysis{5,1+jj}{2}/127*100 sub_weight(ii)]),2);
        tab_cell{tab_count,4}=[num2str(data) '|(' num2str(data_predint(1)) ...
            '-' num2str(data_predint(2)) ')'];

        %Calculating average frequency
        data_predint=round(predint(fitresult_frequency_all{4,2,2},...
            [0.9*data_analysis{5,1+jj}{2}/127*100 sub_weight(ii)],0.95,...
            'functional','on'),2);
        data=round(feval(fitresult_frequency_all{4,2,2},...
            [0.9*data_analysis{5,1+jj}{2}/127*100 sub_weight(ii)]),2);
        tab_cell{tab_count,5}=[num2str(data) '|(' num2str(data_predint(1)) ...
            '-' num2str(data_predint(2)) ')'];

        %Determining percent of time STIM was subthreshold

tab_cell{tab_count,6}=round(sum(data_analysis{9,1+jj})/length(data_analysis{9,1+jj})*1
00,2);

tab_cell{tab_count,end}=round(sum(data_analysis{10,1+jj})/length(data_analysis{10,1+jj
})*100,2);

        %Updating counter
        tab_count=tab_count+1;

    end

end

%Exporting cell to table
tab=cell2table(tab_cell(2:end,:),'VariableNames',tab_titles);
writetable(tab,[path_res '5 Measures\THR_Info.csv'],'Delimiter',';');

%% Figures of 0.9*Thresholds
%Organizing data
tab_cell(:,[2 6 7])=[];
for ii = 2:size(tab_cell,1)
    tab_cell{ii,1}=str2double(tab_cell{ii,1}(2:end));
end
for ii = 2:size(tab_cell,1)
    for jj = [3 4]
        idx=strfind(tab_cell{ii,jj},'|');
        tab_cell{ii,jj}=str2double(tab_cell{ii,jj}(1:idx-1));
    end
end
tab_cell_titles=tab_cell(1,:);
tab_data=cell2mat(tab_cell(2:end,:));

%Setting up the figure
fig=figure('Name',['Subject All | Healthy Vibration Sway v2 | '...
    'THR Display'],'Units','Normalized','Outerposition',[0 0 1 1]);

%Titles for plots
title_all={'0.9*THR: THR Value [%]' '0.9*THR: Force [N]' '0.9*THR: Frequency [Hz]'};

%T-test results
```

```
ttest_results=cell(1,3);

for ii = 1:3

    %Preallocating space
    bar_mean=zeros(2,2); bar_stdv=bar_mean;

    %Calculating average and standard deviations
    bar_mean(1,1)=mean(tab_data(tab_data(:,1)<2000,ii+1));
    bar_mean(1,2)=mean(tab_data(tab_data(:,1)>2000,ii+1));
    bar_stdv(1,1)=std(tab_data(tab_data(:,1)<2000,ii+1));
    bar_stdv(1,2)=std(tab_data(tab_data(:,1)>2000,ii+1));

    %Plotting bar plot
    subplot(1,3,ii); BarPlot_KU(bar_mean,bar_stdv,{' ' ' '},{'Old' 'Young + Foam'},...
        'southoutside','horizontal'); ylabel(' '); xlim([0.5 1.5]);
    title(title_all{ii});

    %Statistics

[h,p,ci,stats]=ttest2(tab_data(tab_data(:,1)<2000,ii+1),tab_data(tab_data(:,1)>2000,ii
+1));
    ttest_results{ii}={h,p,ci,stats};

end

%Saving figure
saveas(fig,[path_res '5 Measures\THR_Plots.jpeg']);
close all;

%% Extracting Measures Out of COP Time Series
%Names of the linear measures to be extracted per time series
meas_names={'RMS or 95%Ellipse' 'SE(m=2,R=0.1)' 'DFA_{\alpha}(0.5s-15s)'
'DFA_{R^2}(0.5s-15s)'}';
meas_cols={'AP' 'ML' 'Spatial'};

%DFA calculation parameters (According to Melanie's work)
DFA_t_min=0.5; DFA_t_max=15;

%Sample and approximate entropy parameters (According to Paris)
m=2; R=0.1;

%Going over all the subjects
for ii = 1:length(sub_no)

    %Preallocating space for subject's measures
    meas=cell(length(var_names),5);
    for jj = 1:length(var_names)
        meas{jj,1}=var_names{jj};
    end

    %Loading analysis data for each subject
    load([path_pro 's' num2str(sub_no(ii)) '_DataAnalysis.mat'],'data_analysis');

    %Command window message
    fprintf('Subject: s%d\n',sub_no(ii));

    %Going over all the sessions
    for jj = 1:length(protocol_colors)

        %Command window message
        fprintf('\tSession: %s\n',protocol_colors{jj});
```

```
        %Going over all time series
        for kk = 1:length(var_names)

            %Command window message
            fprintf('\t\t%s: ',data_analysis{kk,1});

            %Preallocating space for linear measures
            meas_temp=zeros(length(meas_names),length(meas_cols));

            %No linear measures for: Zeros, 421 and Button
            if kk == 1 || kk == 5 || kk >= 8

                %Command window message
                fprintf('N/A\n');

            else

                %Grabbing the COP and time series
                if kk == 2      %BL on Ground
                    t=data_analysis{kk,jj+1}(:,1);          %Time series
                    COP_AP=data_analysis{kk,jj+1}(:,2);     %COP_AP
                    COP_ML=data_analysis{kk,jj+1}(:,3);     %COP_ML
                else
                    t=data_analysis{kk,jj+1}{1}(:,1);          %Time series
                    COP_AP=data_analysis{kk,jj+1}{1}(:,2);     %COP_AP
                    COP_ML=data_analysis{kk,jj+1}{1}(:,3);     %COP_ML
                end

                %Downsampling data to 50 Hz for measure extraction
                t=downsample(t,freq_CED(1)/freq_nonlinear);
                COP_AP=downsample(COP_AP,freq_CED(1)/freq_nonlinear);
                COP_ML=downsample(COP_ML,freq_CED(1)/freq_nonlinear);

                %Calculating COP spatial
                COP_Spatial=sqrt(COP_AP.^2+COP_ML.^2);

                %Extracting measures from COP_AP time series
                meas_temp(1,1)=rms(COP_AP);
                meas_temp(2,1)=SampEn_Opt(COP_AP,m,R);

[~,~,~,~,meas_temp(3,1),meas_temp(4,1)]=DFA_KU(COP_AP,freq_nonlinear,DFA_t_min,DFA_t_m
ax,1);

                %Extracting measures from COP_ML time series
                meas_temp(1,2)=rms(COP_ML);
                meas_temp(2,2)=SampEn_Opt(COP_ML,m,R);

[~,~,~,~,meas_temp(3,2),meas_temp(4,2)]=DFA_KU(COP_ML,freq_nonlinear,DFA_t_min,DFA_t_m
ax,1);

                %Extracting measures from COP_Spatial time series
                [meas_temp(1,3),~,~]=Ellip_2D([COP_AP COP_ML],[],[]);
                meas_temp(2,3)=SampEn_Opt(COP_Spatial,m,R);

[~,~,~,~,meas_temp(3,3),meas_temp(4,3)]=DFA_KU(COP_Spatial,freq_nonlinear,DFA_t_min,DF
A_t_max,1);

                %Recording linear measures
                meas{kk,1+jj}=meas_temp;

                %Command window message
                fprintf('Good!\n');
```

```
            end

        end

    end

    %Saving linear measures for each subject -------- FIRST TIME!!!
    save([path_pro 's' num2str(sub_no(ii))
'_Measures.mat'],'meas','meas_names','meas_cols');

end

%% Plotting Measures per Subject
%X-labels for plots
x_label_names={'BL_{GND}' 'BL_{MAT}' 'THR' 'STIM' 'T0'};

%Line Spec for plots
line_specs={'o-k' 'o-m' 'o-r' 'o-b'};

%Going over all the subjects
for ii = 1:length(sub_no)

    %Loading measures for each subject, and removing rows without measures
    load([path_pro 's' num2str(sub_no(ii)) '_Measures.mat'],'meas');
    meas([1 5 8 9 10],:)=[];

    %Loading measure names and other titles
    if ii == 1
        load([path_pro 's' num2str(sub_no(ii))
'_Measures.mat'],'meas_names','meas_cols');
        meas_names{2}='SampEn(m=2,R=0.1)'; meas_names{3}='DFA \alpha(0.5s-15s)';
        meas_names{4}='DFA R^2';
    end

    %Setting up the figure
    fig=figure('Name',['Subject ' num2str(sub_no(ii)) ' | Healthy Vibration Sway v2 |
'...
        'Measures Display'],'Units','Normalized','Outerposition',[0 0 1 1]);

    %Preallocating data for plotting
    plot_data_AP=zeros(length(protocol_colors),length(x_label_names),4);
    plot_data_ML=plot_data_AP;
    plot_data_Spatial=plot_data_AP;

    %Going over all the sessions
    for jj = 1:length(protocol_colors)

        %Going over all stages of protocol
        for kk = 1:length(x_label_names)

            %Going over all measures
            for LL = 1:4

                %Allocating values
                plot_data_AP(jj,kk,LL)=meas{kk,jj+1}(LL,1);
                plot_data_ML(jj,kk,LL)=meas{kk,jj+1}(LL,2);
                plot_data_Spatial(jj,kk,LL)=meas{kk,jj+1}(LL,3);

            end

        end

    end
```

```
%Going over all measures
for LL = 1:4

    %Going over all colors
    for jj = 1:length(protocol_colors)

        %Not R2 of DFA
        if LL < 4

            %Plotting AP measures
            subplot(7,3,6*(LL-1)+[1 4]); plot(1:length(x_label_names),...
                plot_data_AP(jj,:,LL),line_specs{jj}); hold on;

            %Details
            if jj == length(protocol_colors)
                grid; xlim([1-0.5 length(x_label_names)+0.5]);
                xticks(1:length(x_label_names)); ylabel(meas_names{LL});
                xticklabels([]);

                %Title of plots
                if LL == 1
                    title(['s' num2str(sub_no(ii)) ' - ' meas_cols{1}]);
                end

            end

            %Plotting ML measures
            subplot(7,3,6*(LL-1)+[1 4]+1); plot(1:length(x_label_names),...
                plot_data_ML(jj,:,LL),line_specs{jj}); hold on;

            %Details
            if jj == length(protocol_colors)
                grid; xlim([1-0.5 length(x_label_names)+0.5]);
                xticks(1:length(x_label_names)); xticklabels([]);

                %Title of plots
                if LL == 1
                    title(['s' num2str(sub_no(ii)) ' - ' meas_cols{2}]);
                end

            end

            %Plotting Spatial measures
            subplot(7,3,6*(LL-1)+[1 4]+2); plot(1:length(x_label_names),...
                plot_data_Spatial(jj,:,LL),line_specs{jj}); hold on;

            %Details
            if jj == length(protocol_colors)
                grid; xlim([1-0.5 length(x_label_names)+0.5]);
                xticks(1:length(x_label_names)); xticklabels([]);

                %Title of plots
                if LL == 1
                    title(['s' num2str(sub_no(ii)) ' - ' meas_cols{3}]);
                end

            end

        else

            %Plotting AP measures
            subplot(7,3,19); plot(1:length(x_label_names),...
```

```
                    plot_data_AP(jj,:,LL),line_specs{jj}); hold on;

                %Details
                if jj == length(protocol_colors)
                    grid; xlim([1-0.5 length(x_label_names)+0.5]);
                    xticks(1:length(x_label_names)); xticklabels(x_label_names);
                    legend(protocol_colors_abv,'orientation','horizontal',...
                        'location','north'); ylabel(meas_names{LL});
                end

                %Plotting ML measures
                subplot(7,3,20); plot(1:length(x_label_names),...
                    plot_data_ML(jj,:,LL),line_specs{jj}); hold on;

                %Details
                if jj == length(protocol_colors)
                    grid; xlim([1-0.5 length(x_label_names)+0.5]);
                    xticks(1:length(x_label_names)); xticklabels(x_label_names);
                    legend(protocol_colors_abv,'orientation','horizontal',...
                        'location','north');
                end

                %Plotting Spatial measures
                subplot(7,3,21); plot(1:length(x_label_names),...
                    plot_data_Spatial(jj,:,LL),line_specs{jj}); hold on;

                %Details
                if jj == length(protocol_colors)
                    grid; xlim([1-0.5 length(x_label_names)+0.5]);
                    xticks(1:length(x_label_names)); xticklabels(x_label_names);
                    legend(protocol_colors_abv,'orientation','horizontal',...
                        'location','north');
                end

            end

        end

    end

    %Saving plot
    saveas(fig,[path_res '5 Measures\s' num2str(sub_no(ii)) '_Measures.jpeg']);
    close all;

end
```

# Appendix C : Python Codes

```
%Zaccur Nkrumah - Healthy Vibration Sway Study v2 - Statistical Analysis
%University of Kansas - Biodynamics Lab
%Date:March 10, 2022

import scipy.io as spio
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

# import subject measures
```

```
Subject_measures = pd.read_excel('Measures_data.xlsx')
# subject data
subject_d = pd.read_excel('Subject_Data.xlsx')
subject_d.drop([6,10,11,12,13],inplace=True)

#Function to add session number to the data
def session_order(x):
    Subject =x[0]
    Noise = x[1]
    if Subject == 1001:
        if Noise == "'White'":
            return 4
        elif Noise == "'Pink'":
            return 1
        elif Noise == "'Brown'":
            return 2
        else :
            return 3

    elif Subject == 1002:
        if Noise == "'White'":
            return 4
        elif Noise == "'Pink'":
            return 2
        elif Noise == "'Brown'":
            return 3
        else :
            return 1
    elif Subject == 1003:
        if Noise == "'White'":
            return 2
        elif Noise == "'Pink'":
            return 1
        elif Noise == "'Brown'":
            return 3
        else :
            return 4
    elif Subject == 1004:
        if Noise == "'White'":
            return 3
        elif Noise == "'Pink'":
            return 2
        elif Noise == "'Brown'":
            return 1
        else :
            return 4
    elif Subject == 1005:
        if Noise == "'White'":
            return 4
        elif Noise == "'Pink'":
            return 1
        elif Noise == "'Brown'":
            return 2
        else :
            return 3
    elif Subject == 1006:
        if Noise == "'White'":
            return 4
        elif Noise == "'Pink'":
            return 2
        elif Noise == "'Brown'":
            return 3
```

```
        else :
            return 1
    elif Subject == 1008:
        if Noise == "'White'":
            return 3
        elif Noise == "'Pink'":
            return 2
        elif Noise == "'Brown'":
            return 1
        else :
            return 4
    elif Subject == 1009:
        if Noise == "'White'":
            return 4
        elif Noise == "'Pink'":
            return 1
        elif Noise == "'Brown'":
            return 2
        else :
            return 3
    elif Subject == 1010:
        if Noise == "'White'":
            return 1
        elif Noise == "'Pink'":
            return 4
        elif Noise == "'Brown'":
            return 3
        else :
            return 2


#Add session no to the table
Subject_measures['session_order_no']=Subject_measures[['Subject','Noise']].apply(session_order,axis=1)

#Baseline measures
#Baseline measures in the AP direction
sub_BL_AP=Subject_measures[(Subject_measures['Stage']=="'BL_{MAT}'")& (Subject_measures['Time Series']=="'AP'")]

#Baseline measures in the ML direction\
sub_BL_ML=Subject_measures[(Subject_measures['Stage']=="'BL_{MAT}'")& (Subject_measures['Time Series']=="'ML'")]

#Stimulation measures
#Stimulation measures in the AP direction
sub_STIM_AP=Subject_measures[(Subject_measures['Stage']=="'STIM'")& (Subject_measures['Time Series']=="'AP'")]

#Stimulation measures in the ML direction
sub_STIM_ML=Subject_measures[(Subject_measures['Stage']=="'STIM'")& (Subject_measures['Time Series']=="'ML'")]


#Reset indexes for the different measures in all directions
sub_BL_AP.reset_index(inplace=True)
sub_BL_AP.drop('index',axis=1,inplace=True)
sub_BL_ML.reset_index(inplace=True)
sub_BL_ML.drop('index',axis=1,inplace=True)
sub_STIM_AP.reset_index(inplace=True)
sub_STIM_AP.drop('index',axis=1,inplace=True)
sub_STIM_ML.reset_index(inplace=True)
sub_STIM_ML.drop('index',axis=1,inplace=True)

#Boxplots for Baseline Sample Entropy in AP and ML direction
plt.figure(figsize=(10,12))
plt.subplot(2,1,1)
```

```
sns.boxplot(x='Subject',y='SampEn(m=2,R=0.1)',data =sub_BL_AP)
plt.title('AP')
plt.subplot(2,1,2)
sns.boxplot(x='Subject',y='SampEn(m=2,R=0.1)',data =sub_BL_ML)
plt.title('ML')


#Boxplots for Baseline DFA in AP and ML direction
plt.figure(figsize=(10,12))
plt.subplot(2,1,1)
sns.boxplot(x='Subject',y='DFA_alpha(0.5s-15s)',data =sub_BL_AP)
plt.title('AP')
plt.subplot(2,1,2)
sns.boxplot(x='Subject',y='DFA_alpha(0.5s-15s)',data =sub_BL_ML)
plt.title('ML')



#Boxplots for Baseline RMS in AP and ML direction
plt.figure(figsize=(10,12))
plt.subplot(2,1,1)
sns.boxplot(x='Subject',y='RMS or 95%Ellipse',data =sub_BL_AP)
plt.title('AP')
plt.subplot(2,1,2)
sns.boxplot(x='Subject',y='RMS or 95%Ellipse',data =sub_BL_ML)
plt.title('ML')


#Boxplots for Baseline Sample Entropy in all direction for the session orders
plt.figure(figsize=(10,12))
plt.subplot(2,1,1)
sns.boxplot(x='session_order_no',y='SampEn(m=2,R=0.1)',data =sub_BL_AP)
plt.title('AP')
plt.subplot(2,1,2)
sns.boxplot(x='session_order_no',y='SampEn(m=2,R=0.1)',data =sub_BL_ML)
plt.title('ML')


#Boxplots for Baseline DFA in all direction for the session orders
plt.figure(figsize=(10,12))
plt.subplot(2,1,1)
sns.boxplot(x='session_order_no',y='DFA_alpha(0.5s-15s)',data =sub_BL_AP)
plt.title('AP')
plt.subplot(2,1,2)
sns.boxplot(x='session_order_no',y='DFA_alpha(0.5s-15s)',data =sub_BL_ML)
plt.title('ML')


#Boxplots for Baseline RMS or 95%Ellipse in all direction for the session orders
plt.figure(figsize=(10,12))
plt.subplot(2,1,1)
sns.boxplot(x='session_order_no',y='RMS or 95%Ellipse',data =sub_BL_AP)
plt.title('AP')
plt.subplot(2,1,2)
sns.boxplot(x='session_order_no',y='RMS or 95%Ellipse',data =sub_BL_ML)
plt.title('ML')


# Baseline and Stimulation Measures Sample Entropy
sub_BL_STIM_ML=pd.concat([sub_BL_ML,sub_STIM_ML])

#White and ML
```

```
sub_WH_ML=sub_BL_STIM_ML[sub_BL_STIM_ML['Noise']=="'White'"]

# White Baseline and ML
WH_BL_ML = sub_BL_ML[sub_BL_ML['Noise']=="'White'"]

# White Stimulation and ML
WH_STIM_ML = sub_STIM_ML[sub_STIM_ML['Noise']=="'White'"]

WH_BL_STIM_ML
=pd.concat([WH_BL_ML['Subject'],WH_BL_ML['Noise'],WH_BL_ML['SampEn(m=2,R=0.1)'],WH_STIM_ML['SampEn(m=2
,R=0.1)']],axis=1)
WH_BL_STIM_ML.reset_index(inplace=True)
WH_BL_STIM_ML.drop('index',axis=1,inplace=True)
WH_BL_STIM_ML.columns =['Subject','Noise','BL_SE','STIM_SE']

#Pink and ML
sub_PK_ML=sub_BL_STIM_ML[sub_BL_STIM_ML['Noise']=="'Pink'"]

# Pink Baseline and ML
PK_BL_ML = sub_BL_ML[sub_BL_ML['Noise']=="'Pink'"]

# Pink Stimulation and ML
PK_STIM_ML = sub_STIM_ML[sub_STIM_ML['Noise']=="'Pink'"]

PK_BL_STIM_ML
=pd.concat([PK_BL_ML['Subject'],PK_BL_ML['Noise'],PK_BL_ML['SampEn(m=2,R=0.1)'],PK_STIM_ML['SampEn(m=2,R=
0.1)']],axis=1)
PK_BL_STIM_ML.reset_index(inplace=True)
PK_BL_STIM_ML.drop('index',axis=1,inplace=True)
PK_BL_STIM_ML.columns =['Subject','Noise','BL_SE','STIM_SE']

#Brown and ML
sub_BR_ML=sub_BL_STIM_ML[sub_BL_STIM_ML['Noise']=="'Brown'"]

# Brown Baseline and ML
BR_BL_ML = sub_BL_ML[sub_BL_ML['Noise']=="'Brown'"]

# Brown Stimulation and ML
BR_STIM_ML = sub_STIM_ML[sub_STIM_ML['Noise']=="'Brown'"]

BR_BL_STIM_ML
=pd.concat([BR_BL_ML['Subject'],BR_BL_ML['Noise'],BR_BL_ML['SampEn(m=2,R=0.1)'],BR_STIM_ML['SampEn(m=2,R=
0.1)']],axis=1)
BR_BL_STIM_ML.reset_index(inplace=True)
BR_BL_STIM_ML.drop('index',axis=1,inplace=True)
BR_BL_STIM_ML.columns =['Subject','Noise','BL_SE','STIM_SE']

#Placebo and ML
sub_PL_ML=sub_BL_STIM_ML[sub_BL_STIM_ML['Noise']=="'Placebo'"]

# Placebo Baseline and ML
PL_BL_ML = sub_BL_ML[sub_BL_ML['Noise']=="'Placebo'"]

# Placebo Stimulation and ML
PL_STIM_ML = sub_STIM_ML[sub_STIM_ML['Noise']=="'Placebo'"]

PL_BL_STIM_ML
=pd.concat([PL_BL_ML['Subject'],PL_BL_ML['Noise'],PL_BL_ML['SampEn(m=2,R=0.1)'],PL_STIM_ML['SampEn(m=2,R=0
.1)']],axis=1)
PL_BL_STIM_ML.reset_index(inplace=True)
PL_BL_STIM_ML.drop('index',axis=1,inplace=True)
PL_BL_STIM_ML.columns =['Subject','Noise','BL_SE','STIM_SE']
```

```
# Pointplots for Baseline and Stimulation Sample Entropy across vibration treatments
plt.figure(figsize=(15,15))
plt.subplot(4,2,1)
sns.pointplot(x='Stage',y='SampEn(m=2,R=0.1)',data =sub_WH_ML,hue='Subject')
plt.title('White')

plt.subplot(4,2,2)
sns.pointplot(x='Stage',y='SampEn(m=2,R=0.1)',data =sub_PK_ML,hue='Subject')
plt.title('Pink')

plt.subplot(4,2,3)
sns.pointplot(x='Stage',y='SampEn(m=2,R=0.1)',data =sub_BR_ML,hue='Subject')
plt.title('Brown')

plt.subplot(4,2,4)
sns.pointplot(x='Stage',y='SampEn(m=2,R=0.1)',data =sub_PL_ML,hue='Subject')
plt.title('Placebo')
plt.tight_layout(pad=3.0)


# Baseline and Stimulation Measures DFA
#White and ML
sub_WH_ML=sub_BL_STIM_ML[sub_BL_STIM_ML['Noise']=="'White'"]

# White Baseline and ML
WH_BL_ML = sub_BL_ML[sub_BL_ML['Noise']=="'White'"]

# White Stimulation and ML
WH_STIM_ML = sub_STIM_ML[sub_STIM_ML['Noise']=="'White'"]

WH_BL_STIM_ML =pd.concat([WH_BL_ML['Subject'],WH_BL_ML['Noise'],WH_BL_ML['DFA_alpha(0.5s-
15s)'],WH_STIM_ML['DFA_alpha(0.5s-15s)']],axis=1)
WH_BL_STIM_ML.reset_index(inplace=True)
WH_BL_STIM_ML.drop('index',axis=1,inplace=True)
WH_BL_STIM_ML.columns =['Subject','Noise','BL_SE','STIM_SE']

#Pink and ML
sub_PK_ML=sub_BL_STIM_ML[sub_BL_STIM_ML['Noise']=="'Pink'"]

# Pink Baseline and ML
PK_BL_ML = sub_BL_ML[sub_BL_ML['Noise']=="'Pink'"]

# Pink Stimulation and ML
PK_STIM_ML = sub_STIM_ML[sub_STIM_ML['Noise']=="'Pink'"]

PK_BL_STIM_ML =pd.concat([PK_BL_ML['Subject'],PK_BL_ML['Noise'],PK_BL_ML['DFA_alpha(0.5s-
15s)'],PK_STIM_ML['DFA_alpha(0.5s-15s)']],axis=1)
PK_BL_STIM_ML.reset_index(inplace=True)
PK_BL_STIM_ML.drop('index',axis=1,inplace=True)
PK_BL_STIM_ML.columns =['Subject','Noise','BL_SE','STIM_SE']

#Brown and ML
sub_BR_ML=sub_BL_STIM_ML[sub_BL_STIM_ML['Noise']=="'Brown'"]
# Brown Baseline and AP
BR_BL_ML = sub_BL_ML[sub_BL_ML['Noise']=="'Brown'"]

# Brown Stimulation and ML
BR_STIM_ML = sub_STIM_ML[sub_STIM_ML['Noise']=="'Brown'"]

BR_BL_STIM_ML =pd.concat([BR_BL_ML['Subject'],BR_BL_ML['Noise'],BR_BL_ML['DFA_alpha(0.5s-
15s)'],BR_STIM_ML['DFA_alpha(0.5s-15s)']],axis=1)
```

```
BR_BL_STIM_ML.reset_index(inplace=True)
BR_BL_STIM_ML.drop('index',axis=1,inplace=True)
BR_BL_STIM_ML.columns =['Subject','Noise','BL_SE','STIM_SE']

#Placebo and ML
sub_PL_ML=sub_BL_STIM_ML[sub_BL_STIM_ML['Noise']=="'Placebo'"]

# Placebo Baseline and ML
PL_BL_ML = sub_BL_ML[sub_BL_ML['Noise']=="'Placebo'"]

# Placebo Stimulation and ML
PL_STIM_ML = sub_STIM_ML[sub_STIM_ML['Noise']=="'Placebo'"]

PL_BL_STIM_ML =pd.concat([PL_BL_ML['Subject'],PL_BL_ML['Noise'],PL_BL_ML['DFA_alpha(0.5s-
15s)'],PL_STIM_ML['DFA_alpha(0.5s-15s)']],axis=1)
PL_BL_STIM_ML.reset_index(inplace=True)
PL_BL_STIM_ML.drop('index',axis=1,inplace=True)
PL_BL_STIM_ML.columns =['Subject','Noise','BL_SE','STIM_SE']



# Pointplot for Baseline and Stimulation DFA across vibration treatments
plt.figure(figsize=(15,15))
plt.subplot(4,2,1)
sns.pointplot(x='Stage',y='DFA_alpha(0.5s-15s)',data =sub_WH_ML,hue='Subject')
plt.title('White')

plt.subplot(4,2,2)
sns.pointplot(x='Stage',y='DFA_alpha(0.5s-15s)',data =sub_PK_ML,hue='Subject')
plt.title('Pink')

plt.subplot(4,2,3)
sns.pointplot(x='Stage',y='DFA_alpha(0.5s-15s)',data =sub_BR_ML,hue='Subject')
plt.title('Brown')

plt.subplot(4,2,4)
sns.pointplot(x='Stage',y='DFA_alpha(0.5s-15s)',data =sub_PL_ML,hue='Subject')
plt.title('Placebo')
plt.tight_layout(pad=3.0)



#Baseline and Stimulation Measures RMS
#White and ML
sub_WH_ML=sub_BL_STIM_ML[sub_BL_STIM_ML['Noise']=="'White'"]

# White Baseline and ML
WH_BL_ML = sub_BL_ML[sub_BL_ML['Noise']=="'White'"]

# White Stimulation and ML
WH_STIM_ML = sub_STIM_ML[sub_STIM_ML['Noise']=="'White'"]

WH_BL_STIM_ML =pd.concat([WH_BL_ML['Subject'],WH_BL_ML['Noise'],WH_BL_ML['RMS or
95%Ellipse'],WH_STIM_ML['RMS or 95%Ellipse']],axis=1)
WH_BL_STIM_ML.reset_index(inplace=True)
WH_BL_STIM_ML.drop('index',axis=1,inplace=True)
WH_BL_STIM_ML.columns =['Subject','Noise','BL_RMS','STIM_RMS']

#Pink and ML
sub_PK_ML=sub_BL_STIM_ML[sub_BL_STIM_ML['Noise']=="'Pink'"]

# Pink Baseline and ML
```

```
PK_BL_ML = sub_BL_ML[sub_BL_ML['Noise']=="'Pink'"]

# Pink Stimulation and ML
PK_STIM_ML = sub_STIM_ML[sub_STIM_ML['Noise']=="'Pink'"]

PK_BL_STIM_ML =pd.concat([PK_BL_ML['Subject'],PK_BL_ML['Noise'],PK_BL_ML['RMS or
95%Ellipse'],PK_STIM_ML['RMS or 95%Ellipse']],axis=1)
PK_BL_STIM_ML.reset_index(inplace=True)
PK_BL_STIM_ML.drop('index',axis=1,inplace=True)
PK_BL_STIM_ML.columns =['Subject','Noise','BL_RMS','STIM_RMS']

#Brown and ML
sub_BR_ML=sub_BL_STIM_ML[sub_BL_STIM_ML['Noise']=="'Brown'"]
# Brown Baseline and ML
BR_BL_ML = sub_BL_ML[sub_BL_ML['Noise']=="'Brown'"]

# Brown Stimulation and ML
BR_STIM_ML = sub_STIM_ML[sub_STIM_ML['Noise']=="'Brown'"]

BR_BL_STIM_ML =pd.concat([BR_BL_ML['Subject'],BR_BL_ML['Noise'],BR_BL_ML['RMS or
95%Ellipse'],BR_STIM_ML['RMS or 95%Ellipse']],axis=1)
BR_BL_STIM_ML.reset_index(inplace=True)
BR_BL_STIM_ML.drop('index',axis=1,inplace=True)
BR_BL_STIM_ML.columns =['Subject','Noise','BL_RMS','STIM_RMS']

#Placebo and ML
sub_PL_ML=sub_BL_STIM_ML[sub_BL_STIM_ML['Noise']=="'Placebo'"]

# Placebo Baseline and ML
PL_BL_ML = sub_BL_ML[sub_BL_ML['Noise']=="'Placebo'"]

# Placebo Stimulation and ML
PL_STIM_ML = sub_STIM_ML[sub_STIM_ML['Noise']=="'Placebo'"]

PL_BL_STIM_ML =pd.concat([PL_BL_ML['Subject'],PL_BL_ML['Noise'],PL_BL_ML['RMS or
95%Ellipse'],PL_STIM_ML['RMS or 95%Ellipse']],axis=1)
PL_BL_STIM_ML.reset_index(inplace=True)
PL_BL_STIM_ML.drop('index',axis=1,inplace=True)
PL_BL_STIM_ML.columns =['Subject','Noise','BL_RMS','STIM_RMS']



# Pointplot for Baseline and Stimulation RMS across vibration treatments
plt.figure(figsize=(15,15))
plt.subplot(4,2,1)
sns.pointplot(x='Stage',y='RMS or 95%Ellipse',data =sub_WH_ML,hue='Subject')
plt.title('White')

plt.subplot(4,2,2)
sns.pointplot(x='Stage',y='RMS or 95%Ellipse',data =sub_PK_ML,hue='Subject')
plt.title('Pink')

plt.subplot(4,2,3)
sns.pointplot(x='Stage',y='RMS or 95%Ellipse',data =sub_BR_ML,hue='Subject')
plt.title('Brown')

plt.subplot(4,2,4)
sns.pointplot(x='Stage',y='RMS or 95%Ellipse',data =sub_PL_ML,hue='Subject')
plt.title('Placebo')
plt.tight_layout(pad=3.0)
```

```
#Difference between baseline and stimulation measures in ML
# Sample Entropy
Diff_STIM_BL_ML_SE = sub_STIM_ML['SampEn(m=2,R=0.1)'] - sub_BL_ML['SampEn(m=2,R=0.1)']
BL_STIM_NOISE_ML =pd.concat([
sub_BL_ML['Subject'],sub_BL_ML['Noise'],sub_STIM_ML['SampEn(m=2,R=0.1)'],sub_BL_ML['SampEn(m=2,R=0.1)'],Diff_
STIM_BL_ML_SE],axis=1)
BL_STIM_NOISE_ML.columns=['Subject','Noise','STIM_SE','BL_SE_ML','Diff_ST_BL_ML']
BL_STIM_NOISE_ML['Diff_ST_BL_AB']=BL_STIM_NOISE_ML['Diff_ST_BL_ML'].abs()

#DFA
Diff_STIM_BL_ML_DFA = sub_STIM_ML['DFA_alpha(0.5s-15s)'] - sub_BL_ML['DFA_alpha(0.5s-15s)']
BL_STIM_NOISE_DFA_ML =pd.concat([sub_BL_ML['Subject'],sub_BL_ML['Noise'],sub_STIM_ML['DFA_alpha(0.5s-
15s)'],sub_BL_ML['DFA_alpha(0.5s-15s)'],Diff_STIM_BL_ML_DFA],axis=1)
BL_STIM_NOISE_DFA_ML.columns=['Subject','Noise','STIM_DFA_ML','BL_DFA_ML','Diff_ST_BL_DFA_ML']
BL_STIM_NOISE_DFA_ML['Diff_ST_BL_DFA_ML_AB']=BL_STIM_NOISE_DFA_ML['Diff_ST_BL_DFA_ML'].abs()

# RMS
Diff_STIM_BL_ML_rms = sub_STIM_ML['RMS or 95%Ellipse'] - sub_BL_ML['RMS or 95%Ellipse']
BL_STIM_NOISE_rms_ML =pd.concat([ sub_BL_ML['Subject'], sub_BL_ML['Noise'],sub_STIM_ML['RMS or
95%Ellipse'],sub_BL_ML['RMS or 95%Ellipse'],Diff_STIM_BL_ML_rms],axis=1)
BL_STIM_NOISE_rms_ML.columns=['Subject','Noise','STIM_rms','BL_rms','Diff_ST_BL_rms']
BL_STIM_NOISE_rms_ML['Diff_ST_BL_rms_AB']=BL_STIM_NOISE_rms_ML['Diff_ST_BL_rms'].abs()



#Bar plot for Change in Sway measures in ML
plt.figure(figsize=(15,8))
plt.subplot(1,3,1)
sns.barplot(x='Noise',y='Diff_ST_BL_ML',data =BL_STIM_NOISE_ML)
plt.title('Change in Sample Sample Entropy across Vibration Treament')
plt.ylabel('Change in Sample Entropy (ML)')

plt.subplot(1,3,2)
sns.barplot(x='Noise',y='Diff_ST_BL_DFA_ML',data =BL_STIM_NOISE_DFA_ML)
plt.title('Change in DFA across Vibration Treament')
plt.ylabel('Change in DFA (ML)')

plt.subplot(1,3,3)
sns.barplot(x='Noise',y='Diff_ST_BL_rms',data =BL_STIM_NOISE_rms_ML)
plt.title('Change in RMS across Vibration Treament')
plt.ylabel('Change in RMS (ML)')
plt.tight_layout(pad=3.0)


#Bar plot for Absolute Change in Sway measures in ML
plt.figure(figsize=(15,8))
plt.subplot(1,3,1)
sns.barplot(x='Noise',y='Diff_ST_BL_AB',data =BL_STIM_NOISE_ML)
plt.title('Absolute Change in Sample Sample Entropy across Vibration Treament')
plt.ylabel('Absolute Change in Sample Entropy (ML)')

plt.subplot(1,3,2)
sns.barplot(x='Noise',y='Diff_ST_BL_DFA_ML_AB',data =BL_STIM_NOISE_DFA_ML)
plt.title('Absolute Change in DFA across Vibration Treament')
plt.ylabel('Absolute Change in DFA (ML)')

plt.subplot(1,3,3)
sns.barplot(x='Noise',y='Diff_ST_BL_rms_AB',data =BL_STIM_NOISE_rms_ML)
plt.title('Absolute Change in RMS across Vibration Treament')
plt.ylabel('Absolute Change in RMS (ML)')
plt.tight_layout(pad=3.0)
```

```
#Linear regression plot for basiline and the difference between baseline and stimulation measures in ML
#Sample Entropy
sns.lmplot(x='BL_SE_ML',y='Diff_ST_BL_ML',data=BL_STIM_NOISE_ML,hue='Noise')
plt.title(' Change vs Initial' )
plt.xlabel('Initial Sample Entropy (ML)')
plt.ylabel('Change in Sample Entropy (ML)')

#DFA
sns.lmplot(x='BL_DFA_ML',y='Diff_ST_BL_DFA_ML',data=BL_STIM_NOISE_DFA_ML,hue='Noise')
plt.title(' Change Vs Initial (ML)' )
plt.xlabel('Initial DFA (ML)')
plt.ylabel('Change in DFA (ML) ')

#RMS
sns.lmplot(x='BL_rms',y='Diff_ST_BL_rms',data=BL_STIM_NOISE_rms_ML,hue='Noise')
plt.title(' Change vs Initial (ML)' )
plt.xlabel('Initial RMS (ML)')
plt.ylabel('Change in RMS (ML)')


#ANOVA Sample Entropy across subject in AP
import pingouin as pg
se1 = pg.anova(data=sub_BL_AP, dv='SampEn(m=2,R=0.1)', between='Subject', detailed=True)
print(se11)

#ANOVA DFA across subject in AP
import pingouin as pg
dfa1 = pg.anova(data=sub_BL_AP, dv='DFA_alpha(0.5s-15s)', between='Subject', detailed=True)
print(dfa1)

#ANOVA RMS across subject in AP
import pingouin as pg
rms1 = pg.anova(data=sub_BL_AP, dv='RMS or 95%Ellipse', between='Subject', detailed=True)
print(rms1)

#ANOVA Sample Entropy across subject in ML
import pingouin as pg
se2 = pg.anova(data=sub_BL_ML, dv='SampEn(m=2,R=0.1)', between='Subject', detailed=True)
print(se2)

#ANOVA DFA across subject in ML
import pingouin as pg
dfa2= pg.anova(data=sub_BL_ML, dv='DFA_alpha(0.5s-15s)', between='Subject', detailed=True)
print(dfa2)

#ANOVA RMS across subject in ML
import pingouin as pg
rms2 = pg.anova(data=sub_BL_ML, dv='RMS or 95%Ellipse', between='Subject', detailed=True)
print(rms2)



#Repeated measures ANOVA of change in Sample Entropy across vibration types in ML
from statsmodels.stats.anova import AnovaRM
print(AnovaRM(data=BL_STIM_NOISE_ML, depvar='Diff_ST_BL_ML', subject='Subject', within=['Noise']).fit())

#Repeated measures ANOVA of change in Sample Entropy across vibration types in ML
from statsmodels.stats.anova import AnovaRM
print(AnovaRM(data=BL_STIM_NOISE_DFA_ML, depvar='Diff_ST_BL_DFA_ML', subject='Subject', within=['Noise']).fit())
```

```
#Repeated measures ANOVA of change in RMS across vibratuion types in ML
from statsmodels.stats.anova import AnovaRM
print(AnovaRM(data=BL_STIM_NOISE_rms_ML, depvar='Diff_ST_BL_rms', subject='Subject', within=['Noise']).fit())


#Repeated measures ANOVA of Baseline Sample Entropy across sessions types in ML
from statsmodels.stats.anova import AnovaRM
print(AnovaRM(data=sub_BL_ML, depvar='SampEn(m=2,R=0.1)', subject='Subject', within=['session_order_no']).fit())


#Repeated measures ANOVA of Baseline DFA across sessions types in ML
from statsmodels.stats.anova import AnovaRM
print(AnovaRM(data=sub_BL_ML, depvar='DFA_alpha(0.5s-15s)', subject='Subject', within=['session_order_no']).fit())


#Repeated measures ANOVA of Baseline RMS across sessions types in ML
from statsmodels.stats.anova import AnovaRM
print(AnovaRM(data=sub_BL_ML, depvar='RMS or 95%Ellipse', subject='Subject', within=['session_order_no']).fit())
```