Incorporation of Pressure Insoles into Inverse Dynamic Analysis

by

Ahmad Mahmassani

A thesis

presented to the University of Waterloo

in fulfillment of the

thesis requirement for the degree of

Master of Applied Science

in

System Design Engineering

Waterloo, Ontario, Canada, 2021

Author’s Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Estimation of body loads during industrial tasks, such as lifting and weight bearing, is central to workplace ergonomics and the study of the safety and risk factors in work techniques. Evaluating those loads requires data collection of body kinematics and the external forces prevailing during the task under evaluation. Current practice calls for kinematic data to be gathered using optical motion capture systems (OMC) and external forces, primarily ground reaction forces (GRFs), to be gathered using force plates. However, this experimental methodology is confined to laboratory settings.

Modern motion capture systems, such as those based on Inertial Measurement Units (IMUs), pave the way to more versatile motion analysis techniques not confined to labs. Inverse dynamics models have been developed based on IMU kinematic data. In order to eliminate the need for force plates and to make the experimental apparatus fully portable, those models estimate GRFs from measured accelerations.

This study aimed to advance the state-of-the-art on IMU-based inverse dynamics analysis by incorporating pressure insoles as the source of the vertical components of the GRFs, with a view to improving the model fidelity while keeping the experimental apparatus portable. Specifically, it enabled the development of a synchronized and automated inverse dynamics model, comprised of an inertial motion capture suite and pressure insoles, that can estimate net joint forces and moments during manual handling activities.

An experiment was designed to examine whether the GRFs measured by the pressure insole can detect and differentiate among various sizes (and weights) of concrete masonry units (CMUs). The instrumented pressure insoles were consistently able to identify three different CMU block weights (8 kg, 16kg, and 24 kg) during various gait patterns (along circular, square, and linear paths). On the other hand, the results were inconclusive in distinguishing between one-handed and two-handed manual handling of CMUs. An improved inverse dynamic model was introduced to calculate the joint loads workers experience during material manual handling based only on measurements by IMU motion capture suits and pressure insoles.

The outcome of this thesis was the development of a weight detection algorithm with a detection accuracy of 89% across all three sizes of CMUS as well as an integrated inverse dynamic model incorporating data collected by IMUs motion suits and pressure insoles.

Acknowledgements

First, I would like to thank my supervisors, Professor Eihab Abdel-Rahman and Professor Carl T. Haas for their guidance and supervision throughout the period developing this thesis. I would also like to thank my colleagues Dr. JuHyeong Ryu and Tasha McFarland for their assistance and encouragement throughout this duration. Lastly, I would like to extend my thanks to the thesis reviewers; Professor Nima Maftoon and Professor Stewart McLachlin for taking time to read and provide feedback to improve and enhance this thesis.

This research was made possible through the support and funding from the Canadian Masonry Design Center (CMDC), Canadian Concrete Masonry Producers Association (CCMPA), and the Natural Sciences and Engineering Research Council NSERC (CRDPJ 494786-16). I would like to also thank Prof. Bennet Banting and David Stubbs for their time, insight and liaison between our research and the construction sector.

I would like to thank all my friends and colleagues including Prof. Amer Keblawi, Prof. Hussein Daoud, Mr. Ossama Abou-Ali Modad and Mr. Jihad Sinno for their support, advice, and encouragement.

Lastly, I would like to thank my parents and my sister their utmost support, love, and consistent encouragement throughout the entire period of this thesis.

Dedication

A million thank you is never enough…

To my parents,

Issam Mahmassani and Majida Mahmassani

And to my sister,

Lina Mahmassani

Table of Contents

List of Figures

List of Tables

List of Abbreviations

A/P	Anteroposterior
.BVH	Biovision Hierarchy file
BW	Body weight
.Calc	Calc File
CMU	Concrete Masonry Unit
COM	Center of mass
COP	Center of pressure
CPS	Capacitive Pressure Sensors
CS	Coordinate System
dps	Degree per second
FTI	Force-time Integral
g	$1g=9.8\ m/s^2$
IMC	Inertial Motion Capture
IMU	Inertial Measurement Unit
LSB	Least significant bit
M/L	Mediolateral
MH	Metacarpal Head
MH_1	Metatarsal Head
MRI	Magnetic Resonance Imaging
MSD	Musculoskeletal disorders
SC	Surface Coverage
TAI	Total Area of Insole

TAS	Total Area covered by the pressure sensors
WMSD	Work related musculoskeletal disorder

Chapter 1
Introduction

1.1 Problem Statement

The construction industry is one of the industries having the highest musculoskeletal disorders (MSDs). The high rates of MSDs are attributed to physically exerting tasks such as lifting heavy loads. That is, over the past 10 years in Canada, sprains/strains and overexertion were reported as the leading injuries and injury events listed at 44% and 18% respectively. (WSIB, 2020) These occupational tasks include exposing workers to lifting and moving heavy loads which will consequently yield to elevated body loads and awkward postures (Mermarian et al., 2012). The main issue is that the construction industry is heavily reliant on manual labor as the main driver of the sector despite the presence of advanced technological advancements as site complexities prohibit extensive usage and incorporation of automation and robotics.

Hence, multiple ergonomic risk evaluation methods were proposed to prevent WMSDs (Punnett & Wegman, 2004). For instance, postural ergonomic assessment methods such as Rapid Upper Limb Assessment (RULA), (McAtamney and Corlett, 1993), Rapid Entire Body Assessment (REBA) (Hignett & McAtamney, 2000), and Ovako Working posture Analysing System (OWAS) (Karhu et al., 1977) rely on an experts' manual observation to measure posture and body movement which is subjective and imprecise.

Motion capture systems provide a potential alternative for expert observation, as they can provide automatic, continuous, and full body motion data collecting. Motion capture technology can not only provide postural and motion analysis, but also suport analysis of physical demands and loads on body segments through biomechanical modelling (Radwin et al., 2001). In construction, two main motion capture methods are utilized: optoelectronic motion capture (OMC) systems and inertial motion capture (IMC) systems. Marker based OMCs are considered the 'gold standard' for kinematic data collection (Kim & Nussbaum, 2013). However, their use in field studies is impractical due to their setup requirements, procedures, and constraints such as illumination, sensitivity to outdoor conditions, required direct line of sight to the test subject, and a restricted required space. Hence OMC usage is confined to laboratory settings. Other camera-based systems that rely on remote sensing or computer vision algorithms could be utilized; their systems have the advantage of marker-less tracking (van der

Kruk & Reijne, 2018). However, these systems rely on cameras thus constraining their utility to direct line of sight and illumination effects (Seo et al., 2015).

Compared to OMCs, inertial motion capture systems are more flexible to utilize as they are not only portable, lightweight, cost-efficient, and not dependent on line of sight or illumination (Alwasel et al., 2017a; Bolink et al., 2016; Morrow et al., 2017; Robert-Lachaine et al., 2017; van der Kruk & Reijne, 2018), but are also accurate in measuring kinematics due to advancements in accelerometer technologies (Bolink et al., 2016; Morrow et al., 2017; Robert Lachaine et al., 2017).

In biomechanics, inverse dynamics is used to predict the forces and the moments exerted by the body muscles and joints under a certain activity. The main requirements of such analysis are kinematic data, anthropometric measurements, and the external forces applied on the body.

Since the anthropometric data is predetermined based on the participants, the external forces usually consist of the ground reaction forces (GRFs) and hand loads. Hand loads are usually set based on the experimental design, but GRFs are the main force that needs to be measured. GRFs are usually measured by Force Plates (FPs), but FP measurements are confined in laboratory settings due to the following issues: expensive, limited capture area, and have to be installed to the ground. Other techniques utilize GRF estimation based on empirical data (L. Ren et al, 2008), using neural networks (S.E. Oh, et al, 2013), or through kinematic data (Diraneyya, 2019). Other GRF measurement techniques are done by utilizing pressure insoles, as they provide the flexibility to remain wireless, light, and accurate, and have no constraints to laboratory settings.

Hence a quantitative ergonomic assessment tool has been developed that evaluates internal joint forces using IMC systems only, and it has been been utilized to evaluate loads and risk levels experienced by masons (Diraneyya, 2019), however, no synchronized system between IMUs and pressure insoles were used or introduced in masonry.

1.2 Research Objectives and Contributions

The main objective of this thesis was to incorporate pressure insoles into the existing IMU suits to substitute the ground reaction force estimations that were fed into the inverse dynamics model. Furthermore, this research aimed to exploit the use of pressure insole features to develop an automated weight detection algorithm to detect the weight carried by the masons when conducting a manual handling task. The research objectives for this thesis include:

1- Review the literature on the measurement techniques and estimation methods for finding the ground reaction forces.
2- Review the literature on weight detection using plantar pressure detection methods to develop the criteria for developing the weight detection algorithm.
3- Integrate pressure insole GRFs into an existing inverse dynamics model and incorporate the sampling and synchronizing modifications between the IMC system and the pressure insoles.
4- Develop a biomechanical analysis that will evaluate joint loads for participants performing Concrete Masonry Unit (CMUs) carriages of different weights throughout various paths.

The main contribution of the thesis was to update the currently used biomechanical analysis tool by a novel sensor and integration methodology that aimed to improve the fidelity of the current model. In addition, the thesis contributed a weight detecting feature to the biomechanical model that aimed to reduce the complexity of the user interface when used by a non-biomechanics specialist in the masonry training center.

1.3 Scope

The research was conducted under the recommendations and requirements that were set by the Canadian Masonry Design Center (CMDC). The biomechanical assessment tool has been developed to evaluate joint loads and moments for masonry workers under masonry-related tasks. The focus of the biomechanical assessment tool was to update the previously developed assessment tool methodology to incorporate updated methods and tools into the inverse dynamics model embedded in the tool. The improvements were constrained within the scope of sensor integration, data synchronization and processing, as well as improved functionality.

1.4 Thesis Organization

The organization of this thesis is represented below:

Chapter 2 provides a critical review of the previous studies related to the thesis to contextualize the research objective and eventual contribution of this research.

Chapter 3 presents a feasibility study that to detect CMU weights from GRF data exported from the insoles during bilateral hand-load carriage. The chapter also describes the results and limitations that resulted during weight detection using unilateral load carriage.

Chapter 4 introduces the validation study and experimental procedure for CMU weights detection during bilateral hand-load carriage of 3 types of CMUs during 3 gait paths: linear, squared, and circular paths. Lastly, the chapter discusses the performance of the weight detection algorithm through the validation study.

Chapter 5 begins by reviewing the previous biomechanical model that was utilized to evaluate the kinetic parameters of the body joints throughout a masonry task. This is then followed by the adjustments that were utilized to incorporate the pressure insoles into the updated biomechanical model that was utilized in the validation study. The chapter further discusses and evaluates the results of the right and left shoulder moments, right and left elbow moments, as well as the L4/L5 compression force of the lower back.

Chapter 6 concludes the thesis by summarizing the outcomes and contributions of the research and provides limitations of the research and recommendations for future research.

Chapter 2

Literature Review

2.1 Musculoskeletal Disorders in Masonry

The construction industry is a leader in MSDs (Hess et al, 2010; Merlino et al, 2003). Work related MSDs are widely exhibited by construction workers due to task properties such as overexertion, repetitive motion, and awkward postures. This is reinforced by the masonry industry being one of the highest rates of overexertion and back injuries reported to be 43-66.5 and 45.3-75.4 injuries per 10,000 full time equivalents, respectively (Hess et al., 2010; Kincl et al., 2016). Overexertion is a major risk factor in WMSD resulting in days away from work in construction. It is also a leading cause of other nonfatal injuries nationally (CPWR, 2018). In 2015, overexertion during lifting and lowering accounted for 29.9% of the total 20,490 WMSDs in construction. Whereas other overexertion activities such as pushing, pulling, carrying, and holding account for 41.7% of WMSDs in construction in 2017 (CPWR, 2019). These data are shown in Figure 1.

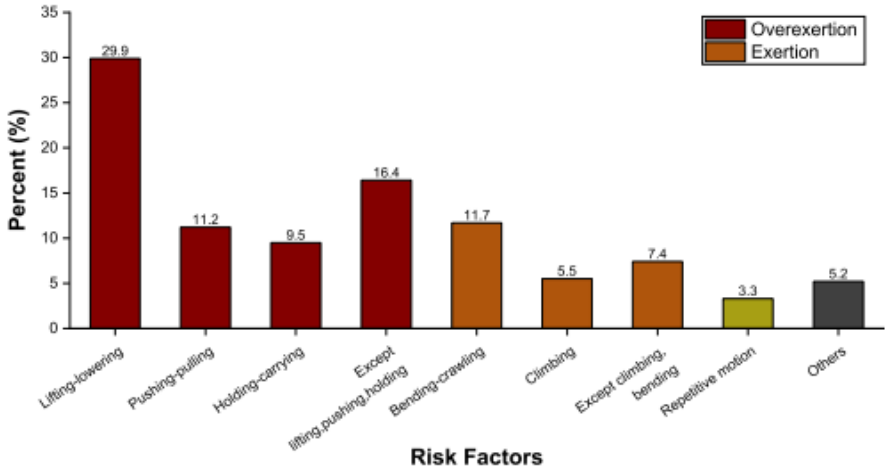

Figure 1: WMSDs risk factors that yield in days away from work, (20,490 cases) (CPWR, 2018)

These high-risk factors mainly originate from manual material handling (MMH) tasks, and therefore they expose the workers to heavy loads, awkward postures, and high repetition (Mermarian et al., 2012; Van Der Molen et al., 2008; Kumar, 2001). As a daily average, masons may lay between 240 to 294 concrete blocks weighing 11-16 kg, resulting in manual handling of cumulative loads up to 3840 kg per day (Gallagher and Heberger, 2013). In brick laying activities, masons may lay down an average

of 1,000 bricks per day (Mitropoulos and Memarian, 2012; Schneider and Susi, 1994). Repetition is a concern since it has been shown to interact with force to increase MSD risk through a fatigue failure process in the tissues (Kumar, 2001; Gallagher and Heberger, 2013).

Consequently, the body part most affected by WMSDs in construction is the back (reported between 2003 and 2017) (CPWR, 2013, 2018, 2019). Back injuries accounted for 41.7% of WMSDs in construction in 2017, which was even higher than all other industries combined (CPWR, 2019)

Financially, high rates of MSDs have burdened the healthcare system in Canada. In Ontario alone, the class of MSDs is the number one cause of lost-time injuries, and costs hundreds of millions of dollars (MOL, 2019c). The total burden of musculoskeletal disorders on the Canadian economy is estimated around $ 22 billion dollars annually. Mainly these disorders are related to workplace hazards (IMHA, 2019).

Similarly in the United States, the National Academy of Social Insurance estimated that the workers' compensation programs paid $61.9 billion in worker benefits. Construction workers received significantly more compensation benefits compared to workers in other industries; the total compensation costs as a percentage of employer spending in construction is nearly three times the average cost for all other industries (3.6% vs. 1.4%) (CPWR, 2018).

The direct costs of workers' compensation for non-fatal claims with more than five days away from work in construction was about $10.4 billion in 2017 (LMRIS, 2020b). Note that overexertion injuries involving outside sources, such as lifting, holding, and carrying objects, amounted to $1.48 billion in direct costs and accounted for 14.2% of the total compensation cost in construction. Overexertion was also ranked the first among the leading causes of disabling injuries for all workplaces in the U.S. in 2017, accounting for $13.98 billion which is 23.5% of the overall national burden (LMRIS, 2020a).

2.2 Injury and Experience in Masonry

Experience plays a role in limiting the number of work-related MSDs in comparison to the less experienced apprentices, mainly since the experienced workers develop essential skills and parameters about the working procedures (Gyekye & Salminen, 2010). The alarming issue is that masonry apprentices are assumed to be healthy due to their new role in the industry, but 78% of apprentices are reporting musculoskeletal symptoms/injuries (Anton et al., 2020). This is mainly because expert workers (known as journeymen) implement strategies that are different than that of first year

apprentices. For instance, when implementing material manual handling tasks, journeymen tend to straighten their back, orient their pelvis, and implement shorter steps (Authier et al. (1995, 1996)). Hence, it can be deduced that comparing kinematic and kinetic variations between first year apprentices and journeymen, one can determine appropriate and safe working parameters. However, there is no quantitative relationship between body loads, level of experience, and working methods under normal working conditions (Ryu, 2021).

2.3 Ergonomic Risk Assessment

Awkward body postures and forced motions are considerable risk factors in causing work-related musculoskeletal disorder (WMSDs), and they are main causes in creating musculoskeletal stresses beyond tissue tolerance (Kumar, 2001). Hence, detecting these postural problems is of importance, and consequently observation-based ergonomic risk assessments were developed, as self-reporting risk factors is very simplistic, and the assessments tend to be subjective and misleading (Plantard et al., 2015; Spielholz et al., 2001).

2.3.1 Observational Risk Assessment

Observational-based risk assessment tend to evaluate risk exposure based on postural variations throughout a task. Hence, evaluators utilize rule-based assessment tools such as Rapid Upper Limb Assessment (RULA) (McAtamney & Corlett, 1993), Rapid Entire Body Assessment (REBA) (Hignett & McAtamney, 2000), and Ovako Working posture Analysing System (OWAS) (Karhu et al., 1977). These tools are the most prevalent observational assessment techniques in the industry (Andreoni et al., 2009; Kee & Karwowski, 2007; Kong et al., 2018a; Lee & Han, 2013; Roman-Liu, 2014).

These tools rely on a whole-body postural scoring system, and based on the grand score, that dictates the level of risk for that posture and suggests recommended actions according to the partial scores of the segments. The difference between the observation assessment tools is the body sectioning of each method. That is, RULA and REBA divide the human body into two sections (i.e., Section 1 is the arms and wrists, and Section 2 is the neck, trunk, and legs) whereas OWAS divides it into three sections (i.e., back, arms, and legs). The RULA-based ergonomic risk assessment is shown in Figure 2:

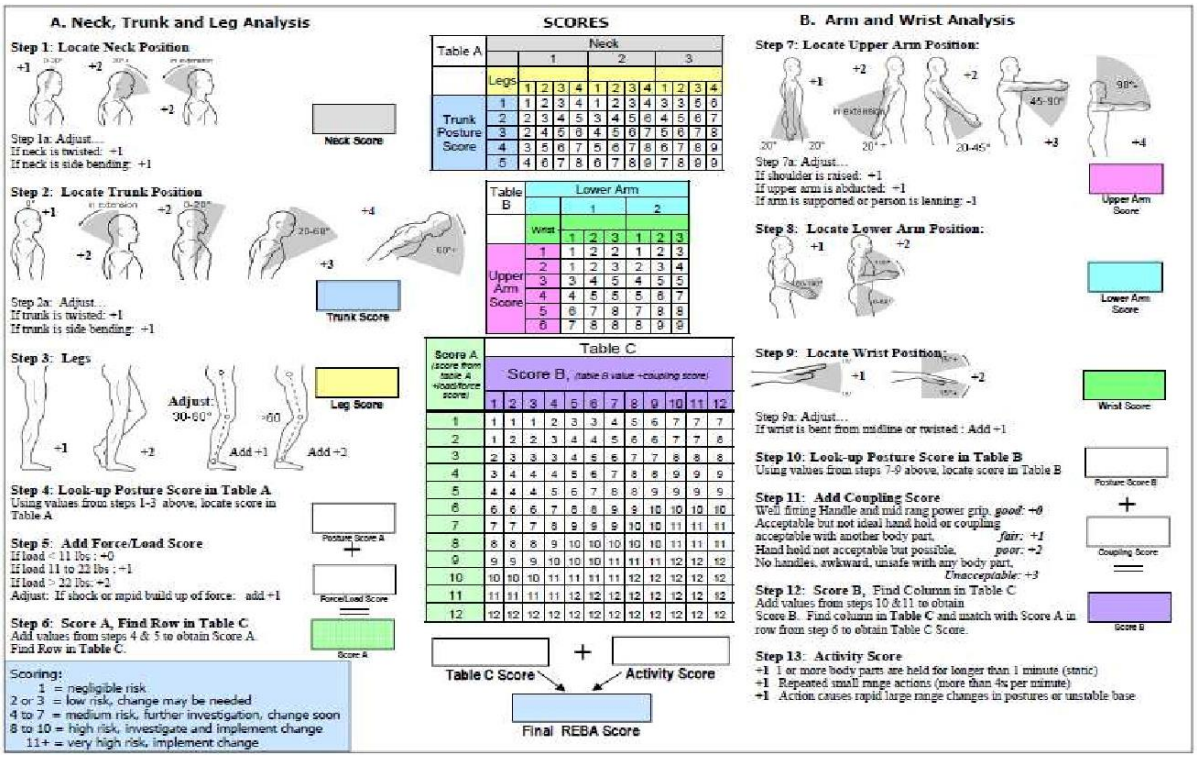

Figure 2: RULA observation-based biomechanical model

These methods are traditionally based on manual observation of joint angles by ergonomic experts which makes the evaluation results prone to inaccuracies and inconsistencies due to human error, such as being unable to distinguish the same joint angle for similar posture. Hence the results are not repeatable and have a high variance.

2.3.2 Computational Risk Assessment

Due to the inaccuracies that the qualitative risk assessment methods provide, automated postural worker data collection has been developed due to the emergence of sensors that enable accurate collection of kinematic and postural data which enhances assessment accuracy and reliability. As a result, direct measurement methods have been widely used on workers with various tasks in the construction industry. One of the main sensors that were utilized in the construction industries are the full body inertial measurement units (IMUs), which are a set of accelerometers, gyroscopes, and magnetometers that will capture 3D worker activities throughout a task.

IMU-based motion capture systems enable accurate joint kinematics data collection in comparison with the the current gold standard, optical motion capture systems (Cuesta-Vargas et al., 2010; Robert-Lachaine et al., 2017, 2020; Schall Jr et al., 2016). Consequently, the IMU data can serve as an input to evaluate postural assessment systems like RULA, REBA and OWAS which are discussed in Section 2.3.1.

For instance, Vignais et al. (2013) combined seven IMUs attached to the upper-body with RULA to compute the risk-level in real-time during light manual tasks such as screwing and unscrewing. Furthermore, Vignais et al. (2017) conducted an IMU based continuous RULA evaluation on cleaning medical materials. Battini et al. (2014) also introduced an ergonomic evaluation system using 17-IMUs suits integrated assessment tools (including RULA and OWAS) for manual material handling tasks in a warehouse setting. Moreover, Valero et al. (2016) developed a system to detect basic unsafe postures of construction workers (e.g., stooping and squatting with back bending) using a wearable IMU suit.

2.4 Quantitative Biomechanical Analysis

Evaluating biomechanical joint loads requires 3D whole-body biomechanical models that will estimate the joint forces and moments of the joints of interests. That is, software packages, such as 3D Static Strength Prediction Program (3DSSPP) (Chaffin et al., 2006), AnyBody (Damsgaard et al., 2006), and OpenSim (Delp et al., 2007), have been developed to carry out biomechanical analysis (Seo et al., 2015). 3DSSPP use body segment parameters, stature, body weight, link lengths, link weights, link centers of gravity, and strength, determined based on values for U.S. industrial population by The Center for Ergonomics at the University of Michigan. It evaluates compression forces in the lumbar joint at L4/L5 disc level, and the joint moments in the elbow, shoulder, L5/S1 disc, hip, and knee joints, using a top-down approach that starts from the forces and moments applied to the hands and ends with the forces and moments applied to the feet. In addition, it utilizes ten L4/L5 level torso muscles, 5 each on the left and right side, to evaluate the L4/L5 lumbar compression force.

Lastly, integrating IMU motion capture data into the inverse dynamics model can automate the biomechanical analysis. For example, Alwasel et al. (2017a) assessed the loads acting on 21 masons' body joints (lower back, shoulders, elbows, hips, and knees) using 3DSSPP driven by obtained IMU-based postural data. On-site biomechanical analysis can, thus, provide explicit and quantitative computation of joint loads, which is a better analytical method compared to the conventional ergonomic assessment methods, such as observation and self-reporting.

2.5 Previously Developed Inverse Dynamic Models

The research described in this thesis builds directly on the previous research conducted to evaluate joint loads and muscle injury risk from biomechanical inertial motion data from masons. (Diraneyya, 2019). IMU sensors are worn by participants while completing a task and the outputs of the IMU sensors are processed by an assessment tool interface. The interface requires multiple inputs that include the participants' mass and height, as well as whether the lift is carried by the left or right hand. The assessment tool utilizes an inverse dynamics solver that estimates the net joint forces and moments, in particular the lower back compression forces, L4/L5 shear forces on the disc, as well as shoulder, elbow, hip, knee, and ankle moments. The assessment tool also provides feedback on critical postures throughout the manual handling task by providing a color-coded stick figure that represents joint risks in the body. This is represented in Figure 3.

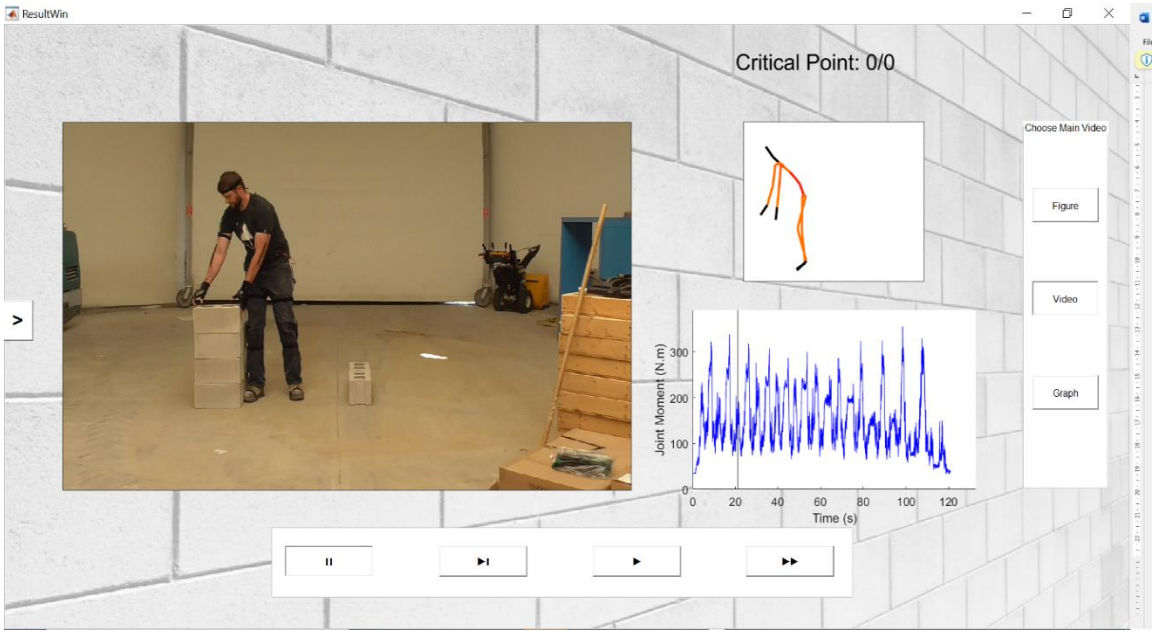

Figure 3: Online assessment tool user interface

Despite the online assessment tool methods are novel compared to other ergonomic assessment methodologies which are qualitative in nature, the assessment tool requires several improvements and features. For instance, in the absence of established thresholds for joint moments and forces, the assessment tool used an arbitrary threshold of 80% of the peak force or moment to act as criteria for defining critical time segments. The tool also reported the loads at the critical points in N or N•m, which

without context. It is not meaningful to masons without expertise in biomechanics (McFarland, 2021). Furthermore, the inverse dynamics model, is fully IMU based, which translates to certain inputs such as the ground reaction forces (GRFs) being estimated (not measured) and fed into the solver which causes errors in the ground reaction forces components. The reason for estimating GRFs is due to the issue that GRFs under a single stance is a determinate problem (1 GRF force required), but it is an indeterminate problem when there is double stance posture (Right and Left GRF variables required), which requires estimation then optimization methods.

2.5.1 Measurement of Ground Reaction Forces

2.5.1.1 Force Plates

GRFs are a kinetic component that can be measured through various methods or be computed analytically based on the type of activity. There are multiple methods to measure GRFs, and one of the main sensor classes that is force plates. In masonry, the main types of conducted activities are in large spaces and are outside lab environments and require a lot of mobility. The main problem with force plates is that they require a lab setting and require a dedicated place for them to be installed in the lab (Ancillao et al, 2018). Furthermore, elaborate analysis of gait under material manual handling tasks requires a lot of space and cannot be covered by the force plates. Two alternatives can be proposed: pressure insoles or estimating GRFs using kinematic data.

2.5.2 Estimation of Ground Reaction Forces

2.5.2.1 Smooth Transition Function Method

There are many attempts to estimate GRFs through kinematic data. One proposed to use a ‘smooth transition functions’ that predict the ratio of the two feet GRFs during the double support phase of the gait cycle. The main assumption is that the tailing foot vertical (y-component) and the lateral (z-component) values of the GRFs drop to 0 as the foot leaves the ground according to the relation (Ren et al, 2008):

$$\frac{F_{y,z}}{F_{y_0,z_0}} = e^{(t/T_{ds})^3}$$

Where F_y and F_z are the instantaneous forces, Fy_0 and F_{z0} are the total forces. The anterior- posterior value of the GRF (GRF_x) is modelled according to the following relation:

$$\frac{F_x}{F_{x_0}} = k_1 e^{-[(t/T_{ds})-(2/3)]^2} - k_2 \frac{t}{T_{ds}}$$

Where T_{ds} is the half the period of the double support phase, and k_1 and k_2 are constants that are determined from the boundary condition at heel strike and toe off.

2.5.2.2 Neural Network Kinematics

A neural network is used to predict GRFs from motion kinematics. The main issue with that method is that the start and the end of the double support phase has been detected using force plates (Oh et al, 2013) prior to running the neural network algorithm to estimate the GRFs. This is not practical to implement on construction sites with many masons as force plates cannot be utilized in non-laboratory settings.

2.5.2.3 Contact Detection Algorithm

A gait event detection algorithm is developed to predict foot contact with the ground using the norms of the heel velocity and the toe threshold velocity. This was utilized with a modified smooth transition function to calculate the right and left total GRFs (Karatsidis et al, 2016). Foot velocity is used to detect foot contact with the ground. For the foot to establish contact with the ground, the contact condition requires that the toe speed V_{toe} drops below a threshold V_{th}. The foot loses contact with the ground when the toe speed is larger than that threshold and the heel acceleration is negative. The velocity threshold protects against noise in the measured toe speed. The condition on heel acceleration enforces an assumption that the heel reaches maximum velocity at toe-off. The representation of the contact detection algorithm is described in Figure 4.

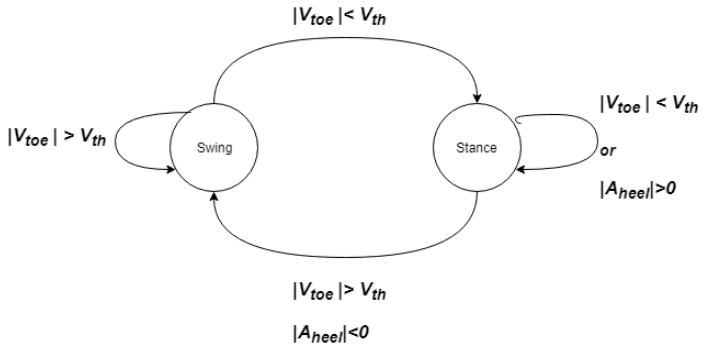

Figure 4: Threshold velocity-based contact detection algorithm (Karatsidis, 2016)

2.5.2.4 Previously Developed GRF Estimation Algorithm

The GRF estimation algorithm was previously developed as part of the biomechanical assessment tool (Diraneyya, 2019) developed in the research program that includes this thesis. The estimation algorithm is based on estimating the ground reaction force as a single force under double stance. Under double stance, finding 2 components of the GRFs is an indeterminate problem. The total GRFs is calculated using inverse dynamics (Vaughan et al, 1982). Then a contact detection algorithm is developed that predicts foot contact with the ground (Karatsidis et al, 2016). Lastly, an optimization algorithm is utilized to estimate the GRFs breakdown between right and left foot.

The optimization algorithm minimizes the squares of the net joint moment magnitudes for the joints along the closed loop from the right to the left foot, namely the right ankle M_{ra}, right knee M_{rk}, right hip M_{rh}, left hip M_{lh}, left knee M_{lk}, and left ankle M_{la} (Diraneyya, 2019).

2.5.3 Pressure Insoles

Pressure insoles are a wearable wireless device that is utilized to determine real time monitoring of plantar pressures. These sensors have a very feasible utilization for health surveillance, injury prevention and athlete training (Tao et al, 2020). There are 3 types of pressure insoles: capacitance, piezoresistive, and piezoelectric devices which play a role of converting the external pressure into electrical signals. Compared to other transmission mechanisms, capacitive pressure sensors have a simpler structure, lower power consumption, and an easy area fabrication (Li et al, 2018).

A lot of effort has been put into improving their sensitivity, increasing their detection range, as well as improving their sampling rate. Pressure insoles are a good substitute to replace GRF estimation as they are robust wearable sensors that dodisturb the worker during a task. They are lightweight, portable, and wireless.

2.6 Conclusion

This chapter reviewed the various issues of ergonomic assessment in the construction industry. It has also described the current state of the art ergonomic risk assessment methodologies that are being implemented. The chapter also reviewed the state-of-the-art sensors that are being deployed in the ergonomic and biomechanical risk assessment tools which is leading to automating biomechanical

analysis. IMU systems have a great potential to be used further in biomechanical risk assessment, as they are lightweight, do not interfere with the workers' productivity and present accurate kinematic estimates. However, other sensors such as pressure insoles will play a role of improving established biomechanical models as they serve to not only provide real time GRF data, but they will help to generate a synchronized system that will enable to enhance the robustness of the biomechanical model and add additional features to the existing model. The following chapter will discuss the research methods that were implemented in this study and will elaborate on not only the importance of pressure insole features in material manual handling tasks, but also on how both IMU motion capture data and pressure insole readings are utilized in the biomechanical model.

Chapter 3
Hand-load Weight Detection

This chapter presents a feasibility study that focuses on detecting the 3 concrete masonry units (CMU) weights or block types during gait. This chapter will focus on: (1) detecting CMU blocks using bilateral load carriage and (2) discuss the results of unilateral load carriage using the same CMU blocks. The ground reaction forces (GRFs) are imported from the Moticon-ReGo pressure insoles as a .txt format and exported to MATLAB where a weight detection algorithm was deployed. This chapter is an exploratory study that serves to inform on the main experiment depicted in Chapter 4.

3.1 Introduction

Determining the amount of load carried during loaded gait is useful, because the vertical component of the GRF provides information about the mechanical stress (Piscoya et al., 2005), which provides indications for conditions such as lower back pain or osteoarthritis. The anteroposterior (A/P) component of the ground reaction force provides indications on shear stress, which provides insight into friction between the foot and the shoe or between the shoe and the ground. (Chang et al., 2011). However, the GRFs do not provide any information about where the forces are being applied on the foot (Castro et al, 2014). Furthermore, the value of the GRF is affected by the variation of the gait cadence when carrying a load. Castro et al. (2015) performed a study on the effect of gait cadence on the ground reaction forces and plantar pressures during load carriage. The load of 20 kg was inserted in a backpack, and the plantar pressure was determined using a pressure insole. Loaded gait was distinguished from the unloaded gait irrespective of the gait cadence level in the metatarsal head area in particular metatarsals 1, 2, and 3 of the leg. The aim of this chapter is to develop a weight detection algorithm based on the concept.

3.2 Moticon ReGo Specifications

The pressure insole is composed of 22 sensors per side foot (Moticon, 2021). The sensors are composed of:

- 16 pressure sensors
- 3 acceleration sensors
- 3 angular rate sensors

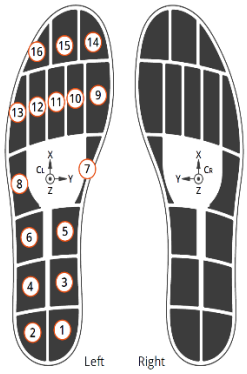

Figure 5: Pressure sensor location. Numbers represent the sensor numbering and $C_{L/R}$ represents the center of coordinate system (Moticon, 2021)

The sensor specifications are as follows: (For further information on the sensor power ratings or advanced sensitivity settings for the acceleration sensor, angular rate sensor or the gyroscope, refer to the LSM6DSL ST module).

3.2.1 Pressure Sensors:

The type of the pressure sensors found in these pressure insoles are capacitive plantar sensors. Their range is from 0-50 N/cm^2. The resolution of the sensors is 0.25 N/cm^2. The pressure sensor geometry has sensor coverage that varies over size, due to areas for wiring, electronics and certain locations that do not scale with size.

3.2.2 Acceleration Sensor

The acceleration sensor is a 3-axis sensor which is localized in the origin of the coordinate system of every side. The range of the sensor is ±16 g with a resolution of 0.488 mg/LSB.

3.2.3 Angular Rate Sensor

Like the acceleration senor, the angular rate sensor is localized at the origin of the coordinate system. The range of the sensor is ±2000 dps with a resolution of 70 mdsp/LSB.

3.2.4 Modes of Operation

There are 4 operation modes of the sensor such as the preview mode, live mode, record mode, and transmit mode. The preview mode transmits fixed basic sensor data wirelessly; the live mode transmits

sensor data wirelessly to a desktop, the record mode is an on-board sensor data recording, and the transmit mode wirelessly transmits data from on-board memory to endpoint.

3.2.5 Data Acquisition

The pressure insole has 4 sampling rates for data acquisition, which are 10, 25, 50 and 100 Hz respectively. The data channels are the timestamp, 16 pressure channels, 3 acceleration channels (X, Y, and Z directions), 3 angular velocity channels, 1 total force channel, and 2 center of pressure (COP) location channels (along x and y direction). Note that these channels are for 1 pressure insole and not for the pair. The sampling rates and the data channel combinations can be selected from the Moticon Science Mobile App (Moticon, 2021). Each data channel can be activated and deactivated manually. The right and left sampling sensors are time synchronized.

The segment coordinate system of the right insole sensor is a right-hand CS whereas the left insole sensor has a left-hand CS. The mode of operation the pressure insole is described in Figure 6.

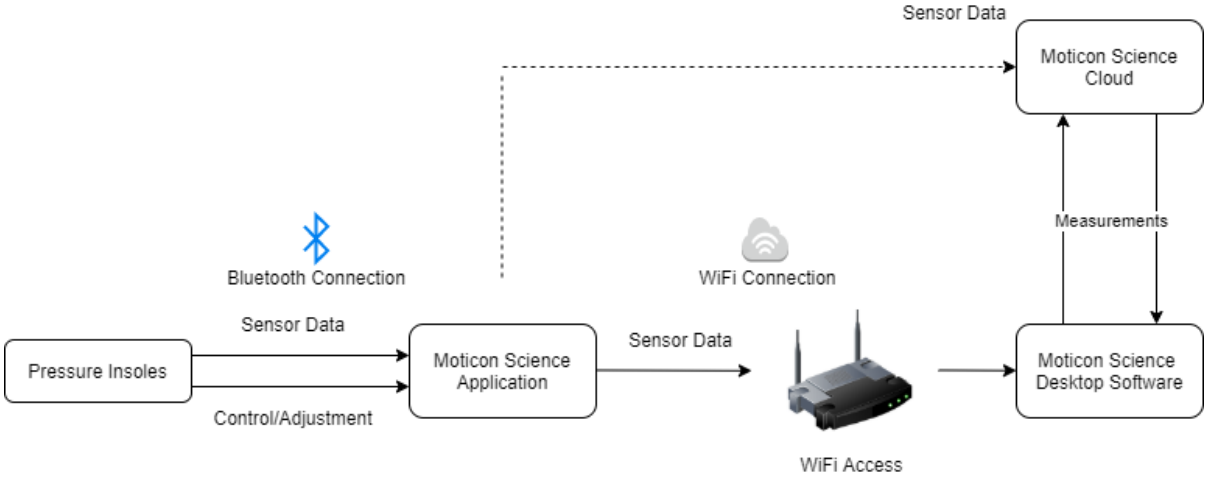

Figure 6: Mode of operation of Moticon pressure insole

Prior to the use of the pressure insole, the system must be calibrated based on the weight of the participant. The calibration process is the following:

1- Insert the participant's weight on the Moticon Science Application.
2- Walk slowly to initially calibrate the CPS.
3- Stand still.

4- Rock the body forward and backward while standing in the same location to calibrate the front and rear CPS patches.

5- Rock the body sideways to calibrate the proximal and the lateral CPS patches.

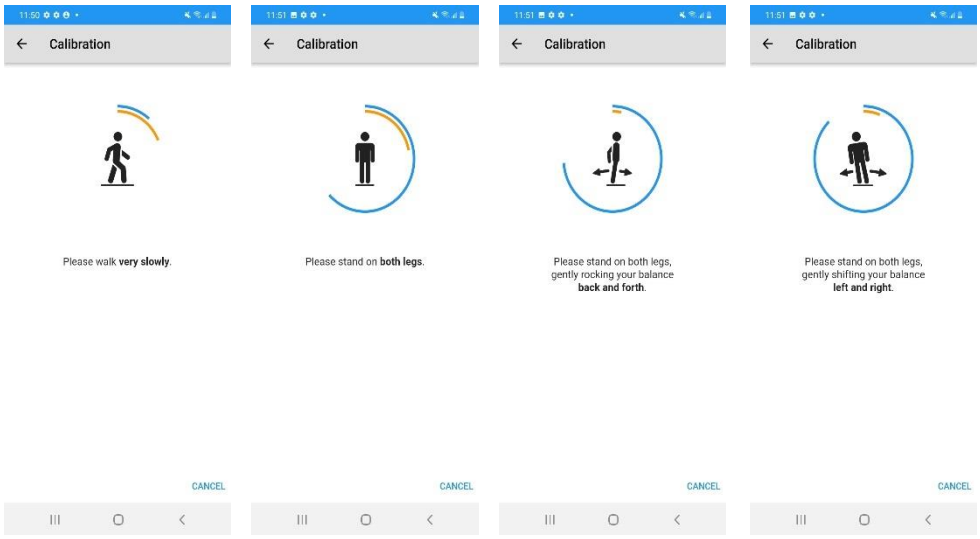

Figure 7: Pressure insole calibration methodology (Moticon, 2021)

3.3 Methods

In the pilot experiment, one subject (male, height: 177 cm, 86 kg) performed 4 tasks, each with one trial, while the right and left GRFs was being measured using a pair of Moticon-ReGo pressure insoles with a sampling rate of 100 Hz. The subject performed 5 slow gait cadence cycles in a straight line while carrying 3 different CMU blocks with both hands. The subject also needed to walk with slow cadence as well as with no weight as a control.

Table 1: CMU block weights

Types of CMU blocks	CMU 1	CMU 2	CMU 3
Weight (in kg)	8	16	24

The first task involved the subject standing in a double stance posture, with no weight, then when instructed, the subject proceeded to move 5 gait cycles in a slow cadence. At this time the pressure insole was recording GRF data through the Moticon Science mobile application that was carried by the

experimenter. The second, third and fourth tasks involved material manual handling where the subject carried CMUs 1,2, and 3 and performed the same exercise done with task 1.

3.4 Data Processing

Pressure measurements have been extracted directly from the pressure insoles and were directly converted into v-GRFs through the following procedure:

- The CPS patch sizes change with the change of size of the pressure insole. To compute the GRF on a pressure sensor, the extracted pressure measurement should be multiplied by a size correcting factor A_i.
- When computing the force over a pressure patch, the values of the pressure measurements must be multiplied by a correction factor p_i to account for the surrounding area of each pressure sensor with the change of the size of the insole. Furthermore, the CPS coverage of the pressure patch is not 100%, this is to account for wiring, electronics, antenna, and borders which do not scale through size. However, the sensor coverage varies with respect to pressure insole sizes as the wiring, electronics, antenna, and borders do not scale with size.

Consequently, the total force F from the pressure values P_i is denoted by the following relationship:

$$GRF_{R,L} = \mathrm{SC}_j \sum_{i=1}^{16} f_i \cdot k_{i,j} = \mathrm{SC}_j \sum_{i=1}^{16} P_i \cdot A_i \cdot k_{i,j}$$

Where:

- f_I is the CPS force of the pressure patch i
- k_i is the scaling factor for the pressure patch per size of the insole j
- SC_j is the surface coverage of the CPS
- P_i is the value of the CPS pressure measurement N/m^2
- A_i is the area of the pressure patch.

The values of the scaling factors, surface coverages and the areas of the pressure patches are displayed in Appendix C.

To detect the weight of the carried CMUs, the following method was utilized:

1- The local right and left v-GRF of the metatarsal head (anatomically they are the first and second metatarsal bones, and the proximal phalanx great toe) which were the pressure patches 9,10 and 14. The metatarsal v-GRF is represented in the following relation:

$$vGRF_{metatarsal_{R,L}} = \mathrm{SC}_j \sum_{i=9,10,14} f_i \cdot k_{i,j} = \mathrm{SC}_j \sum_{i=9,10,14} P_i \cdot A_i \cdot k_{i,j}$$

2- Since the gait cycle and the cadence were not the same throughout the tasks, a normalized force-time integral per gait cycle was plotted for the right and left v-GRFs for each task. This method eliminates the limitation of cadence variation of all gait cycles irrespective of the method utilized.

3.5 Results and Discussion

The plots of the normalized force-time integral for the combined pressure patches 9,10, and 14 of the right and left pressure insoles are represented in Figure 8 and 9 respectively. The variation of the metatarsal head ground reaction force time integral of each leg was measured with respect to the step.

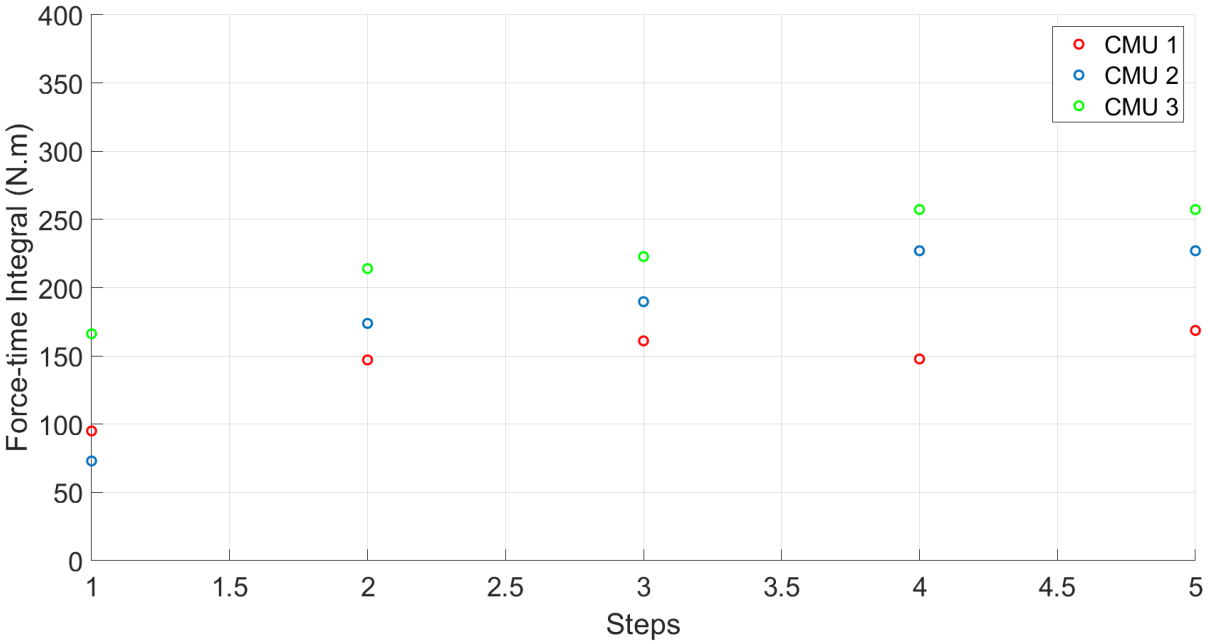

Figure 8: Right v-GRF$_{metatarsal}$ time integral as a function of a step

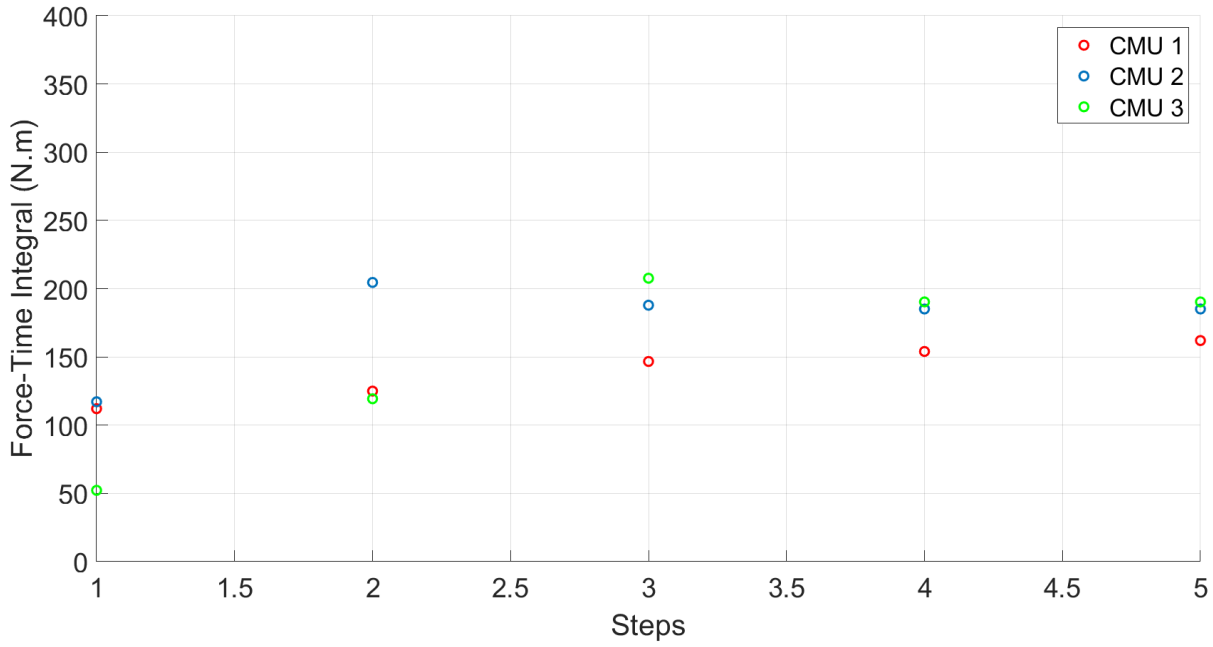

Figure 9: Left v- $GRF_{metatarsal}$ time integral as a function of a step

Since loaded or unloaded gait is not repeatable and the cadence of the tasks are not the same, the force-time interval was done per gait cycle and was normalized with respect to time. What can be noted from both graphs was the ability of the methodology to detect the various CMU block weights throughout the tasks, as the CMU-3 block had the highest magnitude, compared to CMU-2 and CMU-1 respectively. This was clearly shown in Figure 8. However, in Figure 9, the first 2 gait cycles for the right leg did not detect the CMU-3 block to be the highest in magnitude, where the main reason is that the participant hasn't reached steady state gait yet with the CMU in hand. When analyzing gait cycle 3 onwards, the force-time integral was consistent, and the methodology was able to detect all the CMU blocks.

3.6 Unilateral Hand Load Weight Detection

3.6.1 Introduction

A separate study was conducted to determine if it is possible to determine the CMU block weights during unilateral load carriage. During asymmetrical loading, the body implements compensatory adjustments, such as an increase in contralateral trunk flexion (Fowler et al., 2006; Crowe et al., 1993). This results in a higher hip abduction torque due to the greater contraction of the gluteus medius

(Neumann et al., 1996; Neumann et al., 1985). However further understanding of the contributions of the lower segments is unknown, and this hip torque imbalance may be associated with asymmetric load affecting lower limb coordination (Becreanu et al.,2015; Matsuo et al., 2008).

To validate this, Zhang et al. (2009) performed a study to examine the effect of unilateral load carriage on postures and gait symmetry in GRF during walking. They reported that the load conditions included no load, dumbbell (10 and 20% body weight, BW) held in right and left hand, respectively and were instrumented with 12 marker clusters placed over each foot, shank, thigh, forearm, and upper arm bilaterally, sacrum and the sternum for kinematic data collection.

They reported a progressive trunk bending towards contralateral or ipsilateral side, accompanied by decreased stride width as the load in hand increased. A significant increase in the maximum values of contralateral hip adduction and contralateral shoulder abduction was noticed when the trunk bent ipsilaterally, and a decreased ipsilateral hip adduction were seen as loads increased. The symmetry index (SI) in GRF_{medial} was significantly increased while increasing the load weight; the SI in $GRF_{medial/lateral}$ under 20% BW conditions was significantly greater than that under 0 and 10% conditions.

3.6.2 Methods

The methodology and the data processing of this study were the same as the ones utilized in the previous section. The experimental procedure was adjusted in the following manner:

One test subject (male, height: 177 cm, 86 kg) performed 4 tasks, each with one trial, while the right and left GRFs are being measured using a pair of Moticon-ReGo pressure insoles with a sampling rate of 100 Hz (Moticon, 2021). The subject performed 5 slow gait cadence cycles in a straight line while carrying 3 different CMU blocks. The subject also walked with slow cadence as well as with no weight as a control. As the test subject is left hand dominant, they carried these blocks with their dominant hand.

3.6.3 Results

The normalized force-time integral for the combined 9,10, and 14 metatarsal patches of the pressure insoles are presented in Figure 10, and 11 for the right and left feet v-GRFs as functions of the step number, respectively.

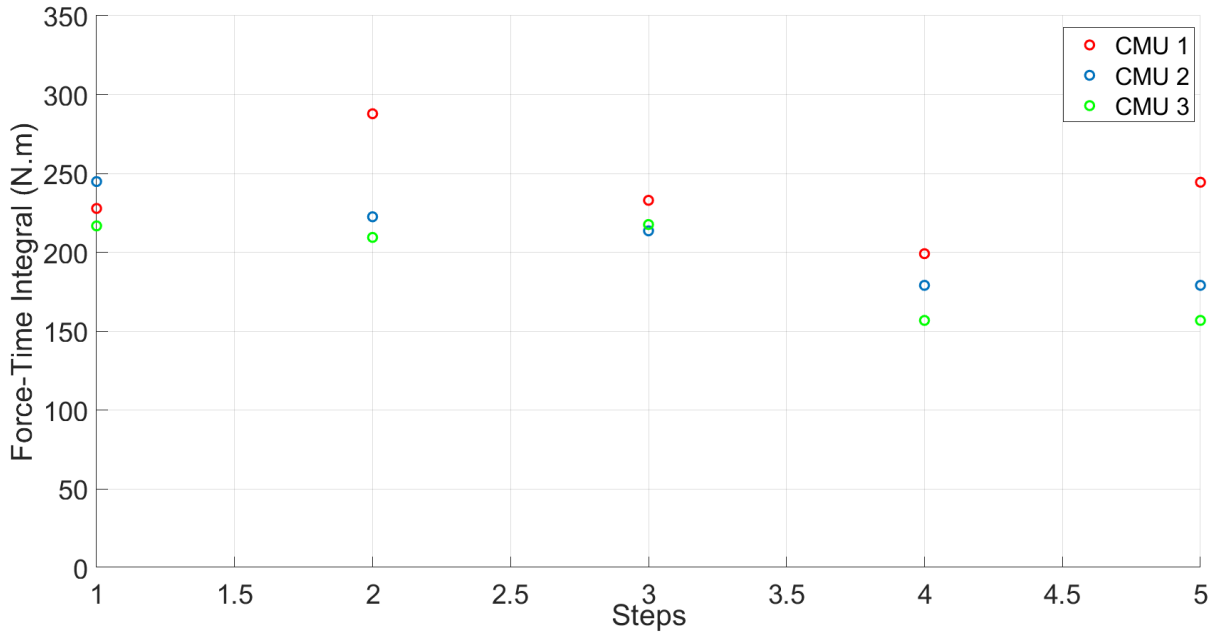

Figure 10: Right v-GRF metatarsal time integral as a function of steps

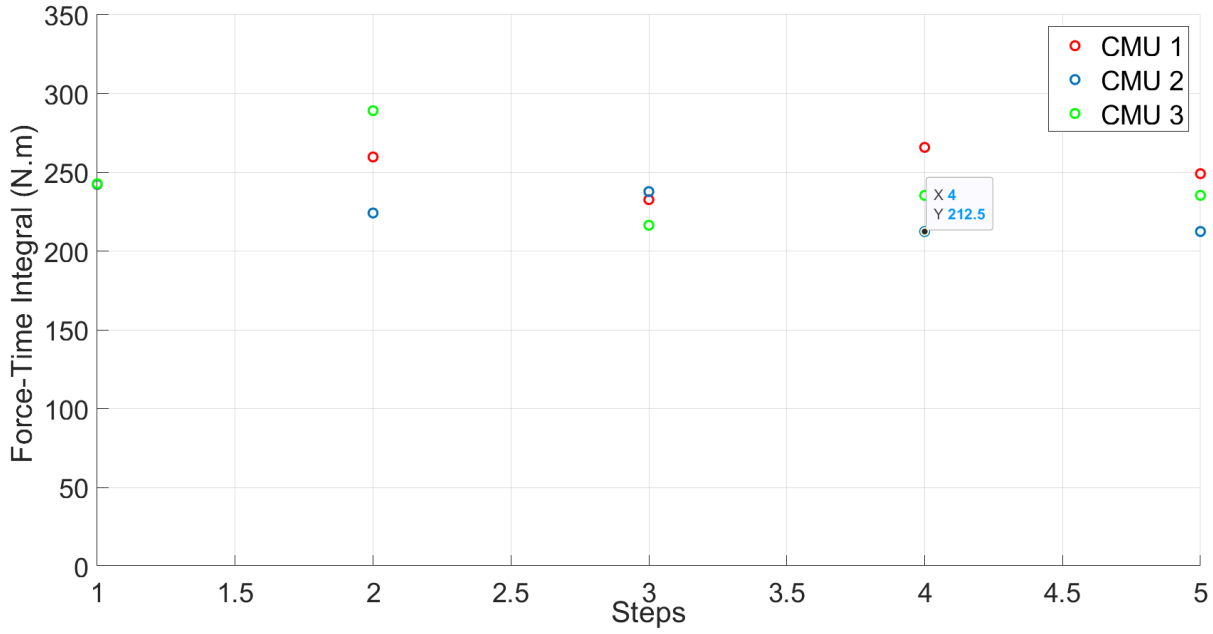

Figure 11: Left v- GRF metatarsal time integral as a function of steps

The unilateral carried v-GRF FTI had no observable direct trend to identify the CMU block weights in either the right or left leg. Furthermore, what can be deduced from the plots is that despite the higher values of the FTI measurements of all the weights compared to the ipsilateral v-GRF FTI values, the contralateral v-GRFs FTI of the lower CMU blocks are higher than the higher CMU block. A potential explanation for that was that the upper torso was losing balance and the body was tipping over towards the load. However, kinematic analysis is required to verify this hypothesis. Hence, given the overall objectives of this study, it was considered unfeasible to further pursue unilateral hand load weight determination during material manual handling.

3.7 Limitations

Some of the limitations of the study were the following:

1- The study has utilized a limited number of steps. More steps are suggested to validate that unilateral load carriage cannot be determined by GRF measurements only during steady state gait.

2- It is suggested to include more steps in the representation of the force-time integral results to ensure that steady state gait has been reached.

Chapter 4
Development and Validation of a Hand-load Weight Detection Method

4.1 Introduction

This aim of this chapter was to validate the feasibility study of detecting CMU block weights while carrying with both hands. The pilot study has shown that for low cadence and traversing in a straight line, the detection algorithm using the localized metatarsal v-GRF measurements has enabled the detection of all CMU blocks. Nonetheless, this has been performed for one subject only undergoing 3 tasks with one trial per task. Hence this validation study examined the performance of the CMU block detection algorithm for various test subjects under different gait paths.

4.2 Experimental Setup

A total of 7 male participants participated in this validation study. The subjects were recruited from within University of Waterloo campus. The study was approved by the University of Waterloo Office of Research Ethics and the Safety Office. The participants' height and weight are summarized in Table 2:

Table 2: Participants' height and weight and pressure insole size selection

Participant	Gender	Height (cm)	Mass (kg)	Pressure Insole Size
1	Male	180	118.4	6
2	Male	176	80.4	6
3	Male	178	88	6
4	Male	183	98	6
5	Male	180	74	7
6	Male	170	82	6
7	Male	177	92	6

The participants were asked to perform 4 gait paths while carrying the 3 CMU blocks as steadily as possible (refer to Table 1 for CMU weights).

1- Walk along a 12.5 m line, pause, and rotate in a counterclockwise direction, and then return to the starting position.
2- Walk in a circular path with a radius of 4 m in a counterclockwise orientation.
3- Walk along a square path with a side length of 6.25 m in a counterclockwise fashion,
4- Repeat the gait paths with no weight in the same suggested orientation.

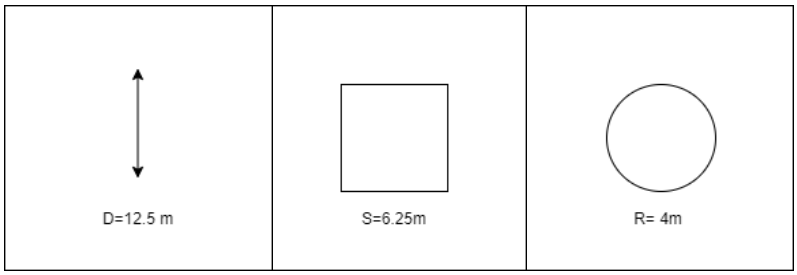

Figure 12: Gait paths

4.3 Data Processing

The experimental procedure and data processing were identical to those of the experimental procedure and data processing listed in Chapter 3. The performance of the detection algorithm performance was depicted by utilizing a confusion table. There are 7 participants who performed 9 tasks through which the number of steps were recorded for every single task. The no-weight gait path is done as a reference for the participants and are excluded from the below count as it is considered as a calibration for each trial. The right and left steps are summed together; as the performance metric of the methodology applies to both legs irrespective of the side.

As there are 3 fixed CMU block weight, the computed FTI variation between the three CMU blocks is sufficient to detect the weights.

4.4 Results and Discussion

The total detection table is shown in Table 3:

Table 3: Table for CMU detection classification

n=1908	**Weight Detection Predicted/step**	**Total**
Actual CMU block detection/step	1683	1683

No CMU Block Detection	225	225
Total	1908	1908

Then, a confusion table was utilized to dissect the weight detection algorithm performance across all CMU blocks. The table details the predicted detection of all CMU blocks compared to the actual detection of all CMU blocks across all steps. The results are shown in Table 4

Table 4: Confusion matrix on CMU detection algorithm

N=1908 (total number of steps)	Detected CMU 1	Detected CMU 2	Detected CMU 3	Total
Actual CMU 1	597	39	0	636
Actual CMU 2	53	564	19	636
Actual CMU 3	0	114	522	636

Based on the weight detection method that is described in Chapter 3, it is expected that the all the CMU blocks are to be detected. The percentage error in insole detection is 11.8%.

Table 4 showed that the steps that noticed no detection in the CMU weights are the skewed towards the 16 kg CMU-2 and the 24 kg CMU-3. Under nominal conditions, weight detection of the higher weights should be much clearer. There are several potential reasons that may cause such phenomena which include the following:

1- The dimensions of the CMU 2-3 are larger than that of CMU-1, hence potentially, the block is hitting the legs throughout the gait cycle and consequently causing the participant to do a premature toe-off following the heel strike. This has been noticed as the metatarsal pressure patches have shown a decrease in v-GRF magnitude in some of the steps compared to the rest in some of the lifts.

2- Another reason for the decrease in values of the v-GRFs across the metatarsal head area was noticed in some of the participants' torso motion. Some of the participants displayed swaying in the upper torso that might generate a reaction moment that will cause the measurement of the GRFs for weight detection to be inconclusive (Fowler et al., 2006; Crowe et al., 1993).

Synchronized motion capture techniques in accordance with v-GRF data must be implemented to quantitatively validate this hypothesis

3- Fatigue may play a role in gait deterioration and a significant decrease of the single support time of the gait cycle, which does not allow the sensors to fully capture the plantar pressure of the foot.

Chapter 5
Incorporating Pressure Insoles into the Ergonomic Assessment Tool

This chapter introduces a previously developed IMU-based biomechanical ergonomic assessment tool that was developed by Diraneyya (2019). The introduction of this chapter will serve as a review of the methodology of Diraneyya (2019), the technology utilized, and their respective necessary assumptions. The second section of this chapter will describe the methodology adopted to incorporate the pressure insole into the biomechanical model.

5.1 Introduction

As discussed in chapter 2, the main challenge of current ergonomic assessments is the lack of a quantitative tool that evaluates whole body kinematics on the construction site. One way to measure whole body motion data is from utilizing wearable IMU suits that derive body posture in 3D such as determining body joints and joint angles. However, as seen in chapter 2, IMU-based assessment tools have been developed further to estimate the joint forces and moments of an individual while doing a task. Hence, Diraneyya (2019) developed an IMU-based ergonomic assessment tool that will compute the joint forces and moments in addition to the required kinematic analysis (Diraneyya, 2019). The methodology is based on the following technical criteria:

1- Extract raw data from inertial measurement unit's (IMU) user interface which are then parsed to obtain the kinematic criteria such as the segment orientations and the sensor locations, accelerations, orientations, and angular velocity.

2- Utilize the participant's gender, height, and weight to perform body segmentation (Dumas et al., 2007) to determine the segment characteristics of each segment such as the segment mass, the center of mass location with respect to the local frame of reference, and the inertia tensor.

3- Gather all types of external forces that are acting on the participant, such as hand loads (used as either unilateral load carriage or bilateral load carriage), external contact forces and ground reaction forces. Ground reaction forces estimation methodology and breakdown can be revisited in section 2.5.2.4.

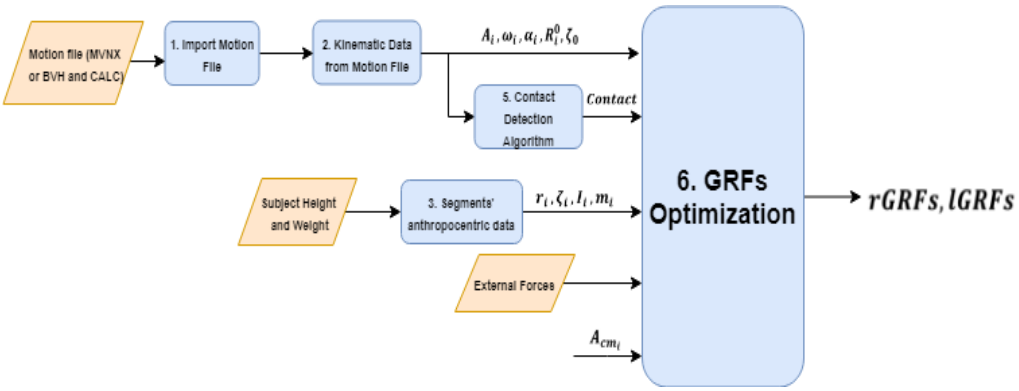

Figure 13: Previous GRF estimation structure (Diraneyya, 2019)

4- Compute the lower back contact forces and the lower back compression forces as well as the joint forces and the moments that are computed from the inverse dynamic solver.

The main issue with the current model is that the v-GRFs are being estimated from the kinematics data of the segmented rigid bodies. Furthermore, the lateral GRFs, that are the mediolateral and the anteroposterior GRFs are estimations from the estimated v-GRF. This makes a major external force input into the inverse dynamics as an estimate rather than a deterministic measurement. However, the pressure insoles generate v-GRFs output, i.e., it is still necessary to perform body segmentation to be able to compute the $GRF_{M/L}$ and $GRF_{A/P}$ components. Hence, the aim of this chapter is to incorporate the pressure insole data into the inverse dynamics model and estimate the mediolateral (M/L) and anteroposterior (A/P) through GRF optimization. This improves the fidelity of the model and decreases the complexity of the model inputs.

5.2 Methods

5.2.1 Full Body Model

The human body is modelled as a multi-body system with joints connecting every segment. The model that will be utilized in this study is a 15-segment full body model. It is comprised of the following: the pelvis, torso, right and left upper arms, right and left lower arms, right and left hands, right and left upper thighs, and right and left lower legs, and right and left feet. The pelvis is taken as the root segment.

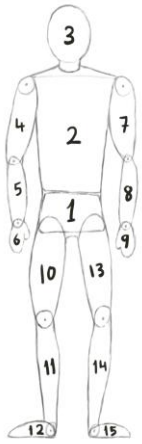

Figure 14: 15-Segment full body model

Note that the higher-order lower body array must be constructed to facilitate numbering the segments with respect to their proximal segment. A top-down approach is used for the upper limb chain and a bottom-up approach is utilized for the lower limb chains. Based on the numbering of the full model mentioned in Figure 14, the higher-order lower body array is established in Table 5.

Table 5: Higher-order lower body array

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$L(i)$	0	1	2	2	4	5	2	7	8	1	10	11	1	13	14
$L^2(i)$	0	0	1	1	2	4	1	2	7	0	1	10	0	1	13
$L^3(i)$	0	0	0	0	1	2	0	1	2	0	0	1	0	0	1
$L^4(i)$	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0
$L^5(i)$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The inertial parameters such as segment length, mass, inertia tensor for each segment is obtained through the literature presented by Dumas et al., (2007). To use the previous parameters, the model follows Dumas et al., segment definitions and their respective anatomical landmarks are shown in Figure 15:

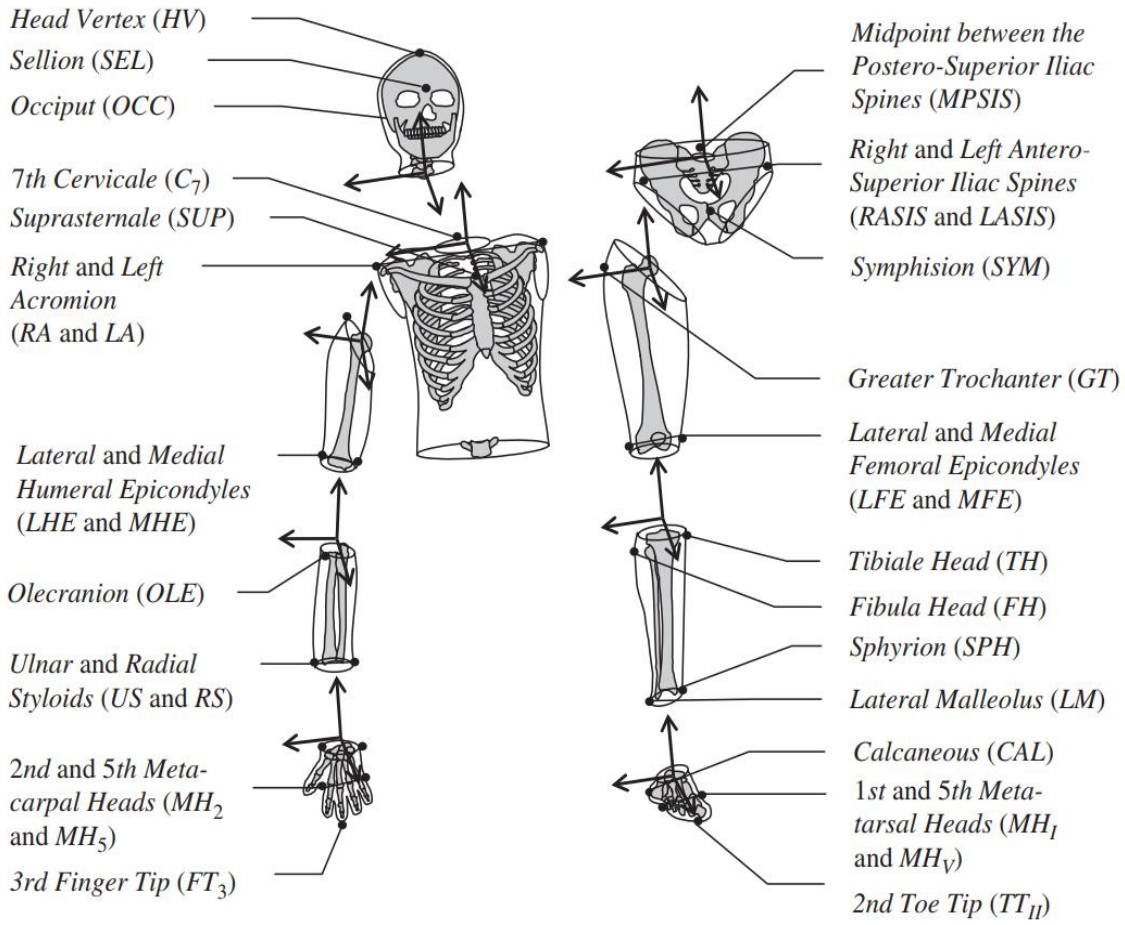

Figure 15: Anatomical landmarks (Dumas et al., 2007) (used with permission)

The origins of the local frames of each segment are placed at the centers of the segment proximal joint except for the root segment frame, pelvis, whose origin is at the Lumbar Joint Center (LJC). The axes of the local frames are labeled following the recommendations of the International Society of Biomechanics (Wu et al.,2002; Wu et al.,2005). Table 6 shows the body segment lengths and their corresponding local frame origins. Note that the coordinate systems have the Y-axis pointing through the length of the segment (pointing cranially) and the Z-axis pointing laterally to the right. The X-axis is the cross-product resultant.

Table 6: Body segments' lengths and frame origins

Segment	Length	Origin
Pelvis	LJC to HJC projection in the sagittal plane	LJC
Torso	CJC to LJC	LJC
Head and Neck	CJC to HV (Head Vertex)	CJC (Cervical Joint Center)
Arm	SJC to EJC	SJC (Shoulder joint Center)
Forearm	EJC to WJC	EJC (Elbow Joint Center)
Hand	WJC to midpoint between MH2 and MH5	WJC (Wrist Joint Center)
Thigh	HJC to KJC	HJC (Hip Joint Center)
Leg	KJC to AJC	KJC (Knee Joint Center)
Foot	AJC to midpoint between $MH_1$1 and $MH_1$5	AJC (Ankle Joint Center)

5.2.2 Model Kinematics

Concerning the locations of the local axes, and their definitions, and directions, please refer to Mr. Diraneyya's thesis for the elaborate tables (Diraneyya, 2019). As for the kinematic data, the segment angular velocities $\omega_i\ (t)$is obtained from the IMC system and is numerically differentiated to obtain the segment angular acceleration $\alpha^i(t)$. However, the acceleration $a^i(t)$is calculated from the measured IMU angular acceleration $a_s^i(t)$, the IMU position with respect to the COM known as $r_s^i(t)$ and the segment angular velocity $\omega^i\ (t)$ defined by the following relation:

$$a^i(t) = a_s^i(t) + \alpha^i(t) \times r_s^i(t) + \omega^i\ (t) \times (\omega^i\ (t) \times r_s^i(t))$$

5.2.3 Inverse Dynamics

Since each single segments' equation of motion can be used to evaluate the net force and moment in its proximal joint, and a top-down approach is used for the upper limbs, and a bottom-up approach is used for the lower limbs, one can be able to propagate all segments inwards toward the root segment (the pelvis) and find all the joint forces and the moments. The types of forces that are acting on any segment *i*, are the proximal joint force F^{ik}, the summation of the distal joint forces $\sum_n F^{jni}$, the total external force acting on the segment F_{ex}^i, the segment weight W^i, and the inertia force of the segment F^{i*}, the equations of motion is $F^{i*} = -m^i a^i$

Using d'Alembert Principle, the Newton's second law for any segment is designated by the following relationship:

$$F^{ik} + \sum_n F^{jni} + F^i_{ex} + W^i + F^{i*} = 0$$

The moment acting on the COM are the net moment of the proximal joint $M^{ik}$of the proximal joint *k*, the moment produced by the net force of the proximal joint around the COM $-r^i \times F^{ik}$, the sum of the net moments of the distal joints $\sum_n M^{jni}$, the moment produced by the net forces of the distal joints around the COM $\sum_n[(\zeta^{jn} - r^i) \times F^{jni}]$, the total external moment M^i_{ex} on the segment, the total external force acting on the segment's COM, $[(r^i_{ex} - r^i) \times F^i_{ex}]$, and the inertia moment of the segment on its center of mass M^{i*}designated by the following:

$$M^{i*} = -\left[I^i.\alpha^i(t) - \omega^i\,(t) \times (\left[I^i\right].\omega^i\,(t))\right].$$

Note that ζ^{jn} is the location of frame F^{jn} with respect to frame F^i origin r^i_{ex} is the location of the point of application of the external force.

The moment equation becomes the following:

$$M^{ik} + \left(-r^i \times F^{ik}\right) + \sum_n M^{jni} + \sum_n \left[(\zeta^{jn} - r^i) \times F^{jni}\right] + M^i_{ex} + \left[\left(r^i_{ex} - r^i\right) \times F^i_{ex}\right] + M^{i*} = 0$$

5.2.4 Lower Back Disk Contact Forces

Internal forces along the back that was used was adopted from Schultz and Anderson (1981). The model is a 3D 10-muscle model which includes the Erector Spinae (ES), Rectus Abdominus (RA), Latissimus Dorsi (LD), Internal Oblique (IO), and External Oblique (EO). The muscle model is shown in Figure 16. It is important to model and study lower back forces in particularly surrounding the L4/L5 and the L5/S1 disks, since these disks are susceptible to high levels of external load particularly in lifting tasks. MRI scanning has shown that L4/L5 and L5/S1 degenerate for workers that tend to lift heavy weights (Battié et al., 1995; Katariina et al., 1998, Katariina et al., 2000) and Alwasel et al. (2017) has found out that the L5/S1 joint was the heavily loaded joint during bricklaying.

Due to the indeterminacy of the problem to find the value of the disc forces, the Bean et al, (1988) linear optimization has been utilized by solving a first linear optimization method to find the maximum muscle intensity needed to balance the exerted moment and a second linear optimization method to find the muscle and contact forces subject to the maximum muscle intensity.

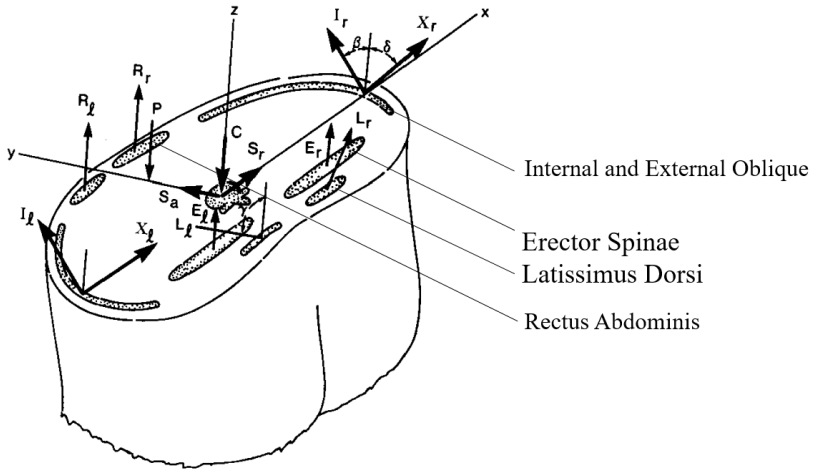

Figure 16: Lower back 10-mucle model (Schlutz and Andersson,1982) (used with permission)

5.2.5 Experimental Setup

A total of 7 male participants took part in this pilot study. The subjects were recruited within University of Waterloo. The study was approved by the University of Waterloo Office of Research Ethics and the Safety Office, Appendix A. The participants' height, weight, and insole size are listed Table 7:

Table 7: Participants' height, weight, and insole size

Participant	Height (cm)	Mass (kg)	Pressure Insole Size
1	180	118.4	6
2	176	80.4	6
3	178	88	6
4	183	98	6
5	180	74	7
6	170	82	6
7	177	92	6

Each participant was equipped with a motion capture suite and a pair of pressure insoles. The motion suit was calibrated following the procedure described in Section 5.2.6 while the insoles were calibrated following the procedure described in Section 3.2.1.5. Each participant proceeded to walk along 4 predefined as described in Section 4.2. Those gait patterns were designed to test the impact of kinematic and kinetic variations in gait during MMH on the performance of the proposed detection algorithm.

Specifically, the linear gait path examines the algorithm's performance where no changes in direction occur from motion initiation until coming back to a full stop. The circular gait path examines the impact of a constant change in direction gait path on detection accuracy. The square gait path examines the impact of abrupt changes direction of detection accuracy.

Figure 17: Experimental setup

5.2.6 Calibration of the Motion Suit

The sensor systems utilized in this study are Moticon ReGo (Moticon, 2021) pressure insoles and Perception Neuron IMC suit (Noitom Ltd, 2017). The IMC suite comprises of 17 IMU's called neurons, each equipped with a three-axis accelerometer, a three-axis gyroscope, and a three-axis magnetometer. The motion capture suite is controlled wirelessly via Axis Neuron user interface (Noitom Ltd, 2017). It is installed by fitting elastic bands containing the IMU's around major body parts, namely the head, upper torso, upper arms, forearms, wrists, lower back, thighs, legs, and feet, Figure 18.

Figure 18: Perception Neuron (Noitom Ltd, 2017)

The calibration procedure. Figure 19 calls for the subject to assume three poses, namely A-pose, T-pose, and S-pose. Their role is to determine the sensor to body alignments and body segment lengths at the beginning of the experiments. The initial steady pose suggested by can be Perception Neuron was not used since the experiment does not require sitting.

Figure 19: Calibration postures (Noitom Ltd, 2021)

The sampling rate of the Perception Neuron suit is listed as 125 Hz. However, throughout the experiment, it was found that sampling rate varied in the range 122-125 Hz. A Kalman filter and a proprietary algorithm use the magnetometer output to prevent drift in the accelerations measured by the accelerometer and the angular velocities measured by the gyroscope (Noitom LTD, 2017) Axis neuron reformulates the gathered data to a 3D human rigid body model based on the estimated 3D spherical adjustments and Euler angles in a global reference frame (Filippeschi et al., 2017; Robert-Lachaine et al., 2020; Roetenberg et al., 2009; Sers et al., 2020). This process results in a down-sampling depending on the quality of the measured kinematic data.

5.2.7 Data Processing and System Synchronization

5.2.7.1 Axis Neuron Motion Files Export

There are two files that need to be exported from Axis Neuron. One of them is the 'filename'.bvh file which it contains all the body segment orientations. This file is not parsed and needs to be processed on Matlab through an open-source code that parses the file. The other file that is required is .calc file, as it contains all the sensor locations, angular velocities, accelerations, and sensor orientations. Another code is developed to read the .calc files for processing.

5.2.7.2 *Pressure Insole Files Export*

Once data are collected on live capture mode on the mobile application, the insole data is automatically transferred to Moticon Open Go, which is the desktop application for the sensor. Hence, the raw file is exported to the Matlab directory via text export for preprocessing.

5.2.7.3 System Synchronization and Up sampling Pressure Insole Data

The first issue to tackle is the sampling rate discrepancy between both signals, where the sampling rate of the pressure insole is 100 Hz, whereas the sampling rate of the Perception Neuron is 125 Hz. The target is to up sample the pressure insole data as it is computationally much more efficient in the designed algorithm. Furthermore, due to the variability of the sampling rate of the Perception Neuron motion suite, it is not recommended to down-sample a fluctuating sampling rate to a fixed sampling rate. Through Matlab, up-sampling is done through resample code, where the function resamples the input sequence (the right and left v-GRF) at a p/q multiplied the original sampling (100 Hz in this case) (p/q ratio should give 125/100, so p=5 and q=4). The resample function utilizes an FIR (Finite impulse response) antialiasing low pass filter to the v-GRF data and compensates for the delay caused by the initial FIR filter called upon by resample. Initially, both .bvh and calc files should be preprocessed i.e., the bvh file should be parsed (code in appendix D) and the .calc file should be processed as well.

As for the synchronization, both. calc, and the v-GRF data are run through a synchronize function. The synchronize function collects the required variables from all input timetables and synchronizes them into a common time vector and outputs all synchronized data in a timetable.

5.3 Results and Discussion

This analysis is limited to 2 subjects out of the 7 participants in the experiment. Results from the other participants were excluded due to faults in the left foot IMU corrupting the motion suit measurement stream. The critical body joints during loaded gait are the lumbar joint as well as the shoulders and the elbows. Hence, the right and left shoulder and elbow net joint moments and the L4/L5 back compression forces were evaluated and are presented below. The results of unloaded gait (hands free) experiments are also presented as experiment control.

5.4 Linear Gait Pattern

The results show an increase in the net shoulder moment for both participants (Figures 20 and 21) as the weight of the CMU blocks increase from the unloaded case to those loaded and similar increase in net moment as the CMU weight increased from 8 kg (CMU-1) to 24kg (CMU-3). The large spikes observed in the left shoulder moment are spurious due to impact events between the left upper arm IMU and the body. Furthermore, since the CMUs were carried bilaterally, the magnitudes of the right and left shoulder moments are similar in magnitude showing no hand preference while carrying the load.

The elbow moments, Figures 22 and 23, increase from the unloaded case to the loaded cases and increase again in the loaded cases as the weight of the CMU increases. Similarly, since the CMUs were carried bilaterally, the magnitudes of the right and left elbow moments are similar showing no hand preference while carrying the loads.

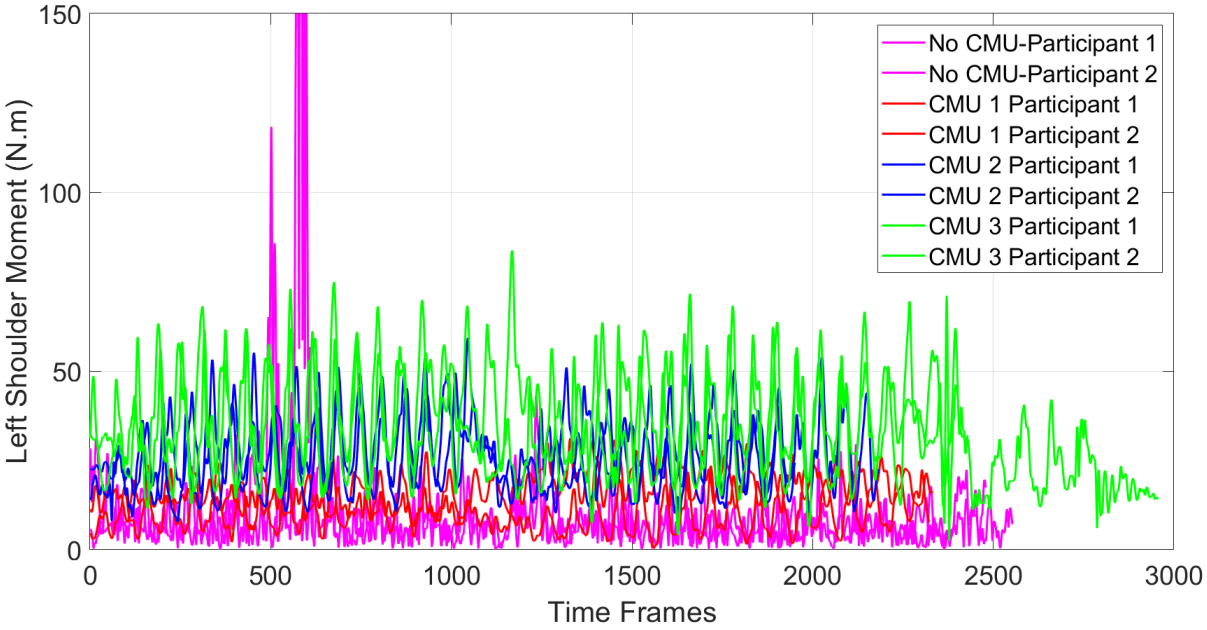

Figure 20: Net moment in the left shoulder joint

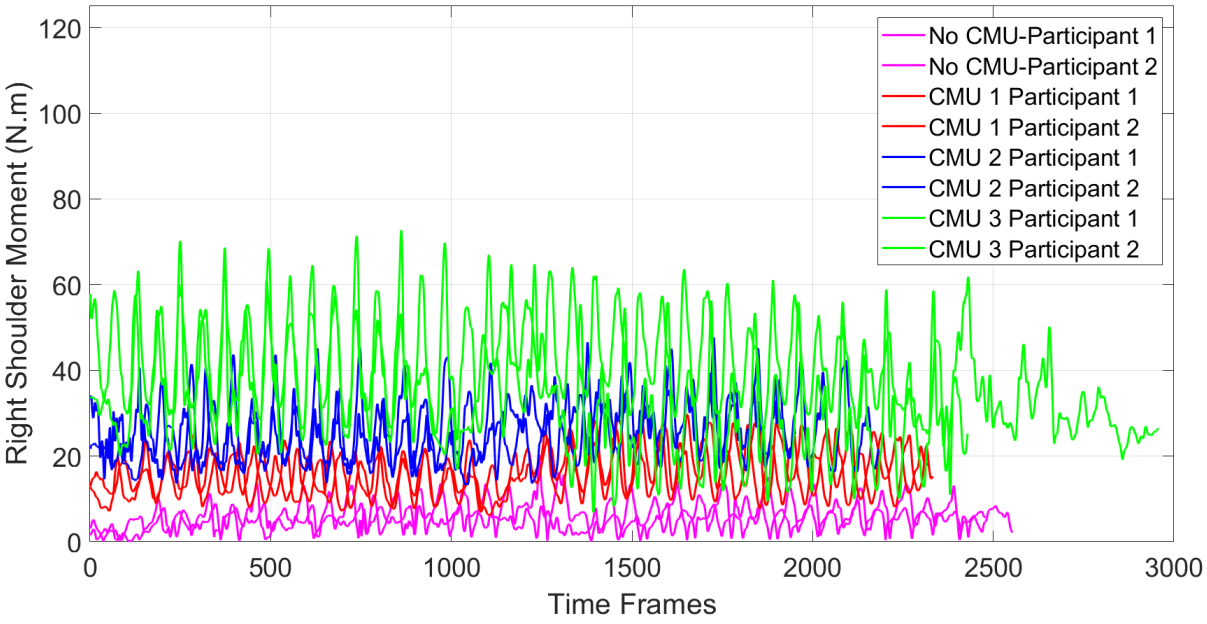

Figure 21: Net moment in the right shoulder joint

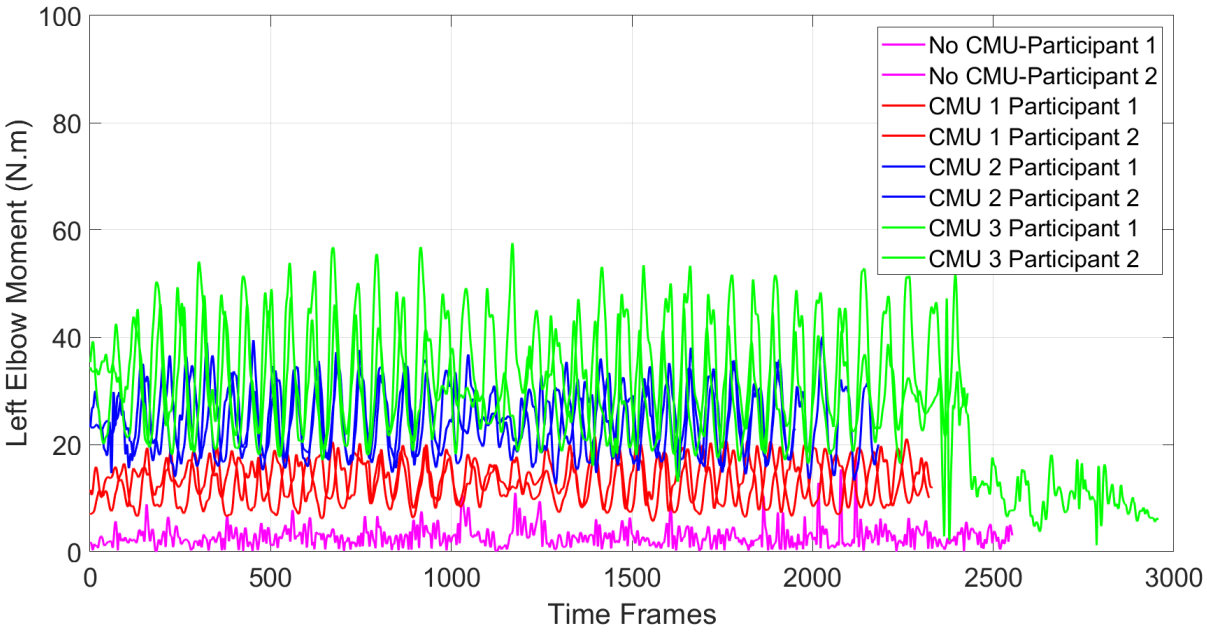

Figure 22: Net moment in the left elbow joint

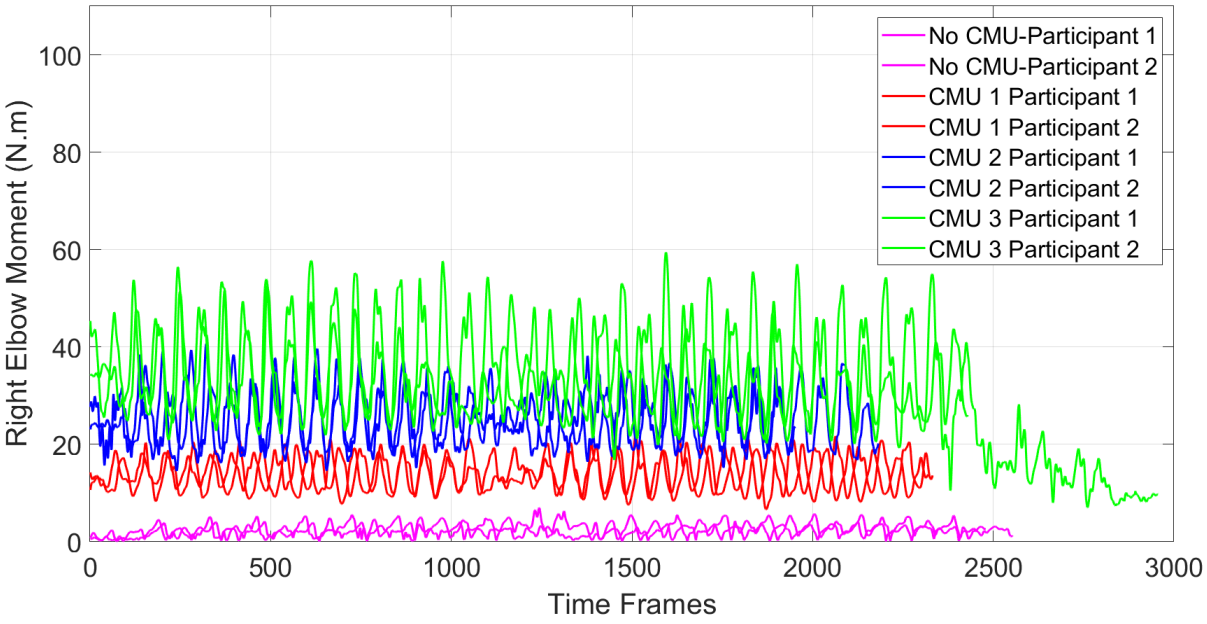

Figure 23: Net moment in the left elbow joint

An increase in L4/L5 back compression force was noticed, Figure 24, as the weight of the CMU increased from the unloaded case to those loaded and as the CMU weight increased from 8 kg (CMU-1) to 24kg (CMU-3).

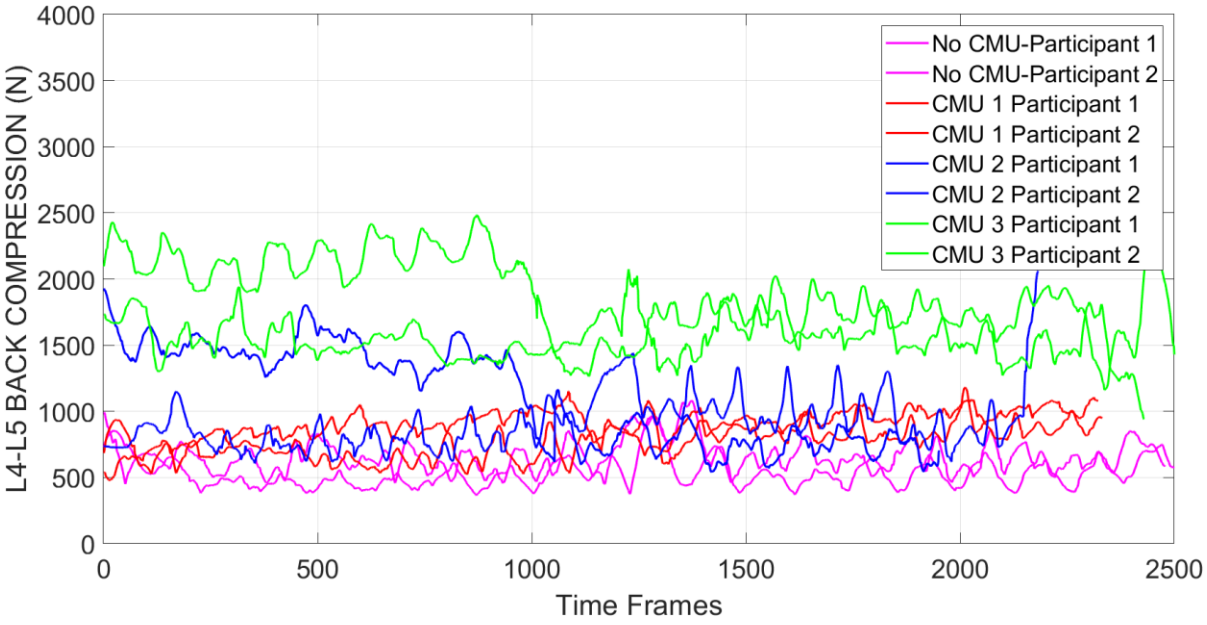

Figure 24: L4/L5 lumbar back compression force

5.5 Circular Gait Pattern

The results show an increase in the net shoulder moment for both participants (Figures 25 and 26) as the weight of the CMU blocks increase from the unloaded case to those loaded and similar increase in net moment as the CMU weight increased from 8 kg (CMU-1) to 24kg (CMU-3). The large spikes observed in the left shoulder moment are spurious due to impact events between the left upper arm IMU and the body. Furthermore, since the CMUs were carried bilaterally, the magnitudes of the right and left shoulder moments are similar in magnitude showing no hand preference while carrying the load.

The elbow moments, Figures 27 and 28, increase from the unloaded case to the loaded cases and increase again in the loaded cases as the weight of the CMU increases. The large spikes observed in the left elbow moment are spurious due to impact events between the left lower arm IMU and the body.

Similarly, since the CMUs were carried bilaterally, the magnitudes of the right and left elbow moments are similar showing no hand preference while carrying the loads.

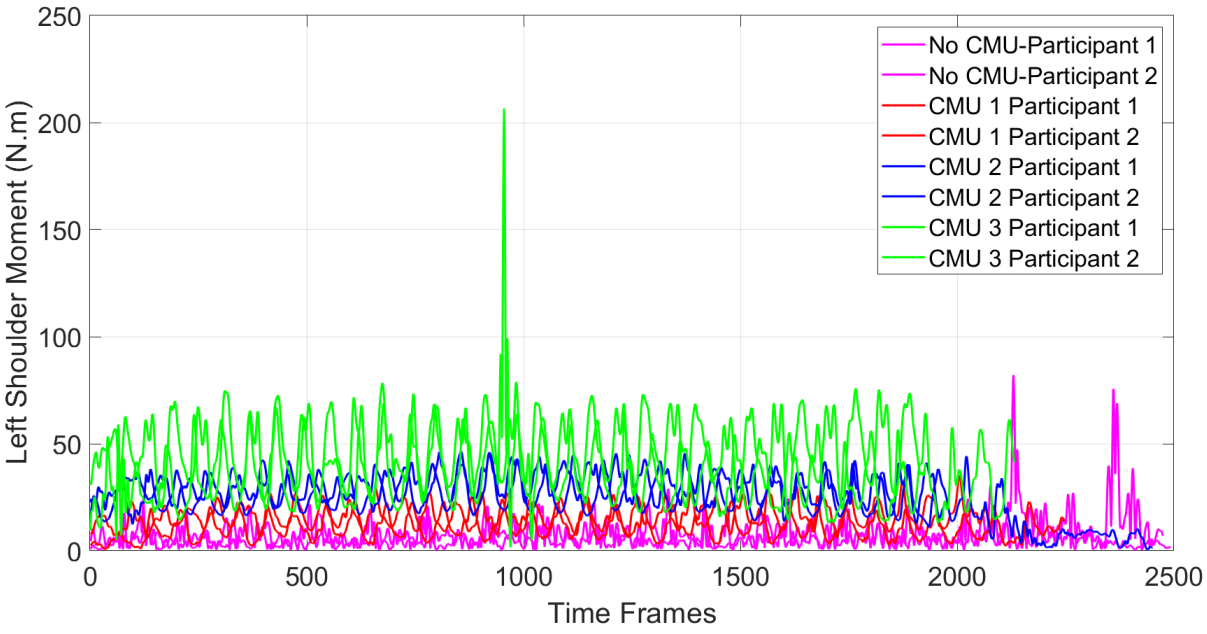

Figure 25: Net moment in the left shoulder joint

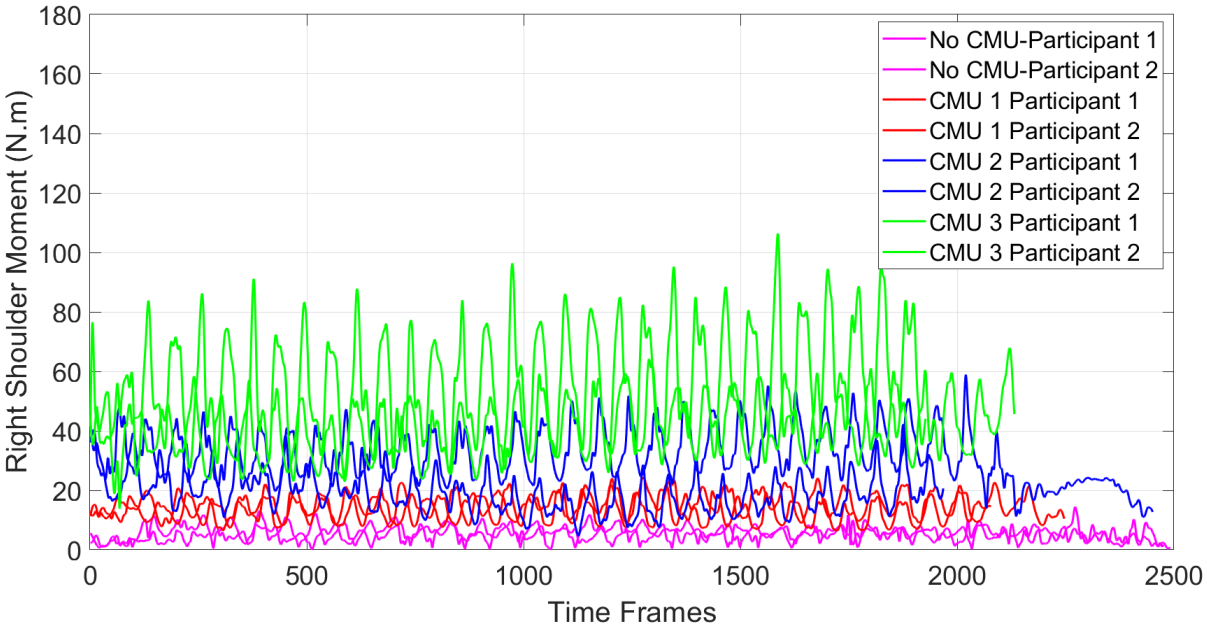

Figure 26: Net moment in the right shoulder joint

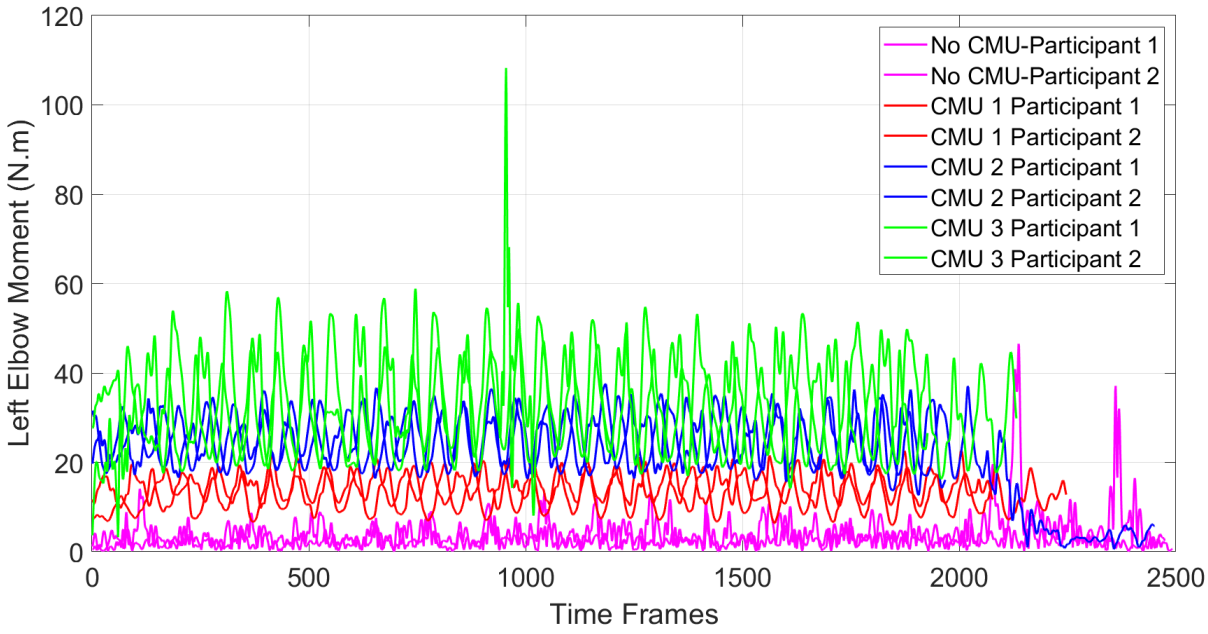

Figure 27: Net moment in the left elbow joint

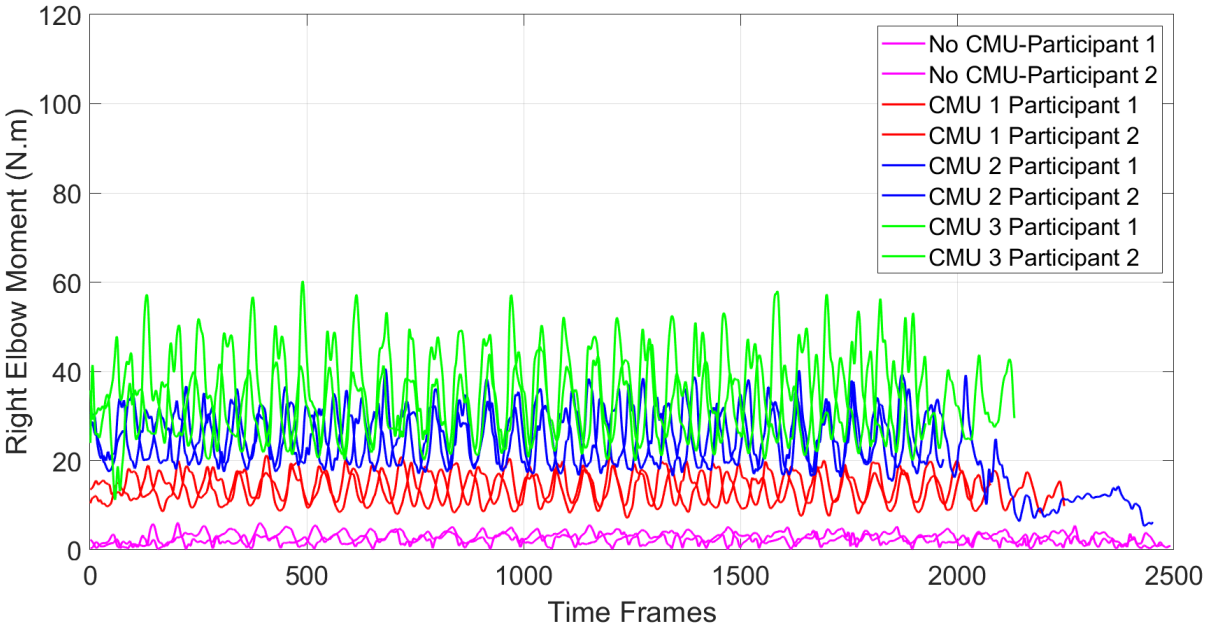

Figure 28: Net moment in the right elbow joint

Concerning the L4/L5 back compression forces, no significant trend was shown in Figure 29. Hence more experimental data gathering should be performed to analyze potential existing trends on the increase of L4/L5 compression force with the increase in CMU loads.

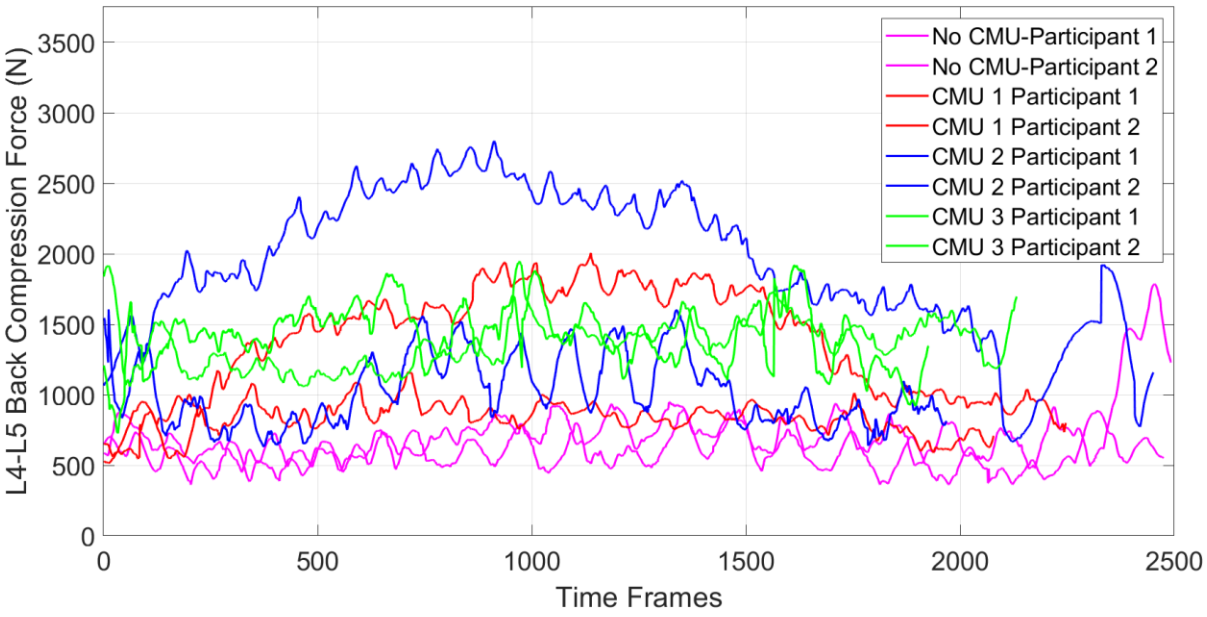

Figure 29: L4/L5 lumbar back compression force

5.6 Square Gait Pattern

The results show an increase in the net shoulder moment for both participants (Figures 30 and 31) as the weight of the CMU blocks increase from the unloaded case to those loaded and similar increase in net moment as the CMU weight increased from 8 kg (CMU-1) to 24kg (CMU-3). The large spikes observed in the left and right shoulder moment are spurious due to impact events between the left and right upper arm IMU and the body. Furthermore, since the CMUs were carried bilaterally, the magnitudes of the right and left shoulder moments are similar in magnitude showing no hand preference while carrying the load.

The elbow moments, Figures 32 and 33, increase from the unloaded case to the loaded cases and increase again in the loaded cases as the weight of the CMU increases. The large spikes observed in the left and right elbow moment are spurious due to impact events between the left and right lower arm IMU and the body. Similarly, since the CMUs were carried bilaterally, the magnitudes of the right and left elbow moments are similar showing no hand preference while carrying the loads.

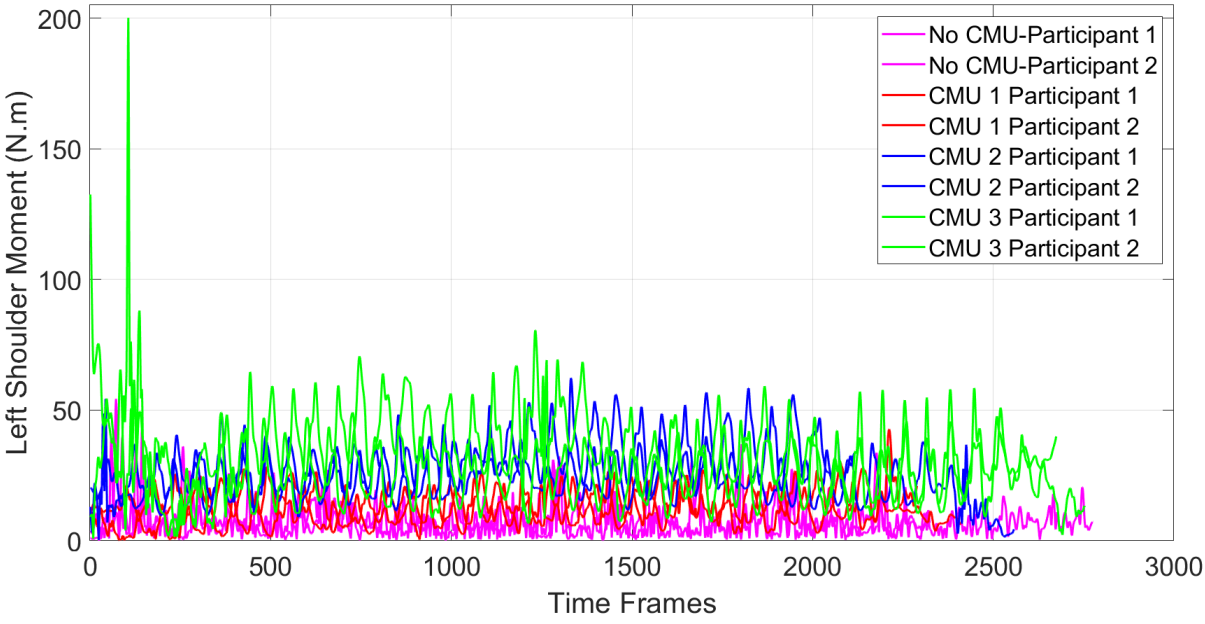

Figure 30: Net moment in the left shoulder joint

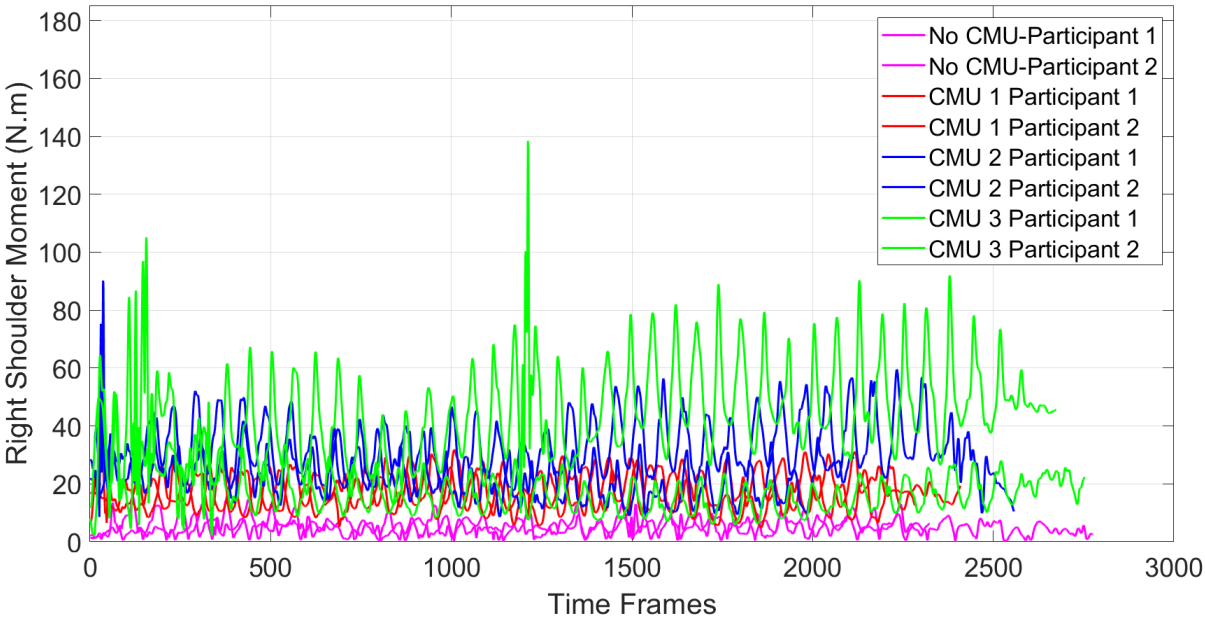

Figure 31: Net moment in the right shoulder joint

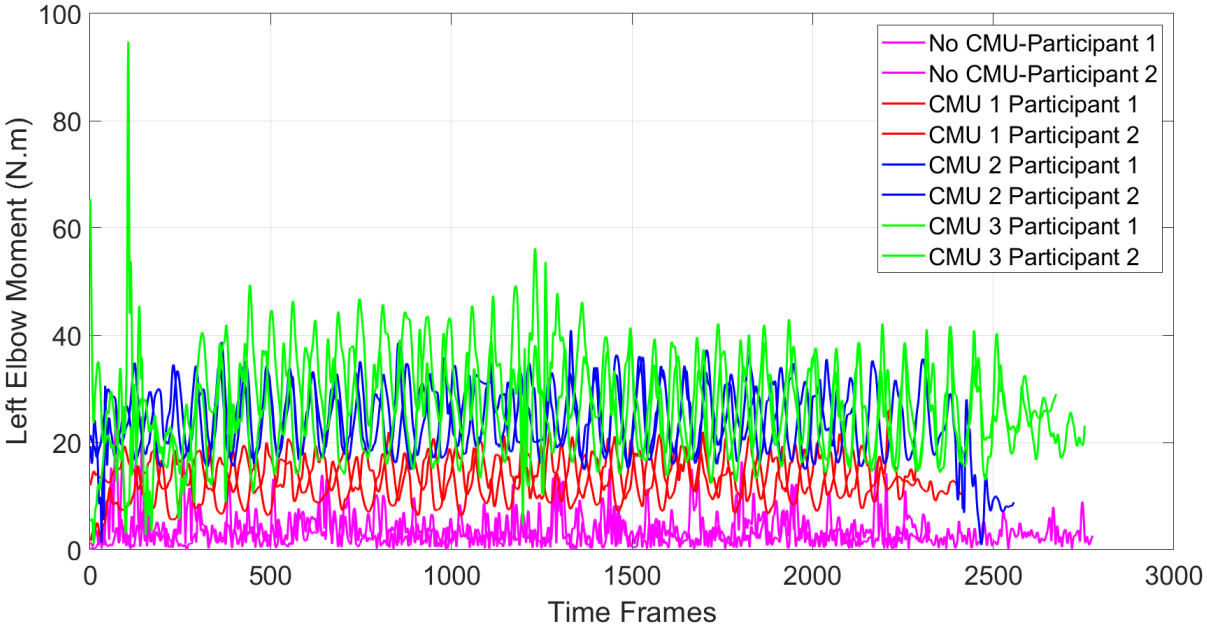

Figure 32: Net moment in the left elbow joint

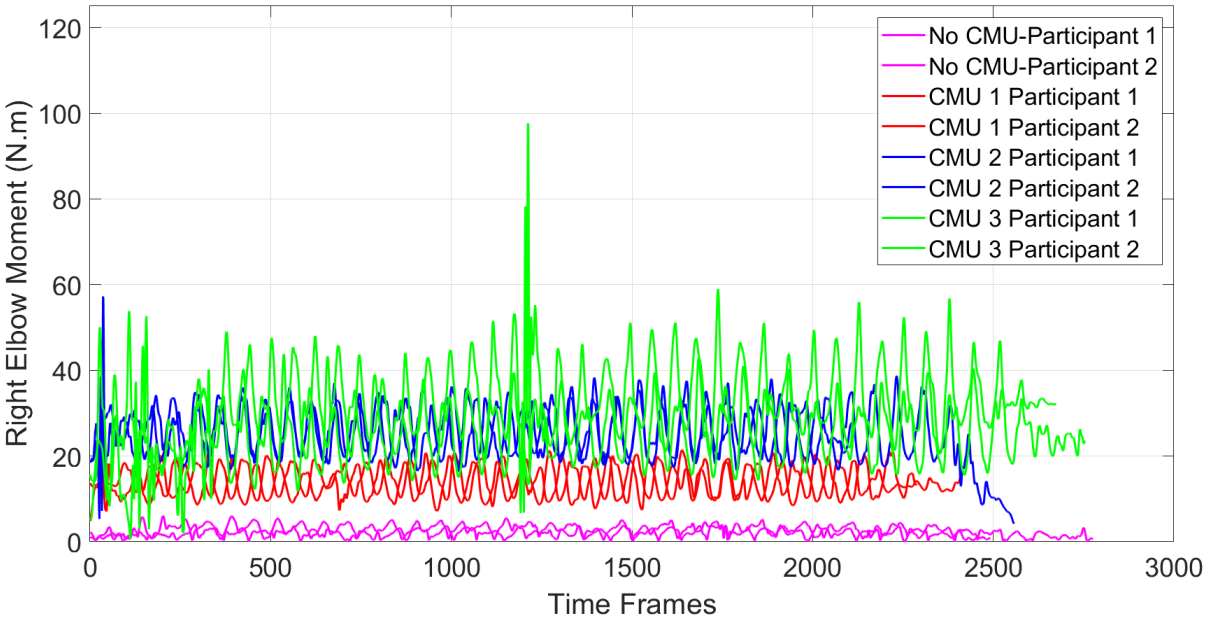

Figure 33: Net moment in the right elbow joint

In the L4/L5 back compression force, Figure 34 an increase in L4/L5 back compression force was noticed with the increase in CMU weights from the unloaded case to those loaded and similar increase

in net compression force as the CMU weight increased from 8 kg (CMU-1) to 24kg (CMU-3). The high initial L4/L5 back compression measurements shown is due to the CMU pick up phase.

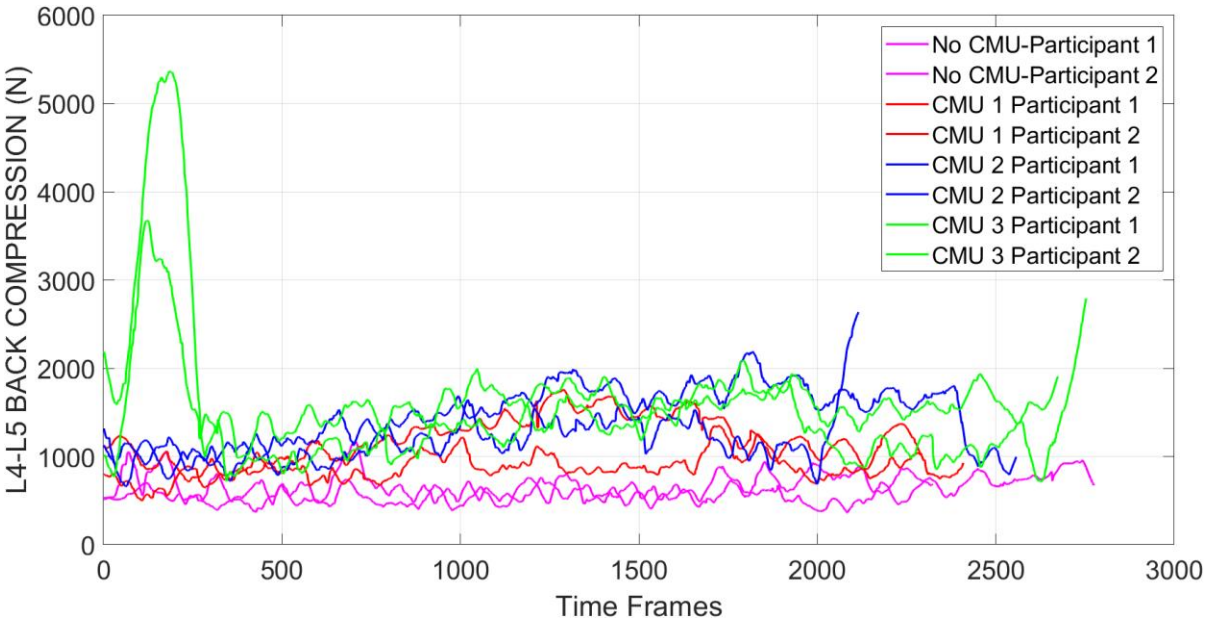

Figure 34: L4/L5 lumbar back compression force

More test subjects are needed to confirm the repeatability of these results. Also, kinematic analysis is needed to correlate variations in body posture including the shoulder and elbow joints and variations in the kinetic joints as the CMU's weight increase. This can also help explore any variations in the upper/lower arm motion patterns with the increase in CMU weight. This analysis may also determine whether a breakdown occurs in upper/lower arm posture due to elevated CMU weight or due to fatigue.

5.7 Conclusion

An extensive experiment is to be conducted to not only validate the repeatability of the results concerning the shoulder and elbow moments for all gait paths. The correlation of the increase in the values of the shoulder and elbow moments with the increase of the CMU block size is to be validated with an ANOVA study with a larger sample size. L4/L5 back compression forces demonstrated a correlation has been detected in the square and linear gait paths but not in the circular gait paths, hence a validation study is suggested to determine if there is a correlation in the circular gait paths.

Chapter 6
Conclusions and Future Work

This thesis has managed to integrate pressure insoles with the existing IMU suites to substitute the ground reaction force estimations that are fed into the inverse dynamics model. Furthermore, this research was essential to optimize the utility of the biomechanical based ergonomic assessment tool by improving its fidelity and maintaining its advantage of staying wireless to still be used in a construction site. Furthermore, this study has exploited the use of pressure insoles that allowed the development of a CMU block weight detection feature that is efficient under bilateral hand carriage. This feature was validated through multiple test subjects and gait paths and has proven to be a reliable detection for CMU blocks. However, this research has a lot of limitations such as the pressure insoles not being able to compute the M/L and the A/P GRF measurements which has still caused to estimate these components. Furthermore, for the weight detection algorithm, even though the insoles managed to detect the CMU blocks at a promising rate, the main constraint of this study is that the weight differential between the CMU block is too high and made the weight detection simpler.

Furthermore, future work for this project may include the following:

1- The biomechanical analysis with the synchronized pressure insole is only performed with 2 test subjects, hence it is essential to elaborate the study to incorporate more test subjects to validate the results.

2- Invest in utilizing updated versions of the pressure insoles that compute all 3 components of the GRFs and incorporate them in the IMU based biomechanical risk assessment model.

3- Utilize more loads with closer weight differential to understand the limitations of the weight detection algorithm. It is also recommended to undergo postural analysis when performing these lifts to correlate torso and hip movements to analyze the contralateral response of the body during bilateral load carriage.

4- As for unilateral load carriage weight detection, it is essential to perform biomechanical analysis to understand the body response to counteract the contralateral moment that is generated by the non-weight bearing side of the lifter. It is important to perform a risk assessment study on all joints to identify the locations of high injury risks.

5- The biomechanical assessment study was performed on common individuals and not masons, hence they might not be physically qualified to perform this type of manual handling activities. So, fatigue might come to play especially under higher CMU block weight carriage. Hence, it is crucial that this study is relayed to CMDC and potentially perform this experiment on a whole shift for all levels of masons (1st year apprentices, 2nd year apprentices, journeymen etc…) this would relay a good benchmark on the effect of fatigue when carrying these blocks. Also, it is essential to incorporate working environmental conditions to replicate nominal masonry behavior when performing the experiment.

Letters of Copyright Permission

ELSEVIER LICENSE
TERMS AND CONDITIONS

Jan 20, 2022

This Agreement between University of Waterloo -- Ahmad Mahmassani ("You") and Elsevier ("Elsevier") consists of your license details and the terms and conditions provided by Elsevier and Copyright Clearance Center.

License Number	5233310257518
License date	Jan 20, 2022
Licensed Content Publisher	Elsevier
Licensed Content Publication	Journal of Biomechanics
Licensed Content Title	Adjustments to McConville et al. and Young et al. body segment inertial parameters
Licensed Content Author	R. Dumas,L. Chèze,J.-P. Verriest
Licensed Content Date	Jan 1, 2007
Licensed Content Volume	40
Licensed Content Issue	3
Licensed Content Pages	11
Start Page	543
End Page	553
Type of Use	reuse in a thesis/dissertation

ELSEVIER LICENSE
TERMS AND CONDITIONS

Jan 20, 2022

This Agreement between University of Waterloo -- Ahmad Mahmassani ("You") and Elsevier ("Elsevier") consists of your license details and the terms and conditions provided by Elsevier and Copyright Clearance Center.

License Number	5233310077496
License date	Jan 20, 2022
Licensed Content Publisher	Elsevier
Licensed Content Publication	Journal of Biomechanics
Licensed Content Title	Analysis and measurement of lumbar trunk loads in tasks involving bends and twists
Licensed Content Author	A.B. Schultz,G.B.J. Andersson,K. Haderspeck,R. Örtengren,M. Nordin,R. Björk
Licensed Content Date	Jan 1, 1982
Licensed Content Volume	15
Licensed Content Issue	9
Licensed Content Pages	7
Start Page	669
End Page	675

References

1. A. Ancillao, S. Tedesco, J. Barton, and B. OFlynn, Indirect measurement of ground reaction forces and moments by means of wearable inertial sensors: a systematic review," Sensors, vol. 18, no. 8, p. 2564, 2018.
2. Berceanu C, Marghitu D., Gudavalli M., Raju P., Vikas Y. Gait analysis parameters of healthy human subjects with asymmetric loads. *Computer methods in biomechanics and biomedical engineering*. 2016;19(8):855-863. doi:10.1080/10255842.2015.1075008
3. Castro, M.P., Meucci, M., Soares, D.P., Fonseca, P., Borgonovo-Santos, M., Sousa, F., Machado, L., Vilas-Boas, J.P., 2014c. Accuracy and repeatability of the gait analysis by the WalkinSense system. Biomed. Res. Int. 2014, 11
4. Chang, W.-R., Chang, C.-C., Matz, S., 2011. The effect of transverse shear force on the required coefficient of friction for level walking. Hum. Factors 53, 461e473.
5. Crowe A, Schiereck P, Keessen W. Gait adaptations of young adult females to hand-held loads determined from ground reaction forces. Gait Posture 1993;1:154–60
6. Diraneyya, M. (2019). *Full-Body Inverse Dynamics Using Inertial Measurement Units* (Master's thesis, University of Waterloo).
7. R. Dumas, L. Cheze, and J.-P. Verriest, “Adjustments to McConville et al. and Young et al. body segment inertial parameters," Journal of biomechanics, vol. 40, no. 3, pp. 543-553, 2007.
8. Filippeschi, A., Schmitz, N., Miezal, M., Bleser, G., Ruffaldi, E., & Stricker, D. (2017). Survey of Motion Tracking Methods Based on Inertial Sensors: A Focus on Upper Limb Human Motion. *Sensors (Basel), 17*(6), 1257. doi:10.3390/s17061257
9. Fowler NE, Rodacki AFL, Rodacki CD. Changes in stature and spine kinematics during a loaded walking task. Gait Posture 2006;23: 133–41
10. Ge Wu, Sorin Siegler, Paul Allard, Chris Kirtley, Alberto Leardini, Dieter Rosenbaum, Mike Whittle, Darryl D DLima, Luca Cristofolini, Hartmut Witte, et al. Isb recommendation o definitions of joint coordinate system of various joints for the reporting of human joint motion part i: ankle, hip, and spine. Journal of biomechanics,35(4):543-548, 2002.
11. Ge Wu, Frans CT Van der Helm, HEJ DirkJan Veeger, Mohsen Makhsous, Peter Van Roy, Carolyn Anglin, Jochem Nagels, Andrew R Karduna, Kevin McQuade, Xuguang Wang, et al. ISB recommendation on de_nitions of joint coordinate systems of various joints for the

reporting of human joint motionpart ii: shoulder, elbow, wrist, and hand. Journal of biomechanics, 38(5):981-992, 2005.

12. James C Bean, Don B Chaffin, and Albert B Schultz. Biomechanical model calculation of muscle contraction forces: a double linear programming method. Journal of biomechanics, 21(1):59-66,1988
13. Karamanidis K, Arampatzis A, Brüggemann G-P. 2003. Symmetry and reproducibility of kinematic parameters during various running techniques. Med Sci Sports Exerc. 35(6):1009–1016.
14. A. Karatsidis, G. Bellusci, H. M. Schepers, M. de Zee, M. S. Andersen, and P. H. Veltink, " Esimation of ground reaction forces and moments during gait using only inertial motion capture," Sensors, vol. 17, no. 1, p. 75, 2016.
15. Katariina Luoma, Hilkka Riihimäki, Ritva Luukkonen, Raili Raininko, Eira Viikari Juntura, and Antti Lamminen. Low back pain in relation to lumbar disc degeneration. *Spine*, 25(4):487-492, 2000.
16. Katariina Luoma, Hilkka Riihimäki, Raili Raininko, Ritva Luukkonen, Antti Lamminen, and Eira Viikari-Juntura. Lumbar disc degeneration in relation to occupation. *Scandinavian journal of work, environment & health, pages* 358-366, 1998

17. Li, J., Bao, R., Tao, J., Peng, Y. & Pan, C. Recent progress in flexible pressure sensor arrays: from design to applications. J. Mater. Chem. C 6, 11878–11892 (2018)
18. Memarian, B., and Mitropoulos, P. (2012). Safety incidents and high-risk activities of masonry construction. In *Construction Research Congress 2012: Construction Challenges in a Flat World* (pp. 2510-2519).

19. Michele Crites Battie, Tapio Videman, Laura E Gibbons, Lloyd D Fisher, Hannu Manninen, and Kevin Gill. Determinants of lumbar disc degeneration: a study relating lifetime exposures and magnetic resonance imaging findings in identical twins. Spine, 20(24):2601-2612, 1995.
20. Moticon ReGo. (2021). Moticon_SCIENCE_geometry_v1.2. Retrieved from https://moticon.com/
21. Moticon ReGo. (2021). Moticon-SCIENCE_Instruction-Manual. Retrieved from https://moticon.com/

22. Moticon ReGo. (2021). Moticon-SCIENCE_Instruction-Manual. Retrieved from https://moticon.com/
23. Moticon ReGo. (2021). Moticon-SCIENCE_Sensor-Insole-Product-Specification. Retrieved from https://moticon.com/
24. Noitom Ltd. (2017). "Perception Neuron". Retrieved from https://neuronmocap.com/
25. Noitom Ltd. (2021). Quickstart Guide. Retrieved from https://neuronmocap.com/content/axis-neuron
26. S. E. Oh, A. Choi, and J. H. Mun, Prediction of ground reaction forces during gait based on kinematics and a neural network model," Journal of biomechanics, vol. 46, no. 14, pp. 2372{2380, 2013}.
27. Piscoya, J. L., Fermor, B., Kraus, V. B., Stabler, T. V., & Guilak, F. (2005). The influence of mechanical compression on the induction of osteoarthritis-related biomarkers in articular cartilage explants. *Osteoarthritis and Cartilage*, *13*(12), 1092-1099.
28. Radwin, R. G., Marras, W. S., & Lavender, S. A. (2001). Biomechanical aspects of work-related musculoskeletal disorders. *Theoretical Issues in Ergonomics Science, 2*(2), 153-217. doi:https://doi.org/10.1080/14639220110102044
29. L. Ren, R. K. Jones, and D. Howard, \Whole body inverse dynamics over a complete gait cycle based only on measured kinematics," Journal of biomechanics, vol. 41, no. 12, pp. 2750{2759, 2008.
30. Robert-Lachaine, X., Mecheri, H., Muller, A., Larue, C., & Plamondon, A. (2020). Validation of a low-cost inertial motion capture system for whole-body motion analysis. *J Biomech, 99*, 109520.doi:10.1016/j.jbiomech.2019.109520
31. Roetenberg, D., Luinge, H., &Slycje, P. (2009). Xsens MVN : Full 6DOF Human Motion Tracking Using Miniature Inertial Sensors. *Hand, The*, 1-7 doi:10.1.1.569.9604
32. A.Schultz, G. Andersson, K. Haderspeck, R. Örtengren, M. Nordin, and R. Björk, "Analysis and measurement of lumbar trunk loads in tasks involving bends and twists," Journal of biomechanics, vol. 15, no. 9, pp. 669-675, 1982.
33. Sers, R., Forrester, S., Moss, E., Ward, S., Ma, J., & Zeeca, M., (2020). Validity of the Perceprtion Neuron inertial motion capture system for upper body motion analysis. *Measurement, 149*,107024.

34. Tao, J., Dong, M., Li, L. *et al.* Real-time pressure mapping smart insole system based on a controllable vertical pore dielectric layer. *Microsyst Nanoeng* **6,** 62 (2020).
35. C. L. Vaughan, J. G. Hay, and J. G. Andrews, “Closed loop problems in biomechanics. part ii an optimization approach,” Journal of Biomechanics, vol. 15, no. 3, pp. 201-210,1982.

Appendix A

Consent Form

Department of Systems Design Engineering

Date: July 26, 2021

Title of Project: ***Identification of Good Form among Construction Trade Workers***

Principal Investigator: ***Eihab Abdel-Rahman***
University of Waterloo, Department of ***Systems Design Engineering***
519-888-4567 Ext. ***37737***
eihab@uwaterloo.ca

Principal Investigator: ***Carl Haas***
University of Waterloo, Department of ***Civil and Environmental Engineering***
519-888-4567 Ext. ***35492***
chaas@uwaterloo.ca

Co-Investigator: ***SangHyun Lee***
University of Michigan, USA, Department of ***Civil and Environmental Engineering***
+1 734-764-9420
shdpm@umich.edu

Co-Investigator: ***Abdullatif Alwasel***
King Saud University, Department of ***Biomedical Technology***
+966 11-469-3764
alwasel@ksu.edu.sa

Student investigator: **JuHyeong Ryu**
University of Waterloo, Department of ***Civil and Environmental Engineering***
519-884-4567 Ext. ***33929***
j4ryu@uwaterloo.ca

Student investigator: **Tasha Chun Lien McFarland**

University of Waterloo, Department of ***System Design Engineering***
519-884-4567 Ext. ***33929***
tclmcfarland@uwaterloo.ca
Student investigator: **Ahmad Mahmassani**

University of Waterloo, Department of ***System Design Engineering***
amahmassani@uwaterloo.ca

Purpose of this Study

This study is being carried out as part of the PhD program requirements of Mr. JuHyeong Ryu's. Injury is one of the reasons that remove workers off the workforce early in their careers. These injuries are usually a result of working in dangerous postures that overtime lead to injury. Workers often use these postures as part of their daily work without realizing their long-term implications.

This study hypothesizes that experienced workers have adopted a healthy way of work. Thus, we aim at discovering this 'way of work' to deduce methods that novice workers can use to gain expertise while avoiding injuries and early retirement.

We are seeking to recruit trainees in construction apprenticeship programs to participate in this study. The study will recruit 150 trainees or more from four of experience levels in the program (no experience, first year, second year, and third year trainees). We will recruit 50 expert workers, with 5 or more years of experience, in construction trades and no apparent health challenges. In addition to recruiting construction workers, we will recruit students and faculty at the University of Waterloo for Task E described in the next section.

Procedures Involved in this Study

The project will collect data about how you move during your daily work tasks. Three types of sensors will be used in this study. 1) Whole-body motion data: Wearable commercial motion-tracking suit that will be mounted on each body segment of upper and lower body. Specifically, upper arms, forearms, trunk, thighs, and legs. These are commercially available sensors that measure the joint angles. More information on the suit is available at (http://perceptionmocap.com/). 2) Real-time streaming acceleration data: Wearable and mountable wireless IMUs will be attached at the lower back and each of the thighs. The sensor information is available at (https://mbientlab.com/).

Lastly, 3) Smart pressure insoles that measures external loads carried by users. More information on the sensor is available at (https://www.moticon.de/).

The motion tracking suit units, also known as suit, and the Mbient IMUs will be strapped to body segments using Velcro tape. The smart insoles will be placed inside of the wearer's boots, both sides. No adhesive material will be used for all types of sensors. The suit and Mbient IMUs send the measured motions wirelessly to a nearby computer and smartphone, respectively. The smart insoles also wirelessly record measured data to a nearby computer and smartphone.

The placement of all types of sensors will not need removing of clothing articles. However, you will need to lift your shirt sleeve to expose your upper shoulder for the placement of the IMU sensors on the skin of your shoulder area. After the placement of the sensor your sleeve can be lowered down again.

The suit will be placed on both sides of your body to provide full motion tracking. In addition, one or two video cameras will record how you complete the task.

Prior to the task, you will be asked about your height, age, and weight. You will be asked to participate in one of the described tasks below. The tasks involved are:

A. Complete a wall starting from a lead wall. Mortar and concrete masonry units (CMUs) will be brought to the site of building. The wall is 6 CMUs high and 12 CMUs wide.
B. Lay one or two courses of CMUs in an existing wall.
C. Complete tool and material handling tasks, such as rebar tying of reinforcement walls, drilling of reinforcement walls, and grinding or welds. Material will be laid out before hand for your task.
D. A regular eight-hour construction work shift. We will only use IMU data for the working hours and will exclude data for break times and lunch time. In addition, the collected data will only be used for the analysis of physical exertion.
E. Performing a set of walking exercise hands-free and while carrying 3 types of CMUs weighing 8 kg, 16 kg, and 23 kg. Tracks for the exercise are following dimensions:
    - Walk around a circular path r_{circle} = 4 m (clockwise direction)
    - Walk around a circular path r_{circle} = 4 m (counterclockwise direction)
    - Walk around a square path with D_{side}=6.25 m (clockwise direction)
    - Walk around a square path with D_{side}=6.25 m (counterclockwise direction)
    - Walk along a line, turn, and return (with turn) D_{travel} =12. 5 m

    The dimensions were chosen based on the assumption that the average stride length of a regular male is around 0.7 meters. Each participant will perform a walk in one direction three times with no weight to set the baseline measurement of each individual. Then, each participant will carry each type of CMUs in the order of the

tracks. In between each set of experiments, participants will be offered the option to have a 5-minute break.

Before engaging in the exercises, the researcher(s) will demonstrate the proper CMU handling procedure. Participants will be informed that they are free to stop at any point during the exercise, do not have to continue with the whole program, handle all the CMUs particularly, the 23kg CMU.

As token of our appreciation, a $10 Tim Card will be given for each participant. The amount received is taxable. It is your responsibility to report this amount for income tax purposes.

A regular eight-hour construction work shift. We will only use IMU data for the working hours and will exclude data for break times and lunch time. In addition, the collected data will only be used for the analysis of physical exertion.

Risks to Participation and Associated Safeguards

- There is always a risk of muscle, joint or other injury in any physical work. However, the risks in this study are not anticipated to be greater than those required for your daily work tasks.
- If you are allergic to alcohol swabs used to sanitize the equipment and/or adhesive material used in double-sided tapes you are not be eligible to participate in this study as both materials will be used in this study.
- Sensors are not disposable and will be used for all participants in this study. The sensors will be sanitized using alcohol swabs between uses. The double-sided tape is disposable. Redness or a rash may occur when removing the tape from your skin. This should be temporary and disappear in one or two days.
- Pressure Insoles are not disposable and will be used for all participants in this study.
- Pressure insoles will be sanitized between uses.

Time Commitment

Participation in this study will require approximately 1 hour of your time. All sessions will be scheduled outside of class time.

Changing Your Mind about Participation

You may withdraw from this study at any time without penalty. To do so, indicate this to the researcher or one of the research assistants by saying, "I no longer wish to participate in this study".

Personal Benefits of Participation

There are no direct benefits for participating in the study. However, this study will provide researchers with knowledge about how workers move in their daily tasks thus allowing researchers to design work tasks more safe and efficient.

Confidentiality

To ensure the confidentiality of individuals' data, each participant will be identified by a participant identification code known only to the principal investigators and student investigators. Videotapes will be stored for 7 years, from the day of study anticipated completion (Aug 2021), in a secure area for further research purposes in the future e.g. alerting the worker using video data. No face blurring will be used as the video recording will not be facing the participant, hence, mostly no face recording is done. A separate consent will be requested in order to use the videotapes and/or photographs for teaching, for scientific presentations, or in publications of this work.

Data related to your participation will be submitted to an online data repository. It will be completely anonymized/de-identified by removing names and video recordings before submission. This process is integral to the research process as it allows other researchers to verify results and avoid duplicating research. Other individuals may access this data by downloading data spreadsheets. Should you choose, you may review all data that will be submitted before it is entered into data repository.

Participant Feedback

After the study is completed, you will be provided with an appreciation letter from the research team.

Concerns about Your Participation

This study has been reviewed and received ethics clearance through a University of Waterloo Research Ethics Committee (ORE#30382). If you have questions for the Committee contact the Chief Ethics Officer, Office of Research Ethics, at 1-519-888-4567 ext. 36005 or ore-ceo@uwaterloo.ca.

For all other questions contact Eihab Abdel-Rahman, Carl Haas, JuHyeong Ryu at 519-888-4567 Ext. 37737, 35492, and 33929 respectively.

Questions about the Study

If you have additional questions later or want any other information regarding this study, please contact (Eihab Abdel-Rahman, Carl Haas, JuHyeong Ryu) at 519-888-4567 Ext. 37737, 35492, and 33929 respectively.

CONSENT TO PARTICIPATE

By signing this consent form, you are not waiving your legal rights or releasing the investigator(s) or involved institution(s) from their legal and professional responsibilities.

I agree to take part in a research study being conducted by Dr. Eihab Abdel-Rahman, Dr. Carl Haas, and Juhyeong Ryu of the Department of Systems Design Engineering and Civil and Environmental Engineering, University of Waterloo.

I have made this decision based on the information I have read in the Information letter. All the procedures, any risks and benefits have been explained to me. I have had the opportunity to ask any questions and to receive any additional details I wanted about the study. If I have questions later about the study, I can ask one of the researchers (Eihab Abdel-Rahman, Department of Systems Design Engineering, Carl Haas, JuHyeong Ryu, Department of Civil and Environmental Engineering at 519-888-4567 exts. 33737, 35492, 33929 respectively).

I understand that I may withdraw from the study at any time without penalty by telling the researcher.

This study has been reviewed and received ethics clearance through a University of Waterloo Research Ethics Committee (ORE#30382). If you have questions for the Committee contact the Chief Ethics Officer, Office of Research Ethics, at 1-519-888-4567 ext. 36005 or ore-ceo@uwaterloo.ca.

Do you want to review data before it is stored in data repository?

Yes ☐ No ☐

Printed Name of Participant _____ Signature of Participant _____

_____ _____

Dated at Waterloo, Ontario Witnessed

Consent to Use Video and/or Photographs

Sometimes a certain photograph and/or part of a video-tape clearly shows a particular feature or detail that would be helpful in teaching or when presenting the study results in a scientific presentation or publication. If you grant permission for photographs or videotapes in which you appear to be used in this manner, please complete the following section.

I agree to allow video and/or photographs to be used in teaching or scientific presentations, or published in scientific journals or professional publications of this work without identifying me by name.

_____ _____

Printed Name of Participant Signature of Participant

_____ _____

Dated at Waterloo, Ontario Witnessed

Appendix B

Metatarsal Head Bilateral Handload Weight Detection Methodology (One Test Subject and One Gait Path Example)

```
%%Metatarsal Head  Pressure Patches 9, 10, and 14 no weight
fileNames = dir('*.xlsx')
for i = 1:length(fileNames)
   File = fileNames(i).name;

   data{i} = xlsread(File);
end
% File Imports //DONOT USE xlsread next time you process, use dlmread/
AL0=xlsread('a-l-0.xlsx');
CL0=xlsread('ca-l-0.xlsx');
JL0=xlsread('ju-l-0.xlsx');
ML0=xlsread('Mo-l-0.xlsx');
RL0=xlsread('r-l-0.xlsx');
SL0=xlsread('sa-l-0.xlsx');

AL1=xlsread('a-l-1.xlsx');
CL1=xlsread('ca-l-1.xlsx');
JL1=xlsread('ju-l-1.xlsx');
ML1=xlsread('Mo-l-1.xlsx');
RL1=xlsread('r-l-1.xlsx');
SL1=xlsread('sa-l-1.xlsx');

AL2=xlsread('a-l-2.xlsx');
CL2=xlsread('ca-l-2.xlsx');
JL2=xlsread('ju-l-2.xlsx');
ML2=xlsread('Mo-l-2.xlsx');
RL2=xlsread('r-l-2.xlsx');
```

```
SL2=xlsread('sa-l-2.xlsx');

AL3=xlsread('a-l-3.xlsx');
CL3=xlsread('ca-l-3.xlsx');
JL3=xlsread('ju-l-3.xlsx');
ML3=xlsread('Mo-l-3.xlsx');
RL3=xlsread('r-l-3.xlsx');
SL3=xlsread('sa-l-3.xlsx');


%% Both Handed 1 Lift linear allwieghts all test subjects
%Preprocessing methods//first, separate Right and Left pressure measurements from rawfile
   AL0pressureLboth_0_1=xlsread('a-l-0.xlsx',1,'B100:Q1900');
   AL0pressureRboth_0_1=xlsread('a-l-0.xlsx',1,'AA100:AP1900');
   AL1pressureLboth_0_1=xlsread('a-l-1.xlsx',1,'B100:Q1900');
   AL1pressureRboth_0_1=xlsread('a-l-1.xlsx',1,'AA100:AP1900');
   AL2pressureLboth_0_1=xlsread('a-l-2.xlsx',1,'B100:Q1900');
   AL2pressureRboth_0_1=xlsread('a-l-2.xlsx',1,'AA100:AP1900');
   AL3pressureLboth_0_1=xlsread('a-l-3.xlsx',1,'B100:Q1900');
   AL3pressureRboth_0_1=xlsread('a-l-3.xlsx',1,'AA100:AP1900');

   CL0pressureLboth_0_1=xlsread('ca-l0.xlsx',1,'B100:Q1900');
   CL0pressureRboth_0_1=xlsread('ca-l-0.xlsx',1,'AA100:AP1900');
   CL1pressureLboth_0_1=xlsread('ca-l-1.xlsx',1,'B100:Q1900');
   CL1pressureRboth_0_1=xlsread('ca-l-1.xlsx',1,'AA100:AP1900');
   CL2pressureLboth_0_1=xlsread('ca-l-2.xlsx',1,'B100:Q1900');
   CL2pressureRboth_0_1=xlsread('ca-l-2.xlsx',1,'AA100:AP1900');
   CL3pressureLboth_0_1=xlsread('ca-l-3.xlsx',1,'B100:Q1900');
   CL3pressureRboth_0_1=xlsread('ca-l-3.xlsx',1,'AA100:AP1900');

   JL0pressureLboth_0_1=xlsread('ju-l-0.xlsx',1,'B100:Q1900');
   JL0pressureRboth_0_1=xlsread('ju-l-0.xlsx',1,'AA100:AP1900');
```

```
JL1pressureLboth_0_1=xlsread('ju-l-1.xlsx',1,'B100:Q1900');
JL1pressureRboth_0_1=xlsread('ju-l-1.xlsx',1,'AA100:AP1900');
JL2pressureLboth_0_1=xlsread('ju-l-2.xlsx',1,'B100:Q1900');
JL2pressureRboth_0_1=xlsread('ju-l-2.xlsx',1,'AA100:AP1900');
JL3pressureLboth_0_1=xlsread('ju-l-3.xlsx',1,'B100:Q1900');
JL3pressureRboth_0_1=xlsread('ju-l-3.xlsx',1,'AA100:AP1900');


ML0pressureLboth_0_1=xlsread('Mo-l-0.xlsx',1,'B100:Q1900');
ML0pressureRboth_0_1=xlsread('Mo-l-0.xlsx',1,'AA100:AP1900');
ML1pressureLboth_0_1=xlsread('Mo-l-1.xlsx',1,'B100:Q1900');
ML1pressureRboth_0_1=xlsread('Mo-l-1.xlsx',1,'AA100:AP1900');
ML2pressureLboth_0_1=xlsread('Mo-l-2.xlsx',1,'B100:Q1900');
ML2pressureRboth_0_1=xlsread('Mo-l-2.xlsx',1,'AA100:AP1900');
ML3pressureLboth_0_1=xlsread('Mo-l-3.xlsx',1,'B100:Q1900');
ML3pressureRboth_0_1=xlsread('Mo-l-3.xlsx',1,'AA100:AP1900');


RL0pressureLboth_0_1=xlsread('r-l-0.xlsx',1,'B100:Q1900');
RL0pressureRboth_0_1=xlsread('r-l-0.xlsx',1,'AA100:AP1900');
RL1pressureLboth_0_1=xlsread('r-l-1.xlsx',1,'B100:Q1900');
RL1pressureRboth_0_1=xlsread('r-l-1.xlsx',1,'AA100:AP1900');
RL2pressureLboth_0_1=xlsread('r-l-2.xlsx',1,'B100:Q1900');
RL2pressureRboth_0_1=xlsread('r-l-2.xlsx',1,'AA100:AP1900');
RL3pressureLboth_0_1=xlsread('r-l-3.xlsx',1,'B100:Q1900');
RL3pressureRboth_0_1=xlsread('r-l-3.xlsx',1,'AA100:AP1900');

SL0pressureLboth_0_1=xlsread('sa-l-0.xlsx',1,'B100:Q1900');
SL0pressureRboth_0_1=xlsread('sa-l-0.xlsx',1,'AA100:AP1900');
SL1pressureLboth_0_1=xlsread('sa-l-1.xlsx',1,'B100:Q1900');
SL1pressureRboth_0_1=xlsread('sa-l-1.xlsx',1,'AA100:AP1900');
SL2pressureLboth_0_1=xlsread('sa-l-2.xlsx',1,'B100:Q1900');
```

```
SL2pressureRboth_0_1=xlsread('sa-l-2.xlsx',1,'AA100:AP1900');
SL3pressureLboth_0_1=xlsread('sa-l-3.xlsx',1,'B100:Q1900');
SL3pressureRboth_0_1=xlsread('sa-l-3.xlsx',1,'AA100:AP1900');
%% Both Handed 1 Lift square all weights all test subjects

AS0pressureLboth_0_1=xlsread('a-s-0.xlsx',1,'B100:Q1900');
AS0pressureRboth_0_1=xlsread('a-s-0.xlsx',1,'AA100:AP1900');
AS1pressureLboth_0_1=xlsread('a-s-1.xlsx',1,'B100:Q1900');
AS1pressureRboth_0_1=xlsread('a-s-1.xlsx',1,'AA100:AP1900');
AS2pressureLboth_0_1=xlsread('a-s-2.xlsx',1,'B100:Q1900');
AS2pressureRboth_0_1=xlsread('a-s-2.xlsx',1,'AA100:AP1900');
AS3pressureLboth_0_1=xlsread('a-s-3.xlsx',1,'B100:Q1900');
AS3pressureRboth_0_1=xlsread('a-s-3.xlsx',1,'AA100:AP1900');

CS0pressureLboth_0_1=xlsread('ca-s-0.xlsx',1,'B100:Q1900');
CS0pressureRboth_0_1=xlsread('ca-s-0.xlsx',1,'AA100:AP1900');
CS1pressureLboth_0_1=xlsread('ca-s-1.xlsx',1,'B100:Q1900');
CS1pressureRboth_0_1=xlsread('ca-s-1.xlsx',1,'AA100:AP1900');
CS2pressureLboth_0_1=xlsread('ca-s-2.xlsx',1,'B100:Q1900');
CS2pressureRboth_0_1=xlsread('ca-s-2.xlsx',1,'AA100:AP1900');
CS3pressureLboth_0_1=xlsread('ca-s-3.xlsx',1,'B100:Q1900');
CS3pressureRboth_0_1=xlsread('ca-s-3.xlsx',1,'AA100:AP1900');

JS0pressureLboth_0_1=xlsread('ju-s-0.xlsx',1,'B100:Q1900');
JS0pressureRboth_0_1=xlsread('ju-s-0.xlsx',1,'AA100:AP1900');
JS1pressureLboth_0_1=xlsread('ju-s-1.xlsx',1,'B100:Q1900');
JS1pressureRboth_0_1=xlsread('ju-s-1.xlsx',1,'AA100:AP1900');
JS2pressureLboth_0_1=xlsread('ju-s-2.xlsx',1,'B100:Q1900');
JS2pressureRboth_0_1=xlsread('ju-s-2.xlsx',1,'AA100:AP1900');
JS3pressureLboth_0_1=xlsread('ju-s-3.xlsx',1,'B100:Q1900');
JS3pressureRboth_0_1=xlsread('ju-s-3.xlsx',1,'AA100:AP1900');
```

```
MS0pressureLboth_0_1=xlsread('Mo-s-0.xlsx',1,'B100:Q1900');
MS0pressureRboth_0_1=xlsread('Mo-s-0.xlsx',1,'AA100:AP1900');
MS1pressureLboth_0_1=xlsread('Mo-s-1.xlsx',1,'B100:Q1900');
MS1pressureRboth_0_1=xlsread('Mo-s-1.xlsx',1,'AA100:AP1900');
MS2pressureLboth_0_1=xlsread('Mo-s-2.xlsx',1,'B100:Q1900');
MS2pressureRboth_0_1=xlsread('Mo-s-2.xlsx',1,'AA100:AP1900');
MS3pressureLboth_0_1=xlsread('Mo-s-3.xlsx',1,'B100:Q1900');
MS3pressureRboth_0_1=xlsread('Mo-s-3.xlsx',1,'AA100:AP1900');

RS0pressureLboth_0_1=xlsread('r-s-0.xlsx',1,'B100:Q1900');
RS0pressureRboth_0_1=xlsread('r-s-0.xlsx',1,'AA100:AP1900');
RS1pressureLboth_0_1=xlsread('r-s-0.xlsx',1,'B100:Q1900');
RS1pressureRboth_0_1=xlsread('r-s-1.xlsx',1,'AA100:AP1900');
RS2pressureLboth_0_1=xlsread('r-s-2.xlsx',1,'B100:Q1900');
RS2pressureRboth_0_1=xlsread('r-s-2.xlsx',1,'AA100:AP1900');
RS3pressureLboth_0_1=xlsread('r-s-3.xlsx',1,'B100:Q1900');
RS3pressureRboth_0_1=xlsread('r-s-3.xlsx',1,'AA100:AP1900');

SS0pressureLboth_0_1=xlsread('sa-s-0.xlsx',1,'B100:Q1900');
SS0pressureRboth_0_1=xlsread('sa-s-0.xlsx',1,'AA100:AP1900');
SS1pressureLboth_0_1=xlsread('sa-s-1.xlsx',1,'B100:Q1900');
SS1pressureRboth_0_1=xlsread('sa-s-1.xlsx',1,'AA100:AP1900');
SS2pressureLboth_0_1=xlsread('sa-s-2.xlsx',1,'B100:Q1900');
SS2pressureRboth_0_1=xlsread('sa-s-2.xlsx',1,'AA100:AP1900');
SS3pressureLboth_0_1=xlsread('sa-s-3.xlsx',1,'B100:Q1900');
SS3pressureRboth_0_1=xlsread('sa-s-3.xlsx',1,'AA100:AP1900');
 %% Both Handed 1 Lift Circular all weights all test subjects

AC0pressureLboth_0_1=xlsread('a-c-0.xlsx',1,'B100:Q1900');
AC0pressureRboth_0_1=xlsread('a-c-0.xlsx',1,'AA100:AP1900');
AC1pressureLboth_0_1=xlsread('a-c-1.xlsx',1,'B100:Q1900');
AC1pressureRboth_0_1=xlsread('a-c-1.xlsx',1,'AA100:AP1900');
```

```
AC2pressureLboth_0_1=xlsread('a-c-2.xlsx',1,'B100:Q1900');
AC2pressureRboth_0_1=xlsread('a-c-2.xlsx',1,'AA100:AP1900');
AC3pressureLboth_0_1=xlsread('a-c-3.xlsx',1,'B100:Q1900');
AC3pressureRboth_0_1=xlsread('a-c-3.xlsx',1,'AA100:AP1900');

CC0pressureLboth_0_1=xlsread('ca-c-0.xlsx',1,'B100:Q1900');
CC0pressureRboth_0_1=xlsread('ca-c-0.xlsx',1,'AA100:AP1900');
CC1pressureLboth_0_1=xlsread('ca-c-1.xlsx',1,'B100:Q1900');
CC1pressureRboth_0_1=xlsread('ca-c-1.xlsx',1,'AA100:AP1900');
CC2pressureLboth_0_1=xlsread('ca-c-2.xlsx',1,'B100:Q1900');
CC2pressureRboth_0_1=xlsread('ca-c-2.xlsx',1,'AA100:AP1900');
CC3pressureLboth_0_1=xlsread('ca-c-3.xlsx',1,'B100:Q1900');
CC3pressureRboth_0_1=xlsread('ca-c-3.xlsx',1,'AA100:AP1900');

JC0pressureLboth_0_1=xlsread('ju-c-0.xlsx',1,'B100:Q1900');
JC0pressureRboth_0_1=xlsread('ju-c-0.xlsx',1,'AA100:AP1900');
JC1pressureLboth_0_1=xlsread('ju-c-1.xlsx',1,'B100:Q1900');
JC1pressureRboth_0_1=xlsread('ju-c-1.xlsx',1,'AA100:AP1900');
JC2pressureLboth_0_1=xlsread('ju-c-2.xlsx',1,'B100:Q1900');
JC2pressureRboth_0_1=xlsread('ju-c-2.xlsx',1,'AA100:AP1900');
JC3pressureLboth_0_1=xlsread('ju-c-3.xlsx',1,'B100:Q1900');
JC3pressureRboth_0_1=xlsread('ju-c-3.xlsx',1,'AA100:AP1900');

MC0pressureLboth_0_1=xlsread('Mo-c-0.xlsx',1,'B100:Q1900');
MC0pressureRboth_0_1=xlsread('Mo-c-0.xlsx',1,'AA100:AP1900');
MC1pressureLboth_0_1=xlsread('Mo-c-1.xlsx',1,'B100:Q1900');
MC1pressureRboth_0_1=xlsread('Mo-c-1.xlsx',1,'AA100:AP1900');
MC2pressureLboth_0_1=xlsread('Mo-c-2.xlsx',1,'B100:Q1900');
MC2pressureRboth_0_1=xlsread('Mo-c-2.xlsx',1,'AA100:AP1900');
MC3pressureLboth_0_1=xlsread('Mo-c-3.xlsx',1,'B100:Q1900');
MC3pressureRboth_0_1=xlsread('Mo-c-3.xlsx',1,'AA100:AP1900');
```

```
RC0pressureLboth_0_1=xlsread('r-s-0.xlsx',1,'B100:Q1900');
RC0pressureRboth_0_1=xlsread('r-s-0.xlsx',1,'AA100:AP1900');
RC1pressureLboth_0_1=xlsread('r-c-1.xlsx',1,'B100:Q1900');
RC1pressureRboth_0_1=xlsread('r-c-1.xlsx',1,'AA100:AP1900');
RC2pressureLboth_0_1=xlsread('r-c-2.xlsx',1,'B100:Q1900');
RC2pressureRboth_0_1=xlsread('r-c-2.xlsx',1,'AA100:AP1900');
RC3pressureLboth_0_1=xlsread('r-c-3.xlsx',1,'B100:Q1900');
RC3pressureRboth_0_1=xlsread('r-c-3.xlsx',1,'AA100:AP1900');

SC0pressureLboth_0_1=xlsread('sa-c-0.xlsx',1,'B100:Q1900');
SC0pressureRboth_0_1=xlsread('sa-c-0.xlsx',1,'AA100:AP1900');
SC1pressureLboth_0_1=xlsread('sa-c-1.xlsx',1,'B100:Q1900');
SC1pressureRboth_0_1=xlsread('sa-c-1.xlsx',1,'AA100:AP1900');
SC2pressureLboth_0_1=xlsread('sa-c-2.xlsx',1,'B100:Q1900');
SC2pressureRboth_0_1=xlsread('sa-c-2.xlsx',1,'AA100:AP1900');
SC3pressureLboth_0_1=xlsread('sa-c-3.xlsx',1,'B100:Q1900');
SC3pressureRboth_0_1=xlsread('sa-c-3.xlsx',1,'AA100:AP1900');

%% Convert pressure readings to force measurements
t=linspace(0,18,1801);
Area6 =[8.87157, 7.97504, 9.07763, 8.96946, 7.63859, 8.31133, 2.91213, 7.56437, 10.70929,
7.74710, 7.77927, 7.85591, 6.77586, 7.34975, 8.53378, 4.38093];
SF6=[1.27455, 1.26371, 1.34455, 1.33676, 1.57054, 1.43161, 5.65405, 2.20099, 1.33617,
1.46691, 1.47022, 1.47398, 1.38862, 1.26216, 1.34270, 1.35611]; %Scaling Factor for the force
depends on the size of the insole

for i=1:length(Area6)


  AL0ForceLboth0_1(:,i)=Area6(i).*(AL0pressureLboth_0_1(:,i));
  AL0ForceRboth0_1(:,i)=Area6(i).*(AL0pressureRboth_0_1(:,i));
  AL1ForceLboth0_1(:,i)=Area6(i).*(AL1pressureLboth_0_1(:,i));
  AL1ForceRboth0_1(:,i)=Area6(i).*(AL1pressureRboth_0_1(:,i));
```

```
AL2ForceLboth0_1(:,i)=Area6(i).*(AL2pressureLboth_0_1(:,i));
AL2ForceRboth0_1(:,i)=Area6(i).*(AL2pressureRboth_0_1(:,i));
AL3ForceLboth0_1(:,i)=Area6(i).*(AL3pressureLboth_0_1(:,i));
AL3ForceRboth0_1(:,i)=Area6(i).*(AL3pressureRboth_0_1(:,i));

AC0ForceLboth0_1(:,i)=Area6(i).*(AC0pressureLboth_0_1(:,i));
AC0ForceRboth0_1(:,i)=Area6(i).*(AC0pressureRboth_0_1(:,i));
AC1ForceLboth0_1(:,i)=Area6(i).*(AC1pressureLboth_0_1(:,i));
AC1ForceRboth0_1(:,i)=Area6(i).*(AC1pressureRboth_0_1(:,i));
AC2ForceLboth0_1(:,i)=Area6(i).*(AC2pressureLboth_0_1(:,i));
AC2ForceRboth0_1(:,i)=Area6(i).*(AC2pressureRboth_0_1(:,i));
AC3ForceLboth0_1(:,i)=Area6(i).*(AC3pressureLboth_0_1(:,i));
AC3ForceRboth0_1(:,i)=Area6(i).*(AC3pressureRboth_0_1(:,i));


AS0ForceLboth0_1(:,i)=Area6(i).*(AS0pressureLboth_0_1(:,i));
AS0ForceRboth0_1(:,i)=Area6(i).*(AS0pressureRboth_0_1(:,i));
AS1ForceLboth0_1(:,i)=Area6(i).*(AS1pressureLboth_0_1(:,i));
AS1ForceRboth0_1(:,i)=Area6(i).*(AS1pressureRboth_0_1(:,i));
AS2ForceLboth0_1(:,i)=Area6(i).*(AS2pressureLboth_0_1(:,i));
AS2ForceRboth0_1(:,i)=Area6(i).*(AS2pressureRboth_0_1(:,i));
AS3ForceLboth0_1(:,i)=Area6(i).*(AS3pressureLboth_0_1(:,i));
AS3ForceRboth0_1(:,i)=Area6(i).*(AS3pressureRboth_0_1(:,i));


CL0ForceLboth0_2(:,i)=Area6(i).*(CL0pressureLboth_0_1(:,i));
CL0ForceRboth0_2(:,i)=Area6(i).*(CL0pressureRboth_0_1(:,i));
CL1ForceLboth0_2(:,i)=Area6(i).*(CL1pressureLboth_0_1(:,i));
CL1ForceRboth0_2(:,i)=Area6(i).*(CL1pressureRboth_0_1(:,i));
CL2ForceLboth0_2(:,i)=Area6(i).*(CL2pressureLboth_0_1(:,i));
CL2ForceRboth0_2(:,i)=Area6(i).*(CL2pressureRboth_0_1(:,i));
CL3ForceLboth0_2(:,i)=Area6(i).*(CL3pressureLboth_0_1(:,i));
```

```
CL3ForceRboth0_2(:,i)=Area6(i).*(CL3pressureRboth_0_1(:,i));

CC0ForceLboth0_2(:,i)=Area6(i).*(CC0pressureLboth_0_1(:,i));
CC0ForceRboth0_2(:,i)=Area6(i).*(CC0pressureRboth_0_1(:,i));
CC1ForceLboth0_2(:,i)=Area6(i).*(CC1pressureLboth_0_1(:,i));
CC1ForceRboth0_2(:,i)=Area6(i).*(CC1pressureRboth_0_1(:,i));
CC2ForceLboth0_2(:,i)=Area6(i).*(CC2pressureLboth_0_1(:,i));
CC2ForceRboth0_2(:,i)=Area6(i).*(CC2pressureRboth_0_1(:,i));
CC3ForceLboth0_2(:,i)=Area6(i).*(CC3pressureLboth_0_1(:,i));
CC3ForceRboth0_2(:,i)=Area6(i).*(CC3pressureRboth_0_1(:,i));

CS0ForceLboth0_2(:,i)=Area6(i).*(CS0pressureLboth_0_1(:,i));
CS0ForceRboth0_2(:,i)=Area6(i).*(CS0pressureRboth_0_1(:,i));
CS1ForceLboth0_2(:,i)=Area6(i).*(CS1pressureLboth_0_1(:,i));
CS1ForceRboth0_2(:,i)=Area6(i).*(CS1pressureRboth_0_1(:,i));
CS2ForceLboth0_2(:,i)=Area6(i).*(CS2pressureLboth_0_1(:,i));
CS2ForceRboth0_2(:,i)=Area6(i).*(CS2pressureRboth_0_1(:,i));
CS3ForceLboth0_2(:,i)=Area6(i).*(CS3pressureLboth_0_1(:,i));
CS3ForceRboth0_2(:,i)=Area6(i).*(CS3pressureRboth_0_1(:,i));

JL0ForceLboth0_3(:,i)=Area6(i).*(JL0pressureLboth_0_1(:,i));
JL0ForceRboth0_3(:,i)=Area6(i).*(JL0pressureRboth_0_1(:,i));
JL1ForceLboth0_3(:,i)=Area6(i).*(JL1pressureLboth_0_1(:,i));
JL1ForceRboth0_3(:,i)=Area6(i).*(JL1pressureRboth_0_1(:,i));
JL2ForceLboth0_3(:,i)=Area6(i).*(JL2pressureLboth_0_1(:,i));
JL2ForceRboth0_3(:,i)=Area6(i).*(JL2pressureRboth_0_1(:,i));
JL3ForceLboth0_3(:,i)=Area6(i).*(JL3pressureLboth_0_1(:,i));
JL3ForceRboth0_3(:,i)=Area6(i).*(JL3pressureRboth_0_1(:,i));

JC0ForceLboth0_3(:,i)=Area6(i).*(JC0pressureLboth_0_1(:,i));
JC0ForceRboth0_3(:,i)=Area6(i).*(JC0pressureRboth_0_1(:,i));
JC1ForceLboth0_3(:,i)=Area6(i).*(JC1pressureLboth_0_1(:,i));
```

```
JC1ForceRboth0_3(:,i)=Area6(i).*(JC1pressureRboth_0_1(:,i));
JC2ForceLboth0_3(:,i)=Area6(i).*(JC2pressureLboth_0_1(:,i));
JC2ForceRboth0_3(:,i)=Area6(i).*(JC2pressureRboth_0_1(:,i));
JC3ForceLboth0_3(:,i)=Area6(i).*(JC3pressureLboth_0_1(:,i));
JC3ForceRboth0_3(:,i)=Area6(i).*(JC3pressureRboth_0_1(:,i));

JS0ForceLboth0_3(:,i)=Area6(i).*(JS0pressureLboth_0_1(:,i));
JS0ForceRboth0_3(:,i)=Area6(i).*(JS0pressureRboth_0_1(:,i));
JS1ForceLboth0_3(:,i)=Area6(i).*(JS1pressureLboth_0_1(:,i));
JS1ForceRboth0_3(:,i)=Area6(i).*(JS1pressureRboth_0_1(:,i));
JS2ForceLboth0_3(:,i)=Area6(i).*(JS2pressureLboth_0_1(:,i));
JS2ForceRboth0_3(:,i)=Area6(i).*(JS2pressureRboth_0_1(:,i));
JS3ForceLboth0_3(:,i)=Area6(i).*(JS3pressureLboth_0_1(:,i));
JS3ForceRboth0_3(:,i)=Area6(i).*(JS3pressureRboth_0_1(:,i));

ML0ForceLboth0_4(:,i)=Area6(i).*(ML0pressureLboth_0_1(:,i));
ML0ForceRboth0_4(:,i)=Area6(i).*(ML0pressureRboth_0_1(:,i));
ML1ForceLboth0_4(:,i)=Area6(i).*(ML1pressureLboth_0_1(:,i));
ML1ForceRboth0_4(:,i)=Area6(i).*(ML1pressureRboth_0_1(:,i));
ML2ForceLboth0_4(:,i)=Area6(i).*(ML2pressureLboth_0_1(:,i));
ML2ForceRboth0_4(:,i)=Area6(i).*(ML2pressureRboth_0_1(:,i));
ML3ForceLboth0_4(:,i)=Area6(i).*(ML3pressureLboth_0_1(:,i));
ML3ForceRboth0_4(:,i)=Area6(i).*(ML3pressureRboth_0_1(:,i));

MC0ForceLboth0_4(:,i)=Area6(i).*(MC0pressureLboth_0_1(:,i));
MC0ForceRboth0_4(:,i)=Area6(i).*(MC0pressureRboth_0_1(:,i));
MC1ForceLboth0_4(:,i)=Area6(i).*(MC1pressureLboth_0_1(:,i));
MC1ForceRboth0_4(:,i)=Area6(i).*(MC1pressureRboth_0_1(:,i));
MC2ForceLboth0_4(:,i)=Area6(i).*(MC2pressureLboth_0_1(:,i));
MC2ForceRboth0_4(:,i)=Area6(i).*(MC2pressureRboth_0_1(:,i));
MC3ForceLboth0_4(:,i)=Area6(i).*(MC3pressureLboth_0_1(:,i));
MC3ForceRboth0_4(:,i)=Area6(i).*(MC3pressureRboth_0_1(:,i));
```

```
MS0ForceLboth0_4(:,i)=Area6(i).*(MS0pressureLboth_0_1(:,i));
MS0ForceRboth0_4(:,i)=Area6(i).*(MS0pressureRboth_0_1(:,i));
MS1ForceLboth0_4(:,i)=Area6(i).*(MS1pressureLboth_0_1(:,i));
MS1ForceRboth0_4(:,i)=Area6(i).*(MS1pressureRboth_0_1(:,i));
MS2ForceLboth0_4(:,i)=Area6(i).*(MS2pressureLboth_0_1(:,i));
MS2ForceRboth0_4(:,i)=Area6(i).*(MS2pressureRboth_0_1(:,i));
MS3ForceLboth0_4(:,i)=Area6(i).*(MS3pressureLboth_0_1(:,i));
MS3ForceRboth0_4(:,i)=Area6(i).*(MS3pressureRboth_0_1(:,i));

RL0ForceLboth0_4(:,i)=Area6(i).*(RL0pressureLboth_0_1(:,i));
RL0ForceRboth0_4(:,i)=Area6(i).*(RL0pressureRboth_0_1(:,i));
RL1ForceLboth0_4(:,i)=Area6(i).*(RL1pressureLboth_0_1(:,i));
RL1ForceRboth0_4(:,i)=Area6(i).*(RL1pressureRboth_0_1(:,i));
RL2ForceLboth0_4(:,i)=Area6(i).*(RL2pressureLboth_0_1(:,i));
RL2ForceRboth0_4(:,i)=Area6(i).*(RL2pressureRboth_0_1(:,i));
RL3ForceLboth0_4(:,i)=Area6(i).*(RL3pressureLboth_0_1(:,i));
RL3ForceRboth0_4(:,i)=Area6(i).*(RL3pressureRboth_0_1(:,i));

RC0ForceLboth0_4(:,i)=Area6(i).*(RC0pressureLboth_0_1(:,i));
RC0ForceRboth0_4(:,i)=Area6(i).*(RC0pressureRboth_0_1(:,i));
RC1ForceLboth0_4(:,i)=Area6(i).*(RC1pressureLboth_0_1(:,i));
RC1ForceRboth0_4(:,i)=Area6(i).*(RC1pressureRboth_0_1(:,i));
RC2ForceLboth0_4(:,i)=Area6(i).*(RC2pressureLboth_0_1(:,i));
RC2ForceRboth0_4(:,i)=Area6(i).*(RC2pressureRboth_0_1(:,i));
RC3ForceLboth0_4(:,i)=Area6(i).*(RC3pressureLboth_0_1(:,i));
RC3ForceRboth0_4(:,i)=Area6(i).*(RC3pressureRboth_0_1(:,i));

RS0ForceLboth0_4(:,i)=Area6(i).*(RS0pressureLboth_0_1(:,i));
RS0ForceRboth0_4(:,i)=Area6(i).*(RS0pressureRboth_0_1(:,i));
RS1ForceLboth0_4(:,i)=Area6(i).*(RS1pressureLboth_0_1(:,i));
RS1ForceRboth0_4(:,i)=Area6(i).*(RS1pressureRboth_0_1(:,i));
```

```
RS2ForceLboth0_4(:,i)=Area6(i).*(RS2pressureLboth_0_1(:,i));
RS2ForceRboth0_4(:,i)=Area6(i).*(RS2pressureRboth_0_1(:,i));
RS3ForceLboth0_4(:,i)=Area6(i).*(RS3pressureLboth_0_1(:,i));
RS3ForceRboth0_4(:,i)=Area6(i).*(RS3pressureRboth_0_1(:,i));



SL0ForceLboth0_4(:,i)=Area6(i).*(SL0pressureLboth_0_1(:,i));
SL0ForceRboth0_4(:,i)=Area6(i).*(SL0pressureRboth_0_1(:,i));
SL1ForceLboth0_4(:,i)=Area6(i).*(SL1pressureLboth_0_1(:,i));
SL1ForceRboth0_4(:,i)=Area6(i).*(SL1pressureRboth_0_1(:,i));
SL2ForceLboth0_4(:,i)=Area6(i).*(SL2pressureLboth_0_1(:,i));
SL2ForceRboth0_4(:,i)=Area6(i).*(SL2pressureRboth_0_1(:,i));
SL3ForceLboth0_4(:,i)=Area6(i).*(SL3pressureLboth_0_1(:,i));
SL3ForceRboth0_4(:,i)=Area6(i).*(SL3pressureRboth_0_1(:,i));

SC0ForceLboth0_4(:,i)=Area6(i).*(SC0pressureLboth_0_1(:,i));
SC0ForceRboth0_4(:,i)=Area6(i).*(SC0pressureRboth_0_1(:,i));
SC1ForceLboth0_4(:,i)=Area6(i).*(SC1pressureLboth_0_1(:,i));
SC1ForceRboth0_4(:,i)=Area6(i).*(SC1pressureRboth_0_1(:,i));
SC2ForceLboth0_4(:,i)=Area6(i).*(SC2pressureLboth_0_1(:,i));
SC2ForceRboth0_4(:,i)=Area6(i).*(SC2pressureRboth_0_1(:,i));
SC3ForceLboth0_4(:,i)=Area6(i).*(SC3pressureLboth_0_1(:,i));
SC3ForceRboth0_4(:,i)=Area6(i).*(SC3pressureRboth_0_1(:,i));

SS0ForceLboth0_4(:,i)=Area6(i).*(SS0pressureLboth_0_1(:,i));
SS0ForceRboth0_4(:,i)=Area6(i).*(SS0pressureRboth_0_1(:,i));
SS1ForceLboth0_4(:,i)=Area6(i).*(SS1pressureLboth_0_1(:,i));
SS1ForceRboth0_4(:,i)=Area6(i).*(SS1pressureRboth_0_1(:,i));
SS2ForceLboth0_4(:,i)=Area6(i).*(SS2pressureLboth_0_1(:,i));
SS2ForceRboth0_4(:,i)=Area6(i).*(SS2pressureRboth_0_1(:,i));
SS3ForceLboth0_4(:,i)=Area6(i).*(SS3pressureLboth_0_1(:,i));
```

```
  SS3ForceRboth0_4(:,i)=Area6(i).*(SS3pressureRboth_0_1(:,i));

   i=i+1;
end
for j=1:length(SF6)

  AL0dForceLboth0_1(:,j)=SF6(j).*(AL0ForceLboth0_1(:,j));
  AL0dForceRboth0_1(:,j)=SF6(j).*(AL0ForceRboth0_1(:,j));
  AL1dForceLboth0_1(:,j)=SF6(j).*(AL1ForceLboth0_1(:,j));
  AL1dForceRboth0_1(:,j)=SF6(j).*(AL1ForceRboth0_1(:,j));
  AL2dForceLboth0_1(:,j)=SF6(j).*(AL2ForceLboth0_1(:,j));
  AL2dForceRboth0_1(:,j)=SF6(j).*(AL2ForceRboth0_1(:,j));
  AL3dForceLboth0_1(:,j)=SF6(j).*(AL3ForceLboth0_1(:,j));
  AL3dForceRboth0_1(:,j)=SF6(j).*(AL3ForceRboth0_1(:,j));

  AC0dForceLboth0_1(:,j)=SF6(j).*(AC0ForceLboth0_1(:,j));
  AC0dForceRboth0_1(:,j)=SF6(j).*(AC0ForceRboth0_1(:,j));
  AC1dForceLboth0_1(:,j)=SF6(j).*(AC1ForceLboth0_1(:,j));
  AC1dForceRboth0_1(:,j)=SF6(j).*(AC1ForceRboth0_1(:,j));
  AC2dForceLboth0_1(:,j)=SF6(j).*(AC2ForceLboth0_1(:,j));
  AC2dForceRboth0_1(:,j)=SF6(j).*(AC2ForceRboth0_1(:,j));
  AC3dForceLboth0_1(:,j)=SF6(j).*(AC3ForceLboth0_1(:,j));
  AC3dForceRboth0_1(:,j)=SF6(j).*(AC3ForceRboth0_1(:,j));

  AS0dForceLboth0_1(:,j)=SF6(j).*(AS0ForceLboth0_1(:,j));
  AS0dForceRboth0_1(:,j)=SF6(j).*(AS0ForceRboth0_1(:,j));
  AS1dForceLboth0_1(:,j)=SF6(j).*(AS1ForceLboth0_1(:,j));
  AS1dForceRboth0_1(:,j)=SF6(j).*(AS1ForceRboth0_1(:,j));
  AS2dForceLboth0_1(:,j)=SF6(j).*(AS2ForceLboth0_1(:,j));
  AS2dForceRboth0_1(:,j)=SF6(j).*(AS2ForceRboth0_1(:,j));
  AS3dForceLboth0_1(:,j)=SF6(j).*(AS3ForceLboth0_1(:,j));
  AS3dForceRboth0_1(:,j)=SF6(j).*(AS3ForceRboth0_1(:,j));
```

```
CL0dForceLboth0_1(:,j)=SF6(j).*(CL0ForceLboth0_2(:,j));
CL0dForceRboth0_1(:,j)=SF6(j).*(CL0ForceRboth0_2(:,j));
CL1dForceLboth0_1(:,j)=SF6(j).*(CL1ForceLboth0_2(:,j));
CL1dForceRboth0_1(:,j)=SF6(j).*(CL1ForceRboth0_2(:,j));
CL2dForceLboth0_1(:,j)=SF6(j).*(CL2ForceLboth0_2(:,j));
CL2dForceRboth0_1(:,j)=SF6(j).*(CL2ForceRboth0_2(:,j));
CL3dForceLboth0_1(:,j)=SF6(j).*(CL3ForceLboth0_2(:,j));
CL3dForceRboth0_1(:,j)=SF6(j).*(CL3ForceRboth0_2(:,j));

CC0dForceLboth0_1(:,j)=SF6(j).*(CC0ForceLboth0_2(:,j));
CC0dForceRboth0_1(:,j)=SF6(j).*(CC0ForceRboth0_2(:,j));
CC1dForceLboth0_1(:,j)=SF6(j).*(CC1ForceLboth0_2(:,j));
CC1dForceRboth0_1(:,j)=SF6(j).*(CC1ForceRboth0_2(:,j));
CC2dForceLboth0_1(:,j)=SF6(j).*(CC2ForceLboth0_2(:,j));
CC2dForceRboth0_1(:,j)=SF6(j).*(CC2ForceRboth0_2(:,j));
CC3dForceLboth0_1(:,j)=SF6(j).*(CC3ForceLboth0_2(:,j));
CC3dForceRboth0_1(:,j)=SF6(j).*(CC3ForceRboth0_2(:,j));

CS0dForceLboth0_1(:,j)=SF6(j).*(CL0ForceLboth0_2(:,j));
CS0dForceRboth0_1(:,j)=SF6(j).*(CS0ForceRboth0_2(:,j));
CS1dForceLboth0_1(:,j)=SF6(j).*(CS1ForceLboth0_2(:,j));
CS1dForceRboth0_1(:,j)=SF6(j).*(CS1ForceRboth0_2(:,j));
CS2dForceLboth0_1(:,j)=SF6(j).*(CS2ForceLboth0_2(:,j));
CS2dForceRboth0_1(:,j)=SF6(j).*(CS2ForceRboth0_2(:,j));
CS3dForceLboth0_1(:,j)=SF6(j).*(CS3ForceLboth0_2(:,j));
CS3dForceRboth0_1(:,j)=SF6(j).*(CS3ForceRboth0_2(:,j));


JL0dForceLboth0_1(:,j)=SF6(j).*(JL0ForceLboth0_3(:,j));
JL0dForceRboth0_1(:,j)=SF6(j).*(JL0ForceRboth0_3(:,j));
JL1dForceLboth0_1(:,j)=SF6(j).*(JL1ForceLboth0_3(:,j));
```

```
JL1dForceRboth0_1(:,j)=SF6(j).*(JL1ForceRboth0_3(:,j));
JL2dForceLboth0_1(:,j)=SF6(j).*(JL2ForceLboth0_3(:,j));
JL2dForceRboth0_1(:,j)=SF6(j).*(JL2ForceRboth0_3(:,j));
JL3dForceLboth0_1(:,j)=SF6(j).*(JL3ForceLboth0_3(:,j));
JL3dForceRboth0_1(:,j)=SF6(j).*(JL3ForceRboth0_3(:,j));

JC0dForceLboth0_1(:,j)=SF6(j).*(JC0ForceLboth0_3(:,j));
JC0dForceRboth0_1(:,j)=SF6(j).*(JC0ForceRboth0_3(:,j));
JC1dForceLboth0_1(:,j)=SF6(j).*(JC1ForceLboth0_3(:,j));
JC1dForceRboth0_1(:,j)=SF6(j).*(JC1ForceRboth0_3(:,j));
JC2dForceLboth0_1(:,j)=SF6(j).*(JC2ForceLboth0_3(:,j));
JC2dForceRboth0_1(:,j)=SF6(j).*(JC2ForceRboth0_3(:,j));
JC3dForceLboth0_1(:,j)=SF6(j).*(JC3ForceLboth0_3(:,j));
JC3dForceRboth0_1(:,j)=SF6(j).*(JC3ForceRboth0_3(:,j));

JS0dForceLboth0_1(:,j)=SF6(j).*(JS0ForceLboth0_3(:,j));
JS0dForceRboth0_1(:,j)=SF6(j).*(JS0ForceRboth0_3(:,j));
JS1dForceLboth0_1(:,j)=SF6(j).*(JS1ForceLboth0_3(:,j));
JS1dForceRboth0_1(:,j)=SF6(j).*(JS1ForceRboth0_3(:,j));
JS2dForceLboth0_1(:,j)=SF6(j).*(JS2ForceLboth0_3(:,j));
JS2dForceRboth0_1(:,j)=SF6(j).*(JS2ForceRboth0_3(:,j));
JS3dForceLboth0_1(:,j)=SF6(j).*(JS3ForceLboth0_3(:,j));
JS3dForceRboth0_1(:,j)=SF6(j).*(JS3ForceRboth0_3(:,j));


ML0dForceLboth0_1(:,j)=SF6(j).*(ML0ForceLboth0_4(:,j));
ML0dForceRboth0_1(:,j)=SF6(j).*(ML0ForceRboth0_4(:,j));
ML1dForceLboth0_1(:,j)=SF6(j).*(ML1ForceLboth0_4(:,j));
ML1dForceRboth0_1(:,j)=SF6(j).*(ML1ForceRboth0_4(:,j));
ML2dForceLboth0_1(:,j)=SF6(j).*(ML2ForceLboth0_4(:,j));
ML2dForceRboth0_1(:,j)=SF6(j).*(ML2ForceRboth0_4(:,j));
ML3dForceLboth0_1(:,j)=SF6(j).*(ML3ForceLboth0_4(:,j));
```

```
ML3dForceRboth0_1(:,j)=SF6(j).*(ML3ForceRboth0_4(:,j));

MC0dForceLboth0_1(:,j)=SF6(j).*(MC0ForceLboth0_4(:,j));
MC0dForceRboth0_1(:,j)=SF6(j).*(MC0ForceRboth0_4(:,j));
MC1dForceLboth0_1(:,j)=SF6(j).*(MC1ForceLboth0_4(:,j));
MC1dForceRboth0_1(:,j)=SF6(j).*(MC1ForceRboth0_4(:,j));
MC2dForceLboth0_1(:,j)=SF6(j).*(MC2ForceLboth0_4(:,j));
MC2dForceRboth0_1(:,j)=SF6(j).*(MC2ForceRboth0_4(:,j));
MC3dForceLboth0_1(:,j)=SF6(j).*(MC3ForceLboth0_4(:,j));
MC3dForceRboth0_1(:,j)=SF6(j).*(MC3ForceRboth0_4(:,j));

MS0dForceLboth0_1(:,j)=SF6(j).*(MS0ForceLboth0_4(:,j));
MS0dForceRboth0_1(:,j)=SF6(j).*(MS0ForceRboth0_4(:,j));
MS1dForceLboth0_1(:,j)=SF6(j).*(MS1ForceLboth0_4(:,j));
MS1dForceRboth0_1(:,j)=SF6(j).*(MS1ForceRboth0_4(:,j));
MS2dForceLboth0_1(:,j)=SF6(j).*(MS2ForceLboth0_4(:,j));
MS2dForceRboth0_1(:,j)=SF6(j).*(MS2ForceRboth0_4(:,j));
MS3dForceLboth0_1(:,j)=SF6(j).*(MS3ForceLboth0_4(:,j));
MS3dForceRboth0_1(:,j)=SF6(j).*(MS3ForceRboth0_4(:,j));


RL0dForceLboth0_1(:,j)=SF6(j).*(RL0ForceLboth0_4(:,j));
RL0dForceRboth0_1(:,j)=SF6(j).*(RL0ForceRboth0_4(:,j));
RL1dForceLboth0_1(:,j)=SF6(j).*(RL1ForceLboth0_4(:,j));
RL1dForceRboth0_1(:,j)=SF6(j).*(RL1ForceRboth0_4(:,j));
RL2dForceLboth0_1(:,j)=SF6(j).*(RL2ForceLboth0_4(:,j));
RL2dForceRboth0_1(:,j)=SF6(j).*(RL2ForceRboth0_4(:,j));
RL3dForceLboth0_1(:,j)=SF6(j).*(RL3ForceLboth0_4(:,j));
RL3dForceRboth0_1(:,j)=SF6(j).*(RL3ForceRboth0_4(:,j));

RC0dForceLboth0_1(:,j)=SF6(j).*(RC0ForceLboth0_4(:,j));
RC0dForceRboth0_1(:,j)=SF6(j).*(RC0ForceRboth0_4(:,j));
```

```
RC1dForceLboth0_1(:,j)=SF6(j).*(RC1ForceLboth0_4(:,j));
RC1dForceRboth0_1(:,j)=SF6(j).*(RC1ForceRboth0_4(:,j));
RC2dForceLboth0_1(:,j)=SF6(j).*(RC2ForceLboth0_4(:,j));
RC2dForceRboth0_1(:,j)=SF6(j).*(RC2ForceRboth0_4(:,j));
RC3dForceLboth0_1(:,j)=SF6(j).*(RC3ForceLboth0_4(:,j));
RC3dForceRboth0_1(:,j)=SF6(j).*(RC3ForceRboth0_4(:,j));

RS0dForceLboth0_1(:,j)=SF6(j).*(RS0ForceLboth0_4(:,j));
RS0dForceRboth0_1(:,j)=SF6(j).*(RS0ForceRboth0_4(:,j));
RS1dForceLboth0_1(:,j)=SF6(j).*(RS1ForceLboth0_4(:,j));
RS1dForceRboth0_1(:,j)=SF6(j).*(RS1ForceRboth0_4(:,j));
RS2dForceLboth0_1(:,j)=SF6(j).*(RS2ForceLboth0_4(:,j));
RS2dForceRboth0_1(:,j)=SF6(j).*(RS2ForceRboth0_4(:,j));
RS3dForceLboth0_1(:,j)=SF6(j).*(RS3ForceLboth0_4(:,j));
RS3dForceRboth0_1(:,j)=SF6(j).*(RS3ForceRboth0_4(:,j));

SL0dForceLboth0_1(:,j)=SF6(j).*(SL0ForceLboth0_4(:,j));
SL0dForceRboth0_1(:,j)=SF6(j).*(SL0ForceRboth0_4(:,j));
SL1dForceLboth0_1(:,j)=SF6(j).*(SL1ForceLboth0_4(:,j));
SL1dForceRboth0_1(:,j)=SF6(j).*(SL1ForceRboth0_4(:,j));
SL2dForceLboth0_1(:,j)=SF6(j).*(SL2ForceLboth0_4(:,j));
SL2dForceRboth0_1(:,j)=SF6(j).*(SL2ForceRboth0_4(:,j));
SL3dForceLboth0_1(:,j)=SF6(j).*(SL3ForceLboth0_4(:,j));
SL3dForceRboth0_1(:,j)=SF6(j).*(SL3ForceRboth0_4(:,j));

SC0dForceLboth0_1(:,j)=SF6(j).*(SC0ForceLboth0_4(:,j));
SC0dForceRboth0_1(:,j)=SF6(j).*(SC0ForceRboth0_4(:,j));
SC1dForceLboth0_1(:,j)=SF6(j).*(SC1ForceLboth0_4(:,j));
SC1dForceRboth0_1(:,j)=SF6(j).*(SC1ForceRboth0_4(:,j));
SC2dForceLboth0_1(:,j)=SF6(j).*(SC2ForceLboth0_4(:,j));
SC2dForceRboth0_1(:,j)=SF6(j).*(SC2ForceRboth0_4(:,j));
SC3dForceLboth0_1(:,j)=SF6(j).*(SC3ForceLboth0_4(:,j));
```

```
        SC3dForceRboth0_1(:,j)=SF6(j).*(SC3ForceRboth0_4(:,j));

        SS0dForceLboth0_1(:,j)=SF6(j).*(SC0ForceLboth0_4(:,j));
        SS0dForceRboth0_1(:,j)=SF6(j).*(SC0ForceRboth0_4(:,j));
        SS1dForceLboth0_1(:,j)=SF6(j).*(SC1ForceLboth0_4(:,j));
        SS1dForceRboth0_1(:,j)=SF6(j).*(SC1ForceRboth0_4(:,j));
        SS2dForceLboth0_1(:,j)=SF6(j).*(SC2ForceLboth0_4(:,j));
        SS2dForceRboth0_1(:,j)=SF6(j).*(SC2ForceRboth0_4(:,j));
        SS3dForceLboth0_1(:,j)=SF6(j).*(SC3ForceLboth0_4(:,j));
        SS3dForceRboth0_1(:,j)=SF6(j).*(SC3ForceRboth0_4(:,j));



        i=i+1;
    end
    %% Find Metatarsal head partial vGRF measurements

AL0TFLboth0_met1=AL0dForceLboth0_1(:,9)+AL0dForceLboth0_1(:,10)+AL0dForceLboth0_1(:,14);

AL0TFRboth0_met1=AL0dForceRboth0_1(:,9)+AL0dForceRboth0_1(:,10)+AL0dForceRboth0_1(:,14);

AL1TFLboth0_met1=AL1dForceLboth0_1(:,9)+AL1dForceLboth0_1(:,10)+AL1dForceLboth0_1(:,14);

AL1TFRboth0_met1=AL1dForceRboth0_1(:,9)+AL1dForceRboth0_1(:,10)+AL1dForceRboth0_1(:,14);

AL2TFLboth0_met1=AL2dForceLboth0_1(:,9)+AL2dForceLboth0_1(:,10)+AL2dForceLboth0_1(:,14);

AL2TFRboth0_met1=AL2dForceRboth0_1(:,9)+AL2dForceRboth0_1(:,10)+AL2dForceRboth0_1(:,14);
```

```
AL3TFLboth0_met1=AL3dForceLboth0_1(:,9)+AL3dForceLboth0_1(:,10)+AL3dForceLboth0_1(:,14);

AL3TFRboth0_met1=AL3dForceRboth0_1(:,9)+AL3dForceRboth0_1(:,10)+AL3dForceRboth0_1(:,14);


AC0TFLboth0_met1=AC0dForceLboth0_1(:,9)+AC0dForceLboth0_1(:,10)+AC0dForceLboth0_1(:,14);

AC0TFRboth0_met1=AC0dForceRboth0_1(:,9)+AC0dForceRboth0_1(:,10)+AC0dForceRboth0_1(:,14);

AC1TFLboth0_met1=AC1dForceLboth0_1(:,9)+AC1dForceLboth0_1(:,10)+AC1dForceLboth0_1(:,14);

AC1TFRboth0_met1=AC1dForceRboth0_1(:,9)+AC1dForceRboth0_1(:,10)+AC1dForceRboth0_1(:,14);

AC2TFLboth0_met1=AC2dForceLboth0_1(:,9)+AC2dForceLboth0_1(:,10)+AC2dForceLboth0_1(:,14);

AC2TFRboth0_met1=AC2dForceRboth0_1(:,9)+AC2dForceRboth0_1(:,10)+AC2dForceRboth0_1(:,14);

AC3TFLboth0_met1=AC3dForceLboth0_1(:,9)+AC3dForceLboth0_1(:,10)+AC3dForceLboth0_1(:,14);

AC3TFRboth0_met1=AC3dForceRboth0_1(:,9)+AC3dForceRboth0_1(:,10)+AC3dForceRboth0_1(:,14);


AS0TFLboth0_met1=AS0dForceLboth0_1(:,9)+AS0dForceLboth0_1(:,10)+AS0dForceLboth0_1(:,14);

AS0TFRboth0_met1=AS0dForceRboth0_1(:,9)+AS0dForceRboth0_1(:,10)+AS0dForceRboth0_1(:,14);
```

AS1TFLboth0_met1=AS1dForceLboth0_1(:,9)+AS1dForceLboth0_1(:,10)+AS1dForceLboth0_1(:,14);

AS1TFRboth0_met1=AS1dForceRboth0_1(:,9)+AS1dForceRboth0_1(:,10)+AS1dForceRboth0_1(:,14);

AS2TFLboth0_met1=AS2dForceLboth0_1(:,9)+AS2dForceLboth0_1(:,10)+AS2dForceLboth0_1(:,14);

AS2TFRboth0_met1=AS2dForceRboth0_1(:,9)+AS2dForceRboth0_1(:,10)+AS2dForceRboth0_1(:,14);

AS3TFLboth0_met1=AS3dForceLboth0_1(:,9)+AS3dForceLboth0_1(:,10)+AS3dForceLboth0_1(:,14);

AS3TFRboth0_met1=AS3dForceRboth0_1(:,9)+AS3dForceRboth0_1(:,10)+AS3dForceRboth0_1(:,14);

CL0TFLboth0_met1=CL0dForceLboth0_1(:,9)+CL0dForceLboth0_1(:,10)+CL0dForceLboth0_1(:,14);

CL0TFRboth0_met1=CL0dForceRboth0_1(:,9)+CL0dForceRboth0_1(:,10)+CL0dForceRboth0_1(:,14);

CL1TFLboth0_met1=CL1dForceLboth0_1(:,9)+CL1dForceLboth0_1(:,10)+CL1dForceLboth0_1(:,14);

CL1TFRboth0_met1=CL1dForceRboth0_1(:,9)+CL1dForceRboth0_1(:,10)+CL1dForceRboth0_1(:,14);

CL2TFLboth0_met1=CL2dForceLboth0_1(:,9)+CL2dForceLboth0_1(:,10)+CL2dForceLboth0_1(:,14);

CL2TFRboth0_met1=CL2dForceRboth0_1(:,9)+CL2dForceRboth0_1(:,10)+CL2dForceRboth0_1(:,14);

CL3TFLboth0_met1=CL3dForceLboth0_1(:,9)+CL3dForceLboth0_1(:,10)+CL3dForceLboth0_1(:,14);

CL3TFRboth0_met1=CL3dForceRboth0_1(:,9)+CL3dForceRboth0_1(:,10)+CL3dForceRboth0_1(:,14);

CS0TFLboth0_met1=CS0dForceLboth0_1(:,9)+CS0dForceLboth0_1(:,10)+CS0dForceLboth0_1(:,14);

CS0TFRboth0_met1=CS0dForceRboth0_1(:,9)+CS0dForceRboth0_1(:,10)+CS0dForceRboth0_1(:,14);

CS1TFLboth0_met1=CS1dForceLboth0_1(:,9)+CS1dForceLboth0_1(:,10)+CS1dForceLboth0_1(:,14);

CS1TFRboth0_met1=CS1dForceRboth0_1(:,9)+CS1dForceRboth0_1(:,10)+CS1dForceRboth0_1(:,14);

CS2TFLboth0_met1=CS2dForceLboth0_1(:,9)+CS2dForceLboth0_1(:,10)+CS2dForceLboth0_1(:,14);

CS2TFRboth0_met1=CS2dForceRboth0_1(:,9)+CS2dForceRboth0_1(:,10)+CS2dForceRboth0_1(:,14);

CS3TFLboth0_met1=CS3dForceLboth0_1(:,9)+CS3dForceLboth0_1(:,10)+CS3dForceLboth0_1(:,14);

CS3TFRboth0_met1=CS3dForceRboth0_1(:,9)+CS3dForceRboth0_1(:,10)+CS3dForceRboth0_1(:,14);

CC0TFLboth0_met1=CC0dForceLboth0_1(:,9)+CC0dForceLboth0_1(:,10)+CC0dForceLboth0_1(:,14);

CC0TFRboth0_met1=CC0dForceRboth0_1(:,9)+CC0dForceRboth0_1(:,10)+CC0dForceRboth0_1(:,14);

CC1TFLboth0_met1=CC1dForceLboth0_1(:,9)+CC1dForceLboth0_1(:,10)+CC1dForceLboth0_1(:,14);

CC1TFRboth0_met1=CC1dForceRboth0_1(:,9)+CC1dForceRboth0_1(:,10)+CC1dForceRboth0_1(:,14);

CC2TFLboth0_met1=CC2dForceLboth0_1(:,9)+CC2dForceLboth0_1(:,10)+CC2dForceLboth0_1(:,14);

CC2TFRboth0_met1=CC2dForceRboth0_1(:,9)+CC2dForceRboth0_1(:,10)+CC2dForceRboth0_1(:,14);

CC3TFLboth0_met1=CC3dForceLboth0_1(:,9)+CC3dForceLboth0_1(:,10)+CC3dForceLboth0_1(:,14);

CC3TFRboth0_met1=CC3dForceRboth0_1(:,9)+CC3dForceRboth0_1(:,10)+CC3dForceRboth0_1(:,14);

JL0TFLboth0_met1=JL0dForceLboth0_1(:,9)+JL0dForceLboth0_1(:,10)+JL0dForceLboth0_1(:,14);

JL0TFRboth0_met1=JL0dForceRboth0_1(:,9)+JL0dForceRboth0_1(:,10)+JL0dForceRboth0_1(:,14);

JL1TFLboth0_met1=JL1dForceLboth0_1(:,9)+JL1dForceLboth0_1(:,10)+JL1dForceLboth0_1(:,14);

JL1TFRboth0_met1=JL1dForceRboth0_1(:,9)+JL1dForceRboth0_1(:,10)+JL1dForceRboth0_1(:,14);

JL2TFLboth0_met1=JL2dForceLboth0_1(:,9)+JL2dForceLboth0_1(:,10)+JL2dForceLboth0_1(:,14);

JL2TFRboth0_met1=JL2dForceRboth0_1(:,9)+JL2dForceRboth0_1(:,10)+JL2dForceRboth0_1(:,14);

JL3TFLboth0_met1=JL3dForceLboth0_1(:,9)+JL3dForceLboth0_1(:,10)+JL3dForceLboth0_1(:,14);

JL3TFRboth0_met1=JL3dForceRboth0_1(:,9)+JL3dForceRboth0_1(:,10)+JL3dForceRboth0_1(:,14);

JC0TFLboth0_met1=JC0dForceLboth0_1(:,9)+JC0dForceLboth0_1(:,10)+JC0dForceLboth0_1(:,14);

JC0TFRboth0_met1=JC0dForceRboth0_1(:,9)+JC0dForceRboth0_1(:,10)+JC0dForceRboth0_1(:,14);

```
JC1TFLboth0_met1=JC1dForceLboth0_1(:,9)+JC1dForceLboth0_1(:,10)+JC1dForceLboth0_1(:,14);

JC1TFRboth0_met1=JC1dForceRboth0_1(:,9)+JC1dForceRboth0_1(:,10)+JC1dForceRboth0_1(:,14)
;

JC2TFLboth0_met1=JC2dForceLboth0_1(:,9)+JC2dForceLboth0_1(:,10)+JC2dForceLboth0_1(:,14);

JC2TFRboth0_met1=JC2dForceRboth0_1(:,9)+JC2dForceRboth0_1(:,10)+JC2dForceRboth0_1(:,14)
;

JC3TFLboth0_met1=JC3dForceLboth0_1(:,9)+JC3dForceLboth0_1(:,10)+JC3dForceLboth0_1(:,14);

JC3TFRboth0_met1=JC3dForceRboth0_1(:,9)+JC3dForceRboth0_1(:,10)+JC3dForceRboth0_1(:,14)
;


JS0TFLboth0_met1=JS0dForceLboth0_1(:,9)+JS0dForceLboth0_1(:,10)+JS0dForceLboth0_1(:,14);

JS0TFRboth0_met1=JS0dForceRboth0_1(:,9)+JS0dForceRboth0_1(:,10)+JS0dForceRboth0_1(:,14);

JS1TFLboth0_met1=JS1dForceLboth0_1(:,9)+JS1dForceLboth0_1(:,10)+JS1dForceLboth0_1(:,14);

JS1TFRboth0_met1=JS1dForceRboth0_1(:,9)+JS1dForceRboth0_1(:,10)+JS1dForceRboth0_1(:,14);

JS2TFLboth0_met1=JS2dForceLboth0_1(:,9)+JS2dForceLboth0_1(:,10)+JS2dForceLboth0_1(:,14);

JS2TFRboth0_met1=JS2dForceRboth0_1(:,9)+JS2dForceRboth0_1(:,10)+JS2dForceRboth0_1(:,14);

JS3TFLboth0_met1=JS3dForceLboth0_1(:,9)+JS3dForceLboth0_1(:,10)+JS3dForceLboth0_1(:,14);

JS3TFRboth0_met1=JS3dForceRboth0_1(:,9)+JS3dForceRboth0_1(:,10)+JS3dForceRboth0_1(:,14);


ML0TFLboth0_met1=ML0dForceLboth0_1(:,9)+ML0dForceLboth0_1(:,10)+ML0dForceLboth0_1(:
,14);
```

ML0TFRboth0_met1=ML0dForceRboth0_1(:,9)+ML0dForceRboth0_1(:,10)+ML0dForceRboth0_1(:,14);

ML1TFLboth0_met1=ML1dForceLboth0_1(:,9)+ML1dForceLboth0_1(:,10)+ML1dForceLboth0_1(:,14);

ML1TFRboth0_met1=ML1dForceRboth0_1(:,9)+ML1dForceRboth0_1(:,10)+ML1dForceRboth0_1(:,14);

ML2TFLboth0_met1=ML2dForceLboth0_1(:,9)+ML2dForceLboth0_1(:,10)+ML2dForceLboth0_1(:,14);

ML2TFRboth0_met1=ML2dForceRboth0_1(:,9)+ML2dForceRboth0_1(:,10)+ML2dForceRboth0_1(:,14);

ML3TFLboth0_met1=ML3dForceLboth0_1(:,9)+ML3dForceLboth0_1(:,10)+ML3dForceLboth0_1(:,14);

ML3TFRboth0_met1=ML3dForceRboth0_1(:,9)+ML3dForceRboth0_1(:,10)+ML3dForceRboth0_1(:,14);

MC0TFLboth0_met1=MC0dForceLboth0_1(:,9)+MC0dForceLboth0_1(:,10)+MC0dForceLboth0_1(:,14);

MC0TFRboth0_met1=MC0dForceRboth0_1(:,9)+MC0dForceRboth0_1(:,10)+MC0dForceRboth0_1(:,14);

MC1TFLboth0_met1=MC1dForceLboth0_1(:,9)+MC1dForceLboth0_1(:,10)+MC1dForceLboth0_1(:,14);

MC1TFRboth0_met1=MC1dForceRboth0_1(:,9)+MC1dForceRboth0_1(:,10)+MC1dForceRboth0_1(:,14);

MC2TFLboth0_met1=MC2dForceLboth0_1(:,9)+MC2dForceLboth0_1(:,10)+MC2dForceLboth0_1(:,14);

MC2TFRboth0_met1=MC2dForceRboth0_1(:,9)+MC2dForceRboth0_1(:,10)+MC2dForceRboth0_1(:,14);

```
MC3TFLboth0_met1=MC3dForceLboth0_1(:,9)+MC3dForceLboth0_1(:,10)+MC3dForceLboth0_1(
:,14);

MC3TFRboth0_met1=MC3dForceRboth0_1(:,9)+MC3dForceRboth0_1(:,10)+MC3dForceRboth0_1
(:,14);



MS0TFLboth0_met1=MS0dForceLboth0_1(:,9)+MS0dForceLboth0_1(:,10)+MS0dForceLboth0_1(:,
14);

MS0TFRboth0_met1=MS0dForceRboth0_1(:,9)+MS0dForceRboth0_1(:,10)+MS0dForceRboth0_1(:
,14);

MS1TFLboth0_met1=MS1dForceLboth0_1(:,9)+MS1dForceLboth0_1(:,10)+MS1dForceLboth0_1(:,
14);

MS1TFRboth0_met1=MS1dForceRboth0_1(:,9)+MS1dForceRboth0_1(:,10)+MS1dForceRboth0_1(:
,14);

MS2TFLboth0_met1=MS2dForceLboth0_1(:,9)+MS2dForceLboth0_1(:,10)+MS2dForceLboth0_1(:,
14);

MS2TFRboth0_met1=MS2dForceRboth0_1(:,9)+MS2dForceRboth0_1(:,10)+MS2dForceRboth0_1(:
,14);

MS3TFLboth0_met1=MS3dForceLboth0_1(:,9)+MS3dForceLboth0_1(:,10)+MS3dForceLboth0_1(:,
14);

MS3TFRboth0_met1=MS3dForceRboth0_1(:,9)+MS3dForceRboth0_1(:,10)+MS3dForceRboth0_1(:
,14);



RL0TFLboth0_met1=RL0dForceLboth0_1(:,9)+RL0dForceLboth0_1(:,10)+RL0dForceLboth0_1(:,1
4);

RL0TFRboth0_met1=RL0dForceRboth0_1(:,9)+RL0dForceRboth0_1(:,10)+RL0dForceRboth0_1(:,1
4);
```

RL1TFLboth0_met1=RL1dForceLboth0_1(:,9)+RL1dForceLboth0_1(:,10)+RL1dForceLboth0_1(:,14);

RL1TFRboth0_met1=RL1dForceRboth0_1(:,9)+RL1dForceRboth0_1(:,10)+RL1dForceRboth0_1(:,14);

RL2TFLboth0_met1=RL2dForceLboth0_1(:,9)+RL2dForceLboth0_1(:,10)+RL2dForceLboth0_1(:,14);

RL2TFRboth0_met1=RL2dForceRboth0_1(:,9)+RL2dForceRboth0_1(:,10)+RL2dForceRboth0_1(:,14);

RL3TFLboth0_met1=RL3dForceLboth0_1(:,9)+RL3dForceLboth0_1(:,10)+RL3dForceLboth0_1(:,14);

RL3TFRboth0_met1=RL3dForceRboth0_1(:,9)+RL3dForceRboth0_1(:,10)+RL3dForceRboth0_1(:,14);

RC0TFLboth0_met1=RC0dForceLboth0_1(:,9)+RC0dForceLboth0_1(:,10)+RC0dForceLboth0_1(:,14);

RC0TFRboth0_met1=RC0dForceRboth0_1(:,9)+RC0dForceRboth0_1(:,10)+RC0dForceRboth0_1(:,14);

RC1TFLboth0_met1=RC1dForceLboth0_1(:,9)+RC1dForceLboth0_1(:,10)+RC1dForceLboth0_1(:,14);

RC1TFRboth0_met1=RC1dForceRboth0_1(:,9)+RC1dForceRboth0_1(:,10)+RC1dForceRboth0_1(:,14);

RC2TFLboth0_met1=RC2dForceLboth0_1(:,9)+RC2dForceLboth0_1(:,10)+RC2dForceLboth0_1(:,14);

RC2TFRboth0_met1=RC2dForceRboth0_1(:,9)+RC2dForceRboth0_1(:,10)+RC2dForceRboth0_1(:,14);

RC3TFLboth0_met1=RC3dForceLboth0_1(:,9)+RC3dForceLboth0_1(:,10)+RC3dForceLboth0_1(:,14);

RC3TFRboth0_met1=RC3dForceRboth0_1(:,9)+RC3dForceRboth0_1(:,10)+RC3dForceRboth0_1(:,14);

RS0TFLboth0_met1=RS0dForceLboth0_1(:,9)+RS0dForceLboth0_1(:,10)+RS0dForceLboth0_1(:,14);

RS0TFRboth0_met1=RS0dForceRboth0_1(:,9)+RS0dForceRboth0_1(:,10)+RS0dForceRboth0_1(:,14);

RS1TFLboth0_met1=RS1dForceLboth0_1(:,9)+RS1dForceLboth0_1(:,10)+RS1dForceLboth0_1(:,14);

RS1TFRboth0_met1=RS1dForceRboth0_1(:,9)+RS1dForceRboth0_1(:,10)+RS1dForceRboth0_1(:,14);

RS2TFLboth0_met1=RS2dForceLboth0_1(:,9)+RS2dForceLboth0_1(:,10)+RS2dForceLboth0_1(:,14);

RS2TFRboth0_met1=RS2dForceRboth0_1(:,9)+RS2dForceRboth0_1(:,10)+RS2dForceRboth0_1(:,14);

RS3TFLboth0_met1=RS3dForceLboth0_1(:,9)+RS3dForceLboth0_1(:,10)+RS3dForceLboth0_1(:,14);

RS3TFRboth0_met1=RS3dForceRboth0_1(:,9)+RS3dForceRboth0_1(:,10)+RS3dForceRboth0_1(:,14);

SL0TFLboth0_met1=SL0dForceLboth0_1(:,9)+SL0dForceLboth0_1(:,10)+SL0dForceLboth0_1(:,14);

SL0TFRboth0_met1=SL0dForceRboth0_1(:,9)+SL0dForceRboth0_1(:,10)+SL0dForceRboth0_1(:,14);

SL1TFLboth0_met1=SL1dForceLboth0_1(:,9)+SL1dForceLboth0_1(:,10)+SL1dForceLboth0_1(:,14);

```
SL1TFRboth0_met1=SL1dForceRboth0_1(:,9)+SL1dForceRboth0_1(:,10)+SL1dForceRboth0_1(:,14
);

SL2TFLboth0_met1=SL2dForceLboth0_1(:,9)+SL2dForceLboth0_1(:,10)+SL2dForceLboth0_1(:,14
);

SL2TFRboth0_met1=SL2dForceRboth0_1(:,9)+SL2dForceRboth0_1(:,10)+SL2dForceRboth0_1(:,14
);

SL3TFLboth0_met1=SL3dForceLboth0_1(:,9)+SL3dForceLboth0_1(:,10)+SL3dForceLboth0_1(:,14
);

SL3TFRboth0_met1=SL3dForceRboth0_1(:,9)+SL3dForceRboth0_1(:,10)+SL3dForceRboth0_1(:,14
);



SC0TFLboth0_met1=SC0dForceLboth0_1(:,9)+SC0dForceLboth0_1(:,10)+SC0dForceLboth0_1(:,14
);

SC0TFRboth0_met1=SC0dForceRboth0_1(:,9)+SC0dForceRboth0_1(:,10)+SC0dForceRboth0_1(:,1
4);

SC1TFLboth0_met1=SC1dForceLboth0_1(:,9)+SC1dForceLboth0_1(:,10)+SC1dForceLboth0_1(:,14
);

SC1TFRboth0_met1=SC1dForceRboth0_1(:,9)+SC1dForceRboth0_1(:,10)+SC1dForceRboth0_1(:,1
4);

SC2TFLboth0_met1=SC2dForceLboth0_1(:,9)+SC2dForceLboth0_1(:,10)+SC2dForceLboth0_1(:,14
);

SC2TFRboth0_met1=SC2dForceRboth0_1(:,9)+SC2dForceRboth0_1(:,10)+SC2dForceRboth0_1(:,1
4);

SC3TFLboth0_met1=SC3dForceLboth0_1(:,9)+SC3dForceLboth0_1(:,10)+SC3dForceLboth0_1(:,14
);

SC3TFRboth0_met1=SC3dForceRboth0_1(:,9)+SC3dForceRboth0_1(:,10)+SC3dForceRboth0_1(:,1
4);
```

```
SS0TFLboth0_met1=SS0dForceLboth0_1(:,9)+SS0dForceLboth0_1(:,10)+SS0dForceLboth0_1(:,14)
;

SS0TFRboth0_met1=SS0dForceRboth0_1(:,9)+SS0dForceRboth0_1(:,10)+SS0dForceRboth0_1(:,14
);

SS1TFLboth0_met1=SS1dForceLboth0_1(:,9)+SS1dForceLboth0_1(:,10)+SS1dForceLboth0_1(:,14)
;

SS1TFRboth0_met1=SS1dForceRboth0_1(:,9)+SS1dForceRboth0_1(:,10)+SS1dForceRboth0_1(:,14
);

SS2TFLboth0_met1=SS2dForceLboth0_1(:,9)+SS2dForceLboth0_1(:,10)+SS2dForceLboth0_1(:,14)
;

SS2TFRboth0_met1=SS2dForceRboth0_1(:,9)+SS2dForceRboth0_1(:,10)+SS2dForceRboth0_1(:,14
);

SS3TFLboth0_met1=SS3dForceLboth0_1(:,9)+SS3dForceLboth0_1(:,10)+SS3dForceLboth0_1(:,14)
;

SS3TFRboth0_met1=SS3dForceRboth0_1(:,9)+SS3dForceRboth0_1(:,10)+SS3dForceRboth0_1(:,14
);


    %% Removing High Frequency noise
    AL0TFLboth0_met_1=movmean(AL0TFLboth0_met1,2);
    AL0TFRboth0_met_1=movmean(AL0TFRboth0_met1,2);
    AL1TFLboth0_met_1=movmean(AL1TFLboth0_met1,2);
    AL1TFRboth0_met_1=movmean(AL1TFRboth0_met1,2);
    AL2TFLboth0_met_1=movmean(AL2TFLboth0_met1,2);
    AL2TFRboth0_met_1=movmean(AL2TFRboth0_met1,2);
    AL3TFLboth0_met_1=movmean(AL3TFLboth0_met1,2);
    AL3TFRboth0_met_1=movmean(AL3TFRboth0_met1,2);

    AC0TFLboth0_met_1=movmean(AC0TFLboth0_met1,2);
```

```
AC0TFRboth0_met_1=movmean(AC0TFRboth0_met1,2);
AC1TFLboth0_met_1=movmean(AC1TFLboth0_met1,2);
AC1TFRboth0_met_1=movmean(AC1TFRboth0_met1,2);
AC2TFLboth0_met_1=movmean(AC2TFLboth0_met1,2);
AC2TFRboth0_met_1=movmean(AC2TFRboth0_met1,2);
AC3TFLboth0_met_1=movmean(AC3TFLboth0_met1,2);
AC3TFRboth0_met_1=movmean(AC3TFRboth0_met1,2);



AS0TFLboth0_met_1=movmean(AS0TFLboth0_met1,2);
AS0TFRboth0_met_1=movmean(AS0TFRboth0_met1,2);
AS1TFLboth0_met_1=movmean(AS1TFLboth0_met1,2);
AS1TFRboth0_met_1=movmean(AS1TFRboth0_met1,2);
AS2TFLboth0_met_1=movmean(AS2TFLboth0_met1,2);
AS2TFRboth0_met_1=movmean(AS2TFRboth0_met1,2);
AS3TFLboth0_met_1=movmean(AS3TFLboth0_met1,2);
AS3TFRboth0_met_1=movmean(AS3TFRboth0_met1,2);



CL0TFLboth0_met_1=movmean(CL0TFLboth0_met1,2);
CL0TFRboth0_met_1=movmean(CL0TFRboth0_met1,2);
CL1TFLboth0_met_1=movmean(CL1TFLboth0_met1,2);
CL1TFRboth0_met_1=movmean(CL1TFRboth0_met1,2);
CL2TFLboth0_met_1=movmean(CL2TFLboth0_met1,2);
CL2TFRboth0_met_1=movmean(CL2TFRboth0_met1,2);
CL3TFLboth0_met_1=movmean(CL3TFLboth0_met1,2);
CL3TFRboth0_met_1=movmean(CL3TFRboth0_met1,2);

CS0TFLboth0_met_1=movmean(CS0TFLboth0_met1,2);
CS0TFRboth0_met_1=movmean(CS0TFRboth0_met1,2);
```

```
CS1TFLboth0_met_1=movmean(CS1TFLboth0_met1,2);
CS1TFRboth0_met_1=movmean(CS1TFRboth0_met1,2);
CS2TFLboth0_met_1=movmean(CS2TFLboth0_met1,2);
CS2TFRboth0_met_1=movmean(CS2TFRboth0_met1,2);
CS3TFLboth0_met_1=movmean(CS3TFLboth0_met1,2);
CS3TFRboth0_met_1=movmean(CS3TFRboth0_met1,2);

CC0TFLboth0_met_1=movmean(CC0TFLboth0_met1,2);
CC0TFRboth0_met_1=movmean(CC0TFRboth0_met1,2);
CC1TFLboth0_met_1=movmean(CC1TFLboth0_met1,2);
CC1TFRboth0_met_1=movmean(CC1TFRboth0_met1,2);
CC2TFLboth0_met_1=movmean(CC2TFLboth0_met1,2);
CC2TFRboth0_met_1=movmean(CC2TFRboth0_met1,2);
CC3TFLboth0_met_1=movmean(CC3TFLboth0_met1,2);
CC3TFRboth0_met_1=movmean(CC3TFRboth0_met1,2);

JL0TFLboth0_met_1=movmean(JL0TFLboth0_met1,2);
JL0TFRboth0_met_1=movmean(JL0TFRboth0_met1,2);
JL1TFLboth0_met_1=movmean(JL1TFLboth0_met1,2);
JL1TFRboth0_met_1=movmean(JL1TFRboth0_met1,2);
JL2TFLboth0_met_1=movmean(JL2TFLboth0_met1,2);
JL2TFRboth0_met_1=movmean(JL2TFRboth0_met1,2);
JL3TFLboth0_met_1=movmean(JL3TFLboth0_met1,2);
JL3TFRboth0_met_1=movmean(JL3TFRboth0_met1,2);

JC0TFLboth0_met_1=movmean(JC0TFLboth0_met1,2);
JC0TFRboth0_met_1=movmean(JC0TFRboth0_met1,2);
JC1TFLboth0_met_1=movmean(JC1TFLboth0_met1,2);
JC1TFRboth0_met_1=movmean(JC1TFRboth0_met1,2);
JC2TFLboth0_met_1=movmean(JC2TFLboth0_met1,2);
JC2TFRboth0_met_1=movmean(JC2TFRboth0_met1,2);
JC3TFLboth0_met_1=movmean(JC3TFLboth0_met1,2);
```

```
JC3TFRboth0_met_1=movmean(JC3TFRboth0_met1,2);

JS0TFLboth0_met_1=movmean(JS0TFLboth0_met1,2);
JS0TFRboth0_met_1=movmean(JS0TFRboth0_met1,2);
JS1TFLboth0_met_1=movmean(JS1TFLboth0_met1,2);
JS1TFRboth0_met_1=movmean(JS1TFRboth0_met1,2);
JS2TFLboth0_met_1=movmean(JS2TFLboth0_met1,2);
JS2TFRboth0_met_1=movmean(JS2TFRboth0_met1,2);
JS3TFLboth0_met_1=movmean(JS3TFLboth0_met1,2);
JS3TFRboth0_met_1=movmean(JS3TFRboth0_met1,2);

ML0TFLboth0_met_1=movmean(ML0TFLboth0_met1,2);
ML0TFRboth0_met_1=movmean(ML0TFRboth0_met1,2);
ML1TFLboth0_met_1=movmean(ML1TFLboth0_met1,2);
ML1TFRboth0_met_1=movmean(ML1TFRboth0_met1,2);
ML2TFLboth0_met_1=movmean(ML2TFLboth0_met1,2);
ML2TFRboth0_met_1=movmean(ML2TFRboth0_met1,2);
ML3TFLboth0_met_1=movmean(ML3TFLboth0_met1,2);
ML3TFRboth0_met_1=movmean(ML3TFRboth0_met1,2);

MC0TFLboth0_met_1=movmean(MC0TFLboth0_met1,2);
MC0TFRboth0_met_1=movmean(MC0TFRboth0_met1,2);
MC1TFLboth0_met_1=movmean(MC1TFLboth0_met1,2);
MC1TFRboth0_met_1=movmean(MC1TFRboth0_met1,2);
MC2TFLboth0_met_1=movmean(MC2TFLboth0_met1,2);
MC2TFRboth0_met_1=movmean(MC2TFRboth0_met1,2);
MC3TFLboth0_met_1=movmean(MC3TFLboth0_met1,2);
MC3TFRboth0_met_1=movmean(MC3TFRboth0_met1,2);

MS0TFLboth0_met_1=movmean(MS0TFLboth0_met1,2);
MS0TFRboth0_met_1=movmean(MS0TFRboth0_met1,2);
MS1TFLboth0_met_1=movmean(MS1TFLboth0_met1,2);
```

```
MS1TFRboth0_met_1=movmean(MS1TFRboth0_met1,2);
MS2TFLboth0_met_1=movmean(MS2TFLboth0_met1,2);
MS2TFRboth0_met_1=movmean(MS2TFRboth0_met1,2);
MS3TFLboth0_met_1=movmean(MS3TFLboth0_met1,2);
MS3TFRboth0_met_1=movmean(MS3TFRboth0_met1,2);


RL0TFLboth0_met_1=movmean(RL0TFLboth0_met1,2);
RL0TFRboth0_met_1=movmean(RL0TFRboth0_met1,2);
RL1TFLboth0_met_1=movmean(RL1TFLboth0_met1,2);
RL1TFRboth0_met_1=movmean(RL1TFRboth0_met1,2);
RL2TFLboth0_met_1=movmean(RL2TFLboth0_met1,2);
RL2TFRboth0_met_1=movmean(RL2TFRboth0_met1,2);
RL3TFLboth0_met_1=movmean(RL3TFLboth0_met1,2);
RL3TFRboth0_met_1=movmean(RL3TFRboth0_met1,2);

RC0TFLboth0_met_1=movmean(RC0TFLboth0_met1,2);
RC0TFRboth0_met_1=movmean(RC0TFRboth0_met1,2);
RC1TFLboth0_met_1=movmean(RC1TFLboth0_met1,2);
RC1TFRboth0_met_1=movmean(RC1TFRboth0_met1,2);
RC2TFLboth0_met_1=movmean(RC2TFLboth0_met1,2);
RC2TFRboth0_met_1=movmean(RC2TFRboth0_met1,2);
RC3TFLboth0_met_1=movmean(RC3TFLboth0_met1,2);
RC3TFRboth0_met_1=movmean(RC3TFRboth0_met1,2);

RS0TFLboth0_met_1=movmean(RS0TFLboth0_met1,2);
RS0TFRboth0_met_1=movmean(RS0TFRboth0_met1,2);
RS1TFLboth0_met_1=movmean(RS1TFLboth0_met1,2);
RS1TFRboth0_met_1=movmean(RS1TFRboth0_met1,2);
RS2TFLboth0_met_1=movmean(RS2TFLboth0_met1,2);
RS2TFRboth0_met_1=movmean(RS2TFRboth0_met1,2);
RS3TFLboth0_met_1=movmean(RS3TFLboth0_met1,2);
```

```
RS3TFRboth0_met_1=movmean(RL3TFRboth0_met1,2);

SL0TFLboth0_met_1=movmean(SL0TFLboth0_met1,2);
SL0TFRboth0_met_1=movmean(SL0TFRboth0_met1,2);
SL1TFLboth0_met_1=movmean(SL1TFLboth0_met1,2);
SL1TFRboth0_met_1=movmean(SL1TFRboth0_met1,2);
SL2TFLboth0_met_1=movmean(SL2TFLboth0_met1,2);
SL2TFRboth0_met_1=movmean(SL2TFRboth0_met1,2);
SL3TFLboth0_met_1=movmean(SL3TFLboth0_met1,2);
SL3TFRboth0_met_1=movmean(SL3TFRboth0_met1,2);

SC0TFLboth0_met_1=movmean(SC0TFLboth0_met1,2);
SC0TFRboth0_met_1=movmean(SC0TFRboth0_met1,2);
SC1TFLboth0_met_1=movmean(SC1TFLboth0_met1,2);
SC1TFRboth0_met_1=movmean(SC1TFRboth0_met1,2);
SC2TFLboth0_met_1=movmean(SC2TFLboth0_met1,2);
SC2TFRboth0_met_1=movmean(SC2TFRboth0_met1,2);
SC3TFLboth0_met_1=movmean(SC3TFLboth0_met1,2);
SC3TFRboth0_met_1=movmean(SC3TFRboth0_met1,2);

SS0TFLboth0_met_1=movmean(SS0TFLboth0_met1,2);
SS0TFRboth0_met_1=movmean(SS0TFRboth0_met1,2);
SS1TFLboth0_met_1=movmean(SS1TFLboth0_met1,2);
SS1TFRboth0_met_1=movmean(SS1TFRboth0_met1,2);
SS2TFLboth0_met_1=movmean(SS2TFLboth0_met1,2);
SS2TFRboth0_met_1=movmean(SS2TFRboth0_met1,2);
SS3TFLboth0_met_1=movmean(SS3TFLboth0_met1,2);
SS3TFRboth0_met_1=movmean(SS3TFRboth0_met1,2);

%% LINE Study
  %%TEST SUBJECT1 0 CMU WEIGHT
```

```
AL0StepL_1=AL0TFLboth0_met_1(1:162);
AL0StepL_2=AL0TFLboth0_met_1(162:281);
AL0StepL_3=AL0TFLboth0_met_1(281:399);
AL0StepL_4=AL0TFLboth0_met_1(399:520);
AL0StepL_5=AL0TFLboth0_met_1(520:638);
AL0StepL_6=AL0TFLboth0_met_1(638:757);
AL0StepL_7=AL0TFLboth0_met_1(757:873);
AL0StepL_8=AL0TFLboth0_met_1(873:992);
AL0StepL_9=AL0TFLboth0_met_1(992:1086);
AL0StepL_10=AL0TFLboth0_met_1(1086:1206);
AL0StepL_11=AL0TFLboth0_met_1(1206:1352);
AL0StepL_12=AL0TFLboth0_met_1(1352:1469);
AL0StepL_13=AL0TFLboth0_met_1(1469:1584);
AL0StepL_14=AL0TFLboth0_met_1(1584:1700);
AL0StepL_15=AL0TFLboth0_met_1(1700:1801);
  AL0StepL_16=AL0TFLboth0_met_1(1630:1734);
%right leg
AL0StepR_1=AL0TFRboth0_met_1(1:134);
AL0StepR_2=AL0TFRboth0_met_1(134:254);
AL0StepR_3=AL0TFRboth0_met_1(254:359);
AL0StepR_4=AL0TFRboth0_met_1(359:465);
AL0StepR_5=AL0TFRboth0_met_1(465:569);
AL0StepR_6=AL0TFRboth0_met_1(569:673);
AL0StepR_7=AL0TFRboth0_met_1(673:780);
AL0StepR_8=AL0TFRboth0_met_1(780:885);
AL0StepR_9=AL0TFRboth0_met_1(885:986);
AL0StepR_10=AL0TFRboth0_met_1(986:1091);
AL0StepR_11=AL0TFRboth0_met_1(1091:1201);
AL0StepR_12=AL0TFRboth0_met_1(1201:1306);
AL0StepR_13=AL0TFRboth0_met_1(1306:1413);
AL0StepR_14=AL0TFRboth0_met_1(1413:1517);
```

```
AL0StepR_15=AL0TFRboth0_met_1(1517:1621);

%%test suject 1 CMU 1 weight
plot (t,AL3TFRboth0_met_1)


AL1StepL_1=AL1TFLboth0_met_1(1:113);
AL1StepL_2=AL1TFLboth0_met_1(113:216);
AL1StepL_3=AL1TFLboth0_met_1(216:313);
AL1StepL_4=AL1TFLboth0_met_1(313:413);
AL1StepL_5=AL1TFLboth0_met_1(413:514);
AL1StepL_6=AL1TFLboth0_met_1(514:611);
AL1StepL_7=AL1TFLboth0_met_1(611:710);
AL1StepL_8=AL1TFLboth0_met_1(710:810);
AL1StepL_9=AL1TFLboth0_met_1(810:910);
AL1StepL_10=AL1TFLboth0_met_1(1037:1143);
AL1StepL_11=AL1TFLboth0_met_1(1143:1245);
AL1StepL_12=AL1TFLboth0_met_1(1245:1344);
AL1StepL_13=AL1TFLboth0_met_1(1344:1448);
AL1StepL_14=AL1TFLboth0_met_1(1551:1657);
AL1StepL_15=AL1TFLboth0_met_1(1657:1759);
%right leg
AL1StepR_1=AL1TFRboth0_met_1(1:143);
AL1StepR_2=AL1TFRboth0_met_1(143:230);
AL1StepR_3=AL1TFRboth0_met_1(230:333);
AL1StepR_4=AL1TFRboth0_met_1(333:430);
AL1StepR_5=AL1TFRboth0_met_1(430:525);
AL1StepR_6=AL1TFRboth0_met_1(525:623);
AL1StepR_7=AL1TFRboth0_met_1(623:723);
AL1StepR_8=AL1TFRboth0_met_1(723:819);
AL1StepR_9=AL1TFRboth0_met_1(819:919);
AL1StepR_10=AL1TFRboth0_met_1(919:1056);
```

```
AL1StepR_11=AL1TFRboth0_met_1(1056:1161);
AL1StepR_12=AL1TFRboth0_met_1(1161:1260);
AL1StepR_13=AL1TFRboth0_met_1(1260:1359);
AL1StepR_14=AL1TFRboth0_met_1(1359:1464);
AL1StepR_15=AL1TFRboth0_met_1(1464:1566);

      %%test suject 1 CMU 2 weight
 plot (t,AL2TFRboth0_met_1)

AL2StepL_1=AL2TFLboth0_met_1(51:153);
AL2StepL_2=AL2TFLboth0_met_1(153:250);
AL2StepL_3=AL2TFLboth0_met_1(250:350);
AL2StepL_4=AL2TFLboth0_met_1(350:445);
AL2StepL_5=AL2TFLboth0_met_1(445:540);
AL2StepL_6=AL2TFLboth0_met_1(540:640);
AL2StepL_7=AL2TFLboth0_met_1(640:740);
AL2StepL_8=AL2TFLboth0_met_1(740:837);
AL2StepL_9=AL2TFLboth0_met_1(837:962);
AL2StepL_10=AL2TFLboth0_met_1(962:1061);
AL2StepL_11=AL2TFLboth0_met_1(1061:1158);
AL2StepL_12=AL2TFLboth0_met_1(1158:1254);
AL2StepL_13=AL2TFLboth0_met_1(1254:1352);
AL2StepL_14=AL2TFLboth0_met_1(1352:1550);
AL2StepL_15=AL2TFLboth0_met_1(1550:1653);
%right leg
AL2StepR_1=AL2TFRboth0_met_1(1:117);
AL2StepR_2=AL2TFRboth0_met_1(117:209);
AL2StepR_3=AL2TFRboth0_met_1(209:308);
AL2StepR_4=AL2TFRboth0_met_1(308:402);
AL2StepR_5=AL2TFRboth0_met_1(402:502);
AL2StepR_6=AL2TFRboth0_met_1(502:600);
AL2StepR_7=AL2TFRboth0_met_1(600:695);
```

```
AL2StepR_8=AL2TFRboth0_met_1(695:790);
AL2StepR_9=AL2TFRboth0_met_1(790:927);
AL2StepR_10=AL2TFRboth0_met_1(927:1026);
AL2StepR_11=AL2TFRboth0_met_1(1026:1117);
AL2StepR_12=AL2TFRboth0_met_1(1117:1213);
AL2StepR_13=AL2TFRboth0_met_1(1213:1310);
AL2StepR_14=AL2TFRboth0_met_1(1310:1409);
AL2StepR_15=AL2TFRboth0_met_1(1406:1508);

% test sub 1 cmu 3
AL3StepL_1=AL3TFLboth0_met_1(37:118);
AL3StepL_2=AL3TFLboth0_met_1(118:221);
AL3StepL_3=AL3TFLboth0_met_1(221:333);
AL3StepL_4=AL3TFLboth0_met_1(333:430);
AL3StepL_5=AL3TFLboth0_met_1(430:526);
AL3StepL_6=AL3TFLboth0_met_1(526:632);
AL3StepL_7=AL3TFLboth0_met_1(632:728);
AL3StepL_8=AL3TFLboth0_met_1(728:831);
AL3StepL_9=AL3TFLboth0_met_1(831:935);
AL3StepL_10=AL3TFLboth0_met_1(935:1028);
AL3StepL_11=AL3TFLboth0_met_1(1028:1143);
AL3StepL_12=AL3TFLboth0_met_1(1143:1242);
AL3StepL_13=AL3TFLboth0_met_1(1254:1345);
AL3StepL_14=AL3TFLboth0_met_1(1345:1443);
AL3StepL_15=AL3TFLboth0_met_1(1443:1542);
%right leg
AL3StepR_1=AL3TFRboth0_met_1(58:173);
AL3StepR_2=AL3TFRboth0_met_1(173:274);
AL3StepR_3=AL3TFRboth0_met_1(274:375);
AL3StepR_4=AL3TFRboth0_met_1(375:482);
AL3StepR_5=AL3TFRboth0_met_1(482:580);
AL3StepR_6=AL3TFRboth0_met_1(580:678);
```

```
AL3StepR_7=AL3TFRboth0_met_1(678:783);
AL3StepR_8=AL3TFRboth0_met_1(783:882);
AL3StepR_9=AL3TFRboth0_met_1(882:985);
AL3StepR_10=AL3TFRboth0_met_1(1091:1191);
AL3StepR_11=AL3TFRboth0_met_1(1191:1295);
AL3StepR_12=AL3TFRboth0_met_1(1395:1493);
AL3StepR_13=AL3TFRboth0_met_1(1493:1600);
AL3StepR_14=AL3TFRboth0_met_1(1600:1692);
AL3StepR_15=AL3TFRboth0_met_1(1692:1792);

 %PERFORM TRAPZ INTEEGRATION TO GET FORCE-TIME INTEGRAL//OR
 %MOMENTUM
% Test Subject 1
IAL0StepL_1=trapz(AL0StepL_1);
IAL0StepL_2=trapz(AL0StepL_2);
IAL0StepL_3=trapz(AL0StepL_3);
IAL0StepL_4=trapz(AL0StepL_4);
IAL0StepL_5=trapz(AL0StepL_5);
IAL0StepL_6=trapz(AL0StepL_6);
IAL0StepL_7=trapz(AL0StepL_7);
IAL0StepL_8=trapz(AL0StepL_8);
IAL0StepL_9=trapz(AL0StepL_9);
IAL0StepL_10=trapz(AL0StepL_10);
IAL0StepL_11=trapz(AL0StepL_11);
IAL0StepL_12=trapz(AL0StepL_12);
IAL0StepL_13=trapz(AL0StepL_13);
IAL0StepL_14=trapz(AL0StepL_14);
IAL0StepL_15=trapz(AL0StepL_15);


IAL0StepR_1=trapz(AL0StepR_10);
IAL0StepR_2=trapz(AL0StepR_2);
```

```
IAL0StepR_3=trapz(AL0StepR_3);
IAL0StepR_4=trapz(AL0StepR_4);
IAL0StepR_5=trapz(AL0StepR_5);
IAL0StepR_6=trapz(AL0StepR_6);
IAL0StepR_7=trapz(AL0StepR_7);
IAL0StepR_8=trapz(AL0StepR_8);
IAL0StepR_9=trapz(AL0StepR_9);
IAL0StepR_10=trapz(AL0StepR_10);
IAL0StepR_11=trapz(AL0StepR_11);
IAL0StepR_12=trapz(AL0StepR_12);
IAL0StepR_13=trapz(AL0StepR_13);
IAL0StepR_14=trapz(AL0StepR_14);
IAL0StepR_15=trapz(AL0StepR_15);

IAL1StepL_1=trapz(AL1StepL_1);
IAL1StepL_2=trapz(AL1StepL_2);
IAL1StepL_3=trapz(AL1StepL_3);
IAL1StepL_4=trapz(AL1StepL_4);
IAL1StepL_5=trapz(AL1StepL_5);
IAL1StepL_6=trapz(AL1StepL_6);
IAL1StepL_7=trapz(AL1StepL_7);
IAL1StepL_8=trapz(AL1StepL_8);
IAL1StepL_9=trapz(AL1StepL_9);
IAL1StepL_10=trapz(AL1StepL_10);
IAL1StepL_11=trapz(AL1StepL_11);
IAL1StepL_12=trapz(AL1StepL_12);
IAL1StepL_13=trapz(AL1StepL_13);
IAL1StepL_14=trapz(AL1StepL_14);
IAL1StepL_15=trapz(AL1StepL_15);


IAL1StepR_1=trapz(AL1StepR_1);
```

```
IAL1StepR_1=trapz(AL1StepR_2);
IAL1StepR_1=trapz(AL1StepR_3);
IAL1StepR_1=trapz(AL1StepR_4);
IAL1StepR_1=trapz(AL1StepR_5);
IAL1StepR_1=trapz(AL1StepR_6);
IAL1StepR_1=trapz(AL1StepR_7);
IAL1StepR_1=trapz(AL1StepR_8);
IAL1StepR_1=trapz(AL1StepR_9);
IAL1StepR_1=trapz(AL1StepR_10);
IAL1StepR_1=trapz(AL1StepR_11);
IAL1StepR_1=trapz(AL1StepR_12);
IAL1StepR_1=trapz(AL1StepR_13);
IAL1StepR_1=trapz(AL1StepR_14);
IAL1StepR_1=trapz(AL1StepR_15);


IAL2StepL_1=trapz(AL2StepL_1);
IAL2StepL_2=trapz(AL2StepL_2);
IAL2StepL_3=trapz(AL2StepL_3);
IAL2StepL_4=trapz(AL2StepL_4);
IAL2StepL_5=trapz(AL2StepL_5);
IAL2StepL_6=trapz(AL2StepL_6);
IAL2StepL_7=trapz(AL2StepL_7);
IAL2StepL_8=trapz(AL2StepL_8);
IAL2StepL_9=trapz(AL2StepL_9);
IAL2StepL_10=trapz(AL2StepL_10);
IAL2StepL_11=trapz(AL2StepL_11);
IAL2StepL_12=trapz(AL2StepL_12);
IAL2StepL_13=trapz(AL2StepL_13);
IAL2StepL_14=trapz(AL2StepL_14);
IAL2StepL_15=trapz(AL2StepL_15);
```

```
IAL2StepR_1=trapz(AL2StepR_1);
IAL2StepR_2=trapz(AL2StepR_2);
IAL2StepR_3=trapz(AL2StepR_3);
IAL2StepR_4=trapz(AL2StepR_4);
IAL2StepR_5=trapz(AL2StepR_5);
IAL2StepR_6=trapz(AL2StepR_6);
IAL2StepR_7=trapz(AL2StepR_7);
IAL2StepR_8=trapz(AL2StepR_8);
IAL2StepR_9=trapz(AL2StepR_9);
IAL2StepR_10=trapz(AL2StepR_10);
IAL2StepR_11=trapz(AL2StepR_11);
IAL2StepR_12=trapz(AL2StepR_12);
IAL2StepR_13=trapz(AL2StepR_13);
IAL2StepR_14=trapz(AL2StepR_14);
IAL2StepR_15=trapz(AL2StepR_15);


IAL3StepL_1=trapz(AL3StepL_1);
IAL3StepL_2=trapz(AL3StepL_2);
IAL3StepL_3=trapz(AL3StepL_3);
IAL3StepL_4=trapz(AL3StepL_4);
IAL3StepL_5=trapz(AL3StepL_5);
IAL3StepL_6=trapz(AL3StepL_6);
IAL3StepL_7=trapz(AL3StepL_7);
IAL3StepL_8=trapz(AL3StepL_8);
IAL3StepL_9=trapz(AL3StepL_9);
IAL3StepL_10=trapz(AL3StepL_10);
IAL3StepL_11=trapz(AL3StepL_11);
IAL3StepL_12=trapz(AL3StepL_12);
IAL3StepL_13=trapz(AL3StepL_13);
IAL3StepL_14=trapz(AL3StepL_14);
```

```
IAL3StepL_15=trapz(AL3StepL_15);


IAL3StepR_1=trapz(AL3StepR_1);
IAL3StepR_2=trapz(AL3StepR_2);
IAL3StepR_3=trapz(AL3StepR_3);
IAL3StepR_4=trapz(AL3StepR_4);
IAL3StepR_5=trapz(AL3StepR_5);
IAL3StepR_6=trapz(AL3StepR_6);
IAL3StepR_7=trapz(AL3StepR_7);
IAL3StepR_8=trapz(AL3StepR_8);
IAL3StepR_9=trapz(AL3StepR_9);
IAL3StepR_10=trapz(AL3StepR_10);
IAL3StepR_11=trapz(AL3StepR_11);
IAL3StepR_12=trapz(AL3StepR_12);
IAL3StepR_13=trapz(AL3StepR_13);
IAL3StepR_14=trapz(AL3StepR_14);
IAL3StepR_15=trapz(AL3StepR_15);

% Normalize
% Name coding:(N)ormalize(I)ntegral(participant)-(L,C,S=>path type linear circle or square)
% (1,2,3=>CMU block type)
% (8,16,24)weight,(R,L)hand,trial,(S)tep,(R,L=right or left leg
% measurements)_(step Number)
NIAL0StepL_1= IAL0StepL_1/length(AL0StepL_1);
NIAL0StepL_2= IAL0StepL_2/length(AL0StepL_2);
NIAL0StepL_3= IAL0StepL_3/length(AL0StepL_3);
NIAL0StepL_4= IAL0StepL_4/length(AL0StepL_4);
NIAL0StepL_5= IAL0StepL_5/length(AL0StepL_5);
NIAL0StepL_6= IAL0StepL_6/length(AL0StepL_6);
NIAL0StepL_7= IAL0StepL_7/length(AL0StepL_7);
NIAL0StepL_8= IAL0StepL_8/length(AL0StepL_8);
```

```
NIAL0StepL_9= IAL0StepL_9/length(AL0StepL_9);
NIAL0StepL_10= IAL0StepL_10/length(AL0StepL_10);
NIAL0StepL_11= IAL0StepL_11/length(AL0StepL_11);
NIAL0StepL_12= IAL0StepL_12/length(AL0StepL_12);
NIAL0StepL_13= IAL0StepL_13/length(AL0StepL_13);
NIAL0StepL_14= IAL0StepL_14/length(AL0StepL_14);
NIAL0StepL_15= IAL0StepL_15/length(AL0StepL_15);

NIAL0StepR_1= IAL0StepR_1/length(AR0StepR_1);
NIAL0StepR_2= IAL0StepR_2/length(AR0StepR_2);
NIAL0StepR_3= IAL0StepR_3/length(AR0StepR_3);
NIAL0StepR_4= IAL0StepR_4/length(AR0StepR_4);
NIAL0StepR_5= IAL0StepR_5/length(AR0StepR_5);
NIAL0StepR_6= IAL0StepR_6/length(AR0StepR_6);
NIAL0StepR_7= IAL0StepR_7/length(AR0StepR_7);
NIAL0StepR_8= IAL0StepR_8/length(AR0StepR_8);
NIAL0StepR_9= IAL0StepR_9/length(AR0StepR_9);
NIAL0StepR_10= IAL0StepR_10/length(AR0StepR_10);
NIAL0StepR_11= IAL0StepL_11/length(AR0StepL_11);
NIAL0StepR_12= IAL0StepL_12/length(AL0StepL_12);
NIAL0StepR_13= IAL0StepL_13/length(AL0StepL_13);
NIAL0StepR_14= IAL0StepL_14/length(AL0StepL_14);
NIAL0StepR_15= IAL0StepL_15/length(AL0StepL_15);

NIAL1StepL_1= IAL0StepL_1/length(AL1StepL_1);
NIAL1StepL_2= IAL0StepL_2/length(AL1StepL_2);
NIAL1StepL_3= IAL0StepL_3/length(AL1StepL_3);
NIAL1StepL_4= IAL0StepL_4/length(AL1StepL_4);
NIAL1StepL_5= IAL0StepL_5/length(AL1StepL_5);
NIAL1StepL_6= IAL0StepL_6/length(AL1StepL_6);
NIAL1StepL_7= IAL0StepL_7/length(AL1StepL_7);
NIAL1StepL_8= IAL0StepL_8/length(AL1StepL_8);
```

```
NIAL1StepL_9= IAL0StepL_9/length(AL1StepL_9);
NIAL1StepL_10= IAL0StepL_10/length(AL1StepL_10);
NIAL1StepL_11= IAL0StepL_11/length(AL1StepL_11);
NIAL1StepL_12= IAL0StepL_12/length(AL1StepL_12);
NIAL1StepL_13= IAL0StepL_13/length(AL1StepL_13);
NIAL1StepL_14= IAL0StepL_14/length(AL1StepL_14);
NIAL1StepL_15= IAL0StepL_15/length(AL1StepL_15);

NIAL1StepR_1= IAL1StepR_1/length(AR1StepR_1);
NIAL1StepR_2= IAL1StepR_2/length(AR1StepR_2);
NIAL1StepR_3= IAL1StepR_3/length(AR1StepR_3);
NIAL1StepR_4= IAL1StepR_4/length(AR1StepR_4);
NIAL1StepR_5= IAL1StepR_5/length(AR1StepR_5);
NIAL1StepR_6= IAL1StepR_6/length(AR1StepR_6);
NIAL1StepR_7= IAL1StepR_7/length(AR1StepR_7);
NIAL1StepR_8= IAL1StepR_8/length(AR1StepR_8);
NIAL1StepR_9= IAL1StepR_9/length(AR1StepR_9);
NIAL1StepR_10= IAL1StepR_10/length(AR1StepR_10);
NIAL1StepR_11= IAL1StepL_11/length(AR1StepL_11);
NIAL1StepR_12= IAL1StepL_12/length(AL1StepL_12);
NIAL1StepR_13= IAL1StepL_13/length(AL1StepL_13);
NIAL1StepR_14= IAL1StepL_14/length(AL1StepL_14);
NIAL1StepR_15= IAL1StepL_15/length(AL1StepL_15);

NIAL2StepL_1= IAL2StepL_1/length(AL2StepL_1);
NIAL2StepL_2= IAL2StepL_2/length(AL2StepL_2);
NIAL2StepL_3= IAL2StepL_3/length(AL2StepL_3);
NIAL2StepL_4= IAL2StepL_4/length(AL2StepL_4);
NIAL2StepL_5= IAL2StepL_5/length(AL2StepL_5);
NIAL2StepL_6= IAL2StepL_6/length(AL2StepL_6);
NIAL2StepL_7= IAL2StepL_7/length(AL2StepL_7);
NIAL2StepL_8= IAL2StepL_8/length(AL2StepL_8);
```

```
NIAL2StepL_9= IAL2StepL_9/length(AL2StepL_9);
NIAL2StepL_10= IAL2StepL_10/length(AL2StepL_10);
NIAL2StepL_11= IAL2StepL_11/length(AL2StepL_11);
NIAL2StepL_12= IAL2StepL_12/length(AL2StepL_12);
NIAL2StepL_13= IAL2StepL_13/length(AL2StepL_13);
NIAL2StepL_14= IAL2StepL_14/length(AL2StepL_14);
NIAL2StepL_15= IAL2StepL_15/length(AL2StepL_15);

NIAL2StepR_1= IAL2StepR_1/length(AR2StepR_1);
NIAL2StepR_2= IAL2StepR_2/length(AR2StepR_2);
NIAL2StepR_3= IAL2StepR_3/length(AR2StepR_3);
NIAL2StepR_4= IAL2StepR_4/length(AR2StepR_4);
NIAL2StepR_5= IAL2StepR_5/length(AR2StepR_5);
NIAL2StepR_6= IAL2StepR_6/length(AR2StepR_6);
NIAL2StepR_7= IAL2StepR_7/length(AR2StepR_7);
NIAL2StepR_8= IAL2StepR_8/length(AR2StepR_8);
NIAL2StepR_9= IAL2StepR_9/length(AR2StepR_9);
NIAL2StepR_10= IAL2StepR_10/length(AR2StepR_10);
NIAL2StepR_11= IAL2StepL_11/length(AR2StepL_11);
NIAL2StepR_12= IAL2StepL_12/length(AL2StepL_12);
NIAL2StepR_13= IAL2StepL_13/length(AL2StepL_13);
NIAL2StepR_14= IAL2StepL_14/length(AL2StepL_14);
NIAL2StepR_15= IAL2StepL_15/length(AL2StepL_15);

NIAL3StepL_1= IAL3StepL_1/length(AL3StepL_1);
NIAL3StepL_2= IAL3StepL_2/length(AL3StepL_2);
NIAL3StepL_3= IAL3StepL_3/length(AL3StepL_3);
NIAL3StepL_4= IAL3StepL_4/length(AL3StepL_4);
NIAL3StepL_5= IAL3StepL_5/length(AL3StepL_5);
NIAL3StepL_6= IAL3StepL_6/length(AL3StepL_6);
NIAL3StepL_7= IAL3StepL_7/length(AL3StepL_7);
NIAL3StepL_8= IAL3StepL_8/length(AL3StepL_8);
```

```
    NIAL3StepL_9= IAL3StepL_9/length(AL3StepL_9);
    NIAL3StepL_10= IAL3StepL_10/length(AL3StepL_10);
    NIAL3StepL_11= IAL3StepL_11/length(AL3StepL_11);
    NIAL3StepL_12= IAL3StepL_12/length(AL3StepL_12);
    NIAL3StepL_13= IAL3StepL_13/length(AL3StepL_13);
    NIAL3StepL_14= IAL3StepL_14/length(AL3StepL_14);
    NIAL3StepL_15= IAL3StepL_15/length(AL3StepL_15);

    NIAL3StepR_1= IAL3StepR_1/length(AR3StepR_1);
    NIAL3StepR_2= IAL3StepR_2/length(AR3StepR_2);
    NIAL3StepR_3= IAL3StepR_3/length(AR3StepR_3);
    NIAL3StepR_4= IAL3StepR_4/length(AR3StepR_4);
    NIAL3StepR_5= IAL3StepR_5/length(AR3StepR_5);
    NIAL3StepR_6= IAL3StepR_6/length(AR3StepR_6);
    NIAL3StepR_7= IAL3StepR_7/length(AR3StepR_7);
    NIAL3StepR_8= IAL3StepR_8/length(AR3StepR_8);
    NIAL3StepR_9= IAL3StepR_9/length(AR3StepR_9);
    NIAL3StepR_10= IAL3StepR_10/length(AR3StepR_10);
    NIAL3StepR_11= IAL3StepL_11/length(AR3StepL_11);
    NIAL3StepR_12= IAL3StepL_12/length(AL3StepL_12);
    NIAL3StepR_13= IAL3StepL_13/length(AL3StepL_13);
    NIAL3StepR_14= IAL3StepL_14/length(AL3StepL_14);
NIAL3StepR_15= IAL3StepL_15/length(AL3StepL_15);
```

Appendix C

Pressure Insole Geometric and Sensor Specifications

	Size 1	Size 2	Size 3	Size 4	Size 5	Size 6	Size 7	Size 8	Size 9
TAI(mm^2)	12129	13240	14462	15787	17237	18824	20572	22456	24524
TAS(mm^2)	7578	8330	9179	10109	11129	12241	13696	15042	16525
SC (%)	62.5	62.9	63.5	64	64.6	65	66.6	67	67.4

Table 8: Pressure sensor geometry (Moticon, 2021)

Values of $A_{i,j}$

Size1: [5.63999, 5.03971, 5.72901, 5.65658, 4.4609, 5.1409, 1.71182, 4.1135, 6.70146, 4.72535, 4.74676, 4.79923, 4.35605, 4.71538, 5.43545, 2.80808]

Size2: [6.1744, 5.52418, 6.28194, 6.20342, 4.98368, 5.66192, 1.90741, 4.50528, 7.36236, 5.22065, 5.2439, 5.30050, 4.75849, 5.15237, 5.94863, 3.06889]

Size3: [6.7633, 6.05812, 6.89182, 6.80659, 5.56157, 6.23786, 2.12434, 5.13013, 8.09149, 5.76895, 5.79423, 5.85543, 5.20313, 5.63431, 6.51536, 3.3569]

Size4: [7.40271, 6.63861, 7.5544, 7.46203, 6.19031, 6.86496, 2.36153, 5.8701, 8.88432, 6.36663, 6.39406, 6.4601, 5.68331, 6.15624, 7.12931, 3.66869]

Size5: [8.10396, 7.27661, 8.28128, 8.18138, 6.88073, 7.55448, 2.62373, 6.74505, 9.75514, 7.02432, 7.05401, 7.12509, 6.20415, 6.72578, 7.79909, 4.00834]

Size6: [8.87157, 7.97504, 9.07763, 8.96946, 7.63859, 8.31133, 2.91213, 7.56437, 10.70929, 7.74710, 7.77927, 7.85591, 6.77586, 7.34975, 8.53378, 4.38093]

Size7: [9.87417, 8.86745, 10.15314, 10.02424, 8.66369, 9.32312, 3.36383, 8.52153, 11.93697, 8.66954, 8.70726, 8.79182, 7.54481, 8.13816, 9.49672, 4.88443]

Size8: [10.80294, 9.71255, 11.12091, 10.98165, 9.58739, 10.24559, 3.72449, 9.50622, 13.09233, 9.54795, 9.58874, 9.67998, 8.23314, 8.88777, 10.38316, 5.33475]

Size9: [11.82361, 10.64203, 12.185, 12.03452, 10.6044, 11.26167, 4.12274, 10.58986, 14.36321, 10.51630, 10.56041, 10.65888, 8.98788, 9.7105, 11.35663, 5.82908]

Scaling Factors $k_{i,j}$

Size1: [1.28444, 1.27509, 1.36581, 1.35741, 1.72358, 1.48103, 6.34732, 2.58130, 1.36816, 1.54098, 1.54488, 1.54902, 1.42512, 1.27648, 1.37107, 1.38124]

Size2: [1.28224, 1.27255, 1.36111, 1.35284, 1.68595, 1.47001, 6.18752, 2.57816, 1.36103, 1.52435, 1.52811, 1.53218, 1.41710, 1.27335, 1.36484, 1.37575]

Size3: [1.28008, 1.27006, 1.35653, 1.34839, 1.65196, 1.45941, 6.03892, 2.47764, 1.35420, 1.50848, 1.51213, 1.51612, 1.40937, 1.27028, 1.35878, 1.37042]

Size4: [1.27809, 1.26777, 1.35227, 1.34425, 1.62186, 1.44955, 5.90192, 2.36828, 1.34783, 1.49373, 1.49726, 1.50119, 1.40210, 1.26741, 1.35311, 1.36539]

Size5: [1.27631, 1.26572, 1.34836, 1.34045, 1.59498, 1.44039, 5.77331, 2.25582, 1.34186, 1.47998, 1.48340, 1.48724, 1.39521, 1.26475, 1.34782, 1.36067]

Size6: [1.27455, 1.26371, 1.34455, 1.33676, 1.57054, 1.43161, 5.65405, 2.20099, 1.33617, 1.46691, 1.47022, 1.47398, 1.38862, 1.26216, 1.34270, 1.35611]

Size7: [1.25346, 1.24515, 1.31551, 1.30922, 1.51502, 1.39669, 5.32575, 2.13843, 1.31156, 1.43327, 1.43604, 1.43956, 1.35677, 1.24458, 1.31650, 1.32635]

Size8: [1.25194, 1.24339, 1.31224, 1.30601, 1.49592, 1.38915, 5.22746, 2.09680, 1.30662, 1.42203, 1.42471, 1.42813, 1.35107, 1.24231, 1.31207, 1.32242]

Size9: [1.25048, 1.24171, 1.30912, 1.30295, 1.47843, 1.38198, 5.13495, 2.05973, 1.30192, 1.41135, 1.41396, 1.41728, 1.34561, 1.24013, 1.30782, 1.31862]

Appendix D

Load BVH File Matlab Code

```
%% Copy rights
%{
   Copyright 2012 Will Robertson and The University of Adelaide

   Released under the terms and conditions of the Apache License, v2. http://www.apache.org/licenses/LICENSE-2.0

   https://github.com/wspr/bvh-matlab
   %}

   %%
   function [skeleton,time] = loadbvh(fname,varargin)
   %% LOADBVH  Load a .bvh (Biovision) file.
   %
   % Loads BVH file specified by FNAME (with or without .bvh extension)
   % and parses the file, calculating joint kinematics and storing the
   % output in SKELETON.
   %
   % Optional argument 'delim' allows for setting the delimiter between
   % fields. E.g., for a tab-separated BVH file:
   %
   %    skeleton = loadbvh('louise.bvh','delim','\t')
```

```
%
% By default 'delim' is set to the space character.
%
% Some details on the BVH file structure are given in "Motion Capture File
% Formats Explained": http://www.dcs.shef.ac.uk/intranet/research/resmes/CS0111.pdf
% But most of it is fairly self-evident.

%% Options

p = inputParser;
p.addParameter('delim',' ');
parse(p,varargin{:});
opt = p.Results;

%% Load and parse header data
%
% The file is opened for reading, primarily to extract the header data (see
% next section). However, I don't know how to ask Matlab to read only up
% until the line "MOTION", so we're being a bit inefficient here and
% loading the entire file into memory. Oh well.

% add a file extension if necessary:
if ~strncmpi(fliplr(fname),'hvb.',4)
  fname = [fname,'.bvh'];
```

```
end


fid = fopen(fname);
C = textscan(fid,'%s');
fclose(fid);
C = C{1};



%% Parse data
%
% This is a cheap tokeniser, not particularly clever.
% Iterate word-by-word, counting braces and extracting data.

% Initialise:
skeleton = [];
ii = 1;
nn = 0;
brace_count = 1;

while ~strcmp( C{ii} , 'MOTION' )

  ii = ii+1;
  token = C{ii};
```

```
if strcmp( token , '{' )

  brace_count = brace_count + 1;

elseif strcmp( token , '}' )

  brace_count = brace_count - 1;

elseif strcmp( token , 'OFFSET' )

  skeleton(nn).offset = [str2double(C(ii+1)) ; str2double(C(ii+2)) ; str2double(C(ii+3))];
  ii = ii+3;

elseif strcmp( token , 'CHANNELS' )

  skeleton(nn).Nchannels = str2double(C(ii+1));

  % The 'order' field is an index corresponding to the order of 'X' 'Y' 'Z'.
  % Subtract 87 because char numbers "X" == 88, "Y" == 89, "Z" == 90.
  if skeleton(nn).Nchannels == 3
    skeleton(nn).order = [C{ii+2}(1),C{ii+3}(1),C{ii+4}(1)]-87;
  elseif skeleton(nn).Nchannels == 6
    skeleton(nn).order = [C{ii+5}(1),C{ii+6}(1),C{ii+7}(1)]-87;
  else
```

```
    error('Not sure how to handle not (3 or 6) number of channels.')
  end

  if ~all(sort(skeleton(nn).order)==[1 2 3])
    error('Cannot read channels order correctly. Should be some permutation of ["X" "Y" "Z"].')
  end

  ii = ii + skeleton(nn).Nchannels + 1;

elseif strcmp( token , 'JOINT' ) || strcmp( token , 'ROOT' )
  % Regular joint

  nn = nn+1;

  skeleton(nn).name = C{ii+1};
  skeleton(nn).nestdepth = brace_count;

  if brace_count == 1
    % root node
    skeleton(nn).parent = 0;
  elseif skeleton(nn-1).nestdepth + 1 == brace_count;
    % if I am a child, the previous node is my parent:
    skeleton(nn).parent = nn-1;
  else
```

```
    % if not, what is the node corresponding to this brace count?
    prev_parent = skeleton(nn-1).parent;
    while skeleton(prev_parent).nestdepth+1 ~= brace_count
      prev_parent = skeleton(prev_parent).parent;
    end
    skeleton(nn).parent = prev_parent;
  end

  ii = ii+1;

elseif strcmp( [C{ii},' ',C{ii+1}] , 'End Site' )
  % End effector; unnamed terminating joint
  %
  % N.B. The "two word" token here is why we don't use a switch statement
  % for this code.

  nn = nn+1;

  skeleton(nn).name = ' ';
  skeleton(nn).offset = [str2double(C(ii+4)) ; str2double(C(ii+5)) ; str2double(C(ii+6))];
  skeleton(nn).parent = nn-1; % always the direct child
  skeleton(nn).nestdepth = brace_count;
  skeleton(nn).Nchannels = 0;
```

```
  end

end

%% Initial processing and error checking

Nnodes = numel(skeleton);
Nchannels = sum([skeleton.Nchannels]);
Nchainends = sum([skeleton.Nchannels]==0);

% Calculate number of header lines:
%  - 5 lines per joint
%  - 4 lines per chain end
%  - 4 additional lines (first one and last three)
Nheaderlines = (Nnodes-Nchainends)*5 + Nchainends*4 + 4;

rawdata = importdata(fname,opt.delim,Nheaderlines);

if ~isstruct(rawdata)
   error('Could not parse BVH file %s. Check the delimiter.',fname)
end

index = strncmp(rawdata.textdata,'Frames:',7);
Nframes = sscanf(rawdata.textdata{index},'Frames: %f');
```

```
index = strncmp(rawdata.textdata,'Frame Time:',10);
frame_time = sscanf(rawdata.textdata{index},'Frame Time: %f');

time = frame_time*(0:Nframes-1);

if size(rawdata.data,2) ~= Nchannels
  error('Error reading BVH file: channels count does not match.')
end

if size(rawdata.data,1) ~= Nframes
  warning('LOADBVH:frames_wrong','Error reading BVH file: frames count does not match; continuing anyway.')
  Nframes = size(rawdata.data,1);
end

%% Load motion data into skeleton structure
%
% We have three possibilities for each node we come across:
% (a) a root node that has displacements already defined,
%     for which the transformation matrix can be directly calculated;
% (b) a joint node, for which the transformation matrix must be calculated
%     from the previous points in the chain; and
% (c) an end effector, which only has displacement to calculate from the
```

```
%     previous node's transformation matrix and the offset of the end
%     joint.
%
% These are indicated in the skeleton structure, respectively, by having
% six, three, and zero "channels" of data.
% In this section of the code, the channels are read in where appropriate
% and the relevant arrays are pre-initialised for the subsequent calcs.

channel_count = 0;

for nn = 1:Nnodes

  if skeleton(nn).Nchannels == 6 % root node

    % assume translational data is always ordered XYZ
    skeleton(nn).Dxyz = repmat(skeleton(nn).offset,[1 Nframes]) + rawdata.data(:,channel_count+[1
2 3])';
    skeleton(nn).rxyz(skeleton(nn).order,:) = rawdata.data(:,channel_count+[4 5 6])';

    % Kinematics of the root element:
    skeleton(nn).trans = nan(4,4,Nframes);
    for ff = 1:Nframes
      skeleton(nn).trans(:,:,ff)          =          transformation_matrix(skeleton(nn).Dxyz(:,ff)          ,
skeleton(nn).rxyz(:,ff) , skeleton(nn).order);
```

```
    end

  elseif skeleton(nn).Nchannels == 3 % joint node

    skeleton(nn).rxyz(skeleton(nn).order,:) = rawdata.data(:,channel_count+[1 2 3])';
    skeleton(nn).Dxyz  = nan(3,Nframes);
    skeleton(nn).trans = nan(4,4,Nframes);

  elseif skeleton(nn).Nchannels == 0 % end node
    skeleton(nn).Dxyz  = nan(3,Nframes);
  end

  channel_count = channel_count + skeleton(nn).Nchannels;

end



%% Calculate kinematics
%
% No calculations are required for the root nodes.

% For each joint, calculate the transformation matrix and for convenience
% extract each position in a separate vector.
for nn = find([skeleton.parent] ~= 0 & [skeleton.Nchannels] ~= 0)
```

```
    parent = skeleton(nn).parent;

    for ff = 1:Nframes
      transM = transformation_matrix( skeleton(nn).offset , skeleton(nn).rxyz(:,ff) , skeleton(nn).order
);
      skeleton(nn).trans(:,:,ff) = skeleton(parent).trans(:,:,ff) * transM;
      skeleton(nn).Dxyz(:,ff) = skeleton(nn).trans([1 2 3],4,ff);
    end

  end

  % For an end effector we don't have rotation data;
  % just need to calculate the final position.
  for nn = find([skeleton.Nchannels] == 0)

    parent = skeleton(nn).parent;

    for ff = 1:Nframes
      transM = skeleton(parent).trans(:,:,ff) * [eye(3), skeleton(nn).offset; 0 0 0 1];
      skeleton(nn).Dxyz(:,ff) = transM([1 2 3],4);
    end

  end
```

```
end

function transM = transformation_matrix(displ,rxyz,order)

% Constructs the transformation for given displacement, DISPL, and

% rotations RXYZ. The vector RYXZ is of length three corresponding to

% rotations around the X, Y, Z axes.

%

% The third input, ORDER, is a vector indicating which order to apply

% the planar rotations. E.g., [3 1 2] refers applying rotations RXYZ

% around Z first, then X, then Y.

%

% Years ago we benchmarked that multiplying the separate rotation matrices

% was more efficient than pre-calculating the final rotation matrix

% symbolically, so we don't "optimise" by having a hard-coded rotation

% matrix for, say, 'ZXY' which seems more common in BVH files.

% Should revisit this assumption one day.

%

% Precalculating the cosines and sines saves around 38% in execution time.


c = cosd(rxyz);

s = sind(rxyz);


RxRyRz(:,:,1) = [1 0 0; 0 c(1) -s(1); 0 s(1) c(1)];

RxRyRz(:,:,2) = [c(2) 0 s(2); 0 1 0; -s(2) 0 c(2)];
```

```
RxRyRz(:,:,3) = [c(3) -s(3) 0; s(3) c(3) 0; 0 0 1];


rotM = RxRyRz(:,:,order(1))*RxRyRz(:,:,order(2))*RxRyRz(:,:,order(3));


transM = [rotM, displ; 0 0 0 1];
end
```