**Investigating the Effect of Stochastic Resonance on Balance in Persons with Multiple Sclerosis**

by
Victoria Blackwood

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. Jacob Sosnoff

Date Defended: May 3rd, 2022

The Thesis Committee for Victoria Blackwood certifies that this is the

approved version of the following thesis:

**Investigating the Effect of Stochastic Resonance on Balance in Persons with Multiple Sclerosis**

_____

Chairperson: Dr. Carl Luchies

Date Approved: May 4th, 2022

**Abstract**

Multiple sclerosis (MS) is a chronic neurological disease that affects the central nervous system. It is associated with balance and mobility impairments which increase the risk of falls. Some interventions to improve balance involve endurance and strength training but may prove to be difficult for those with mobility impairments. Stochastic resonance has been explored as a method to improve balance in populations with somatosensory deficits. The effects of stochastic resonance have not been widely studied in MS. The purpose of the study was to investigate the effects of different types of vibration on sway parameters in persons with MS.

The aims of the current study are: 1) to understand how the sensing threshold and linear/non-linear sway parameters of people with MS compared to the healthy old and healthy young on one inch of foam and 2) to investigate the effect of the sub-sensory vibration on linear/non-linear sway parameters in people with MS. The MS participant was in the same age range as the healthy older participants which allowed for a healthy versus disease comparison. The MS participant was compared to the healthy young on foam to understand if using foam as a sensory deficit was comparable to a disease population. It was hypothesized that: 1) MS would exhibit greater center-of-pressure (COP) sway with eyes closed compared to healthy old individuals, 2) sub-sensory vibratory stimulus would improve balance in MS, and 3) healthy young individuals on one inch of foam will display similar sway measures and threshold values to MS.

One older adult with MS participated in the study. Data from four healthy older adults and four healthy young adults on one inch of foam were included in the study from a previous pilot study using the same procedure. Each participant stood on a force plate with their eyes closed for the sway trials. Three different types of vibration (placebo, white, and pink) were

applied to the feet while standing at 90% of their sensing threshold. Linear (RMS and 95% ellipse) and non-linear (sample entropy) measures were extracted and analyzed using MATLAB.

No significant difference was seen in the sensing threshold between the multiple sclerosis participant and healthy older adults. The multiple sclerosis participant demonstrated larger sway variability and unpredictability compared to the healthy older adults and healthy young adults on foam. None of the different types of vibration improved balance of the MS participant. The use of foam for healthy young adult participants is not a representative simulation of severe stages of MS.

When looking at the effects of the different colors of noise, there does not appear to be an effect on sway parameters. Using this data, a preliminary conclusion can be made that sub-threshold vibration might not be a suitable intervention for those with severe stages of MS. However, these assumptions will need to be confirmed using a larger sample size.

**Acknowledgements**

I would like to take this opportunity to thank all the people who have supported me through this two-year long journey. Specifically, I would like to thank Dr. Luchies for being a great mentor and providing me with the support and guidance needed to grow as a student and researcher. My lab members Jess Kirchner, Eryn Gerber, Scott Ring, Camillo Giraldo, Di Bin, Zaccur Nkrumah, and Alex Wilson for being great friends and helping me learn within the lab. My incredible friends Kara Hageman, Savannah Mosier, Lexi Simar, Morgan Riley, and Aya Cloyd for being the best friends I could ask for in the Bioengineering program. Finally, my mom, dad, and three brothers for always being my strongest support system through any of my endeavors and always believing in me and pushing me to be the best I can be.

I would not be where I am today without the help and support of you all, so I am extremely grateful you all have been a part of my graduate school experience.

# Table of Contents

## Chapter 1: Introduction

### Background and Motivation

Multiple Sclerosis (MS) is a chronic neurological disease that affects over 2 million people worldwide (Wajda et al., 2015; Allum et al., 2021; Kalron et al., 2014). It is a progressive disease that affects the central nervous system and leads to demyelination of the myelin sheaths which creates lesions that leave the nerves exposed (Loyd et al., 2019; Wajda et al., 2015; Allum et al., 2021; Correale et al., 2017). The lesions can affect multiple systems in the body responsible for maintaining postural stability, including the somatosensory, cognitive, visual, and vestibular systems. Disturbances in one or more of these systems ultimately lead to balance and mobility impairments. Balance and mobility are the two most reported issues with a prevalence of about 75% in those diagnosed with MS (Wajda et al., 2015; Kalron et al., 2014).

Falls are a prominent issue associated with MS due to the balance and mobility impairments. About 50% of people with MS have experienced at least one fall in their lifetime (Sosnoff et al., 2011; Kasser et al., 2011; Wajda et al., 2015). Different interventions to improve balance, such as aerobic exercises and strength training, are not well suitable for those with mobility impairments so other options have been explored (Inojosa et al., 2020; Freitas et al., 2018).

One potential intervention method is stochastic resonance. Stochastic resonance helps a system to detect and process a weak signal by introducing noise to a non-linear system to boost the signal to surpass a sensory threshold, leading to stimulus detection (White et al., 2019; Costa et al., 2007). Within the literature, this intervention method has been focused on sensory deficit populations because it is a matter of understanding the effect it has on sensation to therefore impact postural stability (Costa et al., 2007). Stochastic resonance has been shown to have

promising effects on the improvement of the vestibular system, motor functions, and postural stability in healthy populations and some pathological groups; however, its effect on MS is still uncertain (White et al., 2019).

## Specific Aims

The first specific aim of the study was to understand how the sensing threshold and linear/non-linear sway parameters of people with MS compared to the healthy old and healthy young on one inch of foam. The second aim of the study was to investigate the effect of the sub-sensory vibration on linear/non-linear sway parameters in people with MS. The MS participant was in the same age range as the healthy older participants which allowed for a healthy versus disease comparison. The MS participant was compared to the healthy young adults on foam to understand if using foam as a sensory deficit was comparable to a disease population.

It was hypothesized that: 1) MS would exhibit greater COP sway with eyes closed compared to healthy old individuals, 2) sub-sensory vibratory stimulus would improve balance in MS, and 3) healthy young individuals on one inch of foam will display similar sway measures and threshold values to MS.

## Thesis Content

The thesis contains four chapters. Chapter 1 is a concise introduction explaining multiple sclerosis and the major issues associated with the disease, as well a short description of stochastic resonance. Chapter 2 is an in-depth background about the disease itself, the effects on postural control, prevalence of falls, and information on the use of stochastic resonance. Chapter 3 includes a manuscript of the pilot study which investigated the effect of stochastic resonance on sway parameters in MS. The chapter includes an introduction, methods, results, and discussion. Chapter 4 summarizes the study and discusses limitations and future studies.

## References


Wajda, D. A., Moon, Y., Motl, R. W., & Sosnoff, J. J. (2015). Preliminary investigation of gait initiation and falls in multiple sclerosis. *Archives of Physical Medicine and Rehabilitation*, *96*(6), 1098–1102. https://doi.org/10.1016/j.apmr.2014.12.011

Allum, J. H. J., Rust, H. M., Lutz, N., Schouenborg, C., Fischer-Barnicol, B., Haller, V., … Yaldizli. (2021). Characteristics of improvements in balance control using vibro-tactile biofeedback of trunk sway for multiple sclerosis patients. *Journal of the Neurological Sciences*, *425*, 117432. https://doi.org/10.1016/j.jns.2021.117432

Kalron, A., & Achiron, A. (2014). The relationship between fear of falling to spatiotemporal gait parameters measured by an instrumented treadmill in people with multiple sclerosis. *Gait and Posture*, *39*(2), 739–744. https://doi.org/10.1016/j.gaitpost.2013.10.012

Loyd, B. J., Fangman, A., Peterson, D. S., Gappmaier, E., Schubert, M. C., Thackery, A., & Dibble, L. (2019). Rehabilitation to improve gaze and postural stability in people with multiple sclerosis: Study protocol for a prospective randomized clinical trial. *BMC Neurology*, *19*(1), 1–8. https://doi.org/10.1186/s12883-019-1353-z

Correale, J., Gaitán, M. I., Ysrraelit, M. C., & Fiol, M. P. (2017). Progressive multiple sclerosis: from pathogenic mechanisms to treatment. *Brain*, *140*(3), 527–546. https://doi.org/10.1093/brain/aww258

Sosnoff, J. J., Socie, M. J., Boes, M. K., Sandroff, B. M., Pula, J. H., Suh, Y., … Motl, R. W. (2011). Mobility, balance and falls in persons with multiple sclerosis. *PLoS ONE*, *6*(11), 2–6. https://doi.org/10.1371/journal.pone.0028021

Kasser, S. L., Jacobs, J. V., Foley, J. T., Cardinal, B. J., & Maddalozzo, G. F. (2011). A prospective evaluation of balance, gait, and strength to predict falling in women with multiple sclerosis. *Archives of Physical Medicine and Rehabilitation*, *92*(11), 1840–1846. https://doi.org/10.1016/j.apmr.2011.06.004

Inojosa, H., Schriefer, D., Trentzsch, K., Klöditz, A., & Ziemssen, T. (2020). Visual Feedback and Postural Control in Multiple Sclerosis. *Journal of Clinical Medicine*, *9*(5), 1291. https://doi.org/10.3390/jcm9051291

Freitas, E. D. S., Frederiksen, C., Miller, R. M., Heishman, A., Anderson, M., Pardo, G., … Bemben, M. G. (2018). Acute and Chronic Effects of Whole-Body Vibration on Balance, Postural Stability, and Mobility in Women With Multiple Sclerosis. *Dose-Response*, *16*(4), 1–13. https://doi.org/10.1177/1559325818816577

White, O., Babic, J., Trenado, C., Johannsen, L., & Goswami, N. (2019). The promise of stochastic resonance in falls prevention. *Frontiers in Physiology*, *10*(JAN), 1–12. https://doi.org/10.3389/fphys.2018.01865

Costa, M., Priplata, A. A., Lipsitz, L. A., Wu, Z., Huang, N. E., Goldberger, A. L., & Peng, C. K. (2007). Noise and poise: Enhancement of postural complexity in the elderly with a stochastic-resonance-based therapy. *Epl*, *77*(6). https://doi.org/10.1209/0295-5075/77/68008

## Chapter 2: Background

### Multiple Sclerosis

Multiple sclerosis (MS) is a chronic inflammatory disease that affects the central nervous system (Loyd et al., 2019; Wajda et al., 2015; Correale et al., 2017). MS leads to demyelination and neurodegeneration as it attacks the nervous system and the myelin sheaths (Allum et al., 2021; Correale et al., 2017). This neurological disease affects approximately 2.2-2.5 million people worldwide and can develop at any age (Wajda et al., 2015; Allum et al., 2021; Kalron et al., 2014). MS can also affect both men and women, but women are twice as likely to develop MS than men (Kasser et al., 2011). Lesions caused by the disease can attack multiple systems in the body such as the somatosensory, cognitive, and visual systems. MS is progressive and variable in nature and affects everyone differently (Loyd et al., 2019; Correale et al., 2017).

MS can progress into one or more stages during an affected person's lifetime. The first stage is typically relapse-remitting MS, then develops into secondary progressive MS, and then into a more severe stage of primary progressive MS (Correale et al., 2017). Disease progression occurs when there is an exceeding amount of axonal loss that the compensatory capacity of the central nervous system cannot withstand (Correale et al., 2017). Damage caused by the progression is irreversible (Correale et al., 2017). Symptoms and timing of the stages of the disease can vary depending on when an individual is diagnosed and when treatment begins to delay progression.

MS presents many symptoms, with severity of these symptoms increasing with disease stage. Some common symptoms include muscle fatigue, vision loss, balance and mobility impairments, dizziness, and cognitive impairment (Loyd et al., 2019). Balance and gait are two of the main issues reported in those with MS (Wajda et al., 2015), with approximately 75%

documenting disturbances in their walking and balance (Kalron et al., 2014). Therefore, this thesis will focus on MS-related balance impairments.

## Postural Control and Mobility in MS

Postural control is an important aspect of daily living. Postural control allows an individual to maintain or restore a state of balance to stand or walk without falling (Inojosa et al., 2020; Costa et al., 2007). The dynamics of postural control are more complex than it appears (Zhou et al., 2016). Quiet standing is achieved through interactions of the somatosensory, vestibular, and visual systems (Costa et al., 2007). Simple walking is a complex process that is dependent on sensorimotor integration (White et al., 2019). The postural control system utilizes a variety of sensory inputs to control body sway, such as plantar skin receptors (White et al., 2019; Zhou et al., 2016). The mechanoreceptors of the soles of the feet serve as a source of information that feed into the control system that can provide information about pressure distribution and shear forces that correlate with walking and quiet standing (White et al., 2019; Zhou et al., 2016). Postural control relies on a healthy sensory system to be successful (White et al., 2019). Postural control involves non-linear processes that are dependent upon several neurological functions including sensory inputs, stretch reflexes, and motor skills (Inojosa et al., 2020). Postural reflexes are activated when a disturbance or perturbation is introduced to the system to allow the person to properly recover and not fall (White et al., 2019). The central nervous system plays an important part by estimating where the body is in space and where in relation the limbs are to each other (White et al., 2019). If any interferences or delays occur in the system, falls could occur.

Falls can be caused by both intrinsic and extrinsic factors in both elderly and impaired individuals (White et al., 2019). Intrinsic factors can include decreased mobility, reduced

proprioception, and other factors (White et al., 2019). Extrinsic factors can include uneven or slippery surfaces, improper footwear, and other environmental obstacles (White et al., 2019). When one or more neurological functions involved in postural control fails, fall risk increases substantially (Inojosa et al., 2020; White et al., 2019).

A neurological disease, such as MS, is commonly associated with poor postural control (Inojosa et al., 2020). As mentioned previously, people diagnosed with MS experience loss of balance and impairment to their gait. The lesions that develop with MS affect various systems that are responsible with maintaining postural control. For instance, the somatosensory system can be affected because the feedback is delayed/interrupted leading to postural instability (Inojosa et al., 2020). As the somatosensory system diminishes, people with MS must rely more heavily on visual feedback since the feedback from the somatosensory system is impaired (Inojosa et al., 2020). A study conducted by Inojosa et al (2020) demonstrated that people with MS have greater center-of-pressure (COP) sway during an eyes closed condition compared to eyes open condition (Inojosa et al., 2020). Symptoms of balance impairment can occur early in the diagnosis of MS (Kalron et al., 2014). As balance and mobility diminish, falls become an increasingly prominent concern (Kalron et al., 2014).

A study conducted by Cattaneo et al. (2014) aimed to assess postural stability during three task-oriented movements in individuals with MS (Cattaneo et al., 2014). The dynamic tasks included chair rising, taking a step, and bending over, which may be difficult in later stages of MS progression (Cattaneo et al., 2014). They discovered that people with MS had trouble stabilizing themselves when completing transitionary movements (Cattaneo et al., 2014). People with MS took longer to reach stability compared to healthy subjects and had more static sway (Cattaneo et al., 2014).

Quantification of Postural Control

Instrumented assessment helps to provide information about balance impairments in various populations. Equipment such as force plates have been used in countless studies to assess gait and quiet stance in healthy and pathologies.

COP can be expressed as a time series to determine how the data fluctuates over time, whether that be analyzing velocity or force for instance. The data can also be expressed in a spatial manner so the COP can be mapped to visualize the displacement and velocity. Postural control mechanisms are assumed to control both the displacement of the COP and the velocity at which the position changes (Costa et al., 2007). The center-of-pressure is measured in both the medio-lateral (left/right) and antero-posterior (front/back) directions (Costa et al., 2007).

## Fall Prevalence

Falls become more prevalent with age due to loss of postural control (Inojosa et al., 2020; Costa et al., 2007). However, those with neurological diseases such as MS can experience falls at any age because of the damage to some of the key functions in maintain postural control (Inojosa et al., 2020).

Approximately 50% of people with MS experience at least one fall during their lifetime (Sosnoff et al., 2011; Kasser et al., 2011; Wajda et al., 2015). Factors that lead to falls could include poor balance, status of disability, and decreased proprioception (Sosnoff et al., 2011). As high as nine falls per year have been reported from people who experience recurrent falls (Kasser et al., 2011). Falls can lead to serious injuries, high medical costs, and lead to the need for assistive devices such as canes or walkers (Sosnoff et al., 2011).

After a fall, gait patterns can change and physical activity can decrease significantly, leading to a loss of autonomy and a decrease in quality of life (Sosnoff et al., 2011; Kasser et al., 2011; Kalron et al., 2014). Long-term decreases in activity level can also lead to decreased muscle force, development of cardiovascular disease, and worsened spasticity (Kalron et al., 2014). Not only do falls have a physical impact, but they can also have a psychological impact and cause people to develop a fear of falling, further worsening activity and quality of life (Kalron et al., 2014).

Falls are more likely to occur during a dynamic event when transitioning from different positions (Wajda et al., 2015; Cattaneo et al., 2014). To mitigate some of this risk, people with MS will often utilize smaller step sizes and slower movements to help prevent another fall from occurring (Kasser et al., 2011; Kalron et al., 2014). Perhaps one of the most negative biomechanical effects of falling in this population is the delay of gait initiation, a crucial component of healthy balance (Wajda et al., 2015).

A study by Sosnoff et al. (2011) explored a relation between fall history and metrics such as sway and mobility in those with MS. It was concluded that people in the falling group demonstrated increased sway velocity in the medio-lateral direction with eyes open and greater overall sway area compared to non-fallers (Sosnoff et al., 2011). In the eyes-closed condition, fallers have greater sway velocity in the medio-lateral and antero-posterior directions (Sosnoff et al., 2011). Those who were classified as fallers also tended to be older, use assistive devices, increased disability, decreased mobility metrics, and had poor overall balance (Sosnoff et al., 2011). Another study investigated gait initiation and falls in people with MS (Wajda et al., 2015). The results from this study and previous studies observed that people with MS had slower COP movements, a reduction in shift from quiet stance to one leg stance when initiating gait, and

had delayed step onset times (Wajda et al., 2015). Kalron and Achiron conducted a study that aimed to understand the relationship between fear of falling and its effect on various parameters of mobility in MS (Kalron et al., 2014). The study found that MS patients who have a fear of falling walk slower, have a shorter stride length, have a wider base of support, and prolong their double support phase (Kalron et al., 2014). They also demonstrate a larger variability of COP during the walking trial (Kalron et al., 2014). Understanding the underlying factors or the effects of falls, can help aid efforts in developing solutions to reduce the amount of falls experienced by different populations.

## Application of Stochastic Resonance

Interventions for improving postural control include endurance and strength training (Inojosa et al., 2020; Loyd et al., 2019), but the positive effects of such existing interventions are limited. In the last decade, the study of stochastic resonance has emerged to explore the effects of noise on biological systems. The goal of stochastic resonance is to help a system detect and process a weak signal (White et al., 2019). To achieve this, the introduction of noise in a non-linear system enhances the quality of the output signal (White et al., 2019; Costa et al., 2007). When an input signal is weak, it is difficult to achieve a neuron's sensory threshold; the addition of low-amplitude, subthreshold noise can provide this signal with an appropriate boost and aid in surpassing this threshold, allowing for sensation (White et al., 2019; Costa et al., 2007).

Stochastic resonance has become an increasingly popular technique applied in research studies to potentially increase postural stability (Costa et al., 2007). Stochastic resonance has been documented to help improve the vestibular system, motor functions, and postural stability (White et al., 2019).

The interventions to improve balance that include resistance and aerobic exercises may be effective for some but not all. For instance, those with MS experience reduced motor control, so it may be difficult to perform those exercises (Freitas et al., 2018). The limited efficacy of these methods necessitates the exploration of alternative interventions. Stochastic resonance has been used in various studies to investigate the effects demonstrated by being applied to skin receptors. Stochastic resonance can be applied electrically or mechanically. The focus of this thesis is the application of a mechanical stimulus (vibration), applied to the plantar receptors of the feet.

Stimulation can be provided at either subthreshold or suprathreshold levels. Subthreshold stimulation is delivered at levels that are below a person's sensory threshold and does not elicit an action potential response, whereas suprathreshold is delivered at levels above a person's threshold and can elicit a response. Suprathreshold stimulation is used more as a "corrective" approach in directing sway, while subthreshold helps to boost the signal. Subthreshold was the chosen level of stimulation based on results of previous research studies. Applying subthreshold noisy stimulation can help improve a system to detect the input signal (Zhou et al., 2016). Using this method can augment the existing signal without providing an excess of noise.

A review done by Bagherzadeh Cham et al. (2016) looked at the benefits of vibratory stimulation delivered at subthreshold levels to the soles of the feet in both older adults and those with diabetes (Bagherzadeh Cham et al., 2016). The results of the review found that vibratory stimulation can improve balance in affected populations. Another study that also investigated effects of subthreshold stimulation, found that participants who had diabetic neuropathy or suffered a stroke had an improvement in their sway stability (Priplata et al., 2006). In addition, a

study that looked at effects in older adults, saw an improvement in balance improvement when subjected to vibrotactile noisy stimulation (Dettmer et al., 2015).

## Conclusion

MS is a debilitating disease that unfortunately has no cure. The MS population experience a high prevalence of falls due to impairments in the systems responsible for maintaining postural control. The balance and mobility impairments caused by the disease can negatively impact peoples lives by decreasing physical activity and autonomy. An important aspect of improving quality of life are the interventions and therapies designed to improve postural stability. Stochastic resonance is becoming a focus within research studies that investigate balance and mobility in elderly and pathological populations. Many studies have analyzed the effects of stochastic resonance in populations with somatosensory deficiencies, but little is known on the effects specific to MS.

The aims of the current study are: 1) to understand how the sensing threshold and linear/non-linear sway parameters of people with MS compared to the healthy old and healthy young on one inch of foam and 2) to investigate the effect of the sub-sensory vibration on linear/non-linear sway parameters in people with MS. The MS participant was in the same age range as the healthy older participants which allowed for a healthy versus disease comparison. The MS participant was compared to the healthy young on foam to understand if using foam as a sensory deficit was comparable to a disease population. It was hypothesized that: 1) MS would exhibit greater COP sway with eyes closed compared to healthy old individuals, 2) sub-sensory vibratory stimulus would improve balance in MS, and 3) healthy young individuals on one inch of foam will display similar sway measures and threshold values to MS.

## References


Loyd, B. J., Fangman, A., Peterson, D. S., Gappmaier, E., Schubert, M. C., Thackery, A., & Dibble, L. (2019). Rehabilitation to improve gaze and postural stability in people with multiple sclerosis: Study protocol for a prospective randomized clinical trial. *BMC Neurology*, *19*(1), 1–8. https://doi.org/10.1186/s12883-019-1353-z

Wajda, D. A., Moon, Y., Motl, R. W., & Sosnoff, J. J. (2015). Preliminary investigation of gait initiation and falls in multiple sclerosis. *Archives of Physical Medicine and Rehabilitation*, *96*(6), 1098–1102. https://doi.org/10.1016/j.apmr.2014.12.011

Correale, J., Gaitán, M. I., Ysrraelit, M. C., & Fiol, M. P. (2017). Progressive multiple sclerosis: from pathogenic mechanisms to treatment. *Brain*, *140*(3), 527–546. https://doi.org/10.1093/brain/aww258

Allum, J. H. J., Rust, H. M., Lutz, N., Schouenborg, C., Fischer-Barnicol, B., Haller, V., … Yaldizli. (2021). Characteristics of improvements in balance control using vibro-tactile biofeedback of trunk sway for multiple sclerosis patients. *Journal of the Neurological Sciences*, *425*, 117432. https://doi.org/10.1016/j.jns.2021.117432

Kalron, A., & Achiron, A. (2014). The relationship between fear of falling to spatiotemporal gait parameters measured by an instrumented treadmill in people with multiple sclerosis. *Gait and Posture*, *39*(2), 739–744. https://doi.org/10.1016/j.gaitpost.2013.10.012

Kasser, S. L., Jacobs, J. V., Foley, J. T., Cardinal, B. J., & Maddalozzo, G. F. (2011). A prospective evaluation of balance, gait, and strength to predict falling in women with multiple sclerosis. *Archives of Physical Medicine and Rehabilitation*, *92*(11), 1840–1846. https://doi.org/10.1016/j.apmr.2011.06.004

Inojosa, H., Schriefer, D., Trentzsch, K., Klöditz, A., & Ziemssen, T. (2020). Visual Feedback and Postural Control in Multiple Sclerosis. *Journal of Clinical Medicine*, *9*(5), 1291. https://doi.org/10.3390/jcm9051291

Costa, M., Priplata, A. A., Lipsitz, L. A., Wu, Z., Huang, N. E., Goldberger, A. L., & Peng, C. K. (2007). Noise and poise: Enhancement of postural complexity in the elderly with a stochastic-resonance-based therapy. *Epl*, *77*(6). https://doi.org/10.1209/0295-5075/77/68008

Zhou, J., Lipsitz, L., Habtemariam, D., & Manor, B. (2016). Sub-sensory vibratory noise augments the physiologic complexity of postural control in older adults. *Journal of NeuroEngineering and Rehabilitation*, *13*(1), 1–8. https://doi.org/10.1186/s12984-016-0152-7

White, O., Babic, J., Trenado, C., Johannsen, L., & Goswami, N. (2019). The promise of stochastic resonance in falls prevention. *Frontiers in Physiology*, *10*(JAN), 1–12. https://doi.org/10.3389/fphys.2018.01865

Cattaneo, D., Rabuffetti, M., Bovi, G., Mevio, E., Jonsdottir, J., & Ferrarin, M. (2014). Assessment of postural stabilization in three task oriented movements in people with multiple sclerosis. *Disability and Rehabilitation*, *36*(26), 2237–2243. https://doi.org/10.3109/09638288.2014.904933

Sosnoff, J. J., Socie, M. J., Boes, M. K., Sandroff, B. M., Pula, J. H., Suh, Y., … Motl, R. W. (2011). Mobility, balance and falls in persons with multiple sclerosis. *PLoS ONE*, *6*(11), 2–6. https://doi.org/10.1371/journal.pone.0028021

Freitas, E. D. S., Frederiksen, C., Miller, R. M., Heishman, A., Anderson, M., Pardo, G., … Bemben, M. G. (2018). Acute and Chronic Effects of Whole-Body Vibration on Balance, Postural Stability, and Mobility in Women With Multiple Sclerosis. *Dose-Response*, *16*(4), 1–13. https://doi.org/10.1177/1559325818816577

Bagherzadeh Cham, M., Mohseni-Bandpei, M. A., Bahramizadeh, M., Kalbasi, S., & Biglarian, A. (2016). The clinical and biomechanical effects of subthreshold random noise on the plantar surface of the foot in diabetic patients and elder people: A systematic review. *Prosthetics and Orthotics International*, *40*(6), 658–667. https://doi.org/10.1177/0309364616631351

Priplata, A. A., Patritti, B. L., Niemi, J. B., Hughes, R., Gravelle, D. C., Lipsitz, L. A., … Collins, J. J. (2006). Noise-enhanced balance control in patients with diabetes and patients with stroke. *Annals of Neurology*, *59*(1), 4–12. https://doi.org/10.1002/ana.20670

Dettmer, M., Pourmoghaddam, A., Lee, B. C., & Layne, C. S. (2015). Effects of aging and tactile stochastic resonance on postural performance and postural control in a sensory conflict task. *Somatosensory and Motor Research*, *32*(2), 128–135. https://doi.org/10.3109/08990220.2015.1004045

# Chapter 3: Investigating the Effect of Stochastic Resonance on Balance in Persons with Multiple Sclerosis

**Abstract**

**Background**: Multiple sclerosis is a chronic neurological disease that affects the central nervous system. It is associated with balance and mobility impairments which increase the risk of falls. Some interventions to improve balance involve endurance and strength training but may prove to be difficult for those with mobility impairments. Stochastic resonance has been explored as a method to improve balance in populations with somatosensory deficits. The effects of stochastic resonance through vibration have not been widely studied in multiple sclerosis.

**Methods**: One older adult with multiple sclerosis (MSOA) participated in the study. Data collected in a previous pilot study from four healthy older adults (HOA) and four healthy young adults (HYA), the latter of which stood on one inch of foam were included in the study to enable a more thorough investigation into the studied effects. Each participant stood on a force plate with their eyes closed for the sway trials. Three different types of vibration (placebo, white, and pink) were applied to the feet while standing at 90% of their sensing threshold. Linear (RMS and 95% ellipse) and non-linear (sample entropy) measures were extracted and analyzed using MATLAB.

**Results**: No significant difference was seen in the sensing threshold between the MSOA and the HOA. The MSOA demonstrated larger sway variability and unpredictability compared to the HOA and HYA on foam. None of the vibration types improved balance during quiet standing of the MSOA. The use of foam for HYA is not an accurate representation of severe stages of the sensory deficit MS population tested.

**Interpretations**: The pilot study looked at various vibrations and the effect of multiple sclerosis on postural sway. While there were no improvements in sway parameters after the vibration stimulus, a larger sample size in the MSOA will be needed to confirm the study results.

Word Count: 282 Key Words: Multiple sclerosis, stochastic resonance, falls, postural sway

## Introduction

Multiple Sclerosis (MS) is a chronic neurological disease that affects over 2 million people worldwide (Wajda et al., 2015; Allum et al., 2021; Kalron et al., 2014). It is a disease that affects the central nervous system and leads to demyelination of the myelin sheaths which creates lesions that leave the nerves exposed (Loyd et al., 2019; Wajda et al., 2015; Allum et al., 2021; Correale et al., 2017). The disease can progress into one or more types depending on timing of diagnosis and treatment. The types include relapse-remitting MS, secondary progressive MS, and primary progressive MS (Correale et al., 2017).

Symptoms of MS can include muscle fatigue, vision loss, and cognitive, balance, and mobility impairments (Loyd et al., 2019). As the disease progresses, the severity of the symptoms also progresses, leading to increased disruption to activities of daily living. Balance and mobility are the two most reported issues with a prevalence of about 75% in those diagnosed with MS (Wajda et al., 2015; Kalron et al., 2014).

The lesions can affect multiple systems in the body responsible for maintaining postural stability, including the somatosensory, cognitive, visual, and vestibular system. Disturbances in one or more systems ultimately lead to balance and mobility issues. When the lesions affect the somatosensory system, the feedback is delayed and ultimately interrupted, which leads to postural instability (Inojosa et al., 2020). Specific to the somatosensory system, mechanoreceptors on the bottom of the feet are considered a source of information for the control system directly involved with motor control tasks such as with walking and standing (White et al., 2019; Zhou et al., 2016). If there are interruptions within any sensory system, postural control will be negatively affected, leading to increased risk of falls.

Falls are a prominent issue in both pathological populations, like those with MS, and the elderly. As people age, the sensory systems gradually degrade, which is associated with a loss of postural control (Inojosa et al., 2020; Costa et al., 2007). People diagnosed with various neurological diseases, such as MS, are at risk of falls due to damage caused by lesions to some of the key functions responsible for maintaining postural control (Inojosa et al., 2020). About 50% of those with MS have experienced at least one fall in their lifetime, while those who experience recurrent falls have had as high as nine falls per year (Sosnoff et al., 2011; Kasser et al., 2011; Wajda et al., 2015). Falls are associated with both physical and psychological impacts. The physical aspects include injuries, decreased physical activity, and altered gait patterns, while the psychological aspect includes a fear of falling (Sosnoff et al., 2011; Kasser et al., 2011; Kalron et al., 2014), the sum of which reduces quality of life.

Since falls pose a significant risk to a variety of populations, it is important to understand the underlying causes and risk factors. Over the years, an abundance of literature has investigated the underlying factors to what could lead to falls in various populations. These factors include the center-of-pressure displacement and velocity during quiet stance and walking, gait initiation, stride length, and the time it takes to stabilize after completing certain tasks. MS is associated with slower gait initiation, shorter stride length, larger COP variability, and poor balance (Wajda et al., 2015; Kalron et al., 2014; Sosnoff et al., 2011).

Given the implications of poor postural control, several interventions have been determined to improve balance. These include endurance and strength training to strengthen the muscles around the ankles and feet (Inojosa et al., 2020). These interventions are not viable options for certain groups of people who have mobility limitations, so other options have been explored to also improve balance (Freitas et al., 2018). One potential intervention is stochastic

resonance; stochastic resonance helps a system to detect and process a weak signal by introducing noise to a non-linear system to enhance the quality of an output signal (White et al., 2019; Costa et al., 2007). The noise helps to boost the signal to surpass a sensory threshold, leading to stimulus detection (White et al., 2019; Costa et al., 2007). This intervention is particularly appealing to sensory deficit populations in research studies to understand how it can affect sensation and therefore postural stability (Costa et al., 2007). Stochastic resonance has been shown to have promising effects on the improvement of the vestibular system, motor functions, and postural stability in healthy populations and some pathological groups (White et al., 2019). However, its effect on MS is still uncertain.

The purpose of this pilot study was to investigate the effect of stochastic resonance on balance in persons with MS. The study aimed to understand how different colors of noise (white and pink) delivered at sub-threshold levels affected balance in a sensory deficit population. Given the limitations during the time of the COVID-19 pandemic and subsequent obstacles in the recruiting process, the data collected during this study was compared to previously collected data from healthy young (18-28 years) participants with simulated-somatosensory deficit and healthy old (60-65 years) participants (Giraldo, 2021).

It was hypothesized that: 1) MS would exhibit greater COP sway with eyes closed compared to healthy old individuals, 2) Sub-sensory vibratory stimulus would improve balance in MS, 3) Healthy young individuals on one inch of foam will display similar sway measures and threshold values to MS.

## Methods

### Participants

One older adult with multiple sclerosis (MSOA: male, age: 63 years) volunteered to participate in the study. Previously collected data from four healthy older adults (HOA: 1 male, age: 63 years; 3 females, age: 62 ± 2 years) and four healthy young adults (HYA: 2 males, age: 25 ± 4.24 years; 2 females, age: 26 ± 2.83 years) participants were included in the study. Each participant was informed of the risks and benefits of the study and gave written consent, as approved by the University of Kansas Institutional Review Board. Inclusion criteria for the study included an official diagnosis of MS from a licensed neurologist, ability to walk with or without assistance from a cane or walker, and the ability to stand for at least two minutes. The inclusion criteria for the HOA and HYA included being between 60 and 65 years of age (only for HOA) and the ability to stand for at least two minutes unassisted. The Exclusion criteria included any surgeries or injuries obtained in the past year and/or being diagnosed with another neurological or musculoskeletal disorder.

### Protocol

The participant was asked to stand on two feet, arms by their side, head straight forward, and with eyes closed. Dots were placed on the ground to instruct the participant where to place their feet. Heels were kept 17 cm apart and their toes pointed forward (McIlroy et al., 1997). The participant stood on two force plates (Advanced Mechanical Technology Inc., Watertown, MA, USA) for the first trial. The participant then stood on a vibrating mat that was placed on top of a singular force plate (Advanced Mechanical Technology Inc., Watertown, MA, USA) for the rest of the trials. The MSOA participant's assistive walking device was readily available in front of them if they felt the need to hold on it. The vibratory mat was developed in the lab and validated

by a previous pilot study using healthy old and healthy young adults (Giraldo, 2021; Whorley 2020). The details of the trials are listed below.

1. Trial 1: Baseline trial on the two force plates [90 seconds].
2. Trial 2: Baseline trial on the mat without any vibration [90 seconds].
3. Trial 3: Sensing threshold was determined by using a modified 421 protocol [2 minutes] (Dyck et al., 1993; Whorley, 2020).
4. Trial 4: Once the sensing threshold was determined, the vibratory stimulus was applied to the participant at 90% of their sensing threshold [90 seconds].
5. Trial 5: Trial that took place post vibratory stimulus [90 seconds].

There was a total of three sessions to accommodate for three different types of vibratory stimuli, including three different colors of vibration: white, pink, and a placebo. White vibration was used because it is a commonly used signal in the study of stochastic facilitation (Kelty-Stephen et al., 2013). Pink vibration was used because of its abundance in natural, biological systems and potential therapeutic benefits (Kelty-Stephen et al., 2013).

The force plates data was collected using Spike2 software (Cambridge Electronic Design, UK) and text files were extracted. Foot-Floor kinetic data was recorded at 100 Hz. The HYA and HOA underwent identical testing conditions, except the HYA stood on one inch foam placed on the mat to simulate age related sensory deficits.

Data Analysis

The data analysis was completed using MATLAB (MathWorks, MA, USA). A low pass Butterworth filter of 20 Hz was used for the sway trials to remove signals generated by motor vibration forces. The filtered data was down-sampled to 50 Hz to extract the desired measures,

including Root Mean Square (RMS), 95% ellipse, and sample entropy. Additionally, the sensing threshold values, established during the 421 protocol, were extracted for each test session. The COP was calculated in the anterior-posterior, medio-lateral, and spatial dimensions.

RMS was extracted for the AP and ML directions of COP to understand the amount of sway variability. The 95% ellipse was used for the spatial COP to assess the overall sway. Sample entropy was used for all directions of COP to understand the sway predictability.

Statistical Analysis

All statistical analyses were completed using Microsoft Excel. A two-sample t-test was done for the sensory threshold percentages. Within the RMS, 95% ellipse, and sample entropy measures, t-tests compared the MSOA participant to the HOA and HYA for the baseline measures. Significance was set to alpha=0.05 for all tests. The sessions for the MSOA were assumed to be independent, while actually being repeated measures, for the purpose of being able to perform statistical analyses for comparisons that were not dependent on vibration type. Given the small sample size for the pilot study, further statistical analyses could not be completed for the post-vibratory trials. However, in future work with larger sample sizes, two different statistical tests could be conducted: (1) t-test to analyze the difference between pre- and post-treatment and (2) repeated measures ANOVA to understand the effect of noise color and patient population. A power analysis was included to determine the sample size needed to maintain a certain power, which can be found in the Appendix A.

## Results

### Sensing Threshold

The sensing thresholds for all three groups are shown in Figure 1. For each group, the sensing threshold values were averaged across all three sessions with the goal to analyze the overall threshold percentages. The averages and standard deviations are included on the bar plot (Figure 1). When comparing the sensing threshold of the MSOA to HYA on one inch of foam, there was a significant difference presented ($p<0.05$). The MSOA had a threshold of 57.57±1.15% and the HYA on foam had a threshold of 38.25±6.05%. Moreover, when comparing sensing thresholds of the HOA, 44.42±17.77%, and HYA on foam, 38.25±6.05%, there was no significant difference ($p>0.05$).

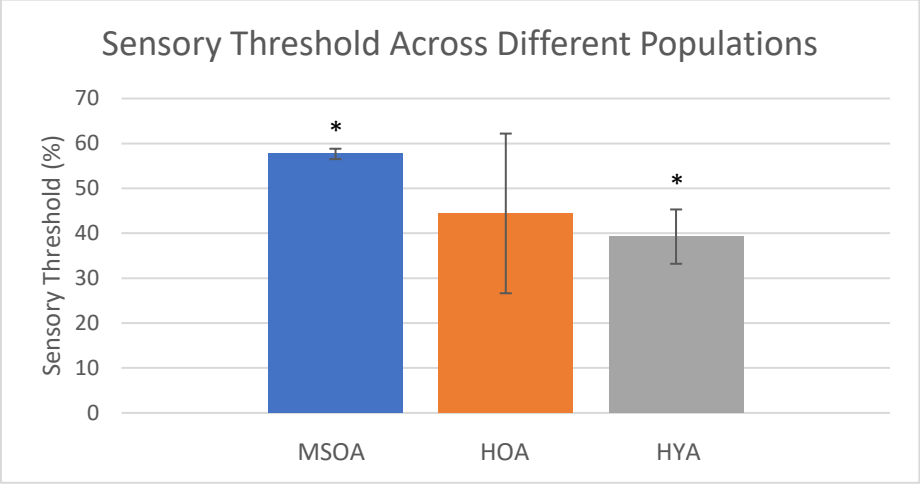


*Figure 1: Averages and standard deviations of sensing thresholds for each group. (MSOA=Older adult with Multiple Sclerosis, HOA=Healthy Older Adults, HYA=Healthy Young Adults on Foam; * = statistical difference between groups)*

Baseline Sway

The baseline sway measures on the mat (without any vibration) were reported for the AP, ML, and spatial directions for all groups. The data for each group was averaged for all sessions to get an understanding of the overall baseline sway across groups with an eyes closed condition. Figures 2 and 3 show the averages and standard deviations of the RMS for the AP and ML directions, respectively. When comparing RMS between the MSOA and HOA in the AP direction, a significant difference was found ($p<0.05$). There was approximately a 200% increase in variability in the AP direction demonstrated by the MSOA compared HOA and HYA. However, in the ML direction, there was no significant difference between HOA and HYA ($p>0.05$). There was a significant difference ($p<0.05$) in both the AP and ML directions between the MSOA and HYA on foam.

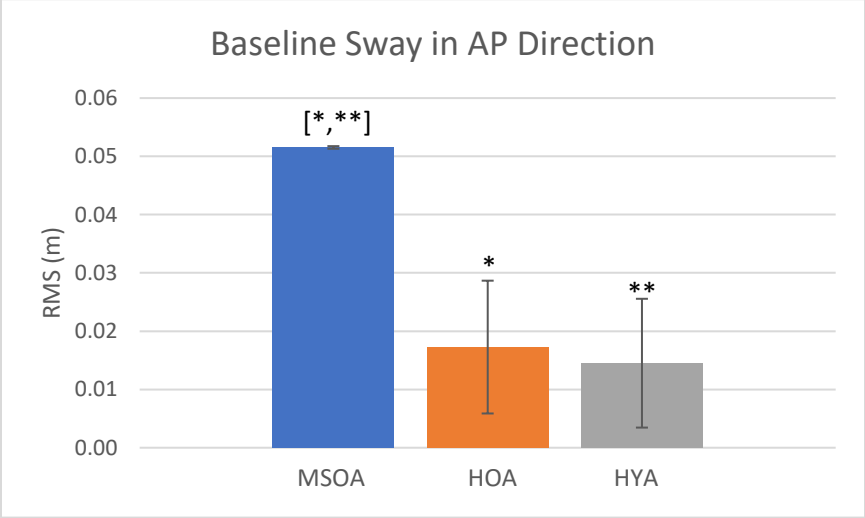


*Figure 2: RMS measure for baseline sway trial in the anterior-posterior direction. (MSOA=Older Adult with Multiple Sclerosis, HOA=Healthy Older Adults, HYA=Healthy Young Adults on Foam; * or **= statistical difference between groups)*

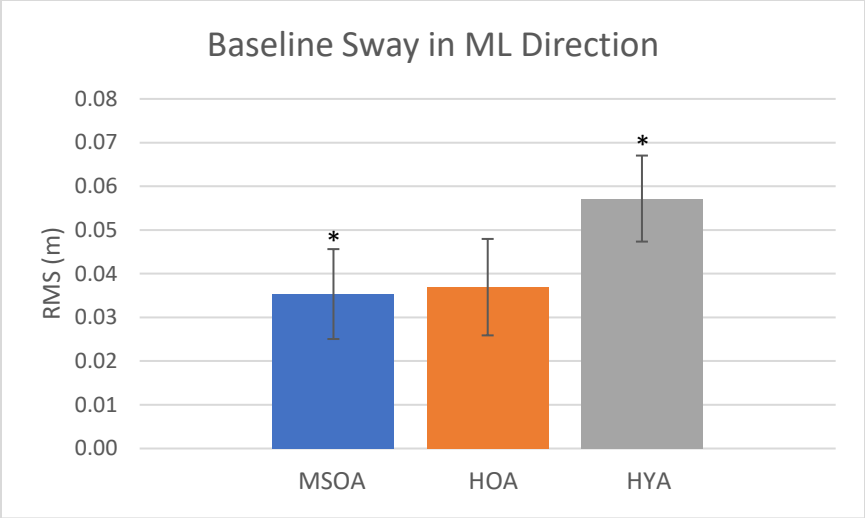


*Figure 3: RMS measure for baseline sway trial in the medio-lateral direction. (MSOA=Older Adult with Multiple Sclerosis, HOA=Healthy Older Adults, HYA=Healthy Young Adults on Foam; * = statistical difference between groups)*

Figure 4 shows the 95% ellipse for the spatial dimension for all groups. Upon analyzing the 95% ellipse for the spatial directions between the MSOA, 0.0071±0.0004 $m^2$, and HOA group, 0.0002±0.00005 $m^2$ , a significant difference was found ($p<0.05$). A significant difference was also seen between the MSOA and HYA on foam, 0.0004±0.00029 $m^2$ ($p<0.05$). Given these values, the MSOA created a larger area with greater COP shifts.

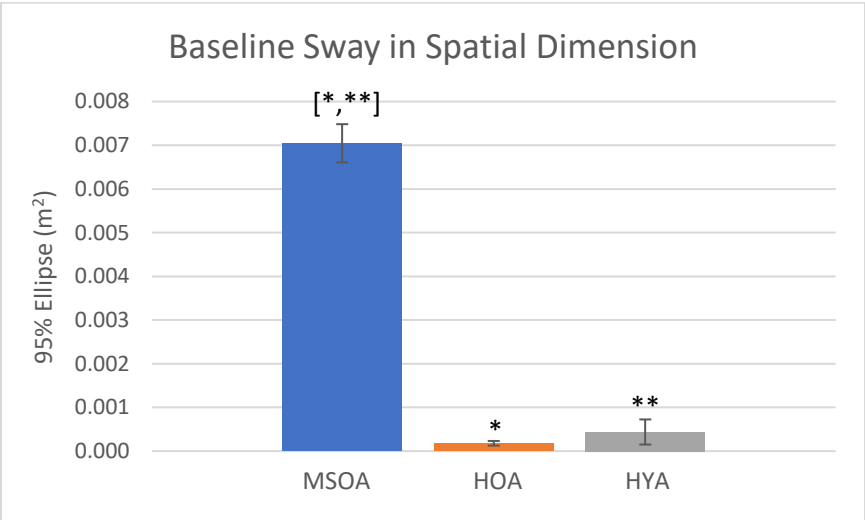


*Figure 4: 95% Ellipse measure for baseline sway trial in the spatial dimension. (MSOA=Older Adult with Multiple Sclerosis, HOA=Healthy Older Adults, HYA=Healthy Young Adults on Foam; * or **= statistical difference between groups)*

Figures 5, 6, and 7 show the averages and standard deviations for the baseline sample entropy measures in all directions. When comparing the MSOA and HOA group, the only significant differences were seen in the AP direction and spatial dimensions. For the AP direction, the MSOA had an average entropy value of 0.55±0.07, whereas the HOA group showed an average entropy value of 0.35±0.13 ($p<0.05$). The MSOA had an entropy value that was approximately twice that of the HOA group for the spatial measure. No significant difference ($p>0.05$) was displayed in the ML direction between the MSOA and HOA. Similar results were seen when comparing the MSOA with the HYA on foam.

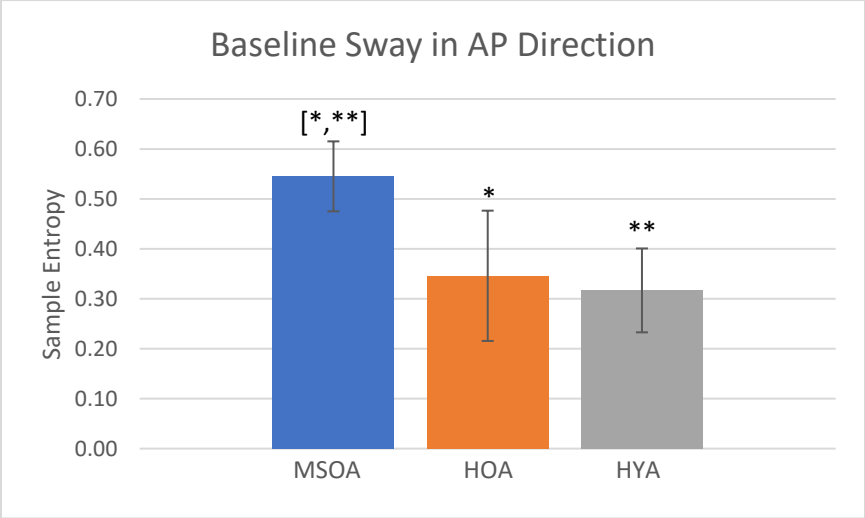


*Figure 5: Sample entropy measure for baseline sway trial in the anterior-posterior direction. (MSOA=Older Adult with Multiple Sclerosis, HOA=Healthy Older Adults, HYA=Healthy Young Adults on Foam; * or **= statistical difference between groups)*

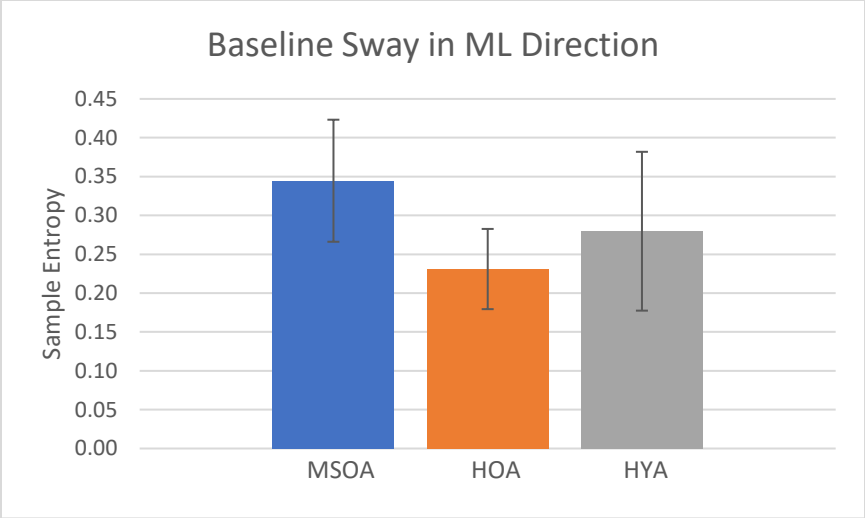


*Figure 6: Sample entropy measure for baseline sway trial in the medio-lateral direction. (MSOA=Older Adult with Multiple Sclerosis, HOA=Healthy Older Adults, HYA=Healthy Young Adults on Foam)*

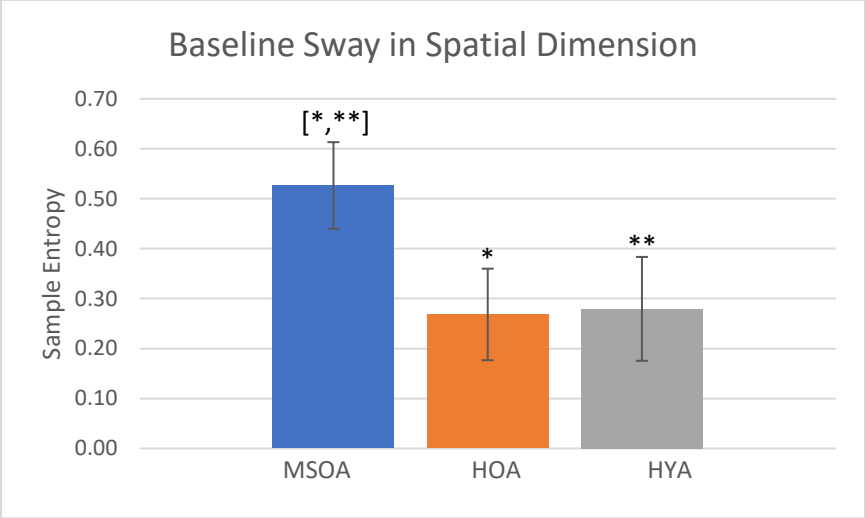


*Figure 7: Sample entropy measure for baseline sway trial in the spatial dimension. (MSOA=Older Adult with Multiple Sclerosis, HOA=Healthy Older Adults, HYA=Healthy Young Adults on Foam; * or **= statistical difference between groups)*

Post-vibratory treatment

After the vibratory stimulus was applied, a quiet stance trial was conducted to analyze the effects of vibration on the linear and non-linear measures of sway. Some limitations should be addressed before detailing the results. The MSOA was only able to complete one session of quiet stance without assistance of holding onto the walker. The color of noise applied during this

session was white noise. During the following two sessions, which included the placebo and pink noise, as the testing session progressed, the participant increasingly needed to rely on assistance. The post-vibration trial was completed with the participant leaning on the walker. Given this obstacle, the effect of vibration for the placebo and pink noise trial were not compared to the respective baseline trials because assistance from the walker affects balance. Instead, the post-vibratory effect was examined by comparing the final trial of the placebo with the final trial of the pink noise. The placebo trial took place prior to the pink noise trial to ensure there was no residual effects of vibration from a previous trial before the data was collected. Statistical tests were not able to be conducted for these set of results given the extremely small sample size of one participant in the MSOA group. Descriptive statistics were provided in lieu of the tests.

Similar to the baseline measures, the post-vibratory measures were also extracted for the AP, ML, and Spatial directions. Figures 8 and 9 display the difference between the baseline and post-vibratory trial for the white noise stimulus in the AP and ML directions of RMS. All three groups were included to examine the effect across all groups for just one noise color. For the AP direction, there was only a slight increase of magnitude, 0.005 meters, between the baseline and post-vibratory trial in the MSOA. The other groups also displayed minor changes, but none with significant differences. In the ML direction, the MSOA had an increase of 0.016 meters between the baseline and post-vibratory trial. The HOA and HYA groups in the ML direction did not display any significant differences between the baseline and post-vibratory trial.

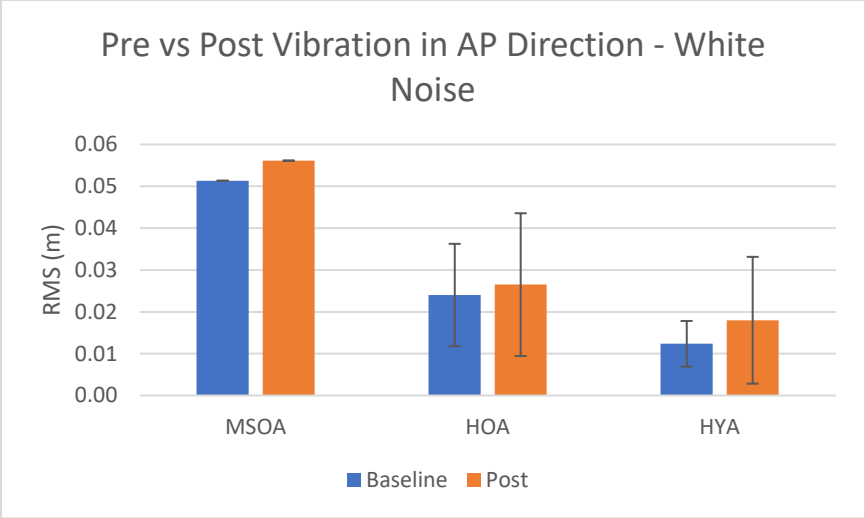


*Figure 8: RMS measure for pre vs post vibration sway trial in the anterior-posterior direction for white noise stimulus. (MSOA=Older Adult with Multiple Sclerosis, HOA=Healthy Older Adults, HYA=Healthy Young Adults on Foam)*

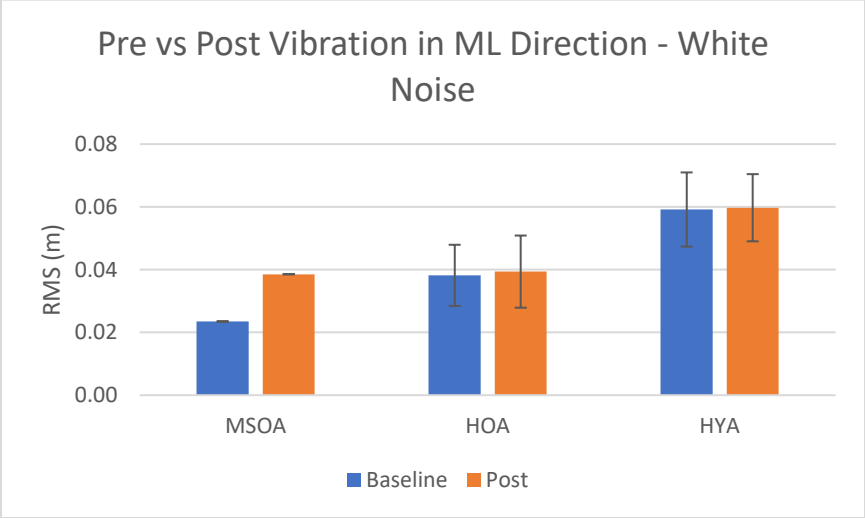


*Figure 9: RMS measure for pre vs post vibration sway trial in the medio-lateral direction for white noise stimulus. (MSOA=Older Adult with Multiple Sclerosis, HOA=Healthy Older Adults, HYA=Healthy Young Adults on Foam)*

Figure 10 below shows the 95% ellipse of the spatial direction for the three groups for only the white noise color. The MSOA had an area decrease of approximately 0.002 $m^2$ between the baseline and post-vibratory trial. The HOA and HYA groups did not demonstrate a notable difference in 95% ellipse between trials.

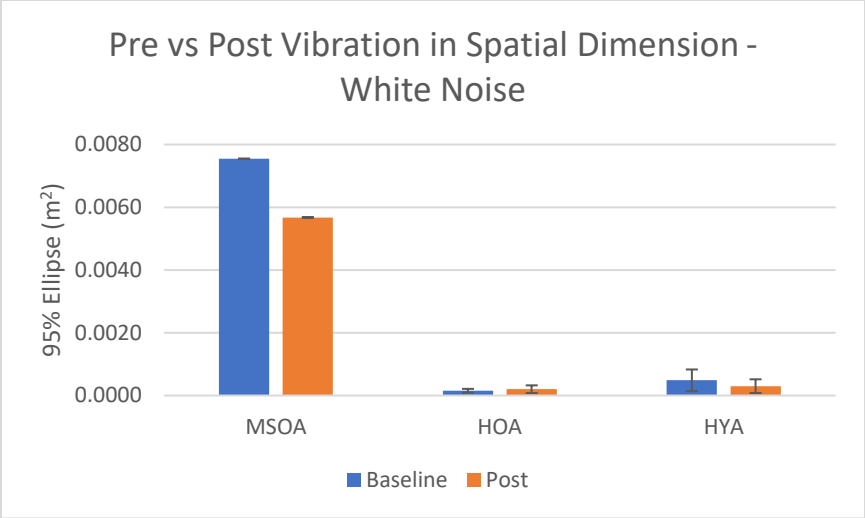


*Figure 10: 95% ellipse measure for pre vs post vibration sway trial in the spatial dimension for white noise stimulus. (MSOA=Older Adult with Multiple Sclerosis, HOA=Healthy Older Adults, HYA=Healthy Young Adults on Foam)*

The following figures display the effects of the sub-threshold vibratory stimulus for the comparison of the placebo and pink noise trials. Figure 11 below shows the RMS for the AP direction. Figure 12 shows the same measure for the ML direction. In the AP and ML direction, the pink noise trial showed a reduction of 0.03 meters compared to the placebo trial.

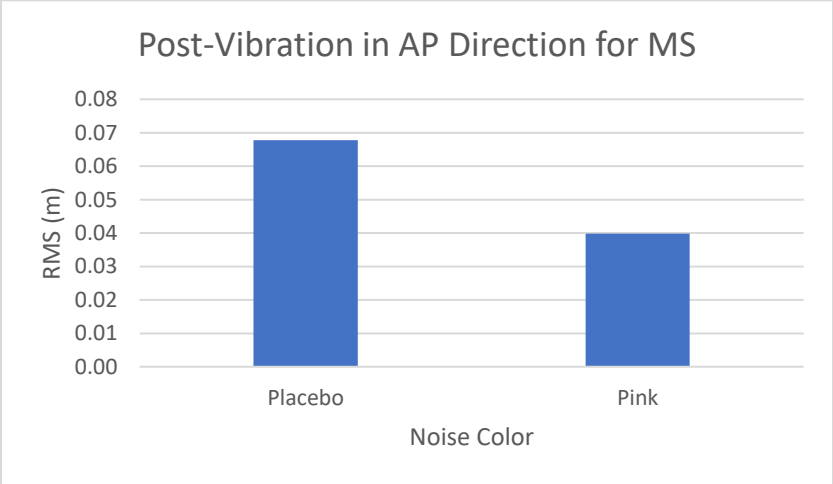


*Figure 11: RMS measure for post vibration sway trial for MS participant in the anterior-posterior direction for placebo and pink noise stimulus.*

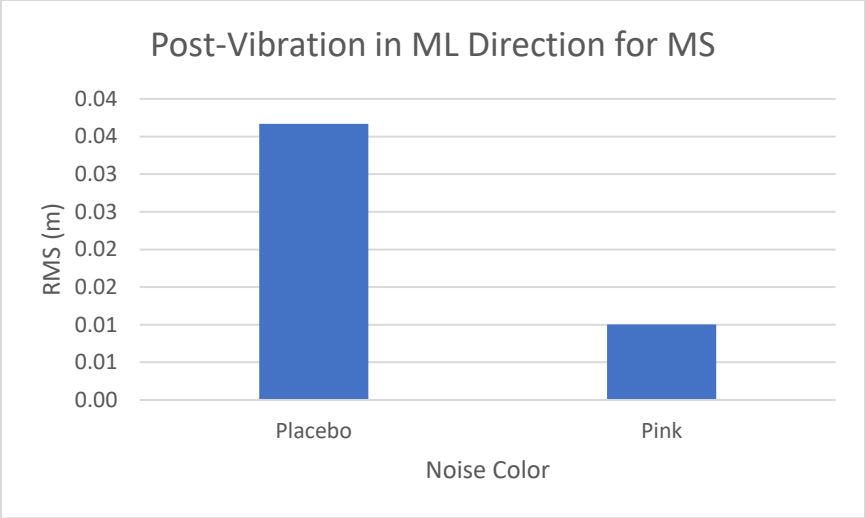


*Figure 12: RMS measure for post vibration sway trial for MS participant in the medio-lateral direction for placebo and pink noise stimulus.*

Figure 13 shows the 95% ellipse for the spatial direction. The pink noise trial displayed an area of 0.00074 $m^2$ compared to the placebo trial with a value of 0.00050 $m^2$.

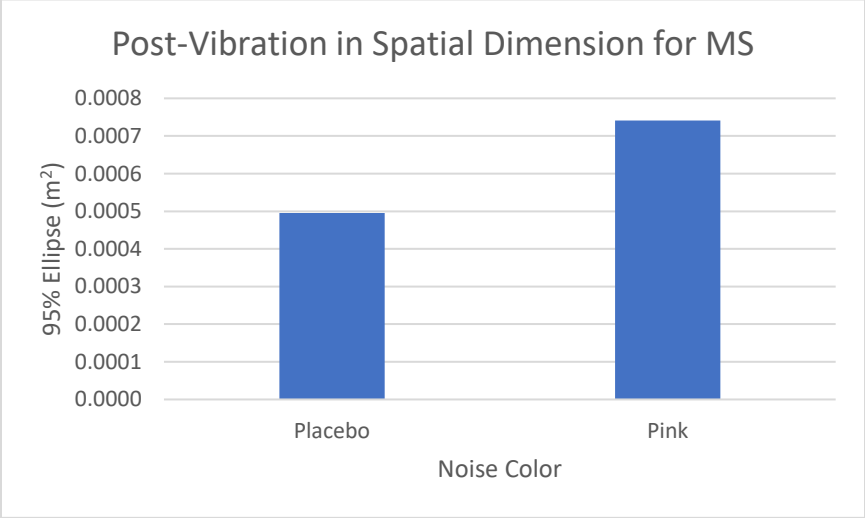


*Figure 13: 95% ellipse measure for post vibration sway trial for MS participant in the spatial dimension for placebo and pink noise stimulus.*

The following figures display the entropy values for the post-vibratory trials of the placebo versus pink noise trial. Figures 14-16 show the entropy values for the AP, ML, and spatial dimension, respectively. In the AP direction, the entropy increased from 0.25 to 0.32. In

the ML direction, the entropy decreased from 0.21 to 0.15. In the spatial dimension, the entropy increased from 0.25 to 0.37.

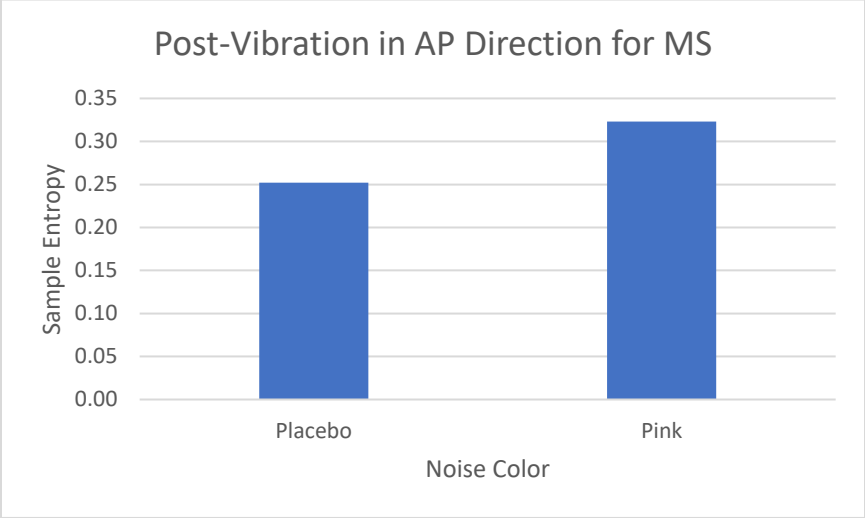


*Figure 14: Sample entropy measure for post vibration sway trial for MS participant in the anterior-posterior direction for placebo and pink noise stimulus.*

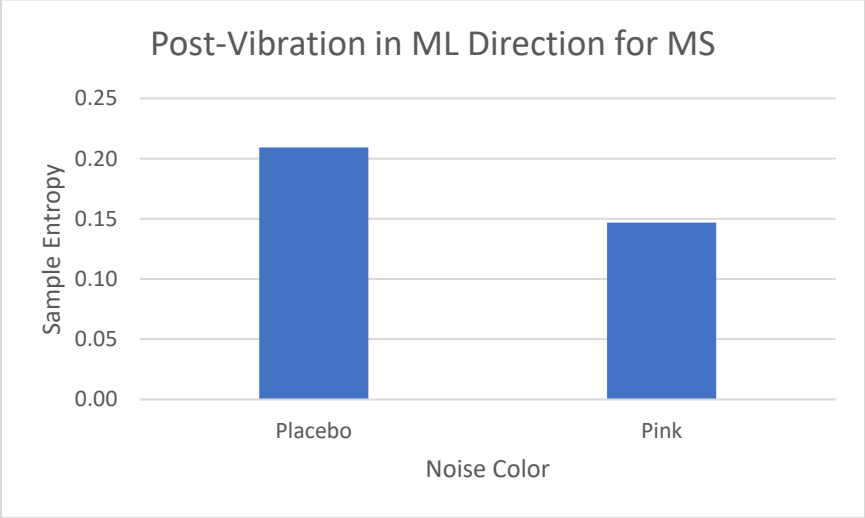


*Figure 15: Sample entropy measure for post vibration sway trial for MS participant in the medio-lateral direction for placebo and pink noise stimulus.*

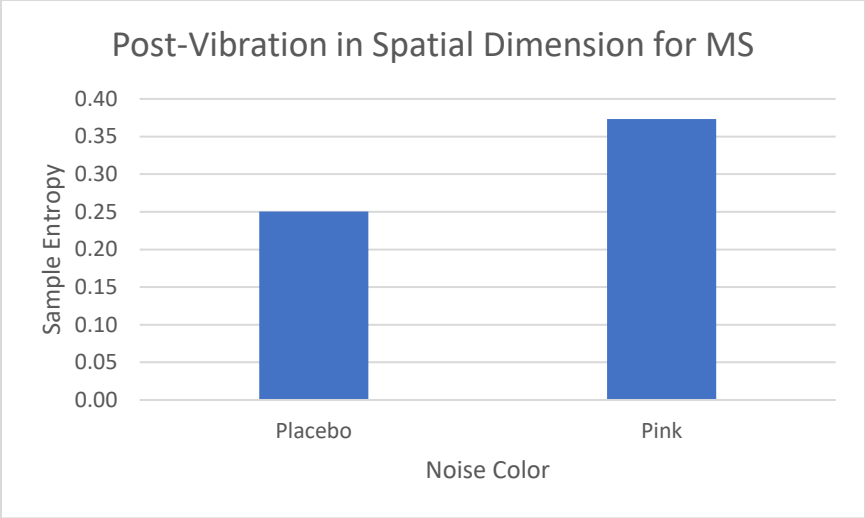


*Figure 16: Sample entropy measure for post vibration sway trial for MS participant in the spatial dimension for placebo and pink noise stimulus.*

## Discussion

The purpose of this pilot study was to investigate the effect of sub-sensory vibration on balance in persons with MS. The mat was recently developed and validated by a previous pilot study conducted in the lab using HOA participants and HYA on one inch of foam (Giraldo, 2021; Whorley, 2020). The current study was conducted during the COVID-19 pandemic; this resulted in substantial limitations during the recruitment process. However, given the limitations, the one participant diagnosed with MS was compared to previous data of the HOA and HYA on foam as initial data to be used to design future studies.

The first aim of the study was to understand how the sensing thresholds and linear/non-linear sway parameters of the MSOA compared to the HOA and HYA on foam. The MSOA was in the same age range as the HOA participants, which allowed for a healthy versus MS comparison. The MSOA was compared to the HYA on foam to understand if using foam as a sensory deficit was comparable to a population with MS. The second aim of the study was to investigate the effect of the sub-threshold vibration on various sway parameters in the MS group.

### Sensing Threshold

The sensory threshold was analyzed across all three groups. It was hypothesized that healthy young adults on one inch of foam will display similar threshold values to MS. The sensory threshold of the MS participant was significantly higher than the HYA on foam, but not the HOA. This indicated that even though the HYA stood on foam, they were still more sensitive than MS. A significant difference was presented between the two healthy groups, so it is assumed that HYA on foam is not a representative simulation of a severe type of MS, such as primary progressive MS. Instead, the HYA on foam provided values that were closer to the HOA. Using the foam more closely resembled age-related sensory deficits rather than severe

stages of MS. The data does not support the hypothesis. A larger MS population and a wider range of MS severities would be needed to confirm this finding.

Baseline

Baseline data was collected while standing on the mat placed on top of the force plate without any vibration applied. Linear and non-linear measures were analyzed in the AP, ML, and Spatial directions. The linear measures included RMS and the 95% ellipse, while the non-linear measure included sample entropy. It was hypothesized that MS would exhibit greater COP sway parameters with eyes closed compared to healthy groups.

The MSOA demonstrated larger RMS values in the AP direction for all three trials compared to the other two healthy groups, meaning the participant demonstrated higher variability sway in the AP direction with their eyes closed. However, in the ML direction, HYA on foam with eyes closed presented more variability with a larger RMS value. Analysis of the 95% ellipse in the spatial dimension showed that the MSOA demonstrated a greater sway area overall compared to the other two healthy groups. This finding is supported by previous studies that found people with MS having more erratic sway with a larger sway area, especially when they have their eyes closed (Cattaneo et al., 2014; Sosnoff et al., 2011; Inojosa et al., 2020).

The MSOA also demonstrated a larger entropy value in the AP and spatial direction compared to the other two groups. The larger entropy values defined the sway pattern of the MSOA to be more unpredictable in the AP direction and overall. This supports conclusions from an existing study, which found that the elderly and those with pathologies may generate less complexity and more random outputs indicating a more unpredictable sway pattern (Costa et al., 2007). Overall, the data supports the hypothesis, except for the magnitude of sway in the ML

direction. A discrepancy to note in the data is the time duration of the baseline trials. The length of the trials for the HOA and HYA participants were 90 seconds; the duration for the MSOA was cut down to 60 seconds for the safety of the participant and to reduce the chance of fatigue.

Effect of vibratory stimulus

The post-vibratory trial took place immediately after receiving the vibration stimulus with the desired color of noise. The MSOA was only able to complete one trial of eyes closed quiet standing without needing assistance for the white noise session. For the other two sessions, assistance was needed for the resulting post-vibratory trials. The participant was informed that they may place their hands on their walker for their safety. Due to this limitation, the baseline and post-vibration data could be compared for only white noise. The placebo and pink noise trials could not be compared to the baseline trial. Instead, the post-vibration trial for placebo was compared to the pink noise to still analyze an effect, if any, on the sway parameters. It was hypothesized that the sub-threshold vibratory stimulus would improve balance in MS.

For the white noise trial, there was a slight increase in RMS magnitude in both the AP and ML direction post-vibration. The 95% ellipse measure showed a decrease as well as the sample entropy making the sway pattern appear more predictable; however, the participant grabbed onto their walker towards the end of the 60 second trial which could have affected the results. This leads to the conclusion that white noise did not improve balance for the participant. Other factors could also affect the sway such as fatigue and the physical exertion required to balance with eyes closed.

When examining the effect of pink noise compared to the placebo for the post-vibratory trial, the pink noise showed a decrease in RMS magnitude in both the AP and ML directions.

However, when analyzing the 95% ellipse measure, the pink noise trial demonstrated a larger sway area than the placebo trial. Upon analyzing the predictability of sway, pink noise resulted in more unpredictable sway than placebo in the AP and spatial directions. This demonstrates that pink noise did not improve balance compared to the placebo trial. Overall, the data does not support the hypothesis. The findings of this pilot study support the results of another study that investigated the effects of acute and chronic whole-body vibration applied to the feet over a 5-week period on balance and saw no significant differences (Freitas et al., 2018).

Additional findings can be observed from the placebo session. As the testing session progressed, the participant required more assistance, from both the research assistant and their walker. Upon analyzing the baseline trial versus the post-vibratory trial, the effect of the assistive device on balance can be summarized. The total sway area was reduced by approximately 93% when holding on to the walker while keeping their eyes closed. This highlights the importance of assistive devices, such as canes and walkers, in pathological populations who suffer from poor postural control. A study that examined postural control in stroke patients with and without the use of a cane found that the sway area decreased by about 54-58% when the cane was used (Maeda et al., 2001). These assistive aids are important in fall-risk populations because they provide improved balance and postural stability.

Given the small sample size and limitations that occurred during the study, only initial observations can be made from the results. These observations can be used as fundamental knowledge to improve the design of future studies conducted using sensory deficit populations in the lab on the vibratory mat. Overall, it was evident that the MS group displayed greater variability and unpredictability of sway compared to age-matched healthy older individuals. The foam used for the healthy young individuals provided results similar to age-related changes, but

not necessarily a pathological population. When looking at the effects of the different colors of noise, the color of the noise does not appear to be an effect on sway parameters. Using this data, a preliminary conclusion can be made that sub-threshold vibration might not be a suitable intervention for those with severe stages of MS. However, these assumptions will need to be confirmed using a larger sample size.

## Limitations

Several limitations were present in this pilot study. First, there were obstacles in the recruiting process. The study took place during the pandemic, so it proved to be a very difficult time to recruit human subjects, especially immunosuppressed, for multiple sessions in research lab space. Recruiting took place through the MS clinic at the University of Kansas Medical Center (KUMC). A neurologist and director of the clinic at KUMC was informed of the study and offered to help recruit for the study by placing recruitment flyers in the exam and waiting rooms. While several people took interest from the flyer, as evidenced by the detachment of study contact information, very few potential participants contacted the study administrator. It is possible that the travel distance between the Lawrence and interested patient's home was an obstacle for some people. Second, there were limitations during the data collections. Some trials had to be shortened from 90 seconds to 60 seconds for the physical safety of the participant and to minimize fatigue that could increase the risk of falling. The participant also required assistance from the walker as the session progressed. Standing with eyes closed appeared to be a demanding task which may fatigued the participant.

## Future Studies

Preliminary findings obtained from this pilot study, can help guide future projects. This project should be replicated with a larger sample size and include more stages of MS to understand how the vibration stimulus affects various severities. With travel distance being a potential obstacle to subject recruitment, collecting data in the clinic using a vibratory mat should help to improve enrollment into the study.

## Conclusion

Multiple sclerosis is associated with poor balance, leading to a greater incidence of falls which proposes a major health risk. Some current interventions used to improve balance are not suitable for those with mobility impairments, like MS. Stochastic resonance has increasingly been explored as a method of improving balance for pathological populations. Sub-threshold stochastic vibration has shown great potential in improving balance in populations such as elderly, stroke, and diabetics. Its effects are not as well known in populations with multiple sclerosis, but this pilot study serves as an initial exploration. More research needs to be done to further understand the effects of different colors of noise on sway parameters in multiple sclerosis. Ultimately, understanding the effects of stochastic resonance on disease populations, such as MS, could provide clinicians with information to innovate new therapies to treat individuals with balance deficits, reducing fall risk and improving overall quality of life for millions.

## References


Wajda, D. A., Moon, Y., Motl, R. W., & Sosnoff, J. J. (2015). Preliminary investigation of gait initiation and falls in multiple sclerosis. *Archives of Physical Medicine and Rehabilitation*, *96*(6), 1098–1102. https://doi.org/10.1016/j.apmr.2014.12.011

Allum, J. H. J., Rust, H. M., Lutz, N., Schouenborg, C., Fischer-Barnicol, B., Haller, V., … Yaldizli. (2021). Characteristics of improvements in balance control using vibro-tactile biofeedback of trunk sway for multiple sclerosis patients. *Journal of the Neurological Sciences*, *425*, 117432. https://doi.org/10.1016/j.jns.2021.117432

Kalron, A., & Achiron, A. (2014). The relationship between fear of falling to spatiotemporal gait parameters measured by an instrumented treadmill in people with multiple sclerosis. *Gait and Posture*, *39*(2), 739–744. https://doi.org/10.1016/j.gaitpost.2013.10.012

Loyd, B. J., Fangman, A., Peterson, D. S., Gappmaier, E., Schubert, M. C., Thackery, A., & Dibble, L. (2019). Rehabilitation to improve gaze and postural stability in people with multiple sclerosis: Study protocol for a prospective randomized clinical trial. *BMC Neurology*, *19*(1), 1–8. https://doi.org/10.1186/s12883-019-1353-z

Correale, J., Gaitán, M. I., Ysrraelit, M. C., & Fiol, M. P. (2017). Progressive multiple sclerosis: from pathogenic mechanisms to treatment. *Brain*, *140*(3), 527–546. https://doi.org/10.1093/brain/aww258

Inojosa, H., Schriefer, D., Trentzsch, K., Klöditz, A., & Ziemssen, T. (2020). Visual Feedback and Postural Control in Multiple Sclerosis. *Journal of Clinical Medicine*, *9*(5), 1291. https://doi.org/10.3390/jcm9051291

White, O., Babic, J., Trenado, C., Johannsen, L., & Goswami, N. (2019). The promise of stochastic resonance in falls prevention. *Frontiers in Physiology*, *10*(JAN), 1–12. https://doi.org/10.3389/fphys.2018.01865

Zhou, J., Lipsitz, L., Habtemariam, D., & Manor, B. (2016). Sub-sensory vibratory noise augments the physiologic complexity of postural control in older adults. *Journal of NeuroEngineering and Rehabilitation*, *13*(1), 1–8. https://doi.org/10.1186/s12984-016-0152-7

Costa, M., Priplata, A. A., Lipsitz, L. A., Wu, Z., Huang, N. E., Goldberger, A. L., & Peng, C. K. (2007). Noise and poise: Enhancement of postural complexity in the elderly with a stochastic-resonance-based therapy. *Epl*, *77*(6). https://doi.org/10.1209/0295-5075/77/68008

Sosnoff, J. J., Socie, M. J., Boes, M. K., Sandroff, B. M., Pula, J. H., Suh, Y., … Motl, R. W. (2011). Mobility, balance and falls in persons with multiple sclerosis. *PLoS ONE*, *6*(11), 2–6. https://doi.org/10.1371/journal.pone.0028021

Kasser, S. L., Jacobs, J. V., Foley, J. T., Cardinal, B. J., & Maddalozzo, G. F. (2011). A prospective evaluation of balance, gait, and strength to predict falling in women with multiple sclerosis. *Archives of Physical Medicine and Rehabilitation*, *92*(11), 1840–1846. https://doi.org/10.1016/j.apmr.2011.06.004

Freitas, E. D. S., Frederiksen, C., Miller, R. M., Heishman, A., Anderson, M., Pardo, G., … Bemben, M. G. (2018). Acute and Chronic Effects of Whole-Body Vibration on Balance, Postural Stability, and Mobility in Women With Multiple Sclerosis. *Dose-Response*, *16*(4), 1–13. https://doi.org/10.1177/1559325818816577

Giraldo, C. (2021). *Design, Validation and Assessment of Vibratory Mat for Stochastic Resonance Applications* (Publication No. 28547245) [Doctoral dissertation, University of Kansas]. ProQuest Dissertations Publishing.

McIlroy, W. E., & Maki, B. E. (1997). Preferred placement of the feet during quiet stance: Development of a standardized foot placement for balance testing. *Clinical Biomechanics*, *12*(1), 66–70. https://doi.org/10.1016/S0268-0033(96)00040-X

Whorley, B. (2020). *A Physiology-Based Approach for Detecting Vibration Perception Threshold in the Plantar Foot* (Publication No. 28029429) [Master's Thesis, University of Kansas]. ProQuest Dissertations Publishing.

Kelty-Stephen, D. G., & Dixon, J. A. (2013). Temporal correlations in postural sway moderate effects of stochastic resonance on postural stability. *Human Movement Science*, *32*(1), 91–105. https://doi.org/10.1016/j.humov.2012.08.006

Dyck, P.J., O'Brien, P.C., Kosanke, J.L., Gillen, D.A., and Karnes, J.L, (1993). A 4, 2, and 1 stepping algorithm for quick and accurate estimation of cutaneous sensation threshold. *Neurology*, *43*(8), 1508-1512.

Cattaneo, D., Rabuffetti, M., Bovi, G., Mevio, E., Jonsdottir, J., & Ferrarin, M. (2014). Assessment of postural stabilization in three task oriented movements in people with multiple sclerosis. *Disability and Rehabilitation*, *36*(26), 2237–2243. https://doi.org/10.3109/09638288.2014.904933

Maeda, A., Nakamura, K., Higuchi, S., Yuasa, T., & Motohashi, Y. (2001). Postural sway during cane use by patients with stroke. *American Journal of Physical Medicine and Rehabilitation*, *80*(12), 903–908. https://doi.org/10.1097/00002060-200112000-00006

## Chapter 4: Summary

### Summary of Study

The purpose of this study was to investigate the effects of stochastic resonance on balance in multiple sclerosis. The pilot study included a participant with primary progressive multiple sclerosis. The participant completed a quiet standing task with their eyes closed and feet placed in a comfortable position. Due to the small sample size of the current study, data from a previous pilot study conducted in the lab using the same procedure, was included in the study which included four healthy older adults and four healthy young adults, the latter which were on one inch of foam. The purpose of the foam was to simulate somatosensory deficiency. Three different types of vibrations (white, pink, placebo) were administered to the participants to analyze how balance is affected by different stimuli. To understand the effect on balance, linear and non-linear sway measures were extracted from the center of pressure time-series. The magnitude of sway was examined using linear measures of the Root Mean Square and 95% ellipse. The predictability of sway was examined using the non-linear measure, sample entropy.

During the baseline trial, the multiple sclerosis participant demonstrated a larger magnitude of sway in the anterior-posterior direction and a greater sway area overall when compared to similarly aged healthy participants. The multiple sclerosis participant also exhibited more unpredictable patterns of sway. The trials that took place after the vibratory stimulus was applied did not result in an improvement for any of the linear or non-linear measures for any of the three vibrations.

## Conclusions

Multiple sclerosis is associated with balance and mobility impairments, which may worsen as the disease progresses. These impairments lead to an increased risk of falls. Interventions to improve balance are an important aspect for reducing falls in affected populations. Stochastic resonance has been studied in sensory deficit populations to investigate how it affects balance. While it has shown promising results in healthy aging, stroke, and diabetic neuropathy populations, little is known about its impact in those with multiple sclerosis. The results concluded from this study help to set a foundation for future studies. Further research should be done to understand how different types of vibrations affect various sway parameters.

## Limitations and Future Work

A main limitation in this study is the small sample size of multiple sclerosis participants. The study took place during the COVID-19 pandemic which was an obstacle in the recruitment process. Another potential reason for limited recruiting could be the distance between the main campus and the clinic where recruitment was focused. In future work, the limitations could be addressed through several solutions. One solution, given adequate time and resources, would be to develop the vibratory mat in a manner that could be portable. While this solution would take time to execute, it would propose a more efficient way to collect data if the mat was in a space closer to the clinic. Another solution is to go through the institutional review board process at the medical center to gain approval of entering the clinic and talking to potential participants directly. Overall the study should be replicated under the same conditions with a larger sample size. Due to the promising results of stochastic resonance in other sensory deficit populations, more research should be conducted to understand if it is a suitable intervention for multiple sclerosis.

## Appendix A: Tables

*Table 1: Means and standard deviations for sensory threshold percentage of all groups. (MSOA=Multiple Sclerosis Older Adult, HOA=Healthy Older Adults, HYA=Healthy Young Adults on foam)*

| | Sensory Threshold (%) | | |
|---|---|---|---|
| | HOA | HYA | MSOA |
| Mean | 44.41667 | 39.25 | 57.66667 |
| SD | 17.77362 | 6.047163 | 1.154701 |

*Table 2: Means and standard deviations of the Root Mean Square (RMS) measure taken from the baseline trial on the mat for the anterior-posterior (AP) and medio-lateral (ML) directions for all three groups. (MSOA=Multiple Sclerosis Older Adult, HOA=Healthy Older Adults, HYA=Healthy Young Adults on foam)*

| | Baseline Mat | | |
|---|---|---|---|
| | RMS (m) | | |
| | MSOA | HOA | HYA |
| AP | 0.0515±0.00022 | 0.0173±0.01140 | 0.0145±0.01105 |
| ML | 0.0353±0.01028 | 0.0369±0.01104 | 0.0572±0.00984 |

*Table 3: Mean and standard deviation of the 95% ellipse measure taken from the baseline trial on the mat in the spatial dimension for all three groups. (MSOA=Multiple Sclerosis Older Adult, HOA=Healthy Older Adults, HYA=Healthy Young Adults on foam)*

| | Baseline Mat | | |
|---|---|---|---|
| | 95% Ellipse ($m^2$) | | |
| | MSOA | HOA | HYA |
| Spatial | 0.0071±0.00044 | 0.0002±0.00 | 0.0004±0.00029 |

*Table 4: Mean and standard deviation of the sample entropy measure taken from the baseline trial on the mat for the anterior-posterior (AP) and medio-lateral (ML) for all three groups. (MSOA=Multiple Sclerosis Older Adult, HOA=Healthy Older Adults, HYA=Healthy Young Adults on foam)*

| | Baseline Mat | | |
|---|---|---|---|
| | Sample Entropy | | |
| | MSOA | HOA | HYA |
| AP | 0.5451±0.07006 | 0.3460±0.13044 | 0.3167±0.08399 |
| ML | 0.3447±0.07856 | 0.2310±0.05162 | 0.2797±0.10222 |
| Spatial | 0.5266±0.08668 | 0.2683±0.09171 | 0.2794±0.10401 |

*Table 5: Post-vibration trial (T0) data for the Root Mean Square (RMS) measure in the anterior-posterior (AP) and medio-lateral (ML) directions for all colors of noise (white, pink, placebo) across all groups. The means and standard deviations (SD) are only given for the healthy populations. (MSOA=Multiple Sclerosis Older Adult, HOA=Healthy Older Adults, HYA=Healthy Young Adults on foam)*

| | T0 | | | | | |
|---|---|---|---|---|---|---|
| | RMS (m) | | | | | |
| | White | | Pink | | Placebo | |
| | AP | ML | AP | ML | AP | ML |
| MSOA | 0.05612 | 0.03853 | 0.03980 | 0.01005 | 0.06781 | 0.03667 |
| HOA-Mean | 0.02651 | 0.03938 | 0.01278 | 0.03667 | 0.00900 | 0.03835 |
| HOA-SD | 0.01706 | 0.01150 | 0.00662 | 0.01260 | 0.00635 | 0.01393 |
| HYA-Mean | 0.01799 | 0.05974 | 0.02088 | 0.05935 | 0.01872 | 0.05269 |
| HYA-SD | 0.01515 | 0.01070 | 0.00888 | 0.01036 | 0.01620 | 0.00648 |

*Table 6: Post-vibration trial (T0) data for the 95% ellipse measure in the spatial dimension for all colors of noise (white, pink, placebo) across all groups. The means and standard deviations (SD) are only given for the healthy populations. (MSOA=Multiple Sclerosis Older Adult, HOA=Healthy Older Adults, HYA=Healthy Young Adults on foam)*

| | T0 | | |
|---|---|---|---|
| | 95% Ellipse ($m^2$) | | |
| | White | Pink | Placebo |
| | Spatial | Spatial | Spatial |
| MSOA | 0.00567 | 0.00074 | 0.00050 |
| HOA-Mean | 0.00020 | 0.00024 | 0.00018 |
| HOA-SD | 0.00012 | 0.00013 | 0.00007 |
| HYA-Mean | 0.00030 | 0.00037 | 0.00048 |
| HYA-SD | 0.00022 | 0.00032 | 0.00045 |

*Table 7: Post-vibration trial (T0) data for the sample entropy measure in the anterior-posterior (AP), medio-lateral (ML) directions, and spatial dimension for all colors of noise (white, pink, placebo) across all groups. The means and standard deviations (SD) are only given for the healthy populations. (MSOA=Multiple Sclerosis Older Adult, HOA=Healthy Older Adults, HYA=Healthy Young Adults on foam)*

| | T0 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Sample Entropy | | | | | | | | |
| | White | | | Pink | | | Placebo | | |
| | AP | ML | Spatial | AP | ML | Spatial | AP | ML | Spatial |
| MSOA | 0.4443 | 0.2919 | 0.4133 | 0.3230 | 0.1468 | 0.3731 | 0.2522 | 0.2092 | 0.2502 |
| HOA-Mean | 0.3686 | 0.2670 | 0.3159 | 0.3462 | 0.2159 | 0.2876 | 0.3231 | 0.2101 | 0.2013 |
| HOA-SD | 0.1263 | 0.0563 | 0.0589 | 0.1115 | 0.0437 | 0.1174 | 0.1806 | 0.0456 | 0.0641 |
| HYA-Mean | 0.3032 | 0.2693 | 0.2589 | 0.3622 | 0.2441 | 0.2533 | 0.2849 | 0.3256 | 0.3262 |
| HYA-SD | 0.0527 | 0.1422 | 0.1388 | 0.1076 | 0.0955 | 0.1114 | 0.0847 | 0.0658 | 0.0600 |

*Table 8: p-values from the t-tests for the comparison of sensory threshold percentages. (MSOA=Multiple Sclerosis Older Adult, HOA=Healthy Older Adults, HYA=Healthy Young Adults on foam)*

| | MSOA vs HYA | HOA vs HYA |
|---|---|---|
| Sensory Threshold | 1.065E-07 | 1.783E-01 |

*Table 9: p-values from the t-tests for the comparison of the Root Mean Square (RMS) in the anterior-posterior (AP) and medio-lateral (ML) direction, as well as the 95% ellipse in the spatial dimension. (MSOA=Multiple Sclerosis Older Adult, HOA=Healthy Older Adults, HYA=Healthy Young Adults on foam)*

| | MSOA vs HOA | MSOA vs HYA |
|---|---|---|
| RMS-AP | 2.475E-07 | 8.268E-08 |
| RMS-ML | 4.148E-01 | 2.255E-02 |
| 95% Ellipse | 6.762E-04 | 8.059E-04 |

*Table 10: p-values from the t-tests for the comparison of the sample entropy (SE) in the anterior-posterior (AP), medio-lateral (ML), and spatial direction. (MSOA=Multiple Sclerosis Older Adult, HOA=Healthy Older Adults, HYA=Healthy Young Adults on foam)*

| | MSOA vs HOA | MSOA vs HYA |
|---|---|---|
| SE-AL | 0.00565 | 0.00419 |
| SE-ML | 0.07012 | 0.14797 |
| SE-Spatial | 0.01085 | 0.00666 |

*Table 11: Power analysis to determine sample size needed for 80% power of each linear/non-linear measure. (MSOA=Multiple Sclerosis Older Adult, HOA=Healthy Older Adults)*

| | AP | | ML | | Spatial | |
|---|---|---|---|---|---|---|
| | MSOA | HOA | MSOA | HOA | MSOA | HOA |
| RMS/95% Ellipse | 1 | 1 | 561 | 561 | 1 | 1 |
| Sample Entropy | 4 | 4 | 5 | 5 | 2 | 2 |

*Table 12: Power analysis to determine sample size needed for 80% power of each linear/non-linear measure. (MSOA=Multiple Sclerosis Older Adult, HYA=Healthy Young Adults on foam)*

| | AP | | ML | | Spatial | |
|---|---|---|---|---|---|---|
| | MSOA | HYA | MSOA | HYA | MSOA | HYA |
| RMS/95% Ellipse | 1 | 1 | 3 | 3 | 1 | 1 |
| Sample Entropy | 3 | 3 | 25 | 25 | 2 | 2 |

## Appendix B: MATLAB Code

## Data Visualization Code

```
%Written by: Camilo Giraldo - Healthy Vibration Sway Study v2 - Data
Visualization
%University of Kansas - Biodynamics Lab
%Updated by: Victoria Blackwood
clear; close all; clc;
%% General
%File paths
path_raw='E:\Research- Codes and Data\Raw Data\';
path_res='E:\Research- Codes and Data\Results\';
path_pro='E:\Research- Codes and Data\Processed Data\';
%Subject numbers
sub_no=[1006,2004];
%Color names
protocol_colors_abv={'WH' 'PK' 'PB'};
protocol_colors={'White' 'Pink' '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','sout
houtside')
                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','sout
houtside')
```

```
                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
```

## Data Filter Code

```
%Written by: Camilo Giraldo - Healthy Vibratio Sway Study v2 - Data Check
(Filt)
%University of Kansas - Biodynamics Lab
%Updated by: Victoria Blackwood

clear; close all; clc;

%% General
%File paths
path_raw='E:\Research- Codes and Data\Raw Data\';
path_res='E:\Research- Codes and Data\Results\';
path_pro='E:\Research- Codes and Data\Processed Data\';

%Subject numbers
sub_no=[2004];

%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)
        %not doing brown color
        if jj==3
            continue
        end

        %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;

%                 %getting rid of NaN data
%                 for LL = 1:size(data,1)
%                     if sum(isnan(data(LL,:))) == 12 %12 or 6
%                         data=data(1:LL-1,:);
%                         fprintf('Not 90 s\t');
%                         break
%                     end
%                 end

                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:size(data_comb,1)+1000);

                    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;

                %getting rid of NaN data
                for LL = 1:size(data,1)
                    if sum(isnan(data(LL,:))) == 6
                        data=data(1:LL-1,:);
```

```
                        fprintf('Not 90 s\t');
                        break
                    end
                end

                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 swaydata
                    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;

                    %getting rid of NaN data
%                     for LL = 1:size(data,1)
%                         if sum(isnan(data(LL,:))) == 7
%                             data=data(1:LL-1,:);
%                             fprintf('NaN Data Removal s\t');
%                             break
%                         end
%                     end

                    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
```

## Data Analysis Code

```
%Written by: Camilo Giraldo - Healthy Vibratio Sway Study v2 - Data Analysis
%University of Kansas - Biodynamics Lab
%Updated by: Victoria Blackwood

clear; close all; clc;

%% General
%File paths
path_raw='E:\Research- Codes and Data\Raw Data\';
path_res='E:\Research- Codes and Data\Results\';
path_pro='E:\Research- Codes and Data\Processed Data\';

%Subject numbers
sub_no=[2004];

%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)
        if jj==3
            continue
        end

        %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));
                %axis([-90 90 -120 50])
                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;

%                 D_BL_gnd=distance(COP(:,1),COP(:,2));
%                 COPx_max= 1000*max(COP(:,1));
%                 COPx_min= 1000*min(COP(:,1));
%                 COPy_max= 1000*max(COP(:,2));
%                 COPy_min= 1000*min(COP(:,2));
%                 COP_range_x=COPx_max-COPx_min;
%                 COP_range_y=COPy_max-COPy_min;

                %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;
                %axis([-90 90 -120 50])
                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;
```

```
%                 D_BL_mat=distance(COP(:,1),COP(:,2));
%                 COPx_max= 1000*max(COP(:,1));
%                 COPx_min= 1000*min(COP(:,1));
%                 COPy_max= 1000*max(COP(:,2));
%                 COPy_min= 1000*min(COP(:,2));
%                 COP_range_x=COPx_max-COPx_min;
%                 COP_range_y=COPy_max-COPy_min;

                %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;
                %axis([-90 90 -120 50])
```

```
                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
```

## Measures Code

```
%Written by: Camilo Giraldo - Healthy Vibratio Sway Study v2 - Linear
Measures
%University of Kansas - Biodynamics Lab
%Updated by: Victoria Blackwood

clear; close all; clc;

%% General
%File paths
path_raw='E:\Research- Codes and Data\Raw Data\';
path_res='E:\Research- Codes and Data\Results\';
path_pro='E:\Research- Codes and Data\Processed Data\';

%Subject numbers
sub_no=[2004];

%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)
        %not doing brown color
        if jj==3
            continue
        end

        %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;

%         %getting rid of NaN data
%                 for LL = 1:size(data,1)
%                     if sum(isnan(data(LL,:))) == 12 %12 or 6
%                         data=data(1:LL-1,:);
%                         fprintf('Not 90 s\t');
%                         break
%                     end
%                 end

        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(['E:\Research- Codes and
Data\codes\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)
        if jj==3
            continue
        end

        %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})*100,2);

tab_cell{tab_count,end}=round(sum(data_analysis{10,1+jj})/length(data_analysi
s{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(2:4,:));

%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)
        if jj==3
            continue
        end

        %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_mi
n,DFA_t_max,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_mi
n,DFA_t_max,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,DFA_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)
        if jj==3
            continue
        end

        %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
```