

Use Authorization

In presenting this thesis in partial fulfillment of the requirements for a graduate degree at Idaho State University, I agree that the Library shall make it freely available for inspection. I further state that permission to download and/or print my thesis for scholarly purposes may be granted by the Dean of the Graduate School, Dean of my academic division, or by the University Librarian. It is understood that any copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Signature _____

Date _____

FORWARD DYNAMIC SIMULATION OF AN UPPER LIMB MODEL WITH
MUSCLE ACTIVATION PATTERNS

By

Adedayo Adekoya

A thesis submitted in partial fulfillment of the requirements

For the degree of

Master of Science

In

Measurement and Control Engineering

Measurement and Control Engineering Research Center

College of Engineering, Idaho State University

May 2015

To the Graduate Faculty:

The members of the committee appointed to examine the thesis of Adedayo Adekoya
find it satisfactory and recommend that it be accepted.

_____ Major Advisor

Dr. Marco P. Schoen

_____ Co-Advisor

Dr. Nancy Devine

_____ Graduate Faculty Representative

Dr. James Mahar

ACKNOWLEDGEMENT

I salute my advisor Dr. Marco Schoen for being a true mentor through the program and for being instrumental in my research and academics. I would also like to mention my Co-advisor Dr. Nancy Devine who gave me inspiration and motivation for this project. I also thank Dr. James Mahar for being my Graduate Faculty Representative and for his timely advice for the student organization at Idaho State University, National Society of Black Engineers.

I would like to thank my parents and parents-in-law for their support and to my lovely wife Samantha Adekoya for being the pillars in my success and accomplishments.

Many thanks go out to the members of Eli M. Oboler Library for guidance in my literature survey, as well as Ellen Combs of the college of Engineering and Donna Plant of the graduate school ISU for providing necessary guidance and support.

TABLE OF CONTENTS

	PAGES
List of Figures.....	vii
List of Tables.....	xiv
Abstract.....	xv
Chapter One: 1.0 Introduction.....	1
1.1 ARWED proposal solution.....	2
1.2 Motion intention.....	3
1.3. sEMG measurement and muscle pattern simulation.....	4
1.4 Past research work.....	5
1.5 Thesis outline.....	6
Chapter Two: 2.0 Muscle anatomy of the upper extremity arm.....	7
2.1 The shoulder.....	10
2.2 The arm.....	12
2.3 The wrist.....	13
2.4 The hand.....	15
Chapter Three: 3.0 OpenSim.....	18
3.1 Background.....	18
3.2 Introduction.....	19
3.3 OpenSim compatibility and models.....	19
3.31 OpenSim models.....	21
3.32 Bodies.....	22
3.33 Body geometry.....	23
3.34 Joints.....	24
3.35 Types of joint.....	26
3.36 Kinematic constraints.....	26
3.37 Forces.....	27

3.38	Controllers.....	27
3.4	Types of model.....	28
3.5	Model editing.....	34
Chapter Four:	4.0 Dynamic upper extremity.....	36
Chapter Five:	5.0 Model motion, simulation and analysis.....	38
5.1	Muscle actions.....	38
5.2	Motion kinematics.....	40
5.3	Building and loading MoBL ARMS model.....	42
5.4	Viewing and animating model.....	43
5.5	Coordinates.....	43
5.6	Loading a motion.....	45
5.7	Motion slider.....	46
5.8	Recording a movie.....	47
5.9	Plotting in OpenSim.....	47
Chapter Six:	6.0 Simulation and testing.....	54
6.1	Model specifics.....	54
6.2	Muscle activation using OpenSim.....	55
6.3	Simulation.....	57
Chapter Seven:	7.0 Results.....	77
7.1	Calculations.....	95
Chapter Eight:	8.0 Conclusion and future work.....	98
References	100
Appendix	107

LIST OF FIGURES

	PAGES
Figure 1a Surface anatomy, posterior muscle view.....	8
Figure 1b Superficial muscles, posterior muscle view.....	8
Figure 1c Origins and insertions, posterior view.....	8
Figure 2a Surface anatomy of the upper limb, anterior muscle view.....	9
Figure 2b Superficial muscles of the upper limb, anterior muscle view.....	9
Figure 2c Anterior view of bones of the upper limb.....	9
Figure 3 Anterior muscle view of the human shoulder.....	11
Figure 4 Anterior muscle view of the upper arm.....	13
Figure 4.1a Posterior muscle view of the forearm.....	15
Figure 4.1b Anterior muscle view of the forearm.....	15
Figure 5a Right hand anterior (palmer) muscle view.....	17
Figure 5b Right hand posterior (dorsal) muscle view.....	17
Figure 6 Opensim musculoskeletal structures.....	21
Figure 7 Opensim Model structure.....	22

	PAGES
Figure 7.1	Body geometry on OpenSim.....24
Figure 7.2	Kinematic relationship between two frames.....24
Figure 7.3	Example of XML Code from Model Arm26 to represent a Joint.....25
Figure 7.4	Controller relationship for arm26 model.....28
Figure 7.5a	OpenSim femur model.....29
Figure 7.5b	Full-body OpenSim model.....29
Figure 7.5c	OpenSim head and neck model.....30
Figure 7.5d	Arnold OpenSim lower limb model.....30
Figure 7.5e	OpenSim lower extremity model.....31
Figure 7.5f	OpenSim lower limb model.....31
Figure 7.5g	OpenSim musculoskeletal lumbar spine model.....32
Figure 7.5h	OpenSim torso and lower extremity model.....32
Figure 7.5i	OpenSim upper and lower body model.....33
Figure 7.5j	OpenSim upper extremity model33
Figure 7.5k	OpenSim wrist model.....34
Figure 8	OpenSim platform for the “Arm26” upper extremity variables.....37
Figure 9	Shows the upper-limb geometry and muscle paths.....37

	PAGES
Figure 10a	Upper extremity lateral view with arm in 45 degrees of flexion.....38
Figure 10b	Upper extremity anterior view with arm in 45 degrees of abduction.....38
Figure 10c	Upper extremity anteromedial view.....38
Figure 10d	Upper extremity posterolateral view, demonstrating the degree of freedom, muscle paths and selected wrapping surfaces.....38
Figure 11	Coordinates and Marker.....40
Figure 12	An OpenSim GUI platform showing MoBL ARMS upper model and list of body path modification tools.....41
Figure 13	Coordinate menu for OpenSim MoBL model and its features.....44
Figure 14	File menu and the load motion.....45
Figure 15	Loaded motion static pose..... 46
Figure 16	Loop play and pause motion control.....48
Figure 17	Motion plot layout and features..... 49
Figure 18	Muscles selection: 50 muscles you can select for a plot.....50
Figure 19	Muscle selection toolbox.....51
Figure 20	GUI representation of the muscle selection..... 52

Figure 21	MoBL ARMS Upper arm model OpenSim active muscles (in blue) in an initial default state of 0 degrees.....	56
Figure 22	Muscle change characteristics of a drink from a cup position.....	75
Figure 23	Typical flexion of the arm showing the distance of moment arm from line of action to joint center.....	77
Figure 24	Motor point of an anterior upper extremity.....	78
Figure 25	Motor point of a posterior upper extremity.....	79
Figure 26	Shows elbow flexion moment arm against elevation angle.....	79
Figure 27	Shows tendon force against elevation angle.....	79
Figure 28	Total fibre force against elevation angle.....	80
Figure 29	Passive fiber force against elevation angle.....	80
Figure 30	Muscle activation reflex for Tendon fibre force at zero degrees elevation angle.....	83
Figure 31	The muscle activation reflex for total tendon force at seventy (70) degrees elevation angle.....	84
Figure 32	The muscle activation reflex for total tendon force at (one hundred and thirty (130) degrees elevation angle.....	85

Figure 33	Muscle activation reflex for Tendon force at Zero (0) degrees elevation angle.....	86
Figure 34	Muscle activation reflex for total tendon force at seventy (70) degrees elevation angle.....	87
Figure 35	Muscle activation reflex for tendon force at one hundred and thirty (130) degrees elevation angle.....	88
Figure 36	Confidence fit of total fibre force by tendon force at zero (o) degrees.....	89
Figure 37	Confidence fit of total fibre force by tendon force at seventy (70) degrees.....	89
Figure 38	Confidence fit of total fibre force by tendon force at one hundred and thirty (130) degrees.....	90
Figure 39	Muscle difference relation of the total fibre force at 70 degrees against total fibre force at 0 degree, with Correlation 0.93115.....	90

Figure 40	Muscle difference relation of the total fibre force at 130 degrees against total fibre force at 0 degree, with Correlation 0.66117.....	91
Figure 41	Muscle difference relation of the total fibre force at 130 degrees against total fibre force at 70 degree, with Correlation 0.7442.....	91
Figure 42	Muscle difference relation of the tendon force at 70 degrees against to tendon force at 0 degree, with Correlation 0.97393.....	92
Figure 43	Muscle difference relation of the tendon force at 130 degrees against to tendon force at 0 degree, with Correlation 0.66268.....	92
Figure 44.	Muscle difference relation of the tendon force at 130 degrees against to tendon force at 70 degrees, with Correlation 0.67042.....	93
Figure 45	Overlay plots for total fibre force at different angle and muscle path.....	94

Figure 46	Overlay plots for tendon force at different angle and muscle path.....	95
-----------	---	----

LIST OF TABLE

	PAGES
Table 1	Passive activation force at 0 degree, 70 degrees and 130 degrees in relations to the muscles existing in the human upper extremity.....81
Table 2	Muscle measurements at points 0, 70 and 130 degrees.....96

ABSTRACT

Engineering and Biomechanics researchers often use multibody models to characterize biological systems. However, the cross relations from biology to mechanics and back can be a problem. In order to overcome some of these problems, a computed muscle control environment is used to rapidly generate forward dynamics simulation of a typical musculoskeletal body.

In this thesis, OpenSim dynamic simulation tool is used to investigate the role of motor control in the production of human movement, specifically the upper arm for reaching and “drinking from a cup” motion. Individual muscle contribution is analyzed for healthy movement which will help for further pathologic development on the augmented wearable device intended to help stroke victims learn and recover from any muscle weakness.

MoBL ARMS model is developed, animations of certain movements are made and relative response plot is analyzed. Also included is the observation from a graphical representation of the muscle activity in relations to forces applied.

Finally, the correlation of muscle activity is used to estimate steady-state activation levels using the computed forces in the muscle and a detailed user manual is designed for further research guide. The analysis included five (5) degree of freedom (forearm, rotation, elbow flexion, shoulder rotation, shoulder elevation and shoulder elevation angle) and 23 muscles crossing the shoulder, elbow and forearm.

CHAPTER ONE

1.0 INTRODUCTION

Stroke is a leading cause of disability worldwide. On a yearly average, every 40 seconds, someone in the United States has a stroke and someone dies of one approximately every 4 minutes (Heart and stroke Facts and statistics. American Heart Association; 2014). More than 795,000 people suffer from new or recurrent stroke (Ischemic or hemorrhagic) in the United States alone. 610 000 of these are first happenings and 185 000 are recurrent stroke happenings and in 2011 about 1 of every 20 deaths in the United State was caused by stroke and 80% of acute stroke survivors lose motor skills of the arm and the hand. (Heart and stroke Facts and statistics. American Heart Association; 2014). The National Science Foundation funded project for engineering research and development aim to develop an innovation for rehabilitation of stroke patients using a new Augmented Reality Wearable Device (ARWED) that taps into the visual information and action execution of stroke patients.

Stroke is caused by a blocked blood vessel or bleeding in the brain. Symptoms can vary depending on whether the stroke is caused by a blood clot or bleeding or where the stroke occurs in the brain and how bad it is. The signs of a stroke may include a sudden severe headache, weakness, numbness, vision problems, confusion, trouble walking or talking, sudden confusion or trouble understanding simple statements, severe headache, dizziness and slurred speech, (Stroke foundation). The risk of stroke increases with age and it is relative to family history, race and gender. It usually happens over time starting with

mild weakness and over time the arm and leg on one side of the body may not be able to move.

1.1 ARWED PROPOSAL SOLUTION

The collaborative research and development of an Augmented Reality Wearable Device (ARWED) is one of the future solutions to the problem of body motor system correction. The objective of the research is to develop a tool for increasing training using a new augmented reality device that taps into the visual information to accomplish task and as a means to help overcome some of the proprioceptive loss stroke patients experience. Researchers in the past years discovered mirror neurons and network (Rizzolatti: 1996) which supports the ability to learn through action imitation and action observation (Rizzolatti: 2001, et al., 2006).

Mirror neurons basically support “action understanding” which means the capacity to recognize that an individual is performing an action, to differentiate this action from other analogous to it and to use the information in order to act appropriately. (Rizzolatti et al., 2001; Gallese et al., 1996). Mirror neuron primarily is the capacity to achieve the internal description of an action and to use it to organize appropriate future behavior. Each time an individual sees an action done by another individual, neurons that represent that action are activated in the observer’s premotor cortex. This automatically induced motor representation of the observed action corresponds to that which is spontaneously generated during active action and whose outcome is known to the acting individual. Hence the mirror system transforms visual information into knowledge (Nelissen et al., 2005).

Research studies show that intensive goal-directed repetition of movement promotes recovery following a stroke (Whitall J., 2000). The possibility in the study gives birth to the present research to develop an Augmented Reality Wearable Device (ARWED) to create a virtual reality environment for the human arm. It will also develop relearning protocols with the ARWED that enhances active range of motion and use issues of the most affected arm of stroke patients.

1.2 MOTION INTENTION

The focus of this work is to access the motion functionalities of the upper extremity arm and provide a detailed observational learning of the human arm-hand motion. The upper extremity hemiparesis is among the most significant and persistent physical disabilities following stroke and it represents a critical barrier to independence, (<https://www.youtube.com/watch?v=4OILUCgEGOG&list=PLGBiefGDmSuR5NbZFuGOrjnTBx--H-H3C>). Since the problem is well recognized there is little evidence demonstrating the most effective approach for promoting functional motor recovery of the hemi_paretic upper extremity. Prominent manifestations of compromised motor control following stroke include: Paired inter-segmental coordination, hyperreflexia or spasticity and weakness, which are mechanical factors associated with the muscle fibre (<https://www.youtube.com/watch?v=4OILUCgEGOG&list=PLGBiefGDmSuR5NbZFuGOrjnTBx--H-H3C>).

The general clinical manifestations of stroke have effects on many body functions, including motor activity, bladder and bowel elimination, intellectual function, spatial-perceptual alterations, personality, sensation, swallowing and communication. These

dysfunctions are directly related to the artery involved and area of the brain it supplies. The computed software defines a means to reflect the characteristic motor deficit which include skilled voluntary movement, impairment of integration of movements, alterations in muscle tone and reflexes. Medical science identifies that motor deficit follows a certain specific pattern because the pyramidal pathway crosses at the level of the medulla, a lesion on one side of the brain which affects motor functions on the opposite side of the body (contralateral).

The arms and legs of affected sides may be weakened or paralyzed to different degrees depending on which part of and to what extent the cerebral circulation is compromised. The ARWED assisted therapy goal is to be equipped with a sensor that records position, velocity and force exerted by the patient, which can be used to measure the patient's motor performance quantitatively, objectively and also perform repetitive tasks consistently and controllably.

1.3. sEMG MEASUREMENT AND MUSCLE PATTERN SIMULATION

To test the effect of the action execution through action observation of the human movement control, the signal analysis (sEMG) of a reduced mobility reach and pinch human arm-hand motions are analyzed. The sequence will reflect a real time symmetry of the healthy limb and the expense of the action-observation priming effect can then be linked to the mirror-neuron systems on healthy subjects and then stroke patients (Laura D. et al., 2005). The beauty of the electromyography (EMG) is to trigger the assisted therapy for stroke patients. The onset of a patient's attempt to move is detected by monitoring EMG signals in selected muscles and the robot assist can help patient perform point to pinch or

reaching movements. Data signals can be recorded and used to better understand the process of recovery from stroke (Laura D. et al., 2005).

1.4 PAST RESEARCH WORK

A lot of research and development projects have been surfacing many years before now and they all make the rehabilitation of stroke victims a central topic around the world. An example is the MIT-MANUS (Kreds H. et al., 1998) which is a back derivable adaptive robot-aided device used for stroke neuro-rehabilitation and recovery therapy. Performed for over 2000 hours of robot-aided therapy with 76 stroke patients and concluded that robot-aided sensorimotor training does have genuine positive effect on reduction of impairment and reorganization of the adult brain (Kreds H. et al., 1998). Another example is the ARM guide which is an “Assisted Rehabilitation and Measurement Guide.” It is a mechatronic device for evaluating arm movement impairment after a brain injury. It is designed to guide and measure reaching movements and force generation and also to study contributions of active muscles and passive tissues restraints. (David R. et al., 1999). Among others is “A model of the upper extremity for simulating musculoskeletal” (Katherine R. S. et al., 1998). It describes a biomedical model of the musculoskeletal system to study neuromuscular control and simulate surgical procedures with OpenSim. With so much development done and ongoing in the neuro- rehabilitation, the success of the MIME project cannot be over emphasized. MIME is a servomechanism device used to move the forearm in a simple planar trajectory, evaluating and measuring abnormal upper-limb motor performance of post stroke patients.

The MIME controls hand position and forearm orientation and it was tested on 13 stroke victims with different impairment levels and a resulting performance data were

generated. (Lum P. et al., 1991). The potential advantages have stimulated an expanding array of research effort using customized robot-assisted therapy. Researchers use observation of another's action to expand the array of research effort of using customized robot-assisted therapy and the potential is stimulating great recovery and positive advantage to stroke victims.

1.5. THESIS OUTLINE

In order to accomplish the development of the ARWED, the priority is to design a goal in computer vision, simulation and modelling will need to be saved in order to enhance an understanding of the effect of the device on the priming of action execution via action observation. OpenSim software helps give a close evaluation of the upper arm-hand human movement control and muscle sequence. These result can then provide a background for the analysis of sEMG signals.

The objectives of this thesis are listed as follows;

1. Understand the principles behind Upper arm musculoskeletal behavior.
2. Develop an OpenSim model.
3. Provide a user manual for OpenSim software use.
4. Make animation of certain movements.
5. Obtain graphical representation of the muscle activity with movements of joints and forces.
6. Correlate muscle activity to analyze steady-state activation levels using the excitation forces which will give other researchers working on stroke victims a guide and directions to muscles associated in fatigue and recovery rate.

CHAPTER TWO

2.0 MUSCLE ANATOMY OF THE UPPER EXTRIMITY ARM

The major exterior arm muscles give the arm its curved shape on the inside and outside. It also gives the arm its strength as they contract or expand with arm movements. Muscles that move the forearm and hand can be seen in Figure 1, however the major surface muscles includes;

- A. BICEP: A large muscle of the upper arm, when flexed it is often viewed as a sign of a person's strength and called biceps brachii muscle. It rest on the humeruss bone and also help rotate the forearm and as well flex the elbow.
- B. TRICEP: It is the large muscle in the back of the upper arm which helps strengthen the arm. It is also called the tricep brachii muscle.
- C. BRACHIORADIALIS: It is located at the top of the forearm near the elbow and helps to rotate the forearm both outwardly and inwardly and sometimes called the 'beer drinker muscle'.
- D. EXTENSOR CARPI RADIALIS LONGUS: It is located next to the brachioradialis and helps move the wrist. It bulges out from the skin when the fist is clenched.
- E. DELTOID: It controls the majority of the shoulder's movement and gives the arm more range of motion.

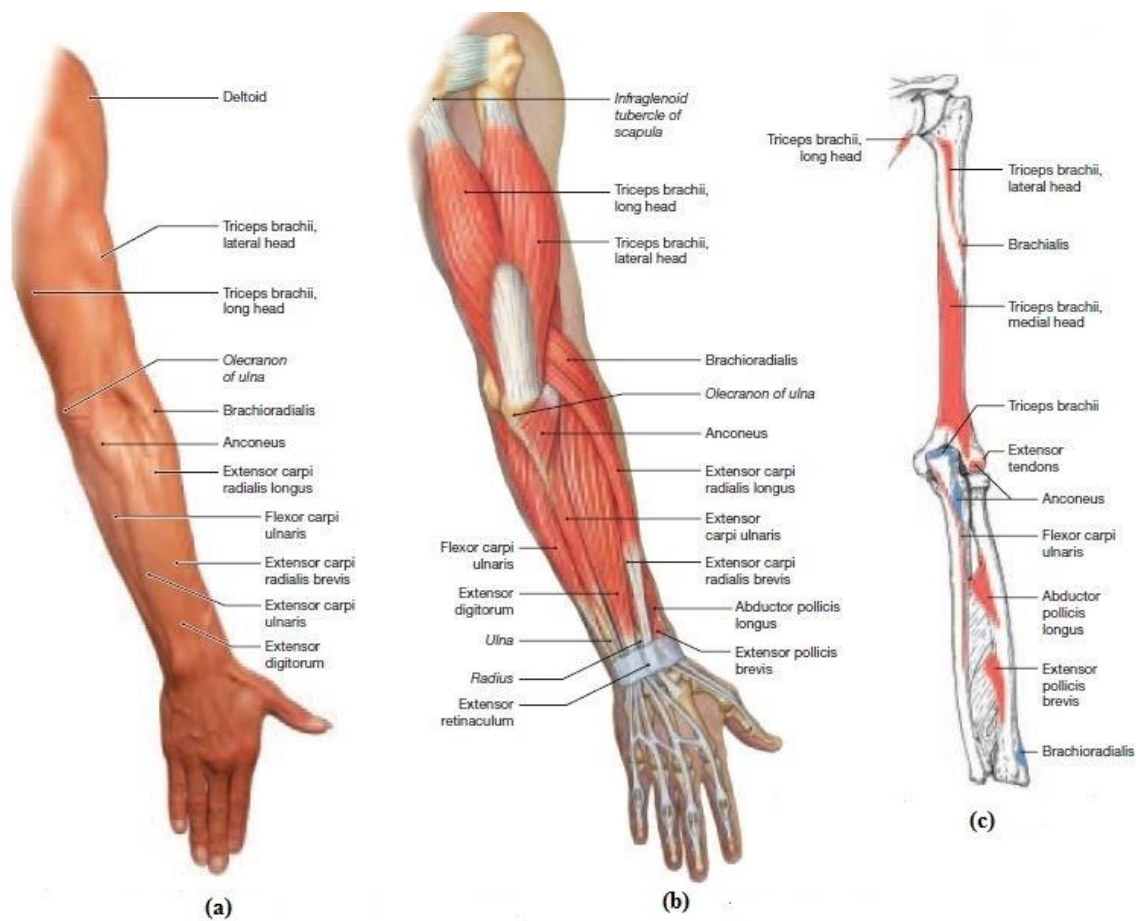


Figure 1. Shows the relationship among the muscles of the upper limb. (a). Surface anatomy, posterior muscle view (b). Superficial muscles, posterior muscle view (c). Origins and insertions, posterior view. Retrieved from <http://www.aw-bc.com/info/martini6e/assets/pdf/chapter11.pdf>

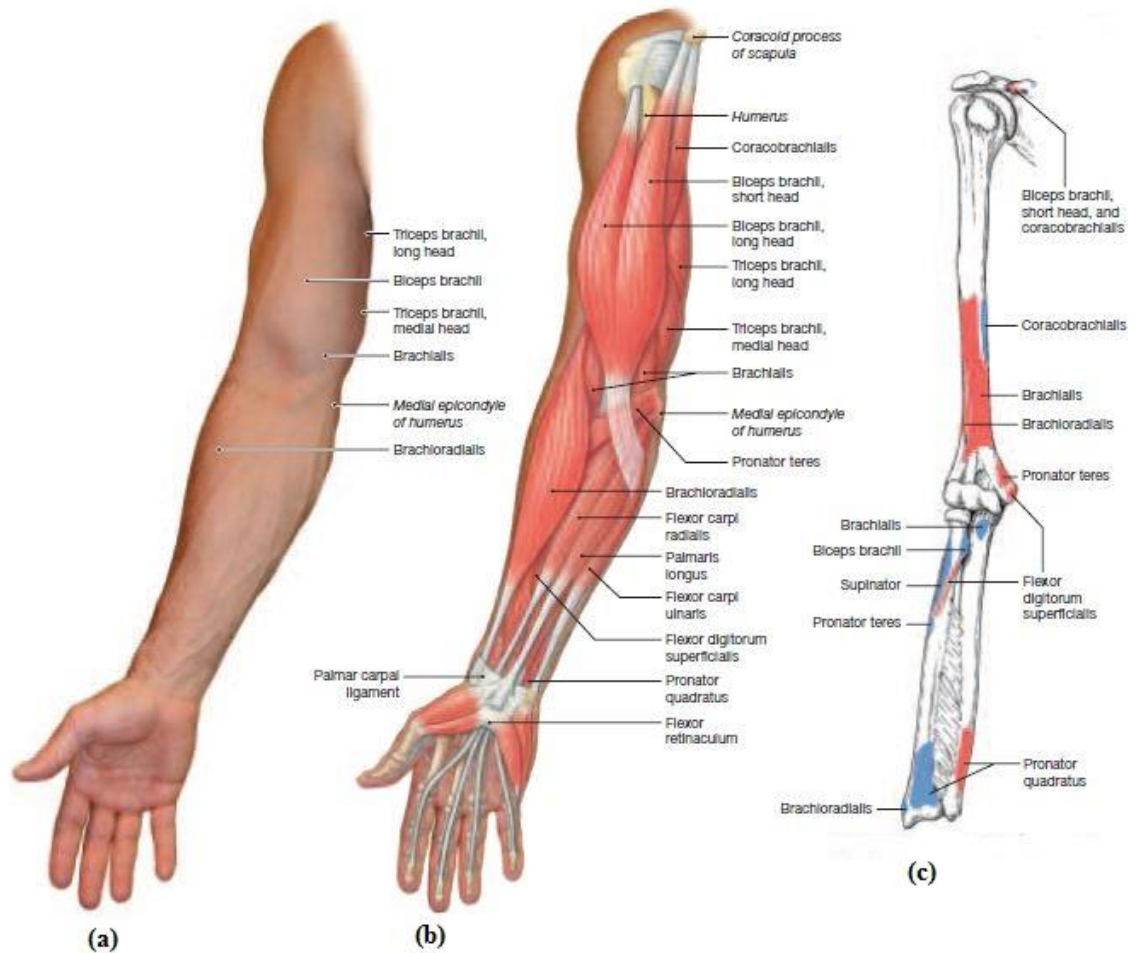


Figure 2. Shows the relationship among the muscles of the upper limb. (a). Surface anatomy of the upper limb, anterior muscle view (b). Superficial muscles of the upper limb, anterior muscle view (c). Anterior view of bones of the upper limb. Retrieved from, <http://www.aw-bc.com/info/martini6e/assets/pdf/chapter11.pdf>

The combination of the shoulder, arm and hand defines the full anatomy of all responses and movement of the upper limb in Figure 1 posterior view and Figure 2 anterior

view shown above. Details of how these sections contribute to the upper limb movement are described in the subsequent paragraphs.

2.1 THE SHOULDER

This is a complex ball and socket joint including the humerus, clavicle (collarbone), and the scapula. Its main motions are as follows;

1. Flexion
2. Extension
3. Adduction
4. Abduction
5. Internal rotation
6. External rotation

Retrieved from, <https://www.youtube.com/watch?v=vG1XQkj3Yx0&index=9&list=PL78D995229606FDE>).

The shoulder flexion is the movement that has to do with moving the shoulder in a forward motion. An example is reaching forward to grasp an object. The action is achieved primarily by combining actions of deltoid muscle in the uppermost extent of the arm, the pectoralis major muscle in the chest, the corabrachialis muscle on the inside of the upper arm and the biceps brachii muscles on the front of the upper arm as shown in Figure 3. The shoulder extension is an opposite movement to flexion. It involves movement of the arm directly behind the body, just as when receiving a baton in a relay race. Retrieved from, <https://www.youtube.com/watch?v=vG1XQkj3Yx0&index=9&list=PL78D995229606FDE>)

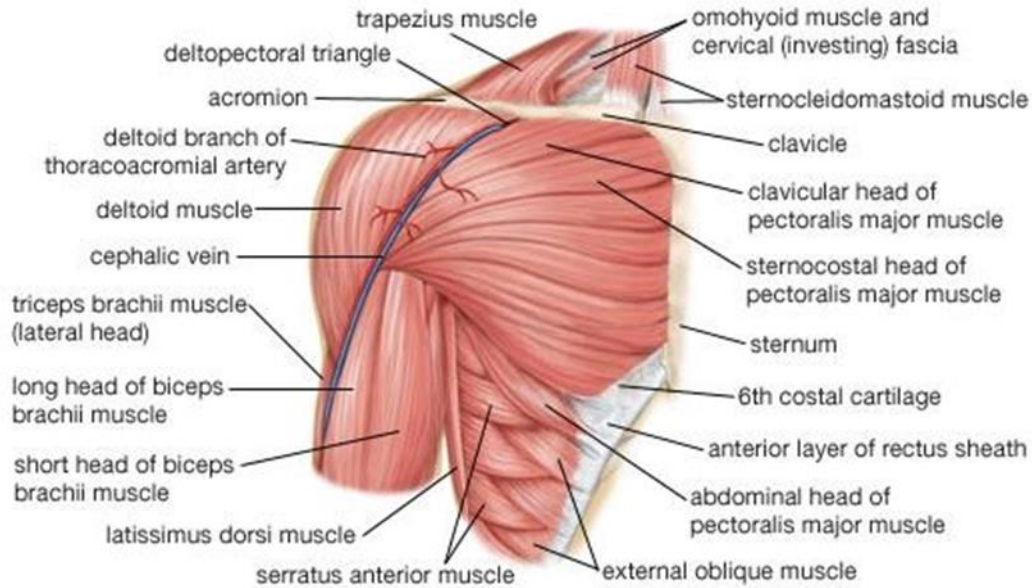


Figure 3. Shows the anterior muscle view of the human shoulder. Retrieved from <http://www.aw-bc.com/info/martini6e/assets/pdf/chapter11.pdf>

The movement is achieved by actions of the deltoid muscle, the latissimus dorsi muscle in the back, the teres major muscle in the armpit area and the triceps muscle in the back of the upper arm. The triceps comprises of three head that originate from different surfaces but share the same insertion at the olecranon process of the ulna (a bone in the forearm), the three heads together act to extend the elbow.

Adduction and abduction of the shoulder respectively behave to lower the arm toward and lift the arm away from the body. A typical example is someone doing a jumping jack workout. In achieving adduction the pectoralis major, latissimus dorsi, teres major, triceps and coracobrachialis interact together while the deltoid and the supraspinatus, a muscle that runs along the scapula in the back forms the main abductors of the shoulder (Jennifer Hicks, 2014).

Tennis backhand stroke describes more about the external rotation of the shoulder and this is effective due to the interaction of the deltoid, teres minor in the armpit area and the infraspinatus muscle, which covers the scapula (Jennifer Hicks, 2014). However the internal rotation of the shoulder is simply the opposite of the external rotation in the sense that the movement is accomplished by the coordination action of the pectoralis major, latissimus dorsi, deltoid, teres major and subscapularis muscles. (The subscapularis is a deep muscle situated on the anterior (front facing) of the surface of the scapula). The ball portion of the ball and socket shoulder joint is stabilized by the teres minor, subscapularis, supraspinatus and infraspinatus muscles which all primarily forms what is called the rotator cuff (Jennifer Hicks, 2014).

The muscles of the rotator cuff are common sites of injury in adults, especially with people who perform overhead motions repeatedly. An example is throwing a baseball or painting a ceiling. Under the acromion (a bony bump at the distal end of the scapula), several of the rotator cuff muscles have tendons running through. The location of the tendons and subacromial bursae (fluid filled sacs located beneath the acromion) at the distal end leaves them vulnerable to compression and pinching, which can result in injury known as shoulder impingement syndrome (SIS).

2.2 THE ARM

The muscles of the upper arm not only support the movement of the shoulder but they also make the forearm move as well. For example, the main muscles involved in forearm flexion are the brachialis, the biceps brachii in the upper arm, and the brachioradialis (as shown in Figure 4 below) which is connected at the humerus bone. For the hand to move away from the shoulder, the opposite effect happens. When the hand

moves toward the shoulder, the angle of the elbow decreases but when it moves away from the shoulder, the angle of the elbow increases. The most important muscle involved in forearm extension is the triceps brachii. Other muscles such as the extensor carpi radialis longus, extensor carpi radialis brevis and the extensor digitorum contribute as well. It is important to note that flexor muscles are used to bring the arm toward the shoulder and extensor muscles are used to bring the arm away from the shoulder with the palm of the hand facing upward.

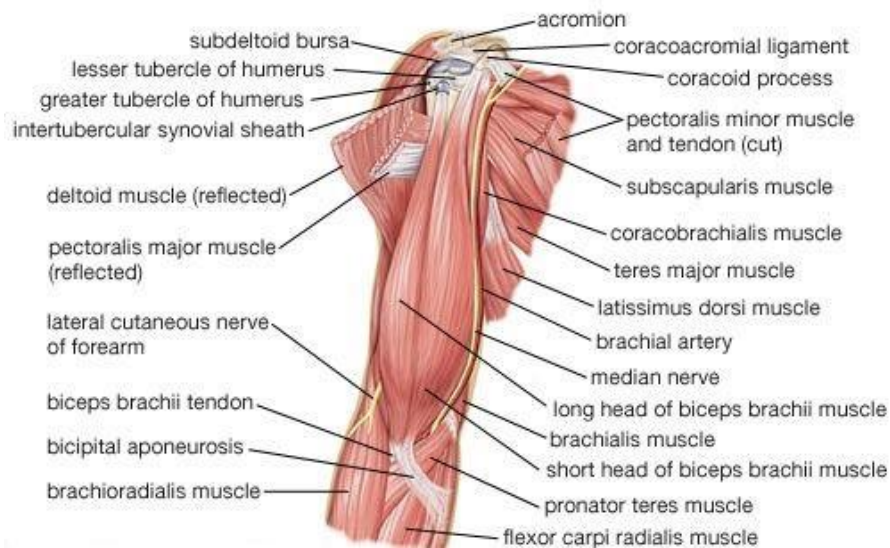


Figure 4. Shows the anterior muscle view of the upper arm. Retrieved from, <http://www.aw-bc.com/info/martini6e/assets/pdf/chapter11.pdf>

2.3 THE WRIST

Wrist flexion refers to movement of the wrist that pulls the palm of the hand downward as if the person is trying to touch their fingertips to the inner portion of their wrist. That action is carried out by the flexor carpi ulnaris, the flexor carpi radialis, the flexor digitorum profundus, the flexor digitorum superficialis, and the flexor pollicis longus.

Wrist extension does the opposite motion where it shortens the angle at the back of the wrist. For example, a person puts their arm and hand straight out in front of them and raises their fingertips to the sky. The muscles in control of that motion are the extensor carpi radialis longus and the extensor carpi radialis brevis. These muscles also abduct the hand at the wrist, where the person moves the hand in the direction of the thumb and the extensor digitorum, which also extends the index to little finger; the extensor digiti minimi, which also extends the little finger and adducts the hand (moves the hand in the direction of the little finger); and the extensor carpi ulnaris, which also adducts the hand. Other small muscles that cross the wrist joint may add to wrist extension, but only in a very minor way.

Wrist supination is the rotation of the wrist that brings the palm facing up. Think of the hand as a bowl of soup, as with how the word “supination” sounds to remember this term. The supinator muscle in the posterior section acts to supinate the forearm. The biceps brachii also adds to supination. Pronation is the opposing action, in which the wrist is rotated so that the palm is facing down. Think of the person dumping out the bowl of soup. The pronator quadratus, a deep muscle in the anterior section, along with the pronator teres, pronates the forearm.

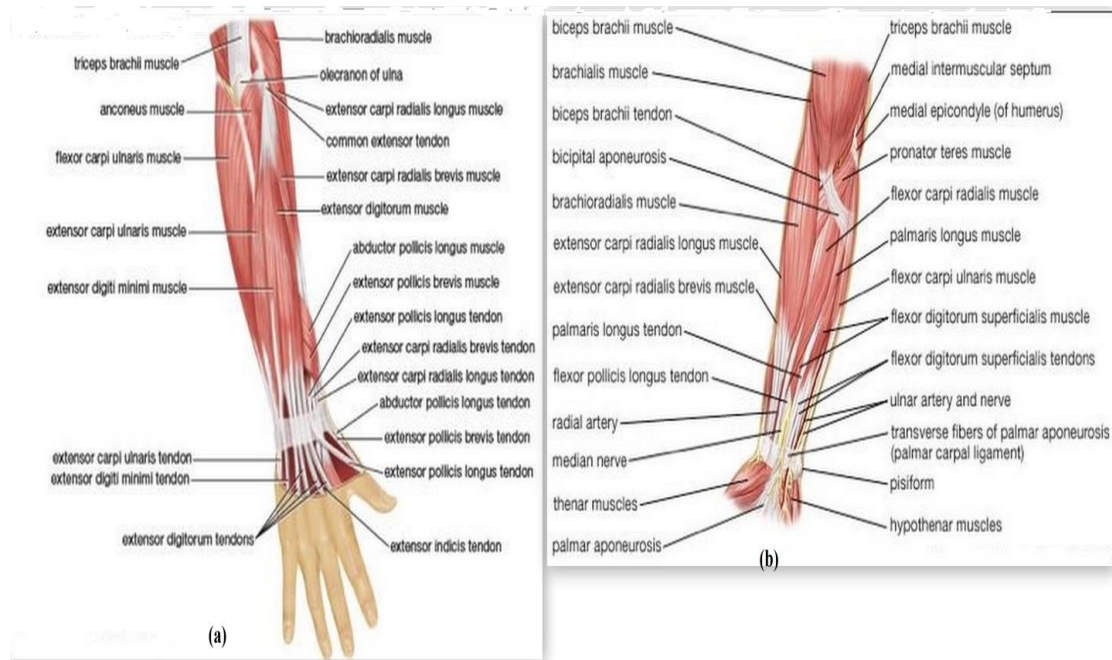


Figure 4.1 Shows (a) posterior muscle view of the forearm and (b) shows the anterior muscle view of the forearm. Retrieved from, <http://www.aw-bc.com/info/martini6e/assets/pdf/chapter11.pdf>

2.4 THE HAND

The hand is full of very small muscles and a lot of nerves which create a multifaceted structure used for fine motor skills such as writing and grasping small objects. The next paragraphs only cover the primary actions of each hand muscle for the purposes of this project. Several muscles that are used in the hand begin at the posterior surface of the ulna or radius bones which are part of the forearm. Those include the abductor pollicis longus, which abducts and extends the thumb; the extensor pollicis brevis, which extends the metacarpophalangeal (MCP) joint of the thumb; the extensor pollicis, which extends the distal phalanx (finger bone) of the thumb; and the extensor indicis, which extends the

index finger at the MCP joint. MCP joints are located between the metacarpal bones, which are situated in the hand, and the phalanges, which are the small bones of the fingers.

There are many small muscles that are located directly in the hand and are not attached to the arm bones which are called intrinsic muscles. They include the palmaris brevis, which assists with grip; the lumbricals, which flex the MCP joints and extend the interphalangeal joints (IPs; the joints between the phalanges) of the fingers; the palmar interossei, which adduct the fingers toward the middle finger (the third digit); and the dorsal interossei, which abducts the fingers away from the middle finger. All the interossei flex the MCP joints and extend the IP joints.

The thenar eminence is located on the palm side of the base of the thumb and is composed of three muscles, the abductor pollicis brevis, the flexor pollicis brevis, and the opponens pollicis, all of which are innervated by the median nerve as shown in Figure 5a below. The abductor pollicis brevis abducts the thumb; the flexor pollicis brevis flexes the MCP joint of the thumb; and the opponens pollicis acts to oppose the thumb to the other fingers. The adductor pollicis, which is not part of the thenar eminence, acts to adduct the thumb.

The hypothenar eminence is located on the palm side of the hand (Figure 5), below the little finger. It contains three muscles that receive their nerves from a branch of the ulnar nerves. The abductor digiti minimi as shown in the Figure 5b below, abducts the little finger, the flexor digiti minimi flexes the little finger and the opponens digiti minimi opposes the little finger with the thumb.

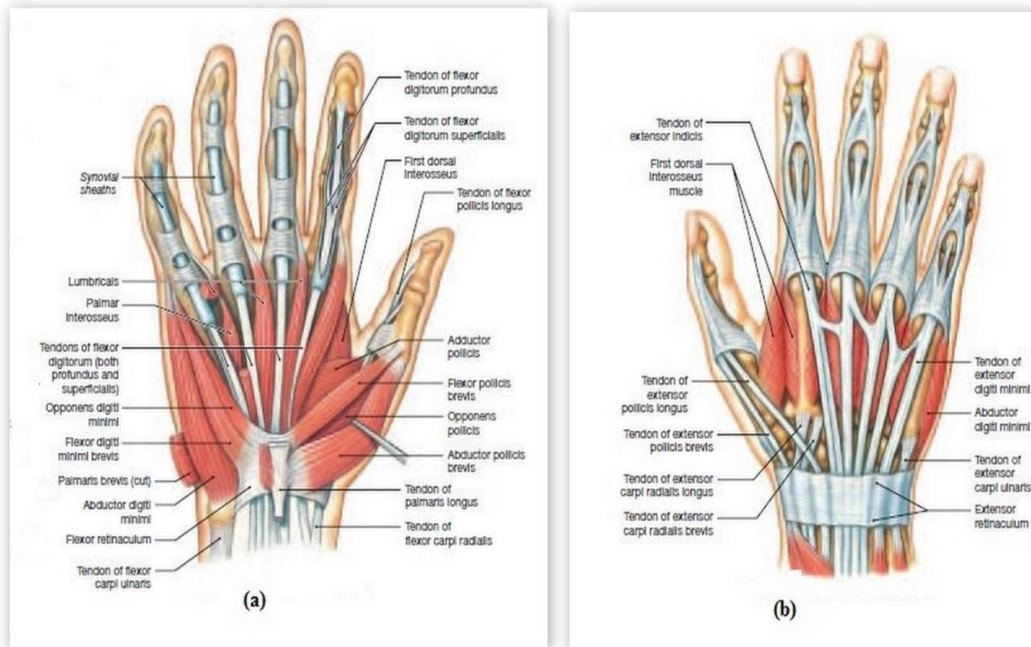


Figure 5. Shows (a) Right hand anterior (palmer) muscle view and (b) Right hand posterior (dorsal) muscle view. Retrieved from, <http://www.aw-bc.com/info/martini6e/assets/pdf/chapter11.pdf>

CHAPTER THREE

3.0 OPENSIM

3.1 BACKGROUND

In solving complex engineering problems, numerical simulations play an important role especially in areas of medical decision making and treatment strategies. Musculoskeletal conditions cost the United States economy over \$849 billion annually and place great demands on the health care systems all over the world (Toby king, 2008). Research and treatments of these conditions can be greatly beneficial from the combination of software tools which offers better insights into neuromuscular biomechanics.

Engineering software such as (ANSYSTM, ADAMSTM, SD-FASTTM, MATLABTM) and musculoskeletal software packages (SIMM, VISUAL 3-D) are the two general categories used for modeling and simulation of biomechanical systems. From research, it is difficult to use built-in mechanical and electrical elements in the engineering software and likewise the musculoskeletal software unfortunately do not share model standards or support data exchange as the engineering packages do. Currently, a freely available modeling and simulation musculoskeletal platform is developed called the OpenSim and can successfully interphase with the MATLAB/SimulinkTM engineering software package. Although user can extend OpenSim by writing their own plug-ins in C++. However the OpenSim software generally lacks the robust design and control components which are needed for real time changes to input controls compared to MATLABTM which is a powerful mathematical computing and control software. Hence MATLABTM is a natural choice to complement OpenSim.

3.2 INTRODUCTION

OpenSim is an open source software that models human muscle movements. It is a freely available software package written in ANSI C++ and the graphical user interphase (GUI) is written in Java. The application builds musculoskeletal models used in research and also provides a platform on the biomechanics community can build a simulation library which can be exchanged, tested, analyzed and improved.

The software is used in many applications, including biomechanics research, medical device design, orthopedics, rehabilitation science, neuroscience research, ergonomic analysis and design, sports science, computer animation, biology, robotics research and education. It is set up such that it can analyze and develop existing or new models of musculoskeletal structures and dynamic simulations from within the GUI and likewise customize controllers, analyses, contact models and muscle models to be shared without need to alter or compile source code. For example is the use of existing computer models of the human lower limb, upper limb, cervical spine and whole body already developed and posted on the OpenSim software platform.

3.3 OPENSIM COMPATIBILITY AND MODELS

OpenSim is a self-reliant modelling and simulation environment that does not need any other additional software to generate its dynamic simulations. The Simbody library which is an open-source dynamic system for creating mathematical models of biological dynamics is the library OpenSim is built on while the commercial package SIMM (software of interactive musculoskeletal modeling) is the platform which OpenSim was developed from. OpenSim amplifies and complements the functionality of SIMM and the joint (.jnt)

and muscles (.msl) files used to describe musculoskeletal models in SIMM can be converted into OpenSim models (*.osim) and brought into the OpenSim framework.

OpenSim is incorporated with Simulink using the MATLABTM S-function mechanism, which provides an interface that combines rapid model-based design, control systems and powerful numerical method strength of MATLAB/SimulinkTM with the simulation and human movement dynamics strength of OpenSim. A Typical case study is the arm model example which successfully uses an interphase that requires both open-loop and closed-loop control systems. The open-loop system uses MATLAB/SimulinkTM to separately reproduce the OpenSim forward dynamic tools while the closed loop system adds the unique feature of feedback control to OpenSim which is necessary for most human movement simulations. Figure 6 describes the OpenSim models of different musculoskeletal structures which can be loaded, viewed, analyzed and plotted with red showing muscles and blue for virtual markers.

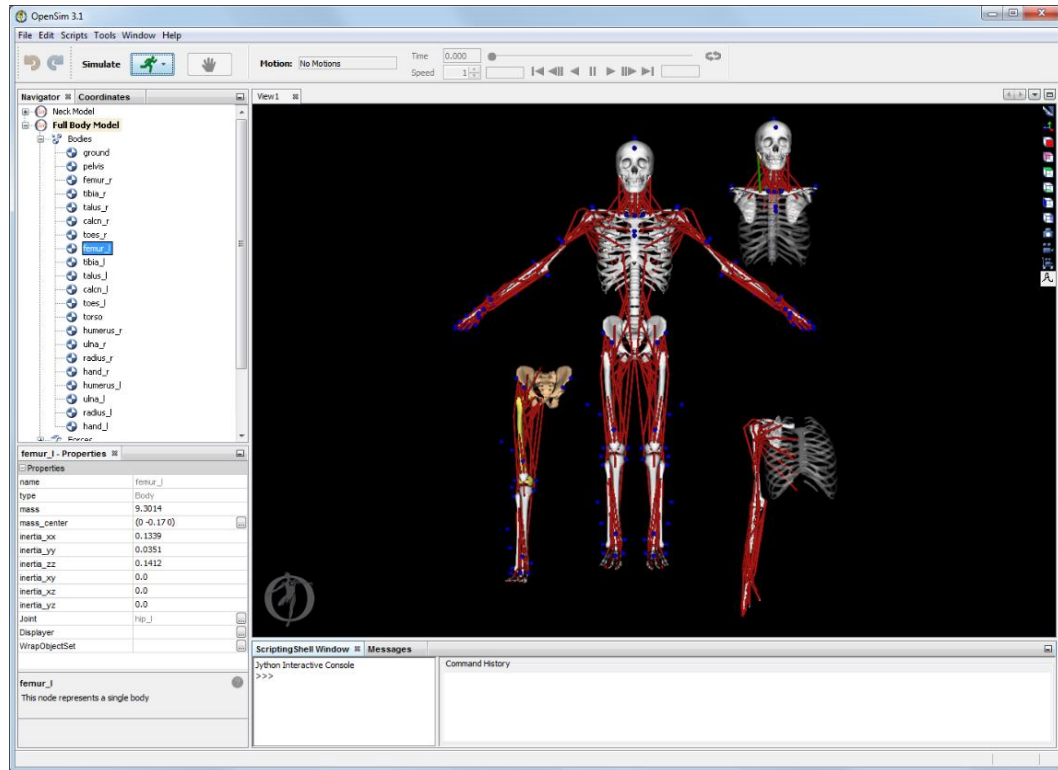


Figure 6. OpenSim musculoskeletal structures (Kevin Xu, 2012).

3.31 OPENSIM MODELS

This describes the dynamics of a system of rigid bodies and joints acted upon by forces to produce motion. The model files such as in Figure 6 include corresponding physical parts such as bodies, joints, forces, markers, constraints, contact geometry and controllers.

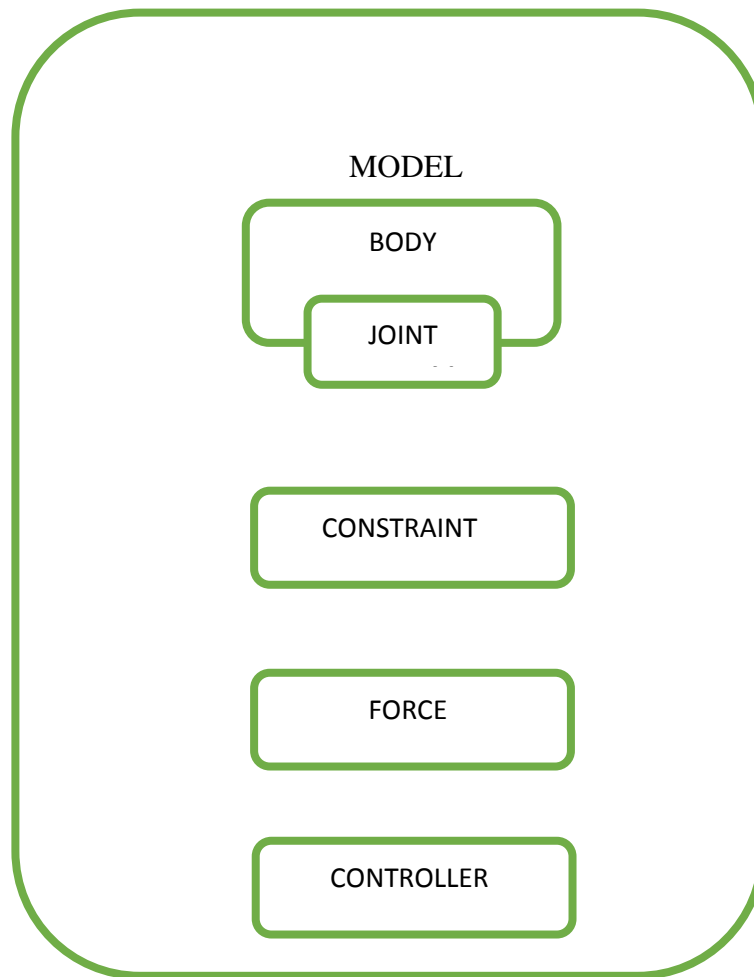


Figure 7. OpenSim Model structure (Jennifer Hicks, 2014).

3.32 BODIES

In order to start a model a set of rigid bodies needs to be defined to represent the system. The group of bodies is defined with the name, mass properties and visible objects associated with each body. Example is an XML code from a model ARM26 which represents a body;

<Body name="r_humerus">

<mass>1.864572</mass>

<mass_center> 0 -0.180496 0</mass_center>

<inertia_xx>0.01481</inertia_xx>

<inertia_yy>0.004551</inertia_yy>

<inertia_zz>0.013193</inertia_zz>

<inertia_xy>0</inertia_xy>

<inertia_xz>0</inertia_xz>

<inertia_yz>0</inertia_yz>

<Joint>

...

3.33 BODY GEOMETRY

In OpenSim the body/segment and the geometry are very different, although the geometry of a body is used for a visualization purpose. The body geometry allows for specifying and positioning of the visual object property associated with a body. In the model, file type .vtp, stl. or .obj are used to visualize geometry while .jpg or .bmp file types specify color and texture.



Figure 7.1 Shows body geometry on OpenSim (Saul k. 2014)

3.34 JOINTS

To define the relationship between the bodies such as the joint definitions to result into the rigid bodies there is need to parameterize the joint coordinates. An example is the kinematic relationship between two frames B and P, see Figure 7.

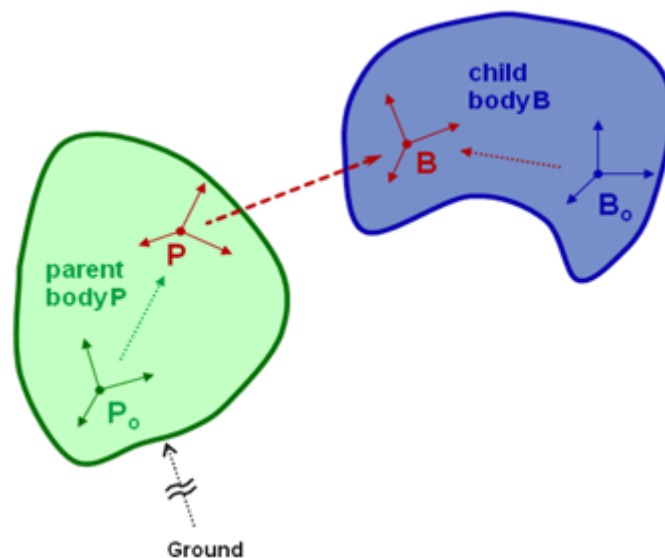


Figure 7.2 Kinematic relationship between two frames (Jennifer Hicks, 2014).

Two joints with parent rigid body P_0 and additional body added B_0 have joint frame B and P. The flexibility in specifying joint frames are not coincident with the body frame due to definitions given to the center-of-mass and inertia moving reference frame B_0 . Following the ARM26 model example, the body $r_humeral$ contains the joint $r_shoulder$. The figure below shows an example from Arm26 defining the $r_shoulder$ joint. Note the key tags, such as `< parent_body>`, `<location_in_parent>`, `<orientation_in_parent>`, `<location>`, `<orientation>`, and `<Coordinate>`.

```

<!--Joint that connects this body with the parent body-->
<Joint>
  <CustomJoint name="r_shoulder">
    <parent_body> ground </parent_body>
    <location_in_parent> -0.01754500 -0.00700000 0.17000000 </location_in_parent>
    <orientation_in_parent> 0.00000000 0.00000000 0.00000000 </orientation_in_parent>
    <location> 0.00000000 0.00000000 0.00000000 </location>
    <orientation> 0.00000000 0.00000000 0.00000000 </orientation>
    <!--Generalized coordinates parameterizing this joint-->
    <CoordinateSet name="">
      <objects>
        <Coordinate name="r_shoulder_elev">
          <!--Coordinate can describe rotational, translational, or coupled values.
          Defaults to rotational-->
          <motion_type> rotational </motion_type>
          <default_value> 0.00000000 </default_value>
          <default_speed_value> 0.00000000 </default_speed_value>
          <initial_value> 0.00000000 </initial_value>
          <range> -1.57079633 3.14159265 </range>
          <clamped> false </clamped>
          <locked> false </locked>
          <prescribed_function/>
        </Coordinate>
      </objects>
      <groups/>
    </CoordinateSet>
    <reverse> false </reverse>
    <SpatialTransform name="">
  </CustomJoint>
</Joint>

```

Figure 7.3 Example of XML Code from Model Arm26 to Represent a Joint (Jennifer Hicks, 2014).

3.35 TYPES OF JOINT

1. Weld Joint: Which introduces no coordinates (degree of freedom) but fuses bodies together.
2. Pin Joint: Has one coordinate about the common Z-axis of parent and child joint frames.
3. Slider Joint: Has one coordinate along common X-axis of parent and child joint frames.
4. Ball Joint: Has three rotational coordinates that are about X, Y, Z of B in P.
5. Ellipsoid Joint: Has three rotational coordinates that are about X, Y, Z of B in P with coupled translations such that B traces an ellipsoid centered at P.
6. Free Joint: Contains six coordinates with three rotational (like the ball) and three translations of B in P.
7. Custom Joint: The user need to specified 1-6 coordinates and also define spatial transform to locate B with respect to P

3.36 KINEMATIC CONSTRAINTS

OpenSim supports three types of built-in constraints;

1. Point constraint: Which fixes a defined point with respect to two bodies giving no relative translation.

2. Weld constraint: This fixes relative location and orientation to two bodies with no translations or rotations.
3. Coordinate Coupler constraint: This is a coordinate's coupler which relates the generalized coordinates of a given joint.

3.37 FORCES

Forces are defined in the <ForceSet> section of the model file and it is the next step after the joints and in order to activate model a definition has to be given to the surrounding forces. Forces come in two varieties;

1. Passive (spring, dampers and contact) and
2. Active (springs, idealized linear or torque actuators and muscles)

The active forces require controls as its input. A user can either supply one or a controller called actuator is introduced which is a subset of the force-set. OpenSim has numerous built-in forces such as;

1. Prescribed Force
2. Spring Generalized Force
3. Bushing Force
4. Hunt Crossley Force
5. Elastic Foundation Force all used to model forces due to contact.

3.38 CONTROLLERS

A controller is simply a default constructor. It computes and sets the value of the controls for the actuators under its control and also adds computational components to the underlying multibody systems. In addition, it handles all of the bookkeeping of the system

indices (See Figure 7.4) and provides access to variable values via their names as strings. It also uses the model component to specify the interface that components must satisfy in order to be part of the system and provides a series of helper methods for adding variables (state, discrete, cache etc.) to the system.

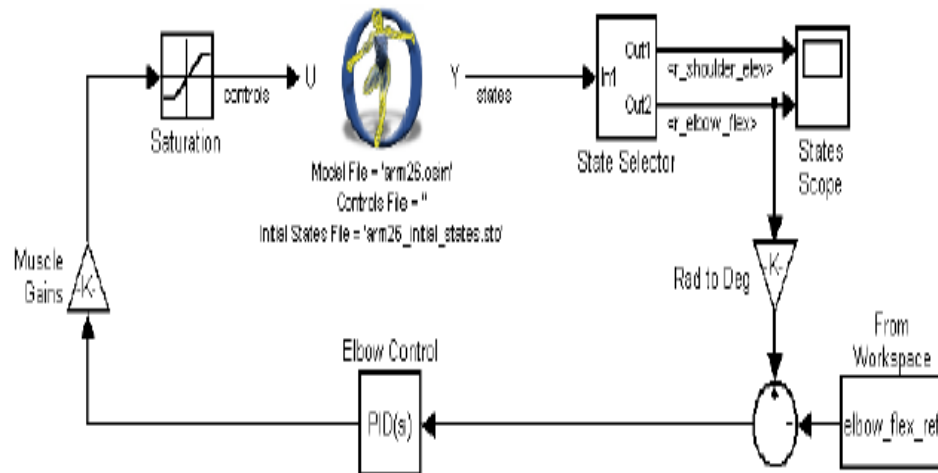


Figure 7.4 Shows a controller relationship for arm26 model (Jennifer Hicks, 2014).

3.4 TYPES OF MODEL

There is a set of pre- defined models within the OpenSim package. These are described below:

1. Deformable Femur Model: This includes a model of the femur that can be deformed to model bone deformities.



Figure 7.5a Shows picture of OpenSim femur model. Retrieved from, <http://simtk-confluence.stanford.edu:8080/display/OpenSim24/Musculoskeletal+Models>.

2. Full-body OpenSim Model: Creates a muscle-actuated simulation of running. The lower-extremity contains the muscles model and the upper-extremity has idealized torque actuators at each degree of freedom (no muscles).



Figure 7.5b Shows picture of a full-body OpenSim model.

Retrieved from, <http://simtkconfluence.stanford.edu:8080/display/OpenSim24/Musculoskeletal+Models>.

3. Head and Neck Musculoskeletal Biomechanics: This model provides neck musculoskeletal models and anatomical data, and a forum for discussion for biomechanics researchers.



Figure 7.5c Shows picture of OpenSim head and neck model.

Retrieved from, <http://simtk-confluence.stanford.edu:8080/display/OpenSim24/Musculoskeletal+Models>.

4. Arnold Lower Limb Model: This model provides an OpenSim lower-extremity model that uses the dataset from 21 cadaver experiment.



Figure 7.5d Shows picture of OpenSim lower limb model. Retrieved from, <http://simtk-confluence.stanford.edu:8080/display/OpenSim24/Musculoskeletal+Models>.

5. Lower Extremity Model: This project provides a lower-extremity model of one or both legs.



Figure 7.5e Shows picture of OpenSim lower extremity model. Retrieved from, <http://simtk-confluence.stanford.edu:8080/display/OpenSim24/>

Musculoskeletal+Models.

6. Lower Limb Model 2010: This model provides a new lower-extremity model with data from a larger set of cadavers that can estimate force and moment generating capacity of the lower limb and reveal relationships between muscle architecture and function.

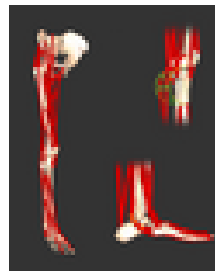


Figure 7.5f Shows picture of OpenSim lower limb model. Retrieved from, <http://simtk-confluence.stanford.edu:8080/display/OpenSim24/Musculoskeletal+Models>.

7. Musculoskeletal Model of the Lumbar Spine: This model includes a musculoskeletal model of the lumbar spine to be used for modeling lumbar spine kinematics.



Figure 7.5g Shows picture of OpenSim musculoskeletal lumbar spine model (<http://simtkconfluence.stanford.edu:8080/display/OpenSim24/Musculoskeletal+Model>).

8. Torso + Lower Extremity Model: This model provides an OpenSim model of lower-extremity and torso for simulating human movement.

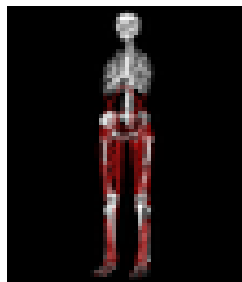


Figure 7.5h Shows picture of OpenSim torso and lower extremity model,

Retrieved from, <http://simtkconfluence.stanford.edu:8080/display/OpenSim24/Musculoskeletal+Model>

9. Upper and Lower Body Model: Provides a starting model for those interested in a full body model.

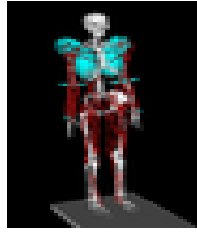


Figure 7.5i Shows picture of OpenSim upper and lower body model. Retrieved from, <http://simtkconfluence.stanford.edu:8080/display/OpenSim24/Musculoskeletal+Model>.

11. Upper Extremity Model: This project provides an upper-extremity model available for download along with accompanying publications.



Figure 7.5j Shows picture of Opensim upper extremity model. Retrieved from, <http://simtkconfluence.stanford.edu:8080/display/OpenSim24/Musculoskeletal+Model>.

12. Wrist Model: This provides a model of the wrist for a SIMM-based musculoskeletal model of the human wrist which can be imported and used in OpenSim.



Figure 7.5k Shows picture of OpenSim wrist model. Retrieved from,
<http://simtkconfluence.stanford.edu:8080/display/OpenSim24/Musculoskeletal+Model>.

3.5 MODEL EDITING

This is more of a generic property editor for objects that makes up an OpenSim model such as bodies, joints, muscles, etc. It is categorized as follows:

1. Property Editor: Allows for opening an object and changing properties.
2. Marker Editor: Used to scale a generic model to fit a particular subject and also used to place landmarks on models, change names of makers, their offsets and bodies they are attached.
3. Muscle Editor: Allows access to models and all parameters of the muscles and other actuators in the model.
4. Path Editor: Path of a muscle is the set of straight lines connecting each pair of adjacent points called fixed points. However, there are three different attachment muscle points ;

1. Via point: These are attachment points that are fixed to a body, but used in the muscle path only when a specified coordinate is in a particular range. Used typically for wrapping such as the knee flexing beyond a certain angle.
 2. Moving muscle point: As the name implies, they are points where X, Y and/or Z offsets in a body's reference frame as a function of coordinates rather than simple constant. It is useful in cases where a muscle path is needed to move as a joint flexes.
 3. Wrap Point: this is a case where OpenSim automatically calculate XYZ offsets in order to wrap a muscle over the surface of a wrap objects such as ellipsoids, spheres, cylinders and tori, which can be used to constrain the paths of the muscle.
- 5 Function Editor: Enables the user to view and modify parameters of a model that are functions such as force length curve of a muscle or a joint constraint function. One can add, delete and move control points and function types.
- 6 Excitation Editor: It is a visual inspection and editor for muscle excitation patterns, such as the inputs for a forward dynamic simulation or outputs of a control algorithm that solves muscles excitations.

CHAPTER FOUR

4.0 DYNAMIC UPPER EXTRIMITY

Imagine the muscle coordination responsible to hold a document in the hand or navigate through the use of computer mouse and keyboard. It is required for the human movement to function with the coordination of many muscles. The transformation between neural control and purposeful movement is highly complex and involves many individual elements such as the musculoskeletal geometry, muscle-tendon dynamics and multibody dynamics.

The upper extremity of the human body includes the shoulder, arm, forearm, wrist, and hand. The modelling and simulation of individual elements of the upper extremity provides estimates of important variables, such as muscles and joint forces. The simulation enables to establish cause and effect relationship and gives insight into muscle function and as well perform “what-if” studies to test hypotheses, predict functional outcomes and identify emergent behaviors. This variable also illustrates how the muscle moment arm, optimal muscle fibre length and tendon slack length influence the variation of muscle force with respect to in-joint angle. See Figure 8 and 9 for detailed graphic example of the upper-limb geometry and muscle paths

On the OpenSim platform, the closest model to the upper extremity is the “arm26” and the “Wrist model”. OpenSim allows for variation with the bodies, joints (constraints and contact geometry), forces (Muscles, actuators, contact forces, other forces), markers (controllers) and probes.

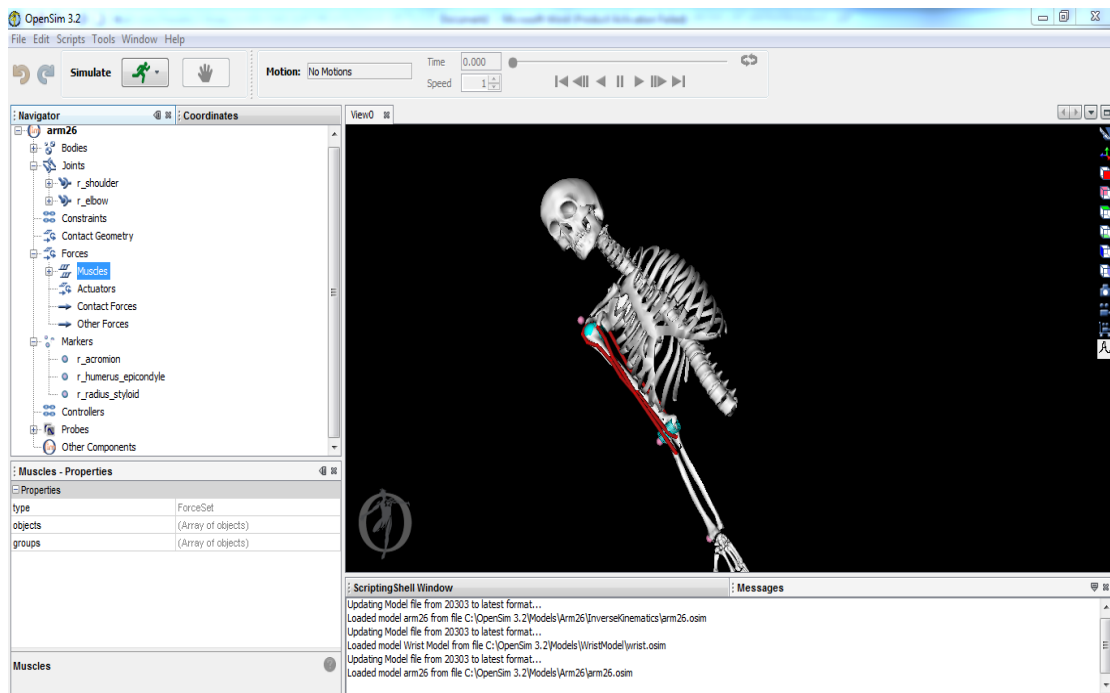


Figure 8. Figure shows the OpenSim platform for the “Arm26” upper extremity variables (Snip-shot from OpenSim software).

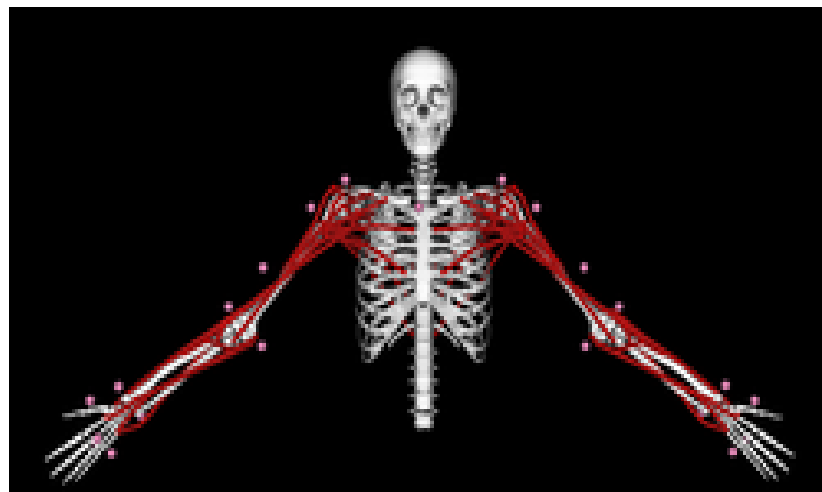


Figure 9. Figure Shows the upper-limb geometry and muscle paths

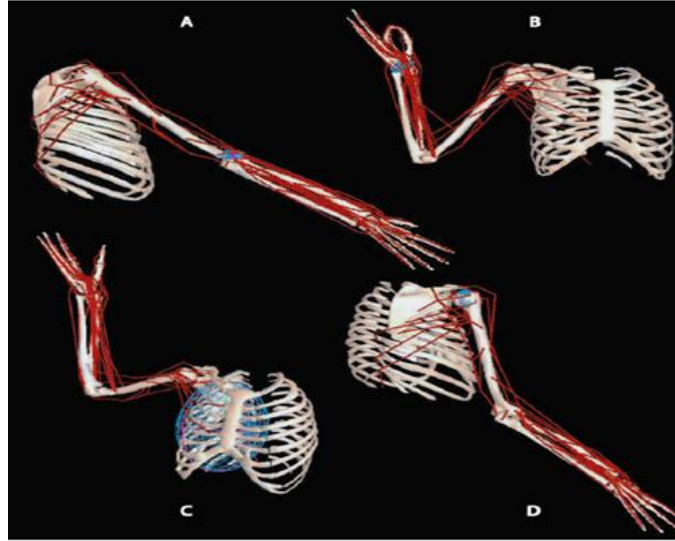


Figure 10. Shows; (a) Upper extremity lateral view with arm in 45 degrees of flexion (90 degrees elevation plane). (b) Upper extremity anterior view with arm in 45 degrees of abduction (zero degree elevation plane), (c) Upper extremity anteromedial view and (d) Upper extremity posterolateral view, demonstrating the degree of freedom, muscle paths and selected wrapping surfaces (blue ellipses) (Katherine R.S et al., 2005).

The upper extremity model is developed considering the general purpose musculoskeletal model which includes graphical representation of the ribs, sternum, clavicle, scapula, humerus, radius, ulna and bones of the wrist and hand as shown in Figure 10. However, the Upper extremity uses the inverse kinematics, inverse dynamics and static optimization to visualize complex movement patterns, probe parameters that are difficult to measure, perform “what-if” studies and identify cause-effect relationships. The overall movement of the joint in the shoulder is determined by the collective motion of the shoulder girdle (clavicle, scapula and humerus) using spherical coordinates. And in relation the wrist rotation is dominated by stiffness (Charles S. k. et al., 2011).

CHAPTER FIVE

5.0 MODEL MOTION, SIMULATION AND ANALYSIS

5.1 MUSCLE ACTIONS

The purpose of this chapter is to develop an understanding and test the upper extremity musculoskeletal system using a computer model. The forward dynamics simulations help understand muscle actions. To implement some of the tools and capabilities of OpenSim, the MoBL ARMS dynamic upper limb model which is the dynamic musculoskeletal model of the human upper extremity is used in this study.

The MoBL ARMS dynamic upper limb model uses the OpenSim 3.1 64 bit release effectively and is likely not to work on any other OpenSim update released after 3.1. The model consists of the torso, upper arm, forearm segment along with the triceps brachia (Long, medial and lateral heads), biceps brachii (long and short heads), and many associated muscles like the deltoid, subscapularis, pectoralis major, brachialis, extensor carpi, flexor carpi and digitorum, palmaris longus, extensor and abductor pollicis muscles. A very similar model to the MoBL ARMS model described in the OpenSim is the Arm 26 model which is shown in Figure 8. It basically shows the two coordinates and corresponding model markers involved in the arm structure. However, the coordinate is simply a joint angle or distance which specifies the relative orientation or location of two body skeletal segments.

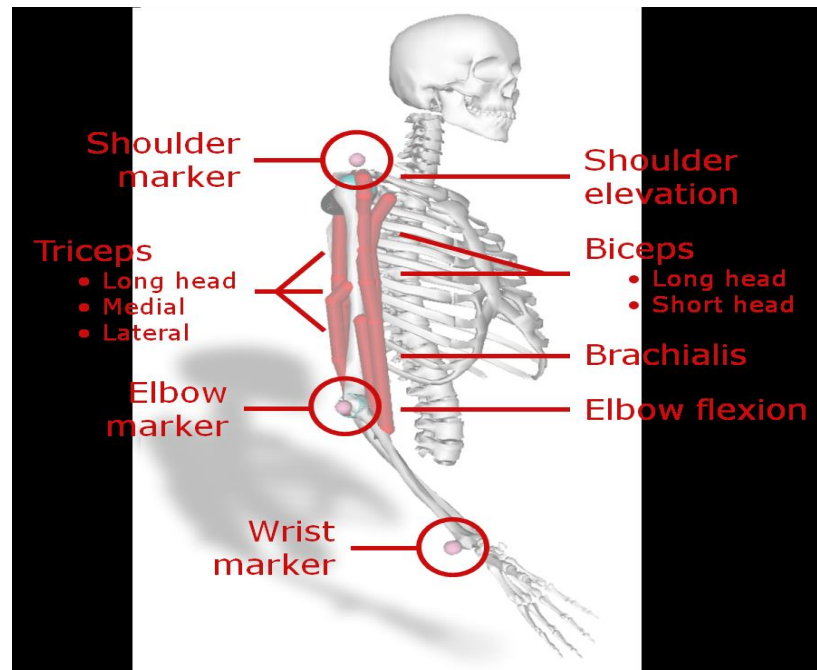


Figure 11. Shows coordinates and Marker (Jeff S. et al., 2011).

5.2 MOTION KINEMATICS

Figure 11 describes the graphical representation of the ribs, sternum, clavicle, scapula, humerus, radius, ulna, bones of the wrist and hand. The general motion of the shoulder is determined by the collective motion of the shoulder girdle (clavicle, scapula and humerus) while the articulation between the humerus and scapula is modelled as a ball and socket, and the movement of the shoulder girdle is simplified to vary only with thoracohumeral (shoulder elevation) (Katherine R.S et al., 2005). Although, the OpenSim software reflects the body tools into the bodies, joints, constraints, forces, and markers (Figure 11) and this helps user to view and edit any paths not needed.

However, the MoBL model represents the right shoulder and arm, and contains the following segments: thorax, which is fixed, clavicle, scapula, humerus, ulna, radius, and hand (attached to the radius). It has eleven degrees of freedom with three at the sternoclavicular joint (clavicle protraction/retraction, elevation/depression, axial rotation),

three at the acromioclavicular joint (scapula protraction/retraction, lateral/medial rotation, tilt), three at the glenohumeral joint (humerus plane of elevation, elevation angle, axle rotation), elbow flexion/extension and forearm pronation/supination (Ed Chadwick et al., 2014).

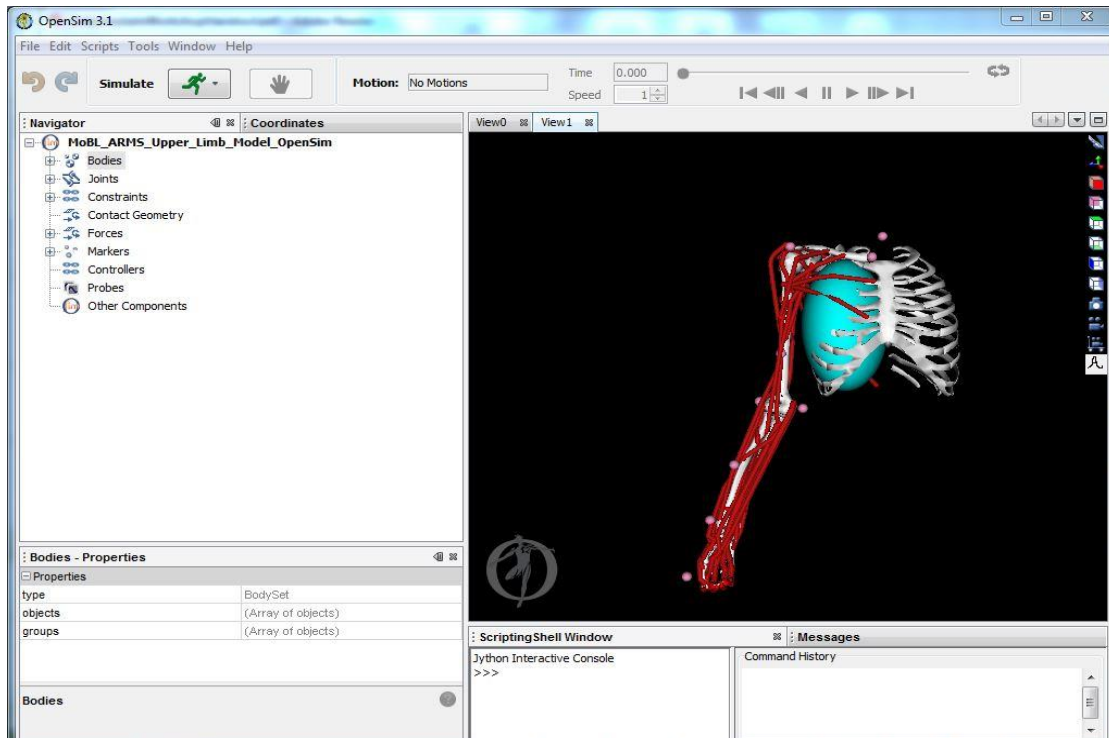


Figure 12. An OpenSim GUI platform showing MoBL ARMS upper model and list of body path modification tools (Snip-shot from OpenSim software)

It is necessary to understand the motion types described in the muscle anatomy of the upper extremity arm (Chapter four). The elevation plane degree of freedom describes the orientation of the vertical plane in which the arm elevates to the frontal plane, the 0° elevation plane is the plane of the shoulder abduction and the shoulder flexion which occurs in the 90° elevation plane and also the elbow flexion is defined from 0° (full extension) to 130° (flexion) and forearm rotation is defined from 90° (pronation) to -90° (supination). The neutral position for the forearm rotation is 0° and is defined when the hand is in the sagittal

plane when the shoulder and wrist are in neutral positions. Wrist flexion motion range from -70° (extension) to 70° (flexion) while deviation ranges from -10° (radial) and 25° (ulnar) (Katherine R.S et al., 2005).

5.3 BUILDING AND LOADING MoBL ARMS MODEL

MoBL Arms dynamic upper limb model has three categories of OpenSim computation layer which includes;

1. Individual elements (Mechanical properties and geometry).
2. Connections (Joint and constraints).
3. Interactions (Internal and external forces).

MoBL Arms is located on OpenSim in the link <https://simtk.org/home/upexdyn>.

It comes in a zip file which can be stored in OpenSim3.1>OpenSim model.osim file. The saved model is titled MoBL_ARMS_Upper_arm_model_OpenSim.osim.

To load this musculoskeletal model into OpenSim:

1. Open the OpenSim software on your computer system (OpenSim 3.1 specifically).
2. Click the file menu and select Open model.
3. Find the model folder, which is located under the OpenSim simulation directory
e.g. C:/OpenSim 3.1.
4. Open the MoBL_ARMS_Upper_arm_model_OpenSim.osim and click open.

After loading the name will appear in the model drop down menu, located in the toolbar, and the navigation window also comes up which shows specific information about the bodies, muscles and joints in the model.

5. Click the window menu and select Navigator and to expand the heading click on the plus icon to its left and likewise is the coordinates, messages, add view, properties and scripting shell.

5.4 VIEWING AND ANIMATING MODEL

Most of the movement orientations can be achieved with the use of the mouse. Along the side of the view window, there are six orienting icons basically for viewing along principal directions. The major directions are as following:

1. Rotate: to rotate the view, click and hold the left mouse button and drag the mouse.
2. Zoom: To zoom, click and hold right mouse button and to zoom in, drag the mouse down and mouse up for zoom out.
3. Translate: To translate the view, click and hold the center mouse button and drag the mouse.

5.5 COORDINATES

To see the coordinate window, click the window menu and select coordinates, this corresponds to joint coordinates in the model.

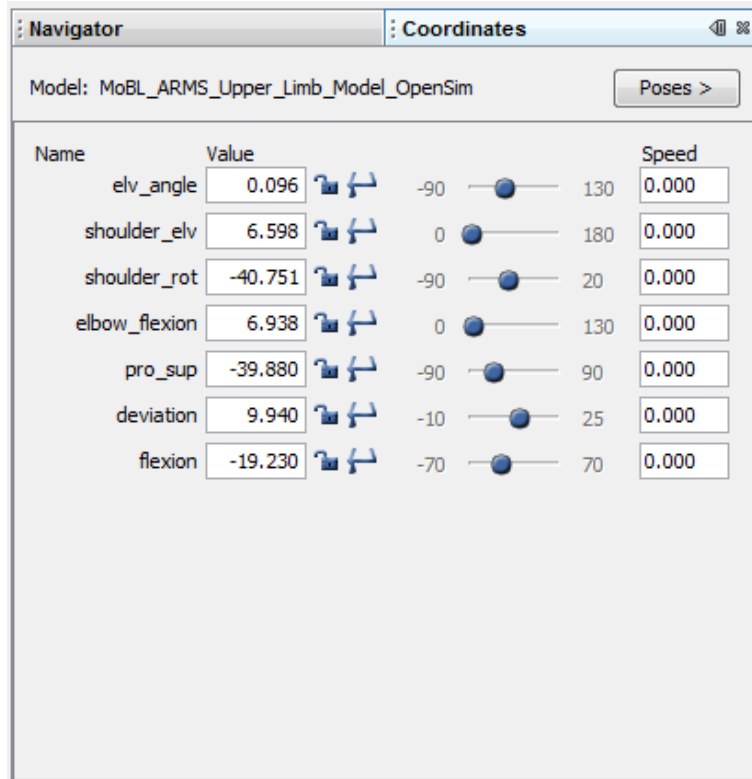


Figure 13. Shows the coordinate menu for OpenSim MoBL model and its features (Snip-shot from OpenSim software).

The coordinate is the model component for managing the access and behaviors of a model's generalized coordinate including its value, speed and acceleration (once the system acceleration have been realized) and the model component enables a coordinate to be locked, prescribed or clamped (limited to a min-to max range). Figure 13 shows the tools which can be described as follows:

1. The first three slider show the rotations of the arm about the X, Y and Z axis of the "shoulder" reference frame. To do this drag the elv angle or shoulder_elv or shoulder_rot tilt slider to the corresponding direction. This rotation is different from rotating the model view.

2. The remaining sliders have to do with joint rotation and control a single degree of freedom. To do this, drag the sliders or type in a desired joint angle in the adjacent textbox.
3. To save a pose or a specific set of joint coordinates:
 1. Restore default joint coordinates by clicking the pose button and selecting default.
 2. If you have to save changes, click the pose button, select new, type the reference change name in the textbox, and click ok. The saved pose will appear under default when one clicks poses.

5.6 LOADING A MOTION

To animate the model, an associated motion file (file type: .mot or .sto) will be loaded into OpenSim by using the file menu as described below

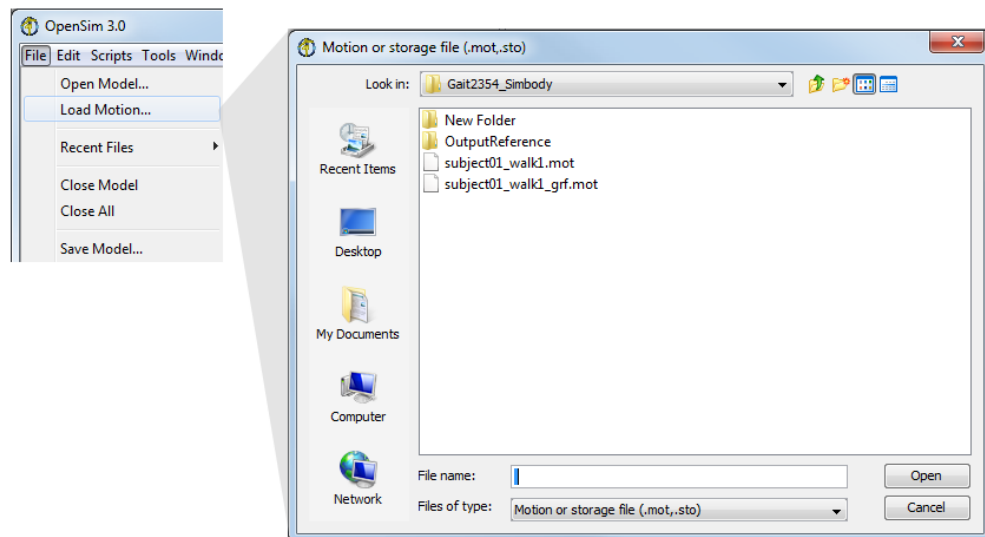


Figure 14. Shows the file menu and the load motion (file type .mot or .sto) (Snip-shot from OpenSim software).

Three options can be used to achieve loading of a motion:

1. Select file > load motion from OpenSim main menu bar. In the window that appears, locate and select the desired motion file (.mot or .ost) and then click open as seen in Figure 15 with motion> static poses. After loading the name of the motion will appear in the motion textbox located on the toolbar
2. Load motions by dragging a motion file from the windows explorer and dropping it into the application, in the 3Dview panel.
3. Load a motion by right clicking on the current model in navigator window and choosing Load Motion from the drop down menu that appears.

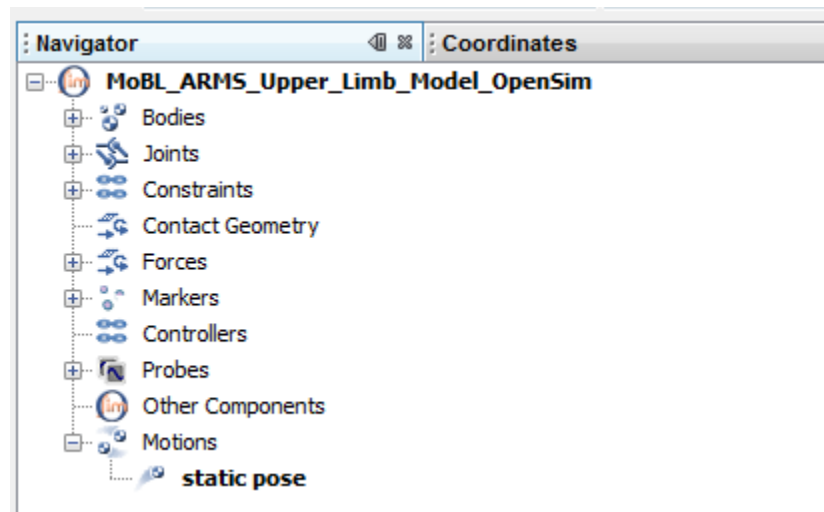


Figure 15. Shows the loaded motion static pose (Snip-shot from OpenSim software).

5.7 MOTION SLIDER

The motion slider corresponds to the current motion file. To execute a particular motion of the model, drag the motion slider depending on the direction of intended motion.

To animate the model, the video control button can help bring a play back to the motion. Also the loop and pause button help control speed of animation.

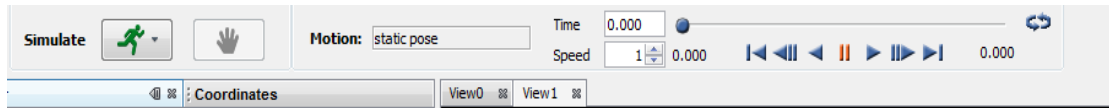


Figure 16. Shows the loop play and pause motion control (Snip-shot from OpenSim software).

5.8 RECORDING A MOVIE

It begins with a click on the movie camera button in the 3D View Toolbar. Enter a file name for the movie to be recorded and click the Open button which will cause recording to be toggled "on." Whatever is displayed in the 3D View window will then be recorded. So any changes made to the camera view or to the model, for example, changing coordinates or playing back a simulation through the play or reverse play buttons in the video controls, will be saved as a movie to the specified file. The movie camera button will be highlighted while recording and to stop recording and saving out the movie file, press the movie camera button again.

5.9 PLOTTING IN OPENSIM

OpenSim plotter allows for the plotting of muscle-tendon properties, length, moment arm, force and joint moment were the SI unit: Meters (length), Newton (force), Newton-meters (moment) are the units used and in achieving a plot the following steps need to be followed:

To open a new plot, click the tool menu and select plot. In the plotter window, click the Y-Quality button and select options ranging from:

- 1 Moment with sub section- (Elv_angle, Shoulder_elv, Shoulder_rot, Elbow_flexion, Pro_sup, Deviation, Flexion).
- 2 Moment arm with sub section- (Elv_angle, Shoulder_elv, Shoulder_rot, Elbow_flexion, Pro_sup, Deviation, Flexion).
- 3 Muscle-tendon length
- 4 Tendon-length
- 5 Normalized fiber-length
- 6 Tendon force
- 7 Active fiber-force
- 8 Passive fiber-force
- 9 Total fiber force

The moment and moment arm having options of elv_angle, shoulder_elv, shoulder_rot, elbow_flexion, pro_sup, deviation, flexion. After the Y-quantity is the muscle selection, which comprises of 50 different muscles used in the MoBL ARM model (Figure 18, 19 and 20), while the X-Quantity follow with the given options as follows:

1. Elv_angle
2. Shoulder_elv
3. Shoulder_rot
4. Elbow_flexion
5. Pro_sup

6. Deviation

7. Flexion

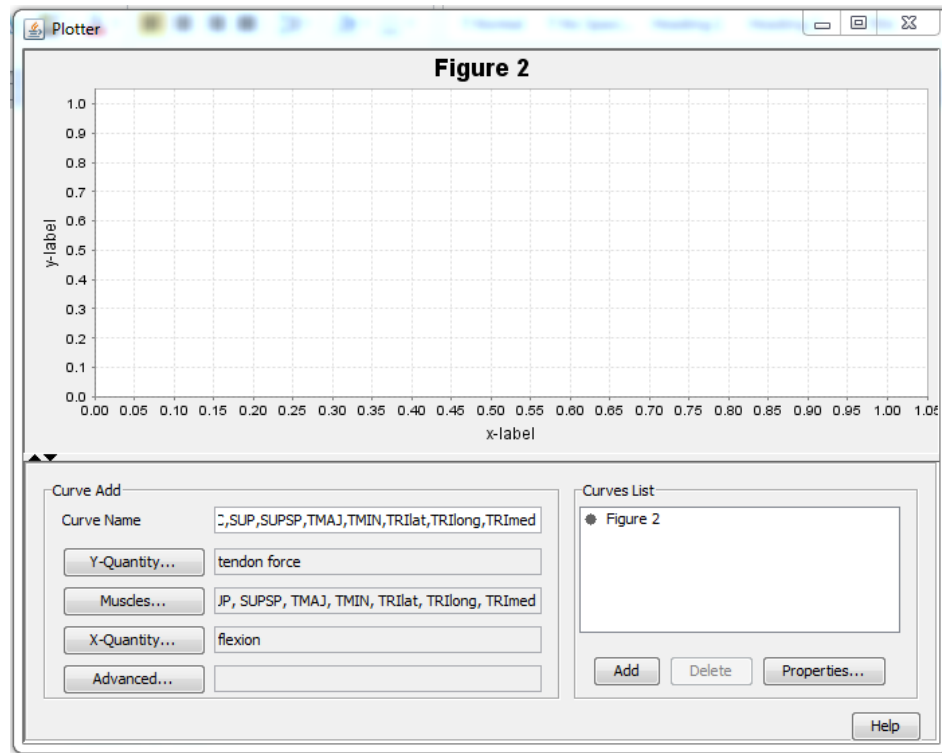


Figure 17. Shows motion plot layout and features (Snip-shot from OpenSim software).

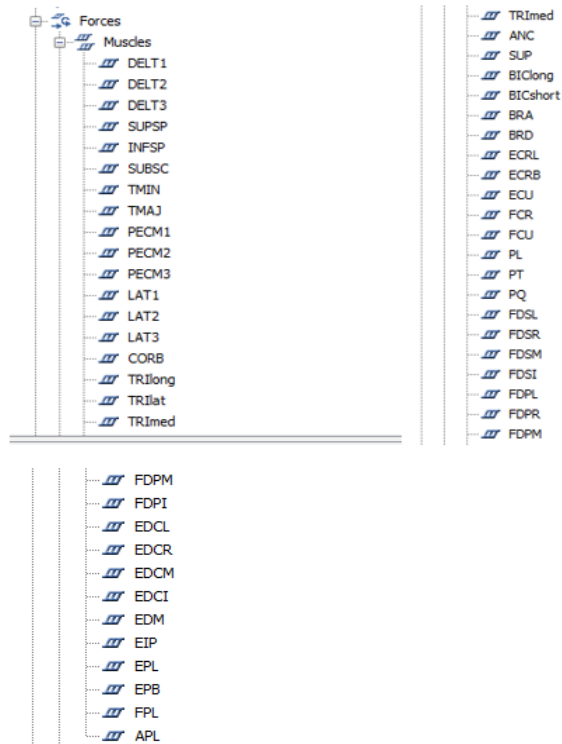


Figure 18. Muscles selection: 50 muscles can be select for a plot. (Snip-shot from OpenSim software).

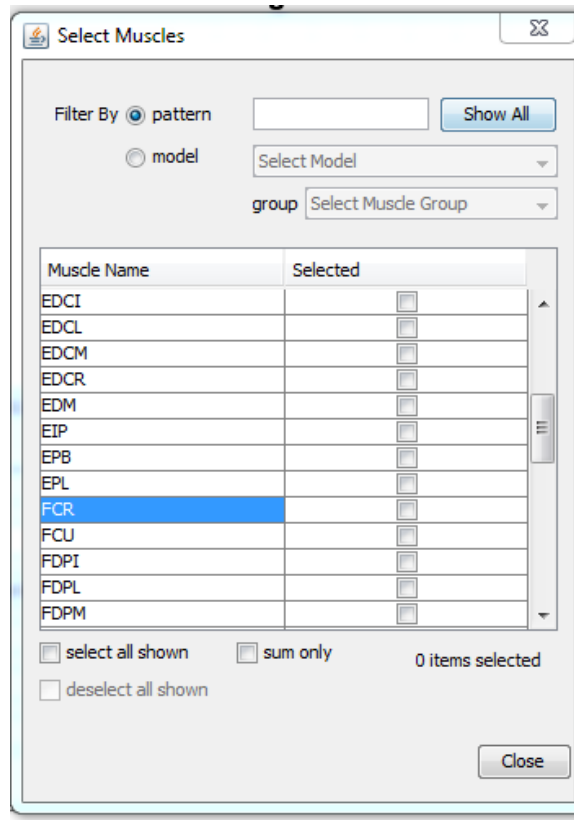


Figure 19. Muscle selection toolbox. (Snip-shot from OpenSim software).

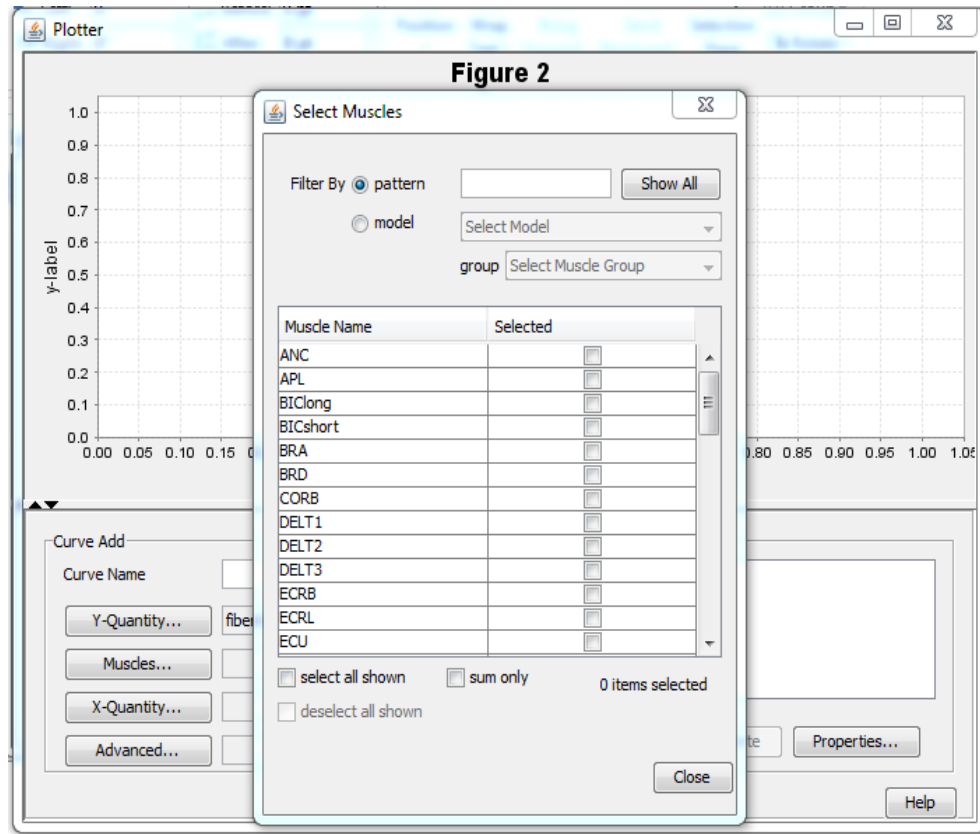


Figure 20. GUI representation of the muscle selection (Snip-shot from OpenSim software).

After selecting an appropriate Y-Quantity, select associated muscles to generate curves and use the group drop down menu which shows only the list of muscles selected. One does not have to close the muscle window even though the plotter window is opened and selections are immediately updated in the muscles textbox. This helps create multiple curves on the same plot. Finally selecting the X-Quantity and the variable will appear on the x-axis as shown in Figure 17. Other features are itemized as follows:

1. To add title to the plot, click on the properties button, type the Y-quantity vs X-quantity into the textbox under the title tab and explore the plot properties, then click ok.
2. To add curve to plot, click add and the changes are reflected on the plot window.

CHAPTER SIX

6.0 SIMULATION AND TESTING

6.1 MODEL SPECIFICS

OpenSim computer simulations play an increasingly important role in solving complex engineering problems and have possibilities to transform experimentally based medical decision-making and treatment design. The simulation provides resources for cause and effect relationship, giving insights into muscles and joint forces. Some quantities can be measured to understand movement dynamics such as muscle activities and reaction forces responsible for movement. OpenSim complements these measurements with estimates of other important variables like ‘what if’ study to test a hypothesis, predict functional outcomes and identify emergent behaviors.

Imagine the muscle coordination necessary to hold a cup in the hand from a table. In order to navigate through it using a computer mouse and a keyboard to carry the cup and get a drink from the cup and replace it back. It requires the coordination of many muscles to achieve this and the transformation between neural control and purposeful movement is highly complex and involves many individual elements such as musculoskeletal geometry, muscle-tendon dynamics and multibody dynamics. These elements make available important but limited understanding of movement dynamics. (Robertson G. et al., 2004).

The upper extremity model is developed using a general purpose musculoskeletal modeling package called OpenSim. The model includes graphical representation of the ribs, sternum, clavicle, humerus, radius, ulna, and bones of wrist and hand. The model consist of five (5) degree of freedom (DOF) and 23 muscles crossing the shoulder, elbow and forearm. The DOF defines the kinematics of the shoulder, elbow and wrist, (Katherine R. S. H. et al., 2005). The shoulder holds its DOF at elevation plane, thoracohumeral angle (also indicated as elevation angle) and shoulder rotation, while the other degrees of freedom are the elbow flexion and forearm rotation. However, other DOF exist but are not as important to this project. Carpometacarpal abduction and flexion, Metacarpophalangeal flexion and Interphalangeal flexion are the other DOF (Katherine R. S. H. et al., 2005).

6.2 MUSCLE ACTIVATION USING OPENSIM

The muscle base of the “MoBL ARMS upper arm model OpenSim” has a number of pure virtual functions. In order to define a property for the rate at which active muscles moment fatigue and recovers starting from a default value of 0.0 and assumed to range 0.0 degrees to 130 degrees and are normalized.

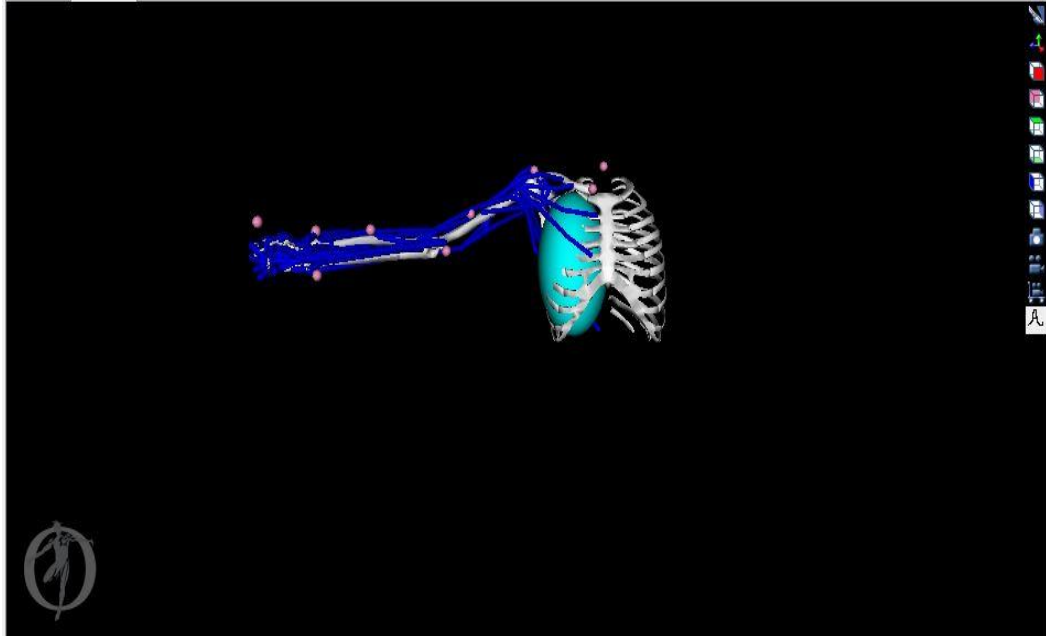


Figure 21. Shows MoBL ARMS Upper arm model OpenSim active muscles (in blue) in an initial default state of 0 degrees. (Snip-shot from OpenSim software).

The scripting for the active state is stated below;

```
_fatigueFactorProp.setName("fatigue_factor");
```

```
_fatigueFactorProp.setValue(0.0);
```

```
_fatigueFactorProp.setComment("percentage of active muscle units that fatigue in unit  
time");
```

```
_propertySet.append(&_fatigueFactorProp, "Parameters");
```

```
_recoveryFactorProp.setName("recovery_factor");
```

```
_recoveryFactorProp.setValue(0.0);
```

```
_recoveryFactorProp.setComment("percentage of fatigued motor units that recover in unit  
time");
```

```
_propertySet.append(&_recoveryFactorProp, "Parameters");
```

Figure 21 describes how the functions equilibrate by computing muscle states assuming the muscle is in an equilibrium state. It initializes the states to reasonable value, releases the Simbody model to the velocity stage and then calls for activation of each muscles and compute isometric force before it begins a dynamic simulation.

6.3 SIMULATION

A run script of one of the simulations is shown below and the time range for the forward simulation is set to from 0 to 0.1 secs;

Running tool MoBL_ARMS_Upper_Limb_Model_OpenSim.

No external loads will be applied (external loads file not specified).

MODEL: MoBL_ARMS_Upper_Limb_Model_OpenSim

ANALYSES (3)

analysis[0] = MuscleAnalysis

analysis[1] = Un-named analysis.

analysis[2] = Un-named analysis.

BODIES (14)

body[0] = ground (mass: 0) (inertia: 1 0 0 0 1 0 0 0 1)

body[1] = clavicle (mass: 0.156) (inertia: 0.00024259 -1.898e-005 -6.994e-005 -1.898e-005 0.00025526 5.371e-005 -6.994e-005 5.371e-005 4.442e-005)

body[2] = clavphant (mass: 0.0001) (inertia: 1 0 0 0 1 0 0 0 1)

body[3] = scapula (mass: 0.70396) (inertia: 0.0012429 0.0004494 0.00040922 0.0004494 0.0011504 0.0002411 0.00040922 0.0002411 0.0013651)

body[4] = scapphant (mass: 0.0001) (inertia: 1 0 0 0 1 0 0 0 1)

body[5] = humphant (mass: 0.0001) (inertia: 1 0 0 0 1 0 0 0 1)

body[6] = humphant1 (mass: 0.0001) (inertia: 1 0 0 0 1 0 0 0 1)

body[7] = humerus (mass: 1.99757) (inertia: 0.0122776 -0.00034741 -0.0002325 -0.00034741 0.00255133 0.0012293 -0.0002325 0.0012293 0.0125789)

body[8] = DELT2pt2a (mass: 0.0001) (inertia: 1e-005 0 0 0 1e-005 0 0 0 1e-005)

body[9] = ulna (mass: 1.1053) (inertia: 0.00541309 0.00031686 -7.615e-005 0.00031686 0.00115318 0.00109169 -7.615e-005 0.00109169 0.00494361)

body[10] = radius (mass: 0.23359) (inertia: 0.00043855 3.014e-005 -4.24e-006 3.014e-005 8.859e-005 6.418e-005 -4.24e-006 6.418e-005 0.00040258)

body[11] = BICpt (mass: 0.0001) (inertia: 1e-005 0 0 0 1e-005 0 0 0 1e-005)

body[12] = proximal_row (mass: 0.0001) (inertia: 1e-005 0 0 0 1e-005 0 0 0 1e-005)

body[13] = hand (mass: 0.5819) (inertia: 0.00011 9e-007 -2e-007 9e-007 6e-005 1.2e-005 -2e-007 1.2e-005 0.00015)

ACTUATORS (50)

actuator[0] = DELT1

actuator[1] = DELT2

actuator[2] = DELT3

actuator[3] = SUPSP

actuator[4] = INFSP

actuator[5] = SUBSC

actuator[6] = TMIN

actuator[7] = TMAJ

actuator[8] = PECM1

actuator[9] = PECM2

actuator[10] = PECM3

actuator[11] = LAT1

actuator[12] = LAT2

actuator[13] = LAT3

actuator[14] = CORB

actuator[15] = TRIlong

actuator[16] = TRllat

actuator[17] = TRImed

actuator[18] = ANC

actuator[19] = SUP

actuator[20] = BIClong

actuator[21] = BICshort

actuator[22] = BRA

actuator[23] = BRD

actuator[24] = ECRL

actuator[25] = ECRB

actuator[26] = ECU

actuator[27] = FCR

actuator[28] = FCU

actuator[29] = PL

actuator[30] = PT

actuator[31] = PQ

actuator[32] = FDSL

actuator[33] = FDSR

actuator[34] = FDSM

actuator[35] = FDSI

actuator[36] = FDPL

actuator[37] = FDPR

actuator[38] = FDPM

actuator[39] = FDPI

actuator[40] = EDCL

actuator[41] = EDCR

actuator[42] = EDCM

actuator[43] = EDCI

actuator[44] = EDM

actuator[45] = EIP

actuator[46] = EPL

actuator[47] = EPB

actuator[48] = FPL

actuator[49] = APL

numStates = 152

numCoordinates = 26

numSpeeds = 26

numActuators = 50

numBodies = 14

numConstraints = 19

numProbes = 0

STATES (152)

y[0] = sternoclavicular_r2

y[1] = sternoclavicular_r3

y[2] = unrotscap_r3

y[3] = unrotscap_r2

y[4] = acromioclavicular_r2

y[5] = acromioclavicular_r3

y[6] = acromioclavicular_r1

y[7] = unrothum_r1

y[8] = unrothum_r3

y[9] = unrothum_r2

y[10] = elv_angle

y[11] = shoulder_elv

y[12] = shoulder1_r2

y[13] = shoulder_rot

y[14] = con_DELT2tx

y[15] = con_DELT2ty

y[16] = con_DELT2tz

y[17] = elbow_flexion

y[18] = pro_sup

y[19] = con_BICtx

y[20] = con_BICty

y[21] = con_BICtz

y[22] = deviation

y[23] = flexion

y[24] = wrist_hand_r1

y[25] = wrist_hand_r3

y[26] = sternoclavicular_r2_u

y[27] = sternoclavicular_r3_u

y[28] = unrotscap_r3_u

y[29] = unrotscap_r2_u

y[30] = acromioclavicular_r2_u

y[31] = acromioclavicular_r3_u

y[32] = acromioclavicular_r1_u

y[33] = unrothum_r1_u

y[34] = unrothum_r3_u

y[35] = unrothum_r2_u

y[36] = elv_angle_u

y[37] = shoulder_elv_u

y[38] = shoulder1_r2_u

y[39] = shoulder_rot_u

y[40] = con_DELT2tx_u

y[41] = con_DELT2ty_u

y[42] = con_DELT2tz_u

y[43] = elbow_flexion_u

y[44] = pro_sup_u

y[45] = con_BICtx_u

y[46] = con_BICty_u

y[47] = con_BICtz_u

y[48] = deviation_u

y[49] = flexion_u

y[50] = wrist_hand_r1_u

y[51] = wrist_hand_r3_u

y[52] = DELT1.activation

y[53] = DELT1.fiber_length

y[54] = DELT2.activation

y[55] = DELT2.fiber_length

y[56] = DELT3.activation

y[57] = DELT3.fiber_length

y[58] = SUPSP.activation

y[59] = SUPSP.fiber_length

y[60] = INFSP.activation

y[61] = INFSP.fiber_length

y[62] = SUBSC.activation

y[63] = SUBSC.fiber_length

y[64] = TMIN.activation

y[65] = TMIN.fiber_length

y[66] = TMAJ.activation

y[67] = TMAJ.fiber_length

y[68] = PECM1.activation

y[69] = PECM1.fiber_length

y[70] = PECM2.activation

y[71] = PECM2.fiber_length

y[72] = PECM3.activation

y[73] = PECM3.fiber_length

y[74] = LAT1.activation

y[75] = LAT1.fiber_length

y[76] = LAT2.activation

y[77] = LAT2.fiber_length

y[78] = LAT3.activation

y[79] = LAT3.fiber_length

y[80] = CORB.activation

y[81] = CORB.fiber_length

y[82] = TRIlong.activation

y[83] = TRIlong.fiber_length

y[84] = TRIlat.activation

y[85] = TRIlat.fiber_length

y[86] = TRImed.activation

y[87] = TRImed.fiber_length

y[88] = ANC.activation

y[89] = ANC.fiber_length

y[90] = SUP.activation

y[91] = SUP.fiber_length

y[92] = BIClong.activation

y[93] = BIClong.fiber_length

y[94] = BICshort.activation

y[95] = BICshort.fiber_length

y[96] = BRA.activation

y[97] = BRA.fiber_length

y[98] = BRD.activation

y[99] = BRD.fiber_length

y[100] = ECRL.activation

y[101] = ECRL.fiber_length

y[102] = ECRB.activation

y[103] = ECRB.fiber_length

y[104] = ECU.activation

y[105] = ECU.fiber_length

y[106] = FCR.activation

y[107] = FCR.fiber_length

y[108] = FCU.activation

y[109] = FCU.fiber_length

y[110] = PL.activation

y[111] = PL.fiber_length

y[112] = PT.activation

y[113] = PT.fiber_length

y[114] = PQ.activation

y[115] = PQ.fiber_length

y[116] = FDSL.activation

y[117] = FDSL.fiber_length

y[118] = FDSR.activation

y[119] = FDSR.fiber_length

y[120] = FDSM.activation

y[121] = FDSM.fiber_length

y[122] = FDSI.activation

y[123] = FDSI.fiber_length

y[124] = FDPL.activation

y[125] = FDPL.fiber_length

y[126] = FDPR.activation

y[127] = FDPR.fiber_length

y[128] = FDPM.activation

y[129] = FDPM.fiber_length

y[130] = FDPI.activation

y[131] = FDPI.fiber_length

y[132] = EDCL.activation

y[133] = EDCL.fiber_length

y[134] = EDCR.activation

y[135] = EDCR.fiber_length

y[136] = EDCM.activation

y[137] = EDCM.fiber_length

y[138] = EDCI.activation

y[139] = EDCI.fiber_length

y[140] = EDM.activation

y[141] = EDM.fiber_length

y[142] = EIP.activation

y[143] = EIP.fiber_length

y[144] = EPL.activation

y[145] = EPL.fiber_length

y[146] = EPB.activation

y[147] = EPB.fiber_length

y[148] = FPL.activation

y[149] = FPL.fiber_length

y[150] = APL.activation

y[151] = APL.fiber_length

Integrating from 0 to 0.1;

Schutte1993Muscle_Deprecated::computeForce, muscle LAT2 force < 0 at time = 0.05

Schutte1993Muscle_Deprecated::computeForce, muscle SUP force < 0 at time = 0.05

Schutte1993Muscle_Deprecated::computeForce, muscle LAT2 force < 0 at time = 0.05

Schutte1993Muscle_Deprecated::computeForce, muscle EPB force < 0 at time =
0.0194975

Schutte1993Muscle_Deprecated::computeForce, muscle DELT2 force < 0 at time =
0.0199367

Schutte1993Muscle_Deprecated::computeForce, muscle TRIlat force < 0 at time = 0.0198879

Schutte1993Muscle_Deprecated::computeForce, muscle TRImed force < 0 at time = 0.0198879

Schutte1993Muscle_Deprecated::computeForce, muscle DELT2 force < 0 at time = 0.0203174

Schutte1993Muscle_Deprecated::computeForce, muscle CORB force < 0 at time = 0.0229717

Schutte1993Muscle_Deprecated::computeForce, muscle TRIlong force < 0 at time = 0.0229717

Schutte1993Muscle_Deprecated::computeForce, muscle TRIlat force < 0 at time = 0.0229717

Schutte1993Muscle_Deprecated::computeForce, muscle TRImed force < 0 at time = 0.0229717

Schutte1993Muscle_Deprecated::computeForce, muscle DELT2 force < 0 at time = 0.0233435

Schutte1993Muscle_Deprecated::computeForce, muscle FDPR force < 0 at time = 0.05

Schutte1993Muscle_Deprecated::computeForce, muscle LAT1 force < 0 at time = 0.05

Schutte1993Muscle_Deprecated::computeForce, muscle EPB force < 0 at time = 0.05

Schutte1993Muscle_Deprecated::computeForce, muscle EPB force < 0 at time = 0.0415332

Schutte1993Muscle_Deprecated::computeForce, muscle DELT2 force < 0 at time = 0.0419859

Schutte1993Muscle_Deprecated::computeForce, muscle TMAJ force < 0 at time = 0.05

Schutte1993Muscle_Deprecated::computeForce, muscle LAT2 force < 0 at time = 0.05

Schutte1993Muscle_Deprecated::computeForce, muscle EDCM force < 0 at time = 0.0473898

Schutte1993Muscle_Deprecated::computeForce, muscle DELT2 force < 0 at time = 0.04792

Schutte1993Muscle_Deprecated::computeForce, muscle LAT2 force < 0 at time = 0.0561219

Schutte1993Muscle_Deprecated::computeForce, muscle TRIl at force < 0 at time = 0.0561219

Schutte1993Muscle_Deprecated::computeForce, muscle TMAJ force < 0 at time = 0.05

Schutte1993Muscle_Deprecated::computeForce, muscle LAT2 force < 0 at time = 0.0677118

Schutte1993Muscle_Deprecated::computeForce, muscle DELT2 force < 0 at time = 0.068278

Schutte1993Muscle_Deprecated::computeForce, muscle TRImed force < 0 at time = 0.05

Schutte1993Muscle_Deprecated::computeForce, muscle LAT2 force < 0 at time = 0.05

Schutte1993Muscle_Deprecated::computeForce, muscle LAT2 force < 0 at time = 0.0811142

Schutte1993Muscle_Deprecated::computeForce, muscle TRImed force < 0 at time = 0.0811142

Schutte1993Muscle_Deprecated::computeForce, muscle LAT2 force < 0 at time = 0.05

Schutte1993Muscle_Deprecated::computeForce, muscle LAT2 force < 0 at time = 0.0877394

To compute actuation, the values of the muscle states and their derivatives are adopted. They are then called by an integrator or other code whenever the musculoskeletal model's state has changes and the muscle will be updated accordingly. The specific tendon force (or maximum muscle stress) can be said to be the maximum force developed per unit cross sectional area which is a frequent parameter used to investigate estimation of muscle forces.

In addition, the musculoskeletal simulations are very sensitive to passive musculotendon properties such as the force relationship that produce muscles and tendon. When muscle and tendon is stretched, these forces play an important role in movement. (Kat M. S. et al., 2014). Considering a change in tendon slack length (i.e. the length at which a tendon starts to develop passive force) by one or two percent. It radically alters the forces from a given muscle for dynamic activities such as reaching and holding to drink

from a cup, which involve specific activation sequences of many muscles. This small change influences muscles drastically, by magnitude and distribution of forces across multiple muscles.

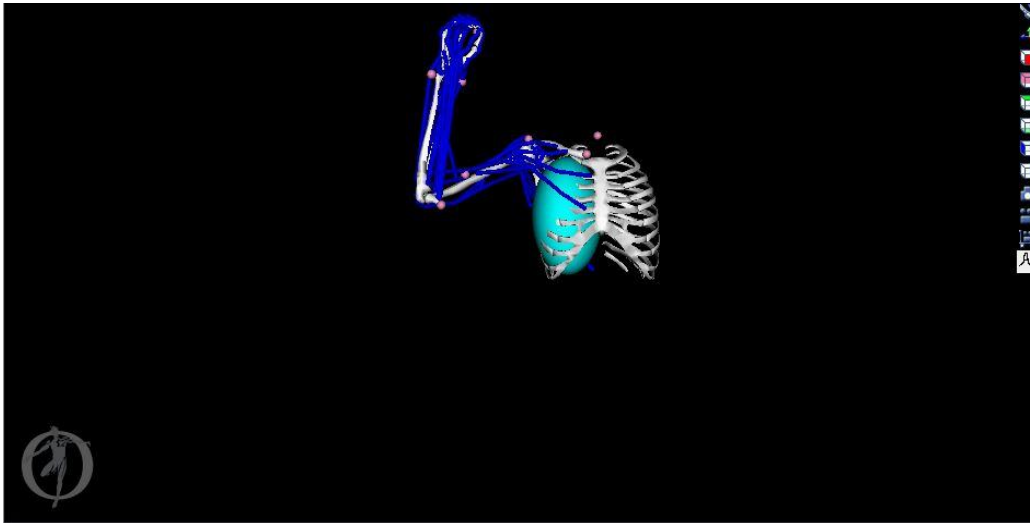


Figure 22. Shows muscle change characteristics of a drink from a cup position. (Snip-shot from OpenSim software).

The muscle moment arm measures the effectiveness of a muscle at contributing to different or particular motion over a range of pattern (Michael A. S. et al., 2013). OpenSim spontaneously analyzes this quantities for any muscle once a model is built and the calculation is the same as the conventional moment arm calculation in mechanical engineering. (Michael A. S. et al., 2013)

$$r_{\theta} = \frac{\tau_{\theta}}{S} \quad (1)$$

Where τ_{θ} is the effective torque acting about θ that is due to scalar tension force $S > 0$ generated by muscle activation; θ is the joint-associated kinematic quantity that the moment arm depends on; and r_{θ} is the muscle moment arm in respect of the kinematic quantity. It is generally taken that in muscle change characteristic as shown in Figure 22, all muscles stay at a constant value until when multiplied by a muscle's cross sectional area, it provides a measure of a muscle's maximum force production.

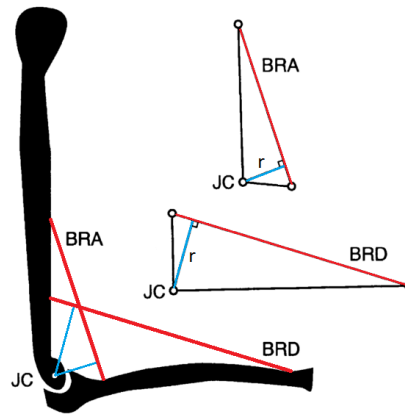


Figure 23. A typical flexion of the arm showing the distance of moment arm from line of action to joint center. (Michael A. S. et al., 2013).

To understand the maximum muscle stress, the moment arm plays an essential role using joint moment measured as a function of joint angle. The moment reflects the specific tension for elbow flexors as shown in Figure 23. The r is the moment arm (unit in length); J_c is the defined joint center; Brachialis (BRA) and Brachioradialis (BRD) muscles which are also modelled as straight-line musculotendon actuators.

CHAPTER SEVEN

7.0 RESULTS

The OpenSim simulation reflects how important the muscle response is based on the desired motion movement. After setting up the MoBL ARM model, the muscle contract and accelerate the arm towards the body and away from the body in a reaching motion as compared to grabbing a cup and drinking from the cup motion as shown in Figure 21 and 22. When the simulation completes the cycle, by reaching the end of the specified time range a plot describing the output directory during the corresponding motion file is shown in Figure 26, 27, 28 and 29 respectively.

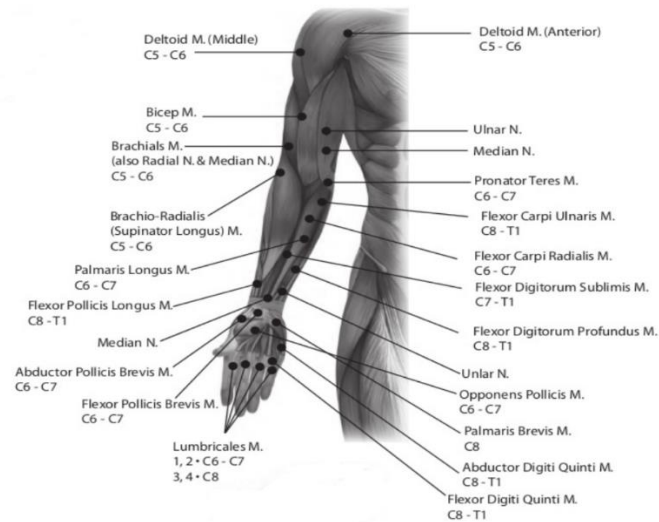


Figure 24. Shows motor point of an anterior upper extremity.

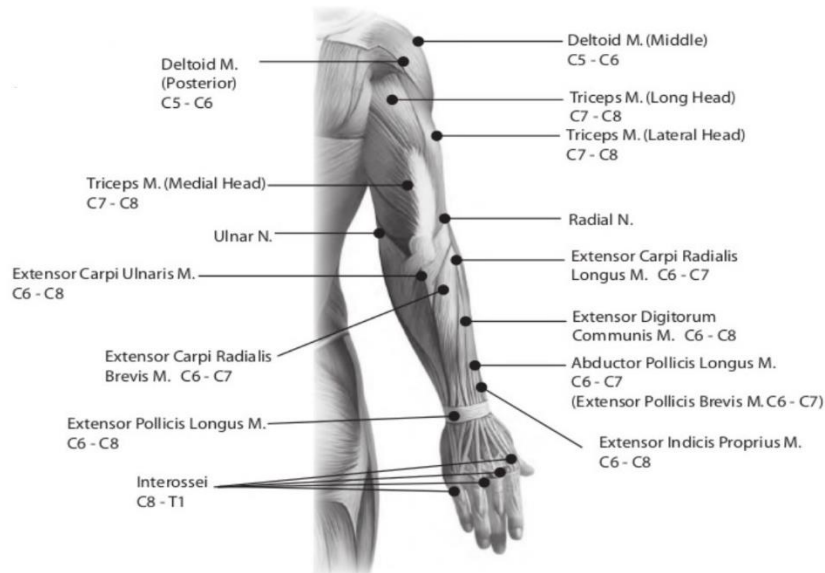


Figure 25. Shows motor point of a posterior upper extremity.

In relations to a number of muscles labelled in a given motor point upper extremity diagram (Figure 24 and 25), the stretch and contraction muscle reflexes of OpenSim MoBL ARM model is compared to these muscles and the reaction observed during the simulation as shown in Figure 26 to 29. Figures 26 and 29 show that as the automatic regulation of skeletal muscle lengthens, the muscle spindle is stretched and its nerves activity increases. This increases the alpha motor neuron activity, causing the muscle fibers to contract and thus resist the stretching. A secondary set of neurons also causes the opposing muscle to relax and the reflex functions to maintain the muscle at a constant length. Variation in the muscle in motion with the relative forces applied to the tendon and tendon fibre is shown on the Table 1 and Figures 30-33. The variation chart for Figure 30 and 31 shows the muscle relationship at standby point of zero degree and the Figure 32 and 33 shows the final point of completing the cycle of motion at 130 degrees of an angle.

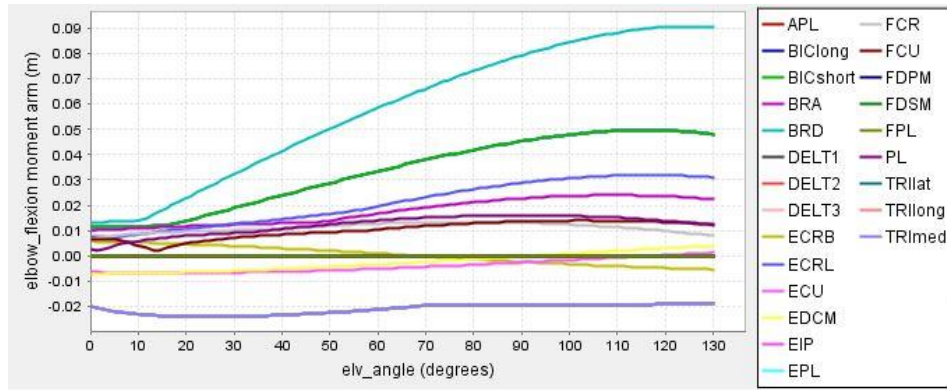


Figure 26. Shows elbow flexion moment arm (m) against elevation angle (degrees)

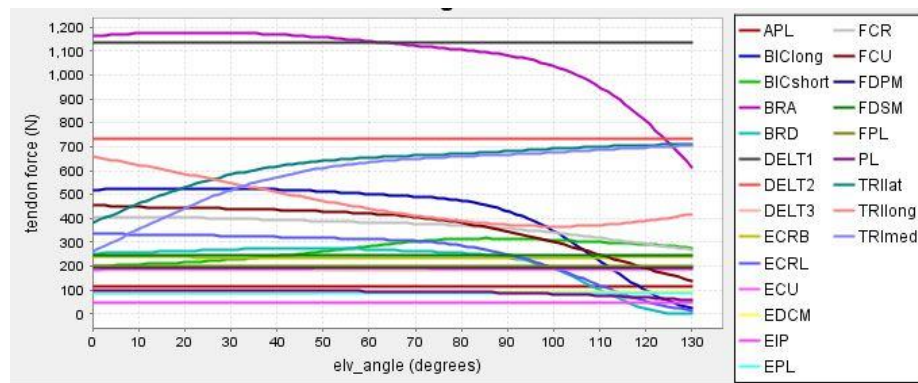


Figure 27. Shows tendon force (N) against elevation angle (degrees)

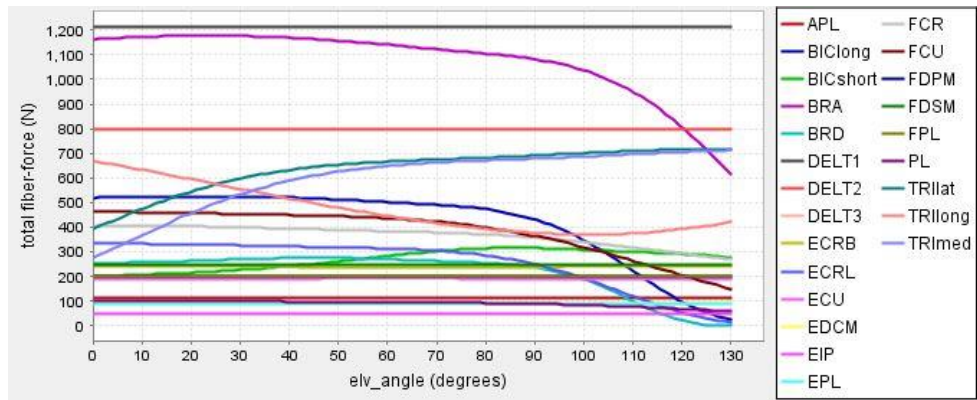


Figure 28. Shows total fibre force (N) against elevation angle (degrees)

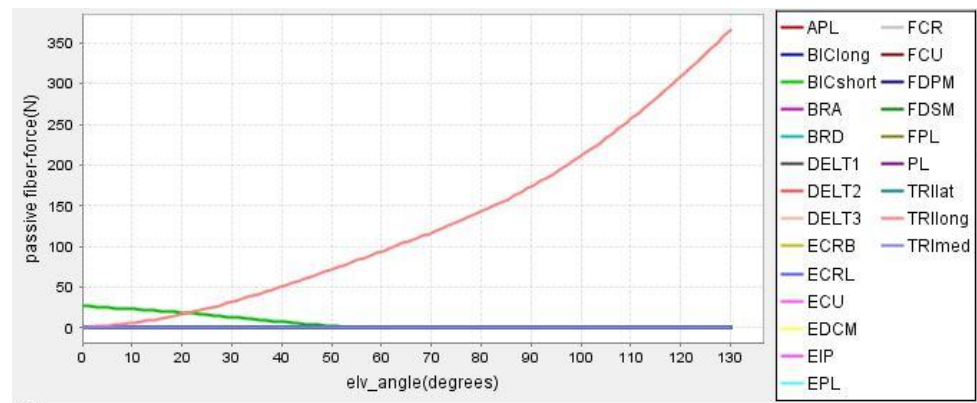


Figure 29. Shows Passive fiber force (N) against elevation angle (degrees)

Table 1. Shows the amount of passive activation force at 0 degree, 70 degrees and 130 degrees in relations to the muscles existing in the human upper extremity.

S/ N	Arm Muscles	Total fibre force (N) at 0 degree	Tendon force (N) at 0 degree	Total fibre force (N) at 70 degrees	Tendon force (N) at 70 degree	Total fibre force (N) at 130 degrees	Tendon force (N) at 130 degree
1	APL	460	450	440	460	120	130
2	BICLONG	510	520	490	510	0	0
3	BICSHORT	250	240	250	250	280	250
4	BRA	1160	1170	1120	1120	610	610
5	BRD	250	250	250	250	0	0
6	DELT1	1210	1130	1210	1130	1210	1130
7	DELT2	800	730	800	730	800	730
8	DELT3	680	660	420	430	420	420
9	ECRB	240	240	240	240	240	240
10	ECRL	340	340	300	300	0	0

11	ECU	180	180	180	180	180	180
12	EDCM	170	240	170	240	170	240
13	EIP	50	50	50	50	50	50
14	EPL	80	80	80	80	80	80
15	FCR	400	400	370	380	200	280
16	FCU	460	200	440	200	110	200
17	FDPM	510	520	480	480	0	0
18	FDSM	250	100	250	100	50	50
19	FPL	260	200	260	200	260	200
20	PL	180	100	180	80	50	50
21	TRILAT	260	250	260	250	710	700
22	TRILONG	680	660	420	400	410	410
23	TRIMED	280	330	660	300	710	700

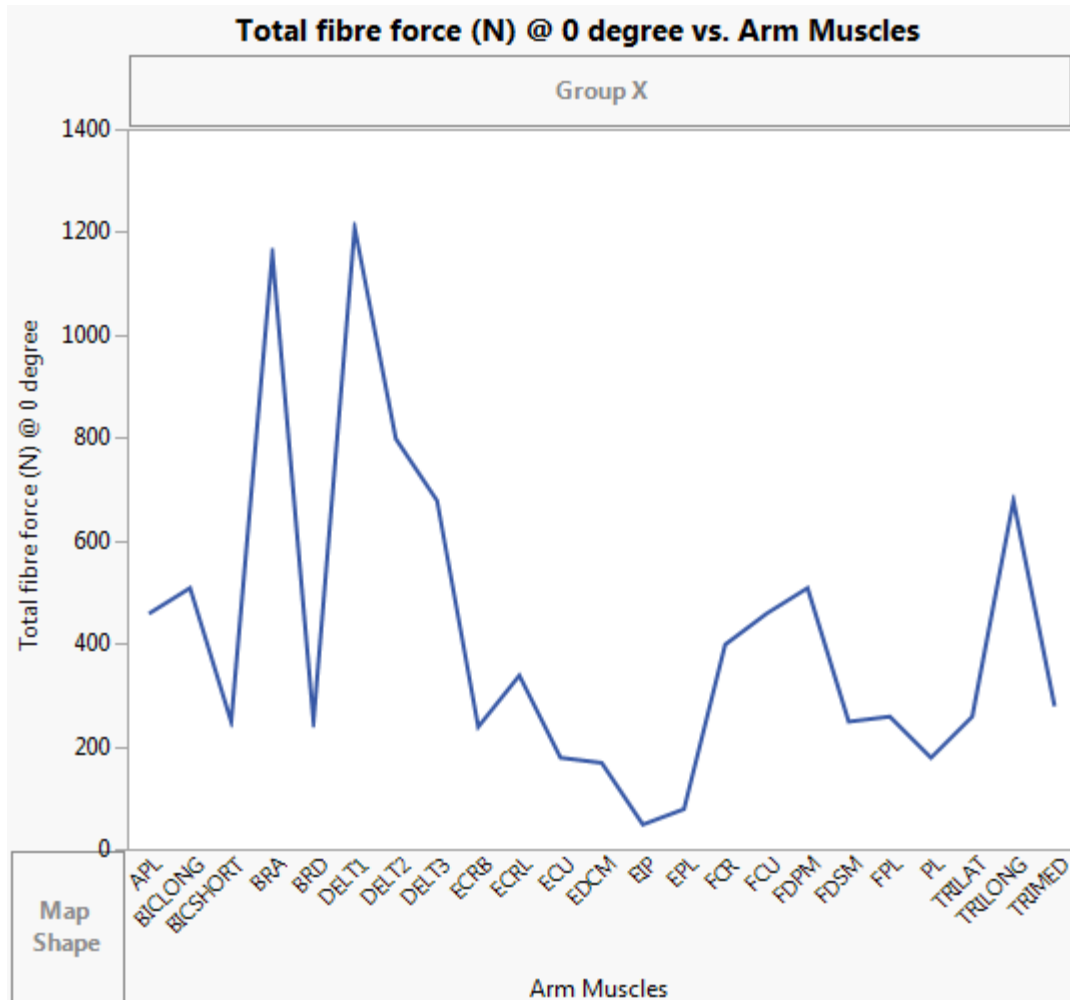


Figure 30. Shows the muscle activation reflex for Tendon fibre force at zero degrees elevation angle.

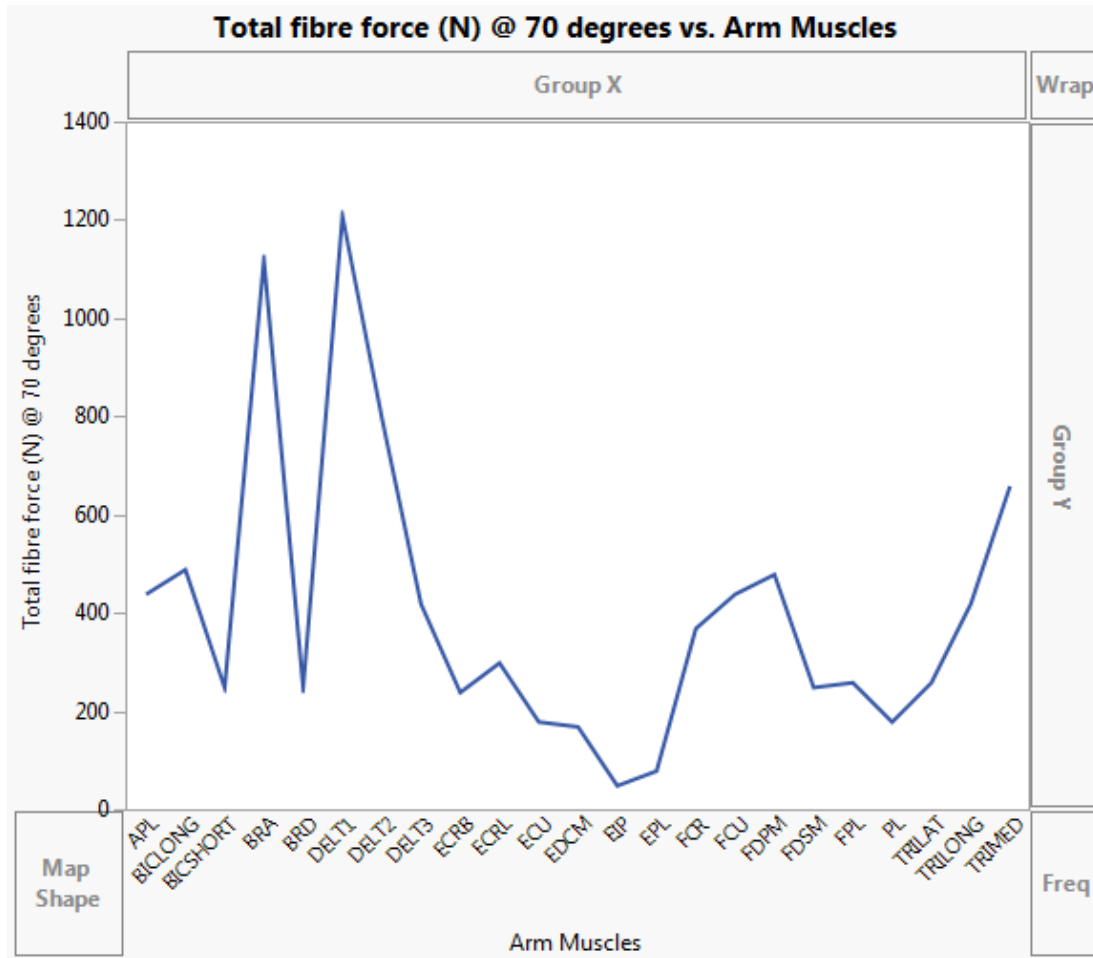


Figure 31. Shows the muscle activation reflex for total tendon force at seventy (70) degrees elevation angle.

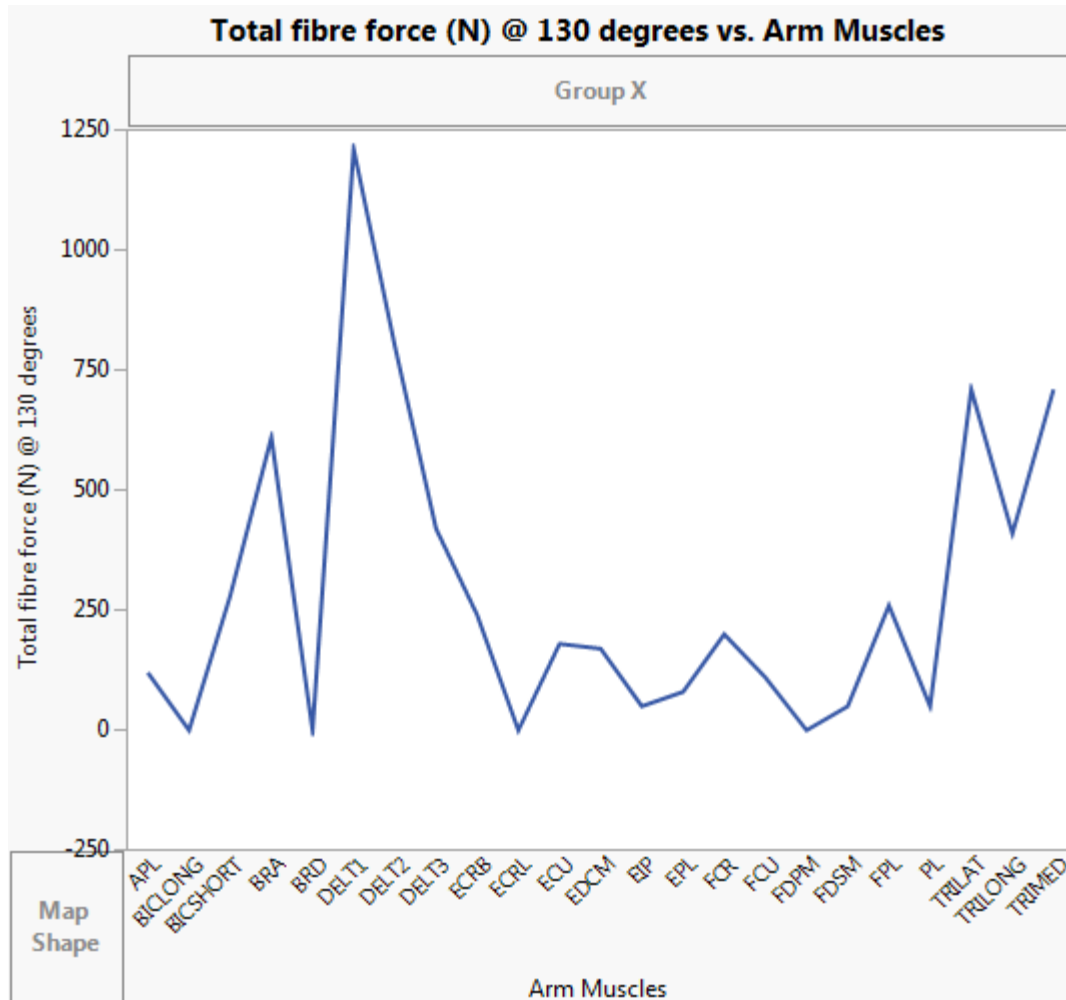


Figure 32. Shows the muscle activation reflex for total tendon force at (one hundred and thirty (130) degrees elevation angle.

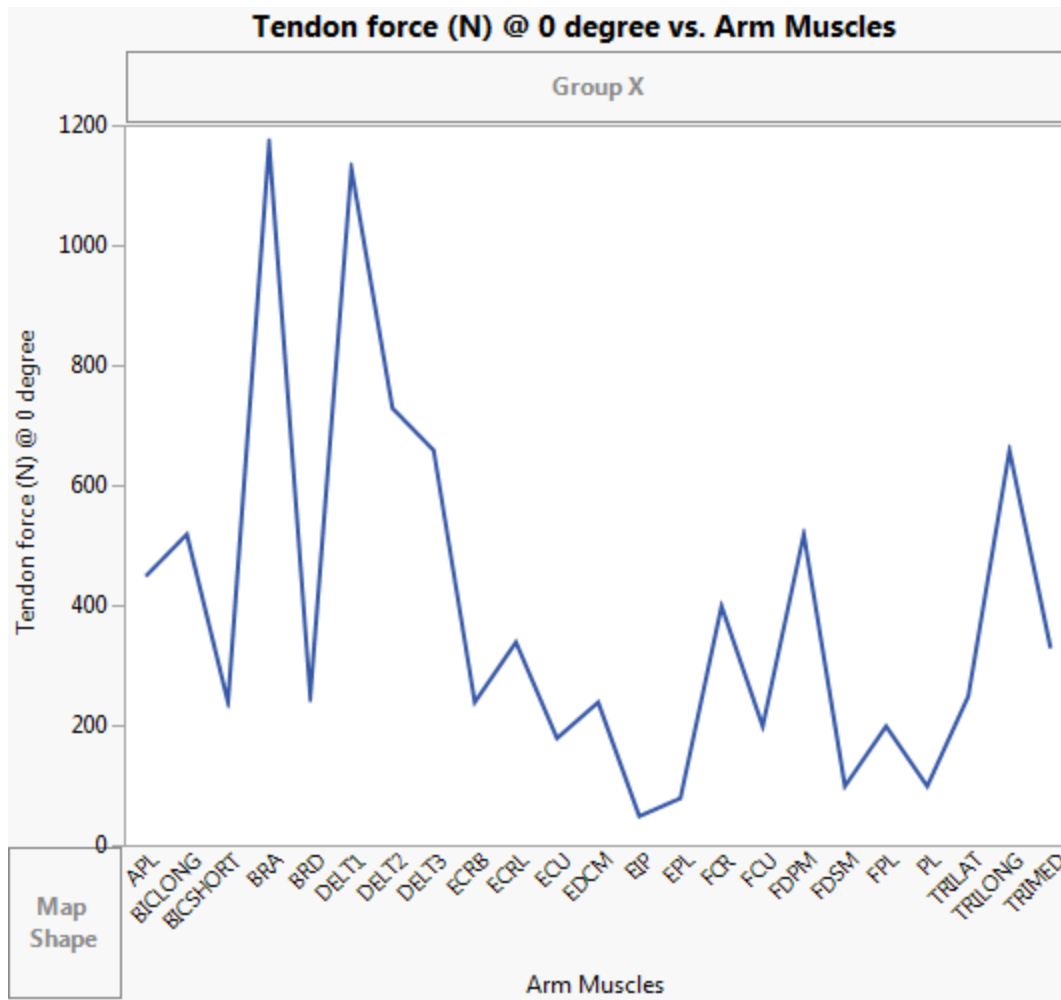


Figure 33. Shows the muscle activation reflex for Tendon force at Zero (0) degrees elevation angle.

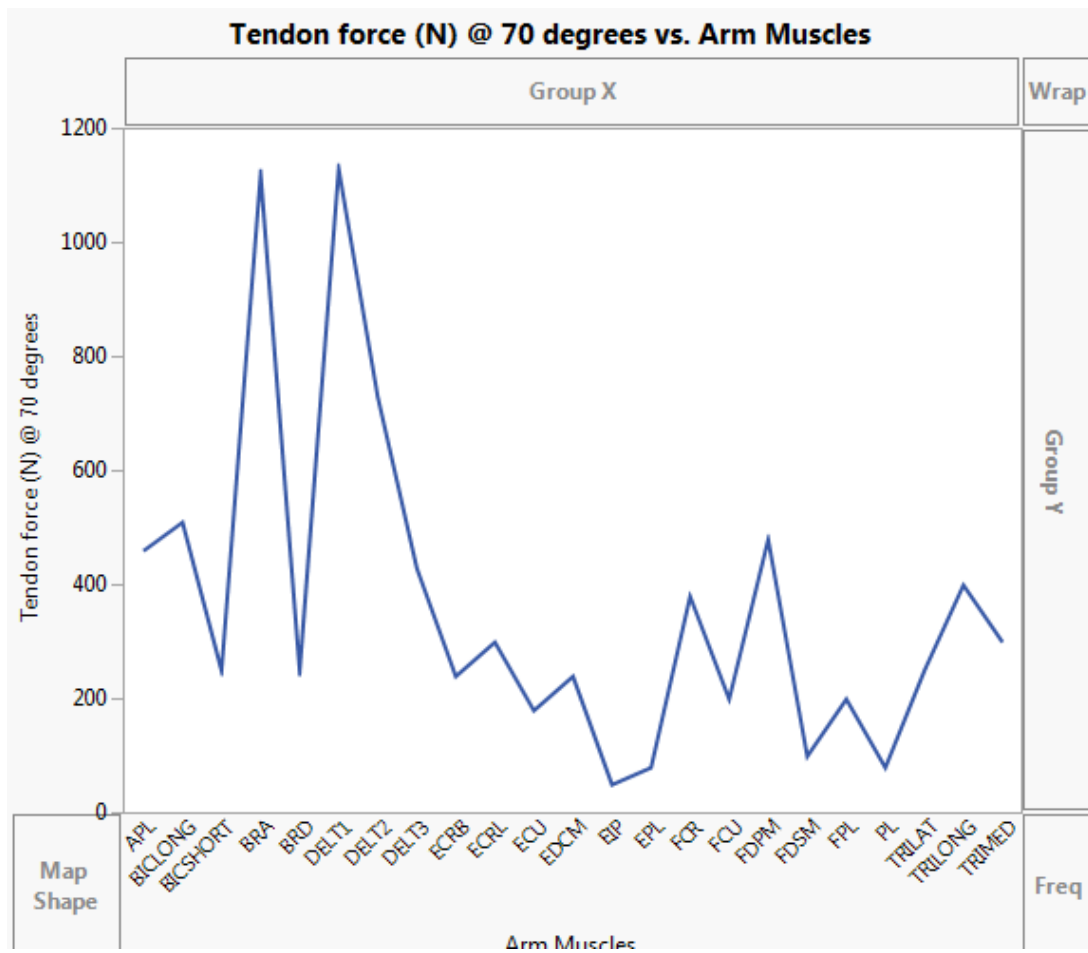


Figure 34. Shows the muscle activation reflex for total tendon force at seventy (70) degrees elevation angle.

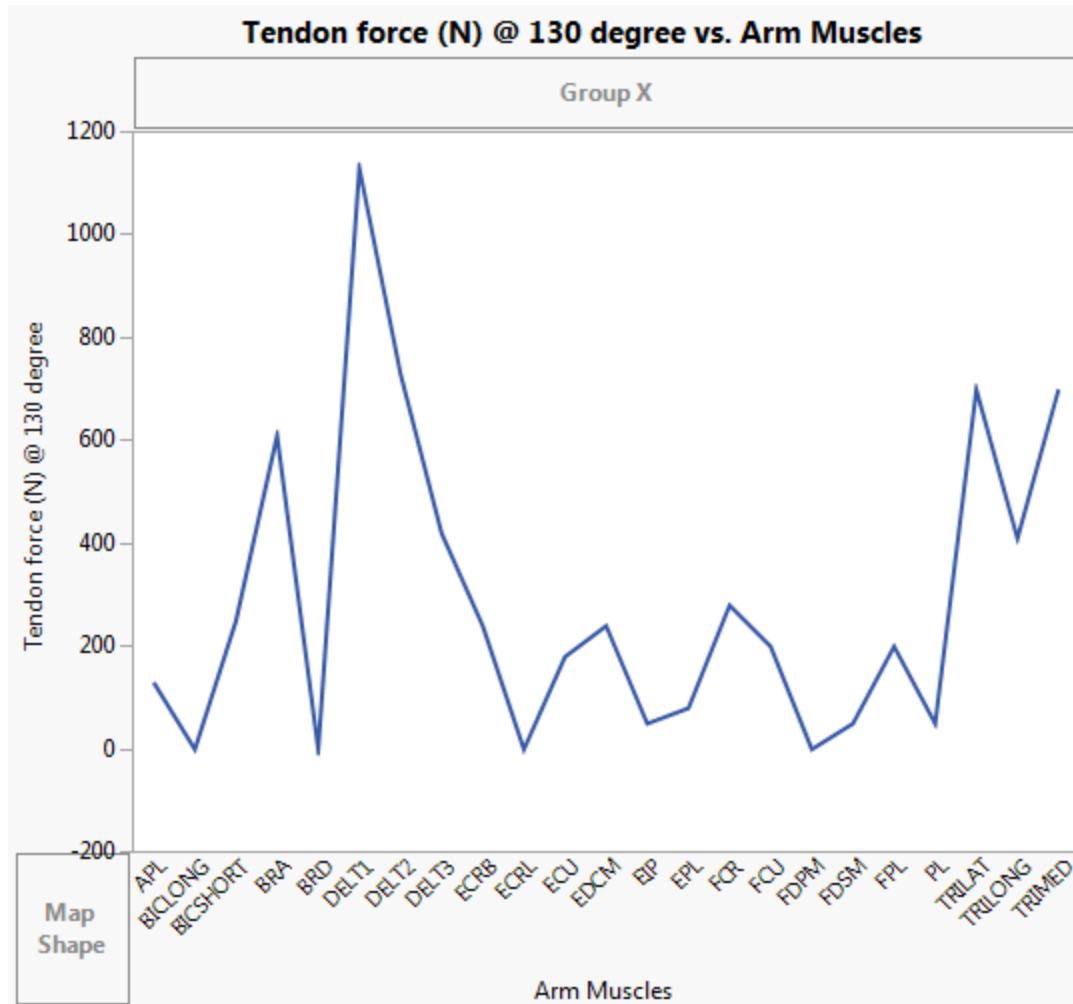


Figure 35. Shows the muscle activation reflex for tendon force at one hundred and thirty (130) degrees elevation angle.

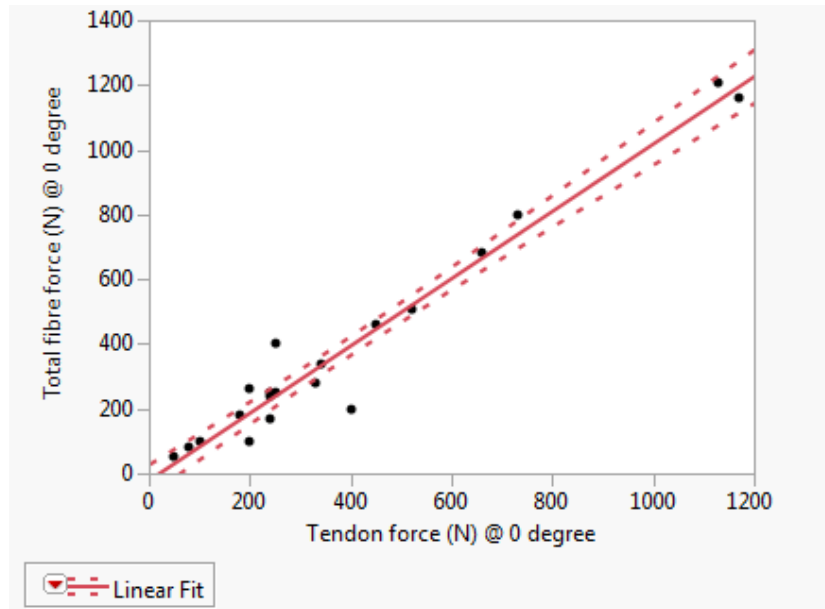


Figure 36. Shows the confidence fit of total fibre force by tendon force at zero (o) degrees.

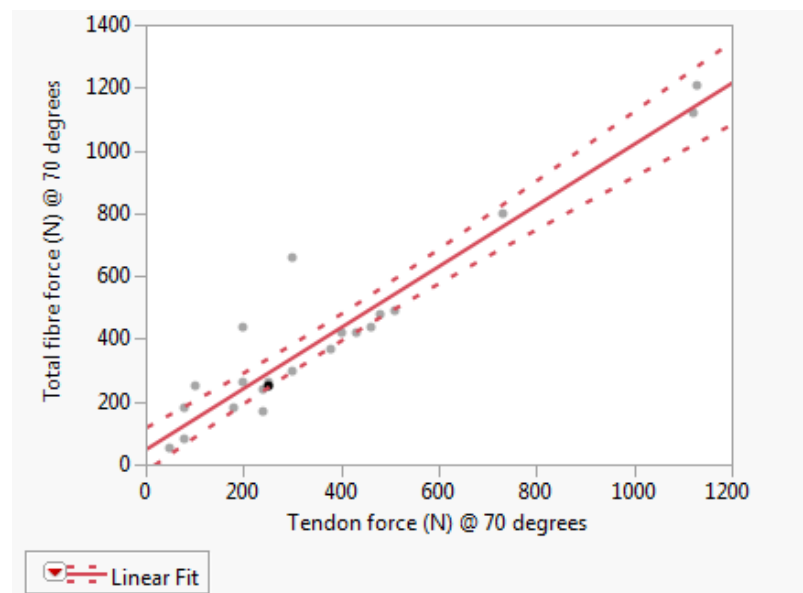


Figure 37. Shows the confidence fit of total fibre force by tendon force at seventy (70) degrees.

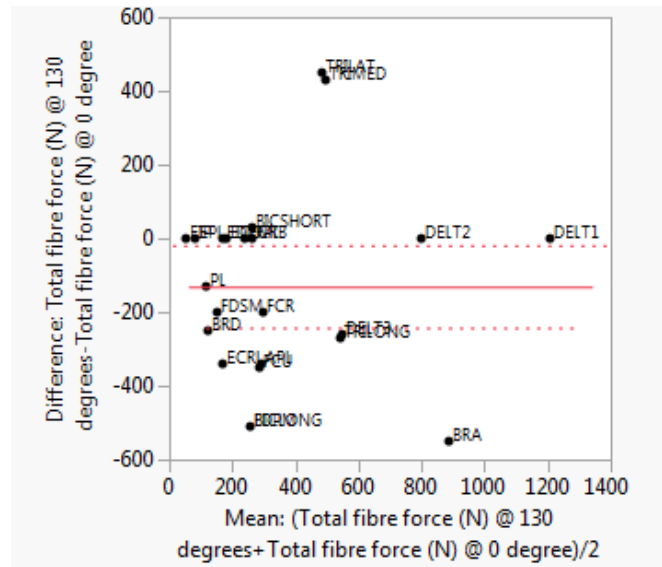


Figure 40. Shows the muscle difference relation of the total fibre force at 130 degrees against total fibre force at 0 degree, with Correlation 0.66117.

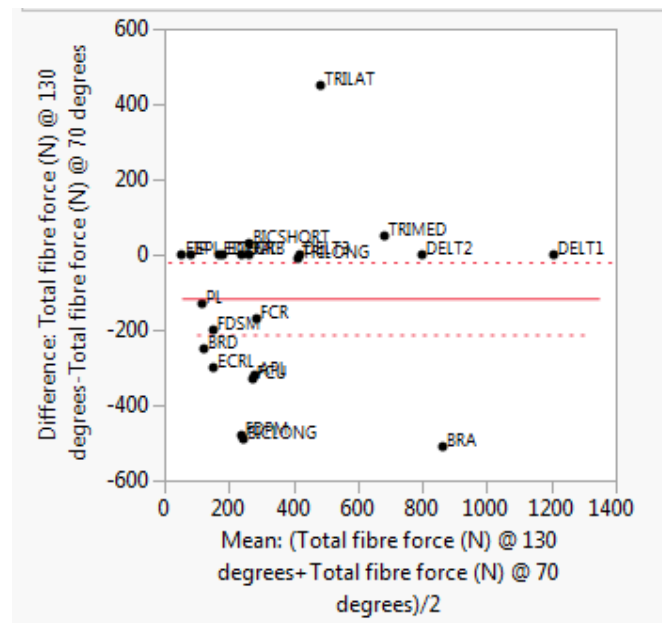
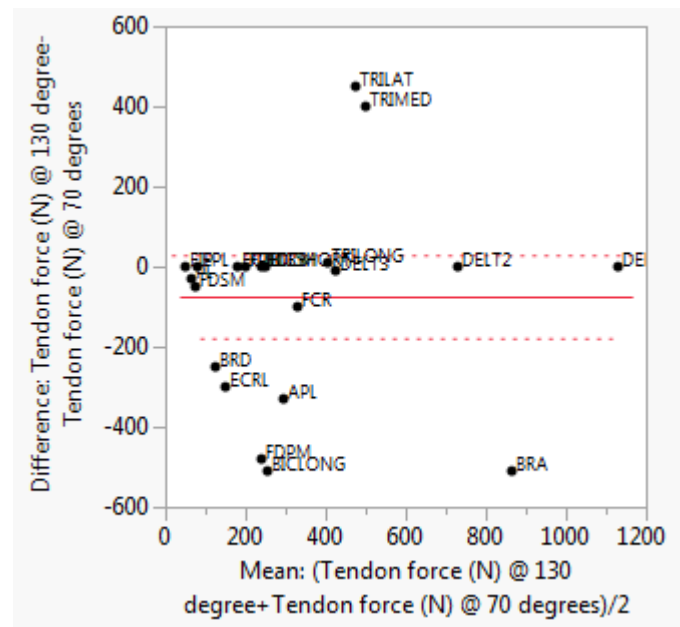


Figure 41. Shows the muscle difference relation of the total fibre force at 130 degrees against total fibre force at 70 degree, with Correlation 0.7442.



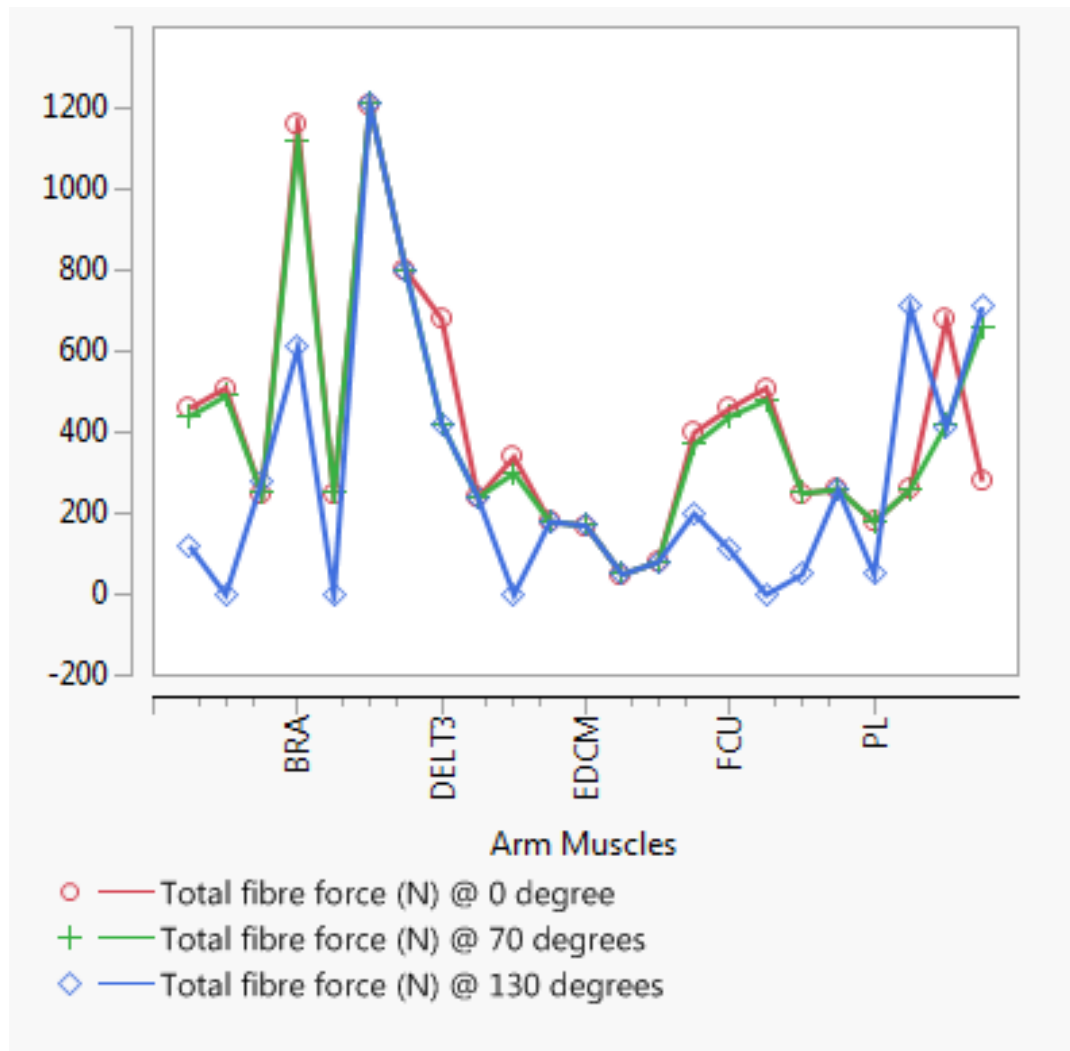


Figure 45. Overlay plots for total fibre force at different angle and muscle path.

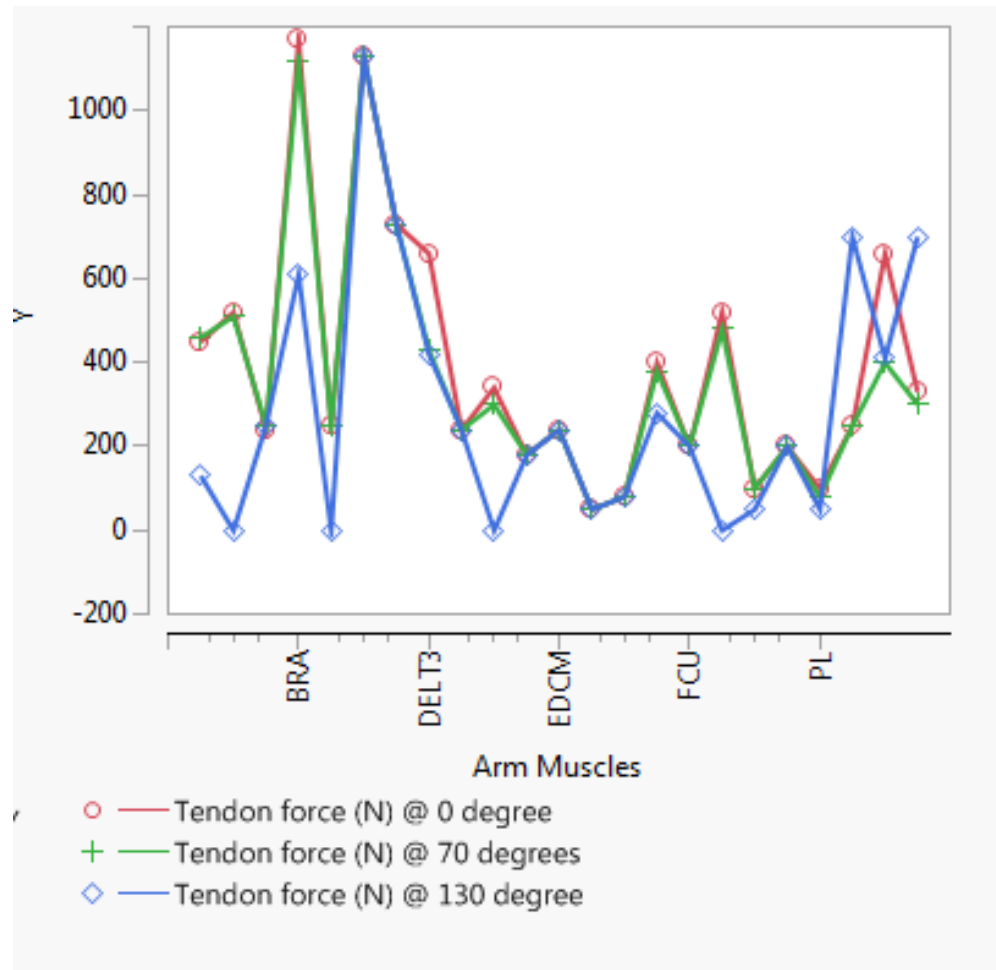


Figure 46. Overlay plots for tendon force at different angle and muscle path.

7.1 CALCULATIONS

The linear fit equation for tendon force against total fibre force at 0 degree is given by;

$$\text{Total fibre force (N) at 0 degree} = -16.6725 + 1.039106 * \text{Tendon force (N) at 0 degree}$$

The linear fit equation for tendon force against total fibre force at 70 degree is given by;

$$\text{Total fibre force (N) at 70 degrees} = 51.677962 + 0.9726563 * \text{Tendon force (N) at 70 degree}$$

The linear fit equation for tendon force against total fibre force at 130 degree is given by;

Total fibre force (N) at 130 degrees = $-16.13003 + 1.0572918 * \text{Tendon force (N) at 130 degree}$

Table 2. Muscle measurements at points 0, 70 and 130 degrees.

	0 degree	70 degrees	130 degrees
R-Square	0.959974	0.897018	0.987092
Mean of response	391.7391	405.2174	289.5652
Observation (Sum of weight)	23	23	23
Tendon force (N)	1.0391006	0.9726563	1.0572918

Apparently, just as muscle activation produces increase in signal intensity which leads to spatial pattern and intensity of muscle activation. The confidence fit chart (Figures 36, 37 and 38) describes this more by showing activity dependent changes from an increase in the relaxation time of muscle response to an activated response. The summary of fit of both forces in relations to muscle response gives an R-square summary of less than 1 (0.959974 for 0 degree, 0.897018 for 70 degrees and 0.987092 for 130 degree), with a mean of response of 391.7391 for 0 degree, 405.2174 for 70 degrees and 289.5652 for 130 degrees just as shown in Table 2 . R-squared is the coefficient of determination or

coefficient of multiple determination of multiple regression. A value less than one is a strong correlation meaning a strong activation sequence.

Conclusively, Figure 39 to 44 gives a secondary approach to comparing the differences in muscle simulation and relative forces at different degrees. The correlation value helps deduce how closely the response is as it tends to 1. However, the overlap plot (Figure 45 and 46) shows in details how the muscles interact all together depending on the angle of interest.

CHAPTER EIGHT

8.0 CONCLUSION AND FUTURE WORK

This research work presents a process of analyzing the muscle activation of the upper extremity using a free source software called OpenSim. The software in the future can be used to identify and select functional muscles of the arm that can effectively support sEMG analysis. It can also help in identifying motions and motor points for a real time augmented reality wearable device on a stroke patient. Especially patient experiencing difficulty in directing the arm or setting a particular reaching movement with the upper arm.

The DELT1, DELT2, BRA muscles has the strongest activating reflexes, which means that shoulder muscle play a significant role in speeding up the arm movement during functional reaching tasks. It may be relatively more important to upper extremity reaching function than forearm and elbow muscle. It can also be deduced that since muscle tissues tolerate electrical imbalances or capabilities which goes across cell membranes by concentrating positive or negative charges on opposite side of the membrane. There exist potentials form of energy stored with activation (such as from a nerve impulse) due to ions crossing the muscle cell membranes, producing activity and resulting in muscle contraction. With reference to this process, it can be further noted that the electromyogram, or sEMG can give further recordings of electrical activity within these muscles. Although inherited and acquired disorders of muscles of the central and peripheral nervous systems can result in abnormal EMG readings, but for stroke related issues OpenSim gives a

foundation of understanding of the muscle sequence and this makes it easy for further development and studies using sEMG signals.

Conclusively, OpenSim is useful in the diagnosis of disorders affecting muscles and the nerves that supply them. Based on the “heart and stroke facts and statistics of the American Heart Association 2014”. The rate of stroke death from 2001 to 2014 has been falling by 35.1% and the actual number of deaths caused by stroke had declined by 21.2%. Thanks to innovative ideas in robotic enhancement projects going on in the United States and all over the world.

REFERENCES

Anatomy Zone. (2012, July 20). *Shoulder joint - glenohumeral joint - 3D anatomy tutorial*.

[Video file]. Retrieved from,

<https://www.youtube.com/watch?v=vG1XQkj3Yx0&index=9&list=PL78D995229606FDE>

Buchanan, T. S. (1995). Evidence that maximum muscle stress is not a constant: differences in specific tension in elbow flexors and extensors. *Med. Eng. Phys.*, 17, 529–536 doi: [http://dx.doi.org/10.1016/1350-4533\(95\)00005-8](http://dx.doi.org/10.1016/1350-4533(95)00005-8)

Chadwick, E. [OpenSimVideos]. (2014, June 24). *Fast simulation of arm dynamics for real time, user in the loop control application*. [Video file]

<https://www.youtube.com/watch?v=ZPChegKKujc&list=PLGBiefGDmSuR5NbZFuGOrjnTBx--H-H3C&feature=share>

Chadwick, E. [OpenSimVideos]. (2014, June 24). *Fast simulation of arm dynamics for real time, user in the loop control application* [Video file].

Retrieved from, <https://www.youtube.com/watch?v=ZPChegKKujc>

Charles, S.K., Formica, D., Zollo, L., Guglielmelli, E., Hogan, N., Krebs, H.I. (2012). The passive stiffness of the wrist and forearm. *Journal of Neurophysiology*. 108(4), 1158–1166. doi: 10.1152/jn.01014.2011

Daly, M., Vidt, M., Holzbaur, K.R.S. (n.d.). Forward dynamic simulation of an upper extremity movement using computed muscle control. *Wake Forest University Baptist Medical Center*. Retrieved from, <http://archive.asbweb.org/conferences/2009/1054.pdf>

Delp, S.L., Anderson, F.C., Arnold, A.S., Loan, P., Habib, A., John, C.T., Thelen, D.G. (2007). OpenSim: Open-source software to create and analyze dynamic simulations of movement. *IEEE Transactions on Biomedical Engineering*, 55, 1940-1950. Retrieved from, <http://simtkconfluence.stanford.edu:8080/display/OpenSim/Tutorial+3++Scaling,+Inverse+Kinematics,+and+Inverse+Dynamics>

Delp, S., Murray, W., Hammer, S. (n.d.). OpenSim tutorial #2: Simulation and analysis of a tendon transfer surgery. Retrieved from, <http://simtkconfluence.stanford.edu:8080/download/attachments/3376123/OpenSimTutorial2.pdf?version=1&modificationDate=1359410743142>

Dipietro, L., Ferraro, M., Palazzolo, J.J., Krebs, H.I., Volpe, B.T., Hogan, N. (2005). Customized interactive robotic treatment for stroke: EMG-triggered therapy. *IEEE Trans Neural Syst Rehabil Eng.* 13(3), 325–334. doi: 10.1109/TNSRE.2005.850423

Hamner, C.A., Guendelman, E., John, C., Reinbolt, J., Delp, Scott. (n.d.). Opensim tutorial #3 Scaling, Inverse kinematics and Inverse dynamics. Retrieved from, <http://simtk-confluence.stanford.edu:8080/download/attachments/3376124/OpenSimTutorial3.pdf?version=2&modificationDate=1352481321866>

Holzbaur, K.R.S., Murray, W.M., Delp, S.L. (2005). A model of the upper extremity for simulating musculoskeletal surgery and analyzing neuromuscular control. *Annals of Biomedical Engineering*, 33(6), 829-840. doi: 10.1007/s10439-005-3320-7

Katherine, R.S., Holzbaur L., Murray, W.M., Delp, S.L. (2005). A model of the upper extremity for simulating musculoskeletal surgery and analyzing neuromuscular control. *Annals of Biomedical Engineering*, 33(6), 829–840. doi: 10.1007/s10439-005-3320-7

- King, T. (2008). How great is the burden of musculoskeletal disease? *American Academy of Orthopaedic Surgeons*. 9(4). Retrieved from, <http://www.aaos.org/news/aaosnow/jan08/youraaos15.asp>
- Krebs, H.I., Volpe, B.T., Aisen, M.L., Hogan, N. (2000). Increasing productivity and quality of care: robot-aided neuro-rehabilitation. *Journal of Rehabilitation Research and Development*. 37(6). Retrieved from, <http://www.rehab.research.va.gov/jour/00/37/6/krebs376.htm>
- Krucik, G. (2013, April 24). Muscles. *Healthline*. Retrieved from, <http://www.healthline.com/human-body-maps/arm-muscles#seoBlock>
- Liepert J., Bauder H., Milner W.H., Taub E., Weiller C. (2000). Treatment-induced cortical reorganization after stroke in humans. *Stroke*. 31, 1210–1216. Retrieved from, <http://stroke.ahajournals.org/content/31/6/1210.full>, doi:10.1161/01.STR.31.6
- Liu, J.Z., Brown, R., Yue, G.H. (2002). A dynamical model of muscle activation, fatigue, and recovery. *Biophysical Journal*, 82(5), 2344-2359. Retrieved from, <http://simtk-confluence.stanford.edu:8080/display/OpenSim24/Custom+Muscle+Model+Part+On>
- Lum P.S., Burgar C.G., Kenney D., Van der Loos, H.F. (1991). Quantification of force abnormalities during passive and active-assisted upper-limb reaching movements

in post-stroke hemiparesis. *IEEE Trans. Biomed. Eng.* 46(6), 652–662. Retrieved from, <http://www.ncbi.nlm.nih.gov/pubmed/10356872>

Mozaffarian, D., Benjamin, E.J., Go, A.S., Arnett, D.K., Blaha, M.J., Cushman, M., Ferranti, S.D. (2014). Heart and stroke facts and statistics. *American Heart Association*. doi 10.1161/CIR.000000000000152

OpenSim Videos (2014, August 19). *Interpreting upper limb dynamic simulations in the context of clinical variability*. [Video file]. Retrieved from, <https://www.youtube.com/watch?v=VyVVPkWEXec&index=3&list=PLGBiefGDmSuR5NbZFuGOrjnTBx--H-H3C>

OpenSim Videos (2014, October 31). *OpenSim for open source musculoskeletal modeling & simulation: roadmap for the future*. [Video file]. Retrieved from, <https://www.youtube.com/watch?v=4OILUCgEGOG&list=PLGBiefGDmSuR5NbZFuGOrjnTBx--H-H3C>

Rickson, C. (Photographer). (n.d.). Hand Muscle Anatomy [digital image]. Retrieved from, <http://www.ehealthideas.com/2014/02/hand-muscles-anatomy.html>

Reinbolta, J.A., Sethb, A., Delpb S.L. (2015). Simulation of human movement: applications using OpenSim. *aMechanical, Aerospace, & Biomedical Engineering*. 2, 186–198. DOI: 10.1016/j.piutam.2011.04.019

Reinkensmeyer D., Schmit B.D., Rymer Z. (1999). Assessment of active and passive restraint during guided reaching after chronic brain injury. *Ann. Biomed. Eng.* 27(6), 805–814. Retrieved from, <http://academic.research.microsoft.com/Paper/169976.aspx>

Rizzolatti G., Fadiga L., Gallese V., Fogassi L. (1996). Premotor cortex and the recognition of motor actions. *Cogn. Brain Res.*, 3, 131-141. Retrieved from, <http://www.ncbi.nlm.nih.gov/pubmed/8713554>

Saul, K. [OpenSim Videos]. (2014, August 19). *Interpreting upper limb dynamic simulations in the context of clinical variability*. [Video file]. Retrieved from, <https://www.youtube.com/watch?v=VyVVPkWEXec&index=3&list=PLGBiefGDmSuR5NbZFuGOrjnTBx--H-H3C>

Upper arm full OpenSim Videos. (2014, June 24). *Fast simulation of arm dynamics for real time, user in the loop control application*. [Video file]. Retrieved from, <https://www.youtube.com/watch?v=ZPChegKKujc>

Whitall J., McCombe L., Waller S., Silver K.H., Macko R.F. (2000). Repetitive bilateral arm training with rhythmic auditory cueing improves motor function in chronic hemiparetic stroke. *Stroke*. 31(10), 2390–2395. doi 10.1161/01.str.31.10.2390

Wikstrom H., Roine R.O., Aronen H.J., Salonen O., Sinkkonen J., Ilmoniemi R.J., Huttunen J. (2000). Specific changes in somatosensory evoked magnetic fields during recovery from sensory motor stroke. *Ann Neurol*. 47, 353–360. doi: 10.4103/1673-5374.139468

APPENDIX

MUSCLE ABBREVIATIONS

DELT1 – Deltoid 1

TMNJ – Teres major

DELT2 – Deltoid 2

APL – Abductor pollicis longus

DELT3 – Deltoid

FPL – Flexor pollicis longus

SUPSP – Supraspinatus

EPB – Extensor pollicis brevis

INFSP – infraglenoid tubercle of scapula

EPL – Extensor pollicis longus

SUBSC – Subscapularis

EIP – Extensor indicis pollicis

TMIN – Teres minor

EDM – Extensor digiti minimi

Pecmi – Pectoralis major

EDCM – Extensor digitorum communis

LAT – Lateral cutaneous nerve of forearm

FDPL – Flexor digitorum profundus

CORB – Coraco brachialis

TRILONG – Triceps brachii long head

TRILAL- Triceps brachii lateral head

TRIMED – Triceps brachi medial head

ANC – Anconeus

SUP – Supinator

BICLONG-Biceps long

BICSHORT – Biceps short

BRA – Brachio radialis

BRD – Brachioradialis

ECRL – Extensor carpi radialis brevis

ECRB – Extensor carpi ulnaris

FCR – Flexor carpi ulnaris

ECU – Extensor carpi ulnaris

FCR – Fexor carpi radialis

FCU – Flexor carpi ulnaris

PI – Palmaris longus

PQ – Pronator quadratus

FDSL – Dlexor digitorum superficialis longus

FDSM-Flexo digitororium superficialis muscles

DOF – Degree of freedom

sEMG- Signal electromyograph

MCP – metacarphophalangeal

IP – Interphalangial joint

ARM – Assisted rehabilitation and measurement

GUI – GRAPHICAL USER INTERPHASE

ARWED- Augmented Reality Wearable device

Human Arm Kinematics

MATLAB program FPK.m

that was used to generate the male FPK

results and animate the arm and muscles (stick figures) to the screen.

```
%-----  
-----  
% FPK.m  
% Forward pose kinematics trajectory for planar 3R  
% adult male human arm shoulder, elbow, wrist  
% ME 4670 & BME 5670, Dr. Bob  
%-----  
-----  
clc; clear;  
% Constants  
DR = pi/180;  
L = [0.315 0.287 0.105]; L1 = L(1); L2 = L(2); L3 = L(3);  
% male arm lengths  
rB = ((3/8)/(1+7/8))*L2;  
% biceps insertion length - scaled from a figure  
rT = ((3/16)/(1+7/8))*L2;  
% triceps insertion length - scaled from a figure  
th1lim = [-140 90]*DR;  
% shoulder pitch joint absolute angle limits  
th2lim = [ 0 145]*DR;  
% elbow pitch joint relative angle limits  
th3lim = [ -70 90]*DR;  
% wrist pitch joint relative angle limits  
th10 = th1lim(1); th1f = th1lim(2);  
th20 = th2lim(1); th2f = th2lim(2);  
th30 = th3lim(1); th3f = th3lim(2);  
N = 60;  
dth1 = (th1f-th10)/(N-1);  
dth2 = (th2f-th20)/(N-1);  
dth3 = (th3f-th30)/(N-1);  
th1 = [th10:dth1:th1f];  
th2 = [th20:dth2:th2f];  
th3 = [th30:dth3:th3f];  
t0 = 0; tf = 3;  
% Artificial time vector
```

```

dt = (tf-t0)/(N-1);
t = [t0:dt:tf];
%
% Loop for FPK motion simulation
%
figure;
for
i = 1:N,
%
% Forward Pose Kinematics
%
c1 = cos(th1(i));
s1 = sin(th1(i));
c12 = cos(th1(i)+th2(i));
s12 = sin(th1(i)+th2(i));
c123 = cos(th1(i)+th2(i)+th3(i));
s123 = sin(th1(i)+th2(i)+th3(i));
xH(i) = L1*c1 + L2*c12 + L3*c123;
% FPK solution
yH(i) = L1*s1 + L2*s12 + L3*s123;
phi(i) = th1(i)+th2(i)+th3(i);
xb(i) = L1*c1 + rB*c12;
% biceps insertion point

yb(i) = L1*s1 + rB*s12;
xt(i) = L1*c1 - rT*c12;
% triceps insertion point
yt(i) = L1*s1 - rT*s12;
% biceps and triceps muscle lengths
Lb(i) = norm([xb(i) yb(i)]); Lt(i) = norm([xt(i) yt(i)]);
% biceps and triceps muscle angles
phib(i) = atan2(-yb(i),-xb(i)); phit(i) = atan2(-yt(i),-
xt(i));
%
% Animate human arm
x1 = [0 L1*c1];
y1 = [0 L1*s1];
x2 = [xt(i) L1*c1 + L2*c12];
y2 = [yt(i) L1*s1 + L2*s12];
x3 = [L1*c1 + L2*c12 xH(i)];
y3 = [L1*s1 + L2*s12 yH(i)];
xb = [0 xb(i)];
yb = [0 yb(i)];
xt = [0 xt(i)];
yt = [0 yt(i)];

```

```

plot(x1,y1,
'b'
,x2,y2,
'b'
,x3,y3,
'b'
,xb,yb,
'r'
,xt,yt,
'r'
); grid;
axis(
'square'
); axis([-0.7 0.7 -0.7 0.7]);
set(gca,
'FontSize'
,18); xlabel(
'\itx' ({\itm}))'
); ylabel(
'\ity' ({\itm}))'
);
pause(dt);
if
i==1
pause;
end
end
%
% Plots
figure;
% input angles trajectory plot
plot(t,th1/DR,
'r'
,t,th2/DR,
'g'
,t,th3/DR,
'b'
); grid;
set(gca,
'FontSize'
,18);
xlabel(
'\ittime' ({\itsec}))'
); ylabel(
'\it\theta_i' ({\itdeg}))'

```

```

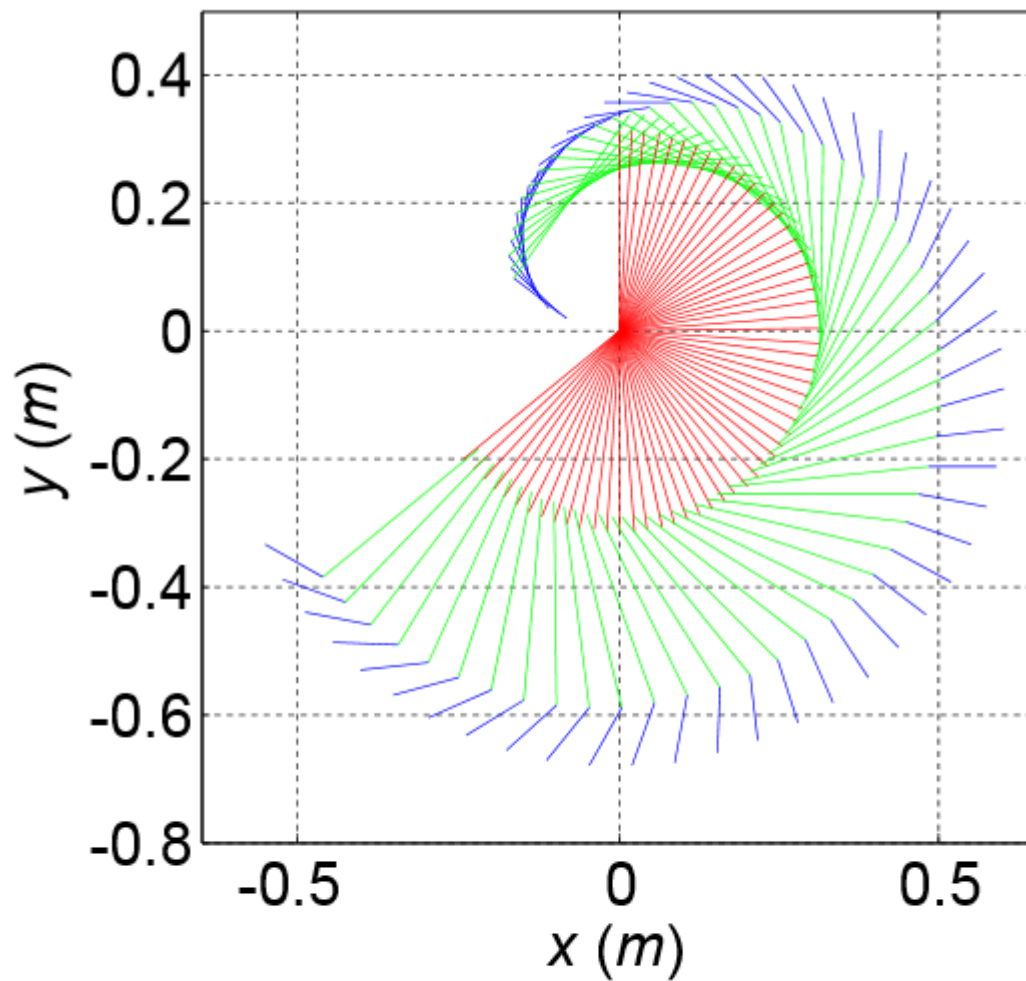
);
legend(
'\it\theta}_1'
,
'\it\theta}_2'
,
'\it\theta}_3'
);
figure;
% Cartesian FPK results x, y, phi plots
subplot(311)
plot(t,xH,
'b'
); grid;
set(gca,
'FontSize'
,18); ylabel(
'\itx} ({\itm})'
);
subplot(312)
plot(t,yH,
'b'
); grid;
set(gca,
'FontSize'
,18); ylabel(
'\ity} ({\itm})'
);
subplot(313)
plot(t,phi/DR,
'b'
); grid; set(gca,
'FontSize'
,18);
xlabel(
'\ittime ({\itsec})'
); ylabel(
'\it\phi} ({\itdeg})'
);
figure;
% Cartesian FPK hand tip trajectory
plot(xH,yH,
'b'
); grid;
axis(

```

```

'square'
); axis([-0.7 0.7 -0.7 0.7]);
set(gca,
'FontSize'
,18); xlabel(
'\itx ({\itm})'
); ylabel(
'\ity ({\itm})'
);
figure;
% muscle lengths plots (angle plots code not shown)
plot(t,Lb,
'r'
,t,Lt,
'g'
); grid; set(gca,
'FontSize'
,18);
xlabel(
'\ittime ({\itsec})'
); ylabel(
'\itL_i ({\itm})'
);
legend(
'\itbicep'
,
'\ittricep'
);

```

Adult Male Human Arm FPK Simulation with all Segments, 60 steps

Upper Arm, Forearm, Hand

Biceps and triceps muscles omitted for clarity

Other simulation with same motion;

SIMULATION ONE

Loaded model MoBL_ARMS_Upper_Limb_Model_OpenSim from file C:\OpenSim
3.1\Models\OpenSim Model\OpenSim
Model\ARMSs_Upper_Limb_Model_OpenSim.osimnew

Running tool MoBL_ARMS_Upper_Limb_Model_OpenSim.

No external loads will be applied (external loads file not specified).

MODEL: MoBL_ARMS_Upper_Limb_Model_OpenSim

ANALYSES (2)

analysis[0] = Un-named analysis.

analysis[1] = Un-named analysis.

BODIES (14)

body[0] = ground (mass: 0) (inertia: 1 0 0 0 1 0 0 0 1)

body[1] = clavicle (mass: 0.156) (inertia: 0.00024259 -1.898e-005 -6.994e-005 -1.898e-005 0.00025526 5.371e-005 -6.994e-005 5.371e-005 4.442e-005)

body[2] = clavphant (mass: 0.0001) (inertia: 1 0 0 0 1 0 0 0 1)

body[3] = scapula (mass: 0.70396) (inertia: 0.0012429 0.0004494 0.00040922 0.0004494 0.0011504 0.0002411 0.00040922 0.0002411 0.0013651)

body[4] = scapphant (mass: 0.0001) (inertia: 1 0 0 0 1 0 0 0 1)

body[5] = humphant (mass: 0.0001) (inertia: 1 0 0 0 1 0 0 0 1)

body[6] = humphant1 (mass: 0.0001) (inertia: 1 0 0 0 1 0 0 0 1)

body[7] = humerus (mass: 1.99757) (inertia: 0.0122776 -0.00034741 -0.0002325 -0.00034741 0.00255133 0.0012293 -0.0002325 0.0012293 0.0125789)

body[8] = DELT2pt2a (mass: 0.0001) (inertia: 1e-005 0 0 0 1e-005 0 0 0 1e-005)

body[9] = ulna (mass: 1.1053) (inertia: 0.00541309 0.00031686 -7.615e-005 0.00031686 0.00115318 0.00109169 -7.615e-005 0.00109169 0.00494361)

body[10] = radius (mass: 0.23359) (inertia: 0.00043855 3.014e-005 -4.24e-006 3.014e-005 8.859e-005 6.418e-005 -4.24e-006 6.418e-005 0.00040258)

body[11] = BICpt (mass: 0.0001) (inertia: 1e-005 0 0 0 1e-005 0 0 0 1e-005)

body[12] = proximal_row (mass: 0.0001) (inertia: 1e-005 0 0 0 1e-005 0 0 0 1e-005)

body[13] = hand (mass: 0.5819) (inertia: 0.00011 9e-007 -2e-007 9e-007 6e-005 1.2e-005 -2e-007 1.2e-005 0.00015)

ACTUATORS (50)

actuator[0] = DELT1

actuator[1] = DELT2

actuator[2] = DELT3

actuator[3] = SUPSP

actuator[4] = INFSP

actuator[5] = SUBSC

actuator[6] = TMIN

actuator[7] = TMAJ

actuator[8] = PECM1

actuator[9] = PECM2

actuator[10] = PECM3

actuator[11] = LAT1

actuator[12] = LAT2

actuator[13] = LAT3

actuator[14] = CORB

actuator[15] = TRIlong

actuator[16] = TRIlat

actuator[17] = TRImed

actuator[18] = ANC

actuator[19] = SUP

actuator[20] = BIClong

actuator[21] = BICshort

actuator[22] = BRA

actuator[23] = BRD
actuator[24] = ECRL
actuator[25] = ECRB
actuator[26] = ECU
actuator[27] = FCR
actuator[28] = FCU
actuator[29] = PL
actuator[30] = PT
actuator[31] = PQ
actuator[32] = FDSL
actuator[33] = FDSR
actuator[34] = FDSM
actuator[35] = FDSI
actuator[36] = FDPL
actuator[37] = FDPR
actuator[38] = FDPMP
actuator[39] = FDPPI
actuator[40] = EDCL
actuator[41] = EDCR
actuator[42] = EDCM
actuator[43] = EDCI
actuator[44] = EDM
actuator[45] = EIP
actuator[46] = EPL
actuator[47] = EPB

```
actuator[48] = FPL
actuator[49] = APL
numStates = 152
numCoordinates = 26
numSpeeds = 26
numActuators = 50
numBodies = 14
numConstraints = 19
numProbes = 0
```

STATES (152)

```
y[0] = sternoclavicular_r2
y[1] = sternoclavicular_r3
y[2] = unrothum_r3
y[3] = unrothum_r2
y[4] = acromioclavicular_r2
y[5] = acromioclavicular_r3
y[6] = acromioclavicular_r1
y[7] = unrothum_r1
y[8] = unrothum_r3
y[9] = unrothum_r2
y[10] = elv_angle
y[11] = shoulder_elv
y[12] = shoulder1_r2
y[13] = shoulder_rot
```

y[14] = con_DELT2tx
y[15] = con_DELT2ty
y[16] = con_DELT2tz
y[17] = elbow_flexion
y[18] = pro_sup
y[19] = con_BICtx
y[20] = con_BICty
y[21] = con_BICtz
y[22] = deviation
y[23] = flexion
y[24] = wrist_hand_r1
y[25] = wrist_hand_r3
y[26] = sternoclavicular_r2_u
y[27] = sternoclavicular_r3_u
y[28] = unrotscap_r3_u
y[29] = unrotscap_r2_u
y[30] = acromioclavicular_r2_u
y[31] = acromioclavicular_r3_u
y[32] = acromioclavicular_r1_u
y[33] = unrothum_r1_u
y[34] = unrothum_r3_u
y[35] = unrothum_r2_u
y[36] = elv_angle_u
y[37] = shoulder_elv_u
y[38] = shoulder1_r2_u

y[39] = shoulder_rot_u
y[40] = con_DELT2tx_u
y[41] = con_DELT2ty_u
y[42] = con_DELT2tz_u
y[43] = elbow_flexion_u
y[44] = pro_sup_u
y[45] = con_BICtx_u
y[46] = con_BICty_u
y[47] = con_BICtz_u
y[48] = deviation_u
y[49] = flexion_u
y[50] = wrist_hand_r1_u
y[51] = wrist_hand_r3_u
y[52] = DELT1.activation
y[53] = DELT1.fiber_length
y[54] = DELT2.activation
y[55] = DELT2.fiber_length
y[56] = DELT3.activation
y[57] = DELT3.fiber_length
y[58] = SUPSP.activation
y[59] = SUPSP.fiber_length
y[60] = INFSP.activation
y[61] = INFSP.fiber_length
y[62] = SUBSC.activation
y[63] = SUBSC.fiber_length

y[64] = TMIN.activation
y[65] = TMIN.fiber_length
y[66] = TMAJ.activation
y[67] = TMAJ.fiber_length
y[68] = PECM1.activation
y[69] = PECM1.fiber_length
y[70] = PECM2.activation
y[71] = PECM2.fiber_length
y[72] = PECM3.activation
y[73] = PECM3.fiber_length
y[74] = LAT1.activation
y[75] = LAT1.fiber_length
y[76] = LAT2.activation
y[77] = LAT2.fiber_length
y[78] = LAT3.activation
y[79] = LAT3.fiber_length
y[80] = CORB.activation
y[81] = CORB.fiber_length
y[82] = TRIlong.activation
y[83] = TRIlong.fiber_length
y[84] = TRIlat.activation
y[85] = TRIlat.fiber_length
y[86] = TRImed.activation
y[87] = TRImed.fiber_length
y[88] = ANC.activation

y[89] = ANC.fiber_length
y[90] = SUP.activation
y[91] = SUP.fiber_length
y[92] = BIClong.activation
y[93] = BIClong.fiber_length
y[94] = BICshort.activation
y[95] = BICshort.fiber_length
y[96] = BRA.activation
y[97] = BRA.fiber_length
y[98] = BRD.activation
y[99] = BRD.fiber_length
y[100] = ECRL.activation
y[101] = ECRL.fiber_length
y[102] = ECRB.activation
y[103] = ECRB.fiber_length
y[104] = ECU.activation
y[105] = ECU.fiber_length
y[106] = FCR.activation
y[107] = FCR.fiber_length
y[108] = FCU.activation
y[109] = FCU.fiber_length
y[110] = PL.activation
y[111] = PL.fiber_length
y[112] = PT.activation
y[113] = PT.fiber_length

y[114] = PQ.activation
y[115] = PQ.fiber_length
y[116] = FDSL.activation
y[117] = FDSL.fiber_length
y[118] = FDSR.activation
y[119] = FDSR.fiber_length
y[120] = FDSM.activation
y[121] = FDSM.fiber_length
y[122] = FDSI.activation
y[123] = FDSI.fiber_length
y[124] = FDPL.activation
y[125] = FDPL.fiber_length
y[126] = FDPR.activation
y[127] = FDPR.fiber_length
y[128] = FDPm.activation
y[129] = FDPm.fiber_length
y[130] = FDPI.activation
y[131] = FDPI.fiber_length
y[132] = EDCL.activation
y[133] = EDCL.fiber_length
y[134] = EDCR.activation
y[135] = EDCR.fiber_length
y[136] = EDCM.activation
y[137] = EDCM.fiber_length
y[138] = EDCI.activation

```
y[139] = EDCI.fiber_length
y[140] = EDM.activation
y[141] = EDM.fiber_length
y[142] = EIP.activation
y[143] = EIP.fiber_length
y[144] = EPL.activation
y[145] = EPL.fiber_length
y[146] = EPB.activation
y[147] = EPB.fiber_length
y[148] = FPL.activation
y[149] = FPL.fiber_length
y[150] = APL.activation
y[151] = APL.fiber_length
```

Integrating from 0 to 0.1

Schutte1993Muscle_Deprecated::computeForce, muscle LAT2 force < 0 at time = 0

Schutte1993Muscle_Deprecated::computeForce, muscle LAT2 force < 0 at time =
0.0199591

Schutte1993Muscle_Deprecated::computeForce, muscle LAT2 force < 0 at time =
0.0199711

Schutte1993Muscle_Deprecated::computeForce, muscle TRIl long force < 0 at time = 0

Schutte1993Muscle_Deprecated::computeForce, muscle TMAJ force < 0 at time =
0.0258274

Schutte1993Muscle_Deprecated::computeForce, muscle LAT2 force < 0 at time =
0.025165

Schutte1993Muscle_Deprecated::computeForce, muscle TRIlat force < 0 at time = 0.0274836

Schutte1993Muscle_Deprecated::computeForce, muscle TMAJ force < 0 at time = 0.0278582

Schutte1993Muscle_Deprecated::computeForce, muscle LAT2 force < 0 at time = 0.0285488

Schutte1993Muscle_Deprecated::computeForce, muscle LAT2 force < 0 at time = 0

Schutte1993Muscle_Deprecated::computeForce, muscle TMAJ force < 0 at time = 0

ForwardTool::run() caught exception

SimTK Exception thrown at AbstractIntegratorRep.cpp:419:

Integrator step failed at time 0.0526389385369668 apparently because:

SimTK Exception thrown at AbstractIntegratorRep.cpp:531:

Error detected by Simbody method AbstractIntegrator::takeOneStep(): Unable to advance time past 0.0526389.

(Required condition 't1 > t0' was not met.)

SIMULATION TWO

Running tool MoBL_ARMS_Upper_Limb_Model_OpenSim.

No external loads will be applied (external loads file not specified).

MODEL: MoBL_ARMS_Upper_Limb_Model_OpenSim

ANALYSES (2)

analysis[0] = Un-named analysis.

analysis[1] = Un-named analysis.

BODIES (14)

body[0] = ground (mass: 0) (inertia: 1 0 0 0 1 0 0 0 1)

body[1] = clavicle (mass: 0.156) (inertia: 0.00024259 -1.898e-005 -6.994e-005 -1.898e-005 0.00025526 5.371e-005 -6.994e-005 5.371e-005 4.442e-005)

body[2] = clavphant (mass: 0.0001) (inertia: 1 0 0 0 1 0 0 0 1)

body[3] = scapula (mass: 0.70396) (inertia: 0.0012429 0.0004494 0.00040922 0.0004494 0.0011504 0.0002411 0.00040922 0.0002411 0.0013651)

body[4] = scapphant (mass: 0.0001) (inertia: 1 0 0 0 1 0 0 0 1)

body[5] = humphant (mass: 0.0001) (inertia: 1 0 0 0 1 0 0 0 1)

body[6] = humphant1 (mass: 0.0001) (inertia: 1 0 0 0 1 0 0 0 1)

body[7] = humerus (mass: 1.99757) (inertia: 0.0122776 -0.00034741 -0.0002325 -0.00034741 0.00255133 0.0012293 -0.0002325 0.0012293 0.0125789)

body[8] = DELT2pt2a (mass: 0.0001) (inertia: 1e-005 0 0 0 1e-005 0 0 0 1e-005)

body[9] = ulna (mass: 1.1053) (inertia: 0.00541309 0.00031686 -7.615e-005 0.00031686 0.00115318 0.00109169 -7.615e-005 0.00109169 0.00494361)

body[10] = radius (mass: 0.23359) (inertia: 0.00043855 3.014e-005 -4.24e-006 3.014e-005 8.859e-005 6.418e-005 -4.24e-006 6.418e-005 0.00040258)

body[11] = BICpt (mass: 0.0001) (inertia: 1e-005 0 0 0 1e-005 0 0 0 1e-005)

body[12] = proximal_row (mass: 0.0001) (inertia: 1e-005 0 0 0 1e-005 0 0 0 1e-005)

body[13] = hand (mass: 0.5819) (inertia: 0.00011 9e-007 -2e-007 9e-007 6e-005 1.2e-005 -2e-007 1.2e-005 0.00015)

ACTUATORS (50)

actuator[0] = DELT1

actuator[1] = DELT2

actuator[2] = DELT3

actuator[3] = SUPSP

actuator[4] = INFSP

actuator[5] = SUBSC

actuator[6] = TMIN
actuator[7] = TMAJ
actuator[8] = PECM1
actuator[9] = PECM2
actuator[10] = PECM3
actuator[11] = LAT1
actuator[12] = LAT2
actuator[13] = LAT3
actuator[14] = CORB
actuator[15] = TRIlong
actuator[16] = TRIlat
actuator[17] = TRImed
actuator[18] = ANC
actuator[19] = SUP
actuator[20] = BIClong
actuator[21] = BICshort
actuator[22] = BRA
actuator[23] = BRD
actuator[24] = ECRL
actuator[25] = ECRB
actuator[26] = ECU
actuator[27] = FCR
actuator[28] = FCU
actuator[29] = PL
actuator[30] = PT

```
actuator[31] = PQ
actuator[32] = FDSL
actuator[33] = FDSR
actuator[34] = FDSM
actuator[35] = FDSI
actuator[36] = FDPL
actuator[37] = FDPR
actuator[38] = FDPM
actuator[39] = FDPI
actuator[40] = EDCL
actuator[41] = EDCR
actuator[42] = EDCM
actuator[43] = EDCI
actuator[44] = EDM
actuator[45] = EIP
actuator[46] = EPL
actuator[47] = EPB
actuator[48] = FPL
actuator[49] = APL
numStates = 152
numCoordinates = 26
numSpeeds = 26
numActuators = 50
numBodies = 14
numConstraints = 19
```

numProbes = 0

STATES (152)

y[0] = sternoclavicular_r2

y[1] = sternoclavicular_r3

y[2] = unrotscap_r3

y[3] = unrotscap_r2

y[4] = acromioclavicular_r2

y[5] = acromioclavicular_r3

y[6] = acromioclavicular_r1

y[7] = unrothum_r1

y[8] = unrothum_r3

y[9] = unrothum_r2

y[10] = elv_angle

y[11] = shoulder_elv

y[12] = shoulder1_r2

y[13] = shoulder_rot

y[14] = con_DELT2tx

y[15] = con_DELT2ty

y[16] = con_DELT2tz

y[17] = elbow_flexion

y[18] = pro_sup

y[19] = con_BICtx

y[20] = con_BICty

y[21] = con_BICtz

y[22] = deviation
y[23] = flexion
y[24] = wrist_hand_r1
y[25] = wrist_hand_r3
y[26] = sternoclavicular_r2_u
y[27] = sternoclavicular_r3_u
y[28] = unrothum_r3_u
y[29] = unrothum_r2_u
y[30] = acromioclavicular_r2_u
y[31] = acromioclavicular_r3_u
y[32] = acromioclavicular_r1_u
y[33] = unrothum_r1_u
y[34] = unrothum_r3_u
y[35] = unrothum_r2_u
y[36] = elv_angle_u
y[37] = shoulder_elv_u
y[38] = shoulder1_r2_u
y[39] = shoulder_rot_u
y[40] = con_DELT2tx_u
y[41] = con_DELT2ty_u
y[42] = con_DELT2tz_u
y[43] = elbow_flexion_u
y[44] = pro_sup_u
y[45] = con_BICtx_u
y[46] = con_BICty_u

y[47] = con_BICtz_u
y[48] = deviation_u
y[49] = flexion_u
y[50] = wrist_hand_r1_u
y[51] = wrist_hand_r3_u
y[52] = DELT1.activation
y[53] = DELT1.fiber_length
y[54] = DELT2.activation
y[55] = DELT2.fiber_length
y[56] = DELT3.activation
y[57] = DELT3.fiber_length
y[58] = SUPSP.activation
y[59] = SUPSP.fiber_length
y[60] = INFSP.activation
y[61] = INFSP.fiber_length
y[62] = SUBSC.activation
y[63] = SUBSC.fiber_length
y[64] = TMIN.activation
y[65] = TMIN.fiber_length
y[66] = TMAJ.activation
y[67] = TMAJ.fiber_length
y[68] = PECM1.activation
y[69] = PECM1.fiber_length
y[70] = PECM2.activation
y[71] = PECM2.fiber_length

y[72] = PECM3.activation
y[73] = PECM3.fiber_length
y[74] = LAT1.activation
y[75] = LAT1.fiber_length
y[76] = LAT2.activation
y[77] = LAT2.fiber_length
y[78] = LAT3.activation
y[79] = LAT3.fiber_length
y[80] = CORB.activation
y[81] = CORB.fiber_length
y[82] = TRIlong.activation
y[83] = TRIlong.fiber_length
y[84] = TRIlat.activation
y[85] = TRIlat.fiber_length
y[86] = TRImed.activation
y[87] = TRImed.fiber_length
y[88] = ANC.activation
y[89] = ANC.fiber_length
y[90] = SUP.activation
y[91] = SUP.fiber_length
y[92] = BIClong.activation
y[93] = BIClong.fiber_length
y[94] = BICshort.activation
y[95] = BICshort.fiber_length
y[96] = BRA.activation

y[97] = BRA.fiber_length
y[98] = BRD.activation
y[99] = BRD.fiber_length
y[100] = ECRL.activation
y[101] = ECRL.fiber_length
y[102] = ECRB.activation
y[103] = ECRB.fiber_length
y[104] = ECU.activation
y[105] = ECU.fiber_length
y[106] = FCR.activation
y[107] = FCR.fiber_length
y[108] = FCU.activation
y[109] = FCU.fiber_length
y[110] = PL.activation
y[111] = PL.fiber_length
y[112] = PT.activation
y[113] = PT.fiber_length
y[114] = PQ.activation
y[115] = PQ.fiber_length
y[116] = FDSL.activation
y[117] = FDSL.fiber_length
y[118] = FDSR.activation
y[119] = FDSR.fiber_length
y[120] = FDSM.activation
y[121] = FDSM.fiber_length

y[122] = FDSI.activation
y[123] = FDSI.fiber_length
y[124] = FDPL.activation
y[125] = FDPL.fiber_length
y[126] = FDPR.activation
y[127] = FDPR.fiber_length
y[128] = FDPM.activation
y[129] = FDPM.fiber_length
y[130] = FDPI.activation
y[131] = FDPI.fiber_length
y[132] = EDCL.activation
y[133] = EDCL.fiber_length
y[134] = EDCR.activation
y[135] = EDCR.fiber_length
y[136] = EDCM.activation
y[137] = EDCM.fiber_length
y[138] = EDCI.activation
y[139] = EDCI.fiber_length
y[140] = EDM.activation
y[141] = EDM.fiber_length
y[142] = EIP.activation
y[143] = EIP.fiber_length
y[144] = EPL.activation
y[145] = EPL.fiber_length
y[146] = EPB.activation

y[147] = EPB.fiber_length

y[148] = FPL.activation

y[149] = FPL.fiber_length

y[150] = APL.activation

y[151] = APL.fiber_length

Integrating from 0 to 0.1

Schutte1993Muscle_Deprecated::computeForce, muscle LAT2 force < 0 at time = 0

Schutte1993Muscle_Deprecated::computeForce, muscle EPB force < 0 at time =
0.0044435

Schutte1993Muscle_Deprecated::computeForce, muscle DELT2 force < 0 at time =
0.00469177

Schutte1993Muscle_Deprecated::computeForce, muscle TMAJ force < 0 at time =
0.00830076

Schutte1993Muscle_Deprecated::computeForce, muscle TRIlong force < 0 at time =
0.00830076

Schutte1993Muscle_Deprecated::computeForce, muscle LAT2 force < 0 at time = 0

Schutte1993Muscle_Deprecated::computeForce, muscle TRIlat force < 0 at time = 0

ForwardTool::run() caught exception

SimTK Exception thrown at AbstractIntegratorRep.cpp:419:

Integrator step failed at time 0.0491637889560721 apparently because:

bad allocation

Using states from input/states file

No external loads will be applied (external loads file not specified).

Executing the analyses from 0 to 0.053...

MuscleAnalysis: WARNING - coordinate elv_angle is locked and can't be varied.

Simulation three

Running tool MoBL_ARMS_Upper_Limb_Model_OpenSim.

No external loads will be applied (external loads file not specified).

MODEL: MoBL_ARMS_Upper_Limb_Model_OpenSim

ANALYSES (3)

analysis[0] = MuscleAnalysis

analysis[1] = Un-named analysis.

analysis[2] = Un-named analysis.

BODIES (14)

body[0] = ground (mass: 0) (inertia: 1 0 0 0 1 0 0 0 1)

body[1] = clavicle (mass: 0.156) (inertia: 0.00024259 -1.898e-005 -6.994e-005 -1.898e-005 0.00025526 5.371e-005 -6.994e-005 5.371e-005 4.442e-005)

body[2] = clavphant (mass: 0.0001) (inertia: 1 0 0 0 1 0 0 0 1)

body[3] = scapula (mass: 0.70396) (inertia: 0.0012429 0.0004494 0.00040922 0.0004494 0.0011504 0.0002411 0.00040922 0.0002411 0.0013651)

body[4] = scapphant (mass: 0.0001) (inertia: 1 0 0 0 1 0 0 0 1)

body[5] = humphant (mass: 0.0001) (inertia: 1 0 0 0 1 0 0 0 1)

body[6] = humphant1 (mass: 0.0001) (inertia: 1 0 0 0 1 0 0 0 1)

body[7] = humerus (mass: 1.99757) (inertia: 0.0122776 -0.00034741 -0.0002325 -0.00034741 0.00255133 0.0012293 -0.0002325 0.0012293 0.0125789)

body[8] = DELT2pt2a (mass: 0.0001) (inertia: 1e-005 0 0 0 1e-005 0 0 0 1e-005)

body[9] = ulna (mass: 1.1053) (inertia: 0.00541309 0.00031686 -7.615e-005 0.00031686 0.00115318 0.00109169 -7.615e-005 0.00109169 0.00494361)

body[10] = radius (mass: 0.23359) (inertia: 0.00043855 3.014e-005 -4.24e-006 3.014e-005 8.859e-005 6.418e-005 -4.24e-006 6.418e-005 0.00040258)

```

body[11] = BICpt (mass: 0.0001) (inertia: 1e-005 0 0 0 1e-005 0 0 0 1e-005)
body[12] = proximal_row (mass: 0.0001) (inertia: 1e-005 0 0 0 1e-005 0 0 0 1e-005)
body[13] = hand (mass: 0.5819) (inertia: 0.00011 9e-007 -2e-007 9e-007 6e-005 1.2e-005
-2e-007 1.2e-005 0.00015)

```

ACTUATORS (50)

```

actuator[0] = DELT1
actuator[1] = DELT2
actuator[2] = DELT3
actuator[3] = SUPSP
actuator[4] = INFSP
actuator[5] = SUBSC
actuator[6] = TMIN
actuator[7] = TMAJ
actuator[8] = PECM1
actuator[9] = PECM2
actuator[10] = PECM3
actuator[11] = LAT1
actuator[12] = LAT2
actuator[13] = LAT3
actuator[14] = CORB
actuator[15] = TRIlong
actuator[16] = TRIlnt
actuator[17] = TRImed
actuator[18] = ANC

```


actuator[19] = SUP
actuator[20] = BIClong
actuator[21] = BICshort
actuator[22] = BRA
actuator[23] = BRD
actuator[24] = ECRL
actuator[25] = ECRB
actuator[26] = ECU
actuator[27] = FCR
actuator[28] = FCU
actuator[29] = PL
actuator[30] = PT
actuator[31] = PQ
actuator[32] = FDSL
actuator[33] = FDSR
actuator[34] = FDSM
actuator[35] = FDSI
actuator[36] = FDPL
actuator[37] = FDPR
actuator[38] = FDPM
actuator[39] = FDPI
actuator[40] = EDCL
actuator[41] = EDCR
actuator[42] = EDCM
actuator[43] = EDCI

```
actuator[44] = EDM
actuator[45] = EIP
actuator[46] = EPL
actuator[47] = EPB
actuator[48] = FPL
actuator[49] = APL

numStates = 152
numCoordinates = 26
numSpeeds = 26
numActuators = 50
numBodies = 14
numConstraints = 19
numProbes = 0
```

STATES (152)

```
y[0] = sternoclavicular_r2
y[1] = sternoclavicular_r3
y[2] = unrothum_r3
y[3] = unrothum_r2
y[4] = acromioclavicular_r2
y[5] = acromioclavicular_r3
y[6] = acromioclavicular_r1
y[7] = unrothum_r1
y[8] = unrothum_r3
y[9] = unrothum_r2
```

y[10] = elv_angle
y[11] = shoulder_elv
y[12] = shoulder1_r2
y[13] = shoulder_rot
y[14] = con_DELT2tx
y[15] = con_DELT2ty
y[16] = con_DELT2tz
y[17] = elbow_flexion
y[18] = pro_sup
y[19] = con_BICtx
y[20] = con_BICty
y[21] = con_BICtz
y[22] = deviation
y[23] = flexion
y[24] = wrist_hand_r1
y[25] = wrist_hand_r3
y[26] = sternoclavicular_r2_u
y[27] = sternoclavicular_r3_u
y[28] = unrotscap_r3_u
y[29] = unrotscap_r2_u
y[30] = acromioclavicular_r2_u
y[31] = acromioclavicular_r3_u
y[32] = acromioclavicular_r1_u
y[33] = unrothum_r1_u
y[34] = unrothum_r3_u

y[35] = unrothum_r2_u
y[36] = elv_angle_u
y[37] = shoulder_elv_u
y[38] = shoulder1_r2_u
y[39] = shoulder_rot_u
y[40] = con_DELT2tx_u
y[41] = con_DELT2ty_u
y[42] = con_DELT2tz_u
y[43] = elbow_flexion_u
y[44] = pro_sup_u
y[45] = con_BICtx_u
y[46] = con_BICty_u
y[47] = con_BICtz_u
y[48] = deviation_u
y[49] = flexion_u
y[50] = wrist_hand_r1_u
y[51] = wrist_hand_r3_u
y[52] = DELT1.activation
y[53] = DELT1.fiber_length
y[54] = DELT2.activation
y[55] = DELT2.fiber_length
y[56] = DELT3.activation
y[57] = DELT3.fiber_length
y[58] = SUPSP.activation
y[59] = SUPSP.fiber_length

y[60] = INFSP.activation
y[61] = INFSP.fiber_length
y[62] = SUBSC.activation
y[63] = SUBSC.fiber_length
y[64] = TMIN.activation
y[65] = TMIN.fiber_length
y[66] = TMAJ.activation
y[67] = TMAJ.fiber_length
y[68] = PECM1.activation
y[69] = PECM1.fiber_length
y[70] = PECM2.activation
y[71] = PECM2.fiber_length
y[72] = PECM3.activation
y[73] = PECM3.fiber_length
y[74] = LAT1.activation
y[75] = LAT1.fiber_length
y[76] = LAT2.activation
y[77] = LAT2.fiber_length
y[78] = LAT3.activation
y[79] = LAT3.fiber_length
y[80] = CORB.activation
y[81] = CORB.fiber_length
y[82] = TRIlong.activation
y[83] = TRIlong.fiber_length
y[84] = TRIlat.activation

y[85] = TRIl.at.fiber_length
y[86] = TRImed.activation
y[87] = TRImed.fiber_length
y[88] = ANC.activation
y[89] = ANC.fiber_length
y[90] = SUP.activation
y[91] = SUP.fiber_length
y[92] = BIClong.activation
y[93] = BIClong.fiber_length
y[94] = BICshort.activation
y[95] = BICshort.fiber_length
y[96] = BRA.activation
y[97] = BRA.fiber_length
y[98] = BRD.activation
y[99] = BRD.fiber_length
y[100] = ECRL.activation
y[101] = ECRL.fiber_length
y[102] = ECRB.activation
y[103] = ECRB.fiber_length
y[104] = ECU.activation
y[105] = ECU.fiber_length
y[106] = FCR.activation
y[107] = FCR.fiber_length
y[108] = FCU.activation
y[109] = FCU.fiber_length

y[110] = PL.activation
y[111] = PL.fiber_length
y[112] = PT.activation
y[113] = PT.fiber_length
y[114] = PQ.activation
y[115] = PQ.fiber_length
y[116] = FDSL.activation
y[117] = FDSL.fiber_length
y[118] = FDSR.activation
y[119] = FDSR.fiber_length
y[120] = FDSM.activation
y[121] = FDSM.fiber_length
y[122] = FDSI.activation
y[123] = FDSI.fiber_length
y[124] = FDPL.activation
y[125] = FDPL.fiber_length
y[126] = FDPR.activation
y[127] = FDPR.fiber_length
y[128] = FDPM.activation
y[129] = FDPM.fiber_length
y[130] = FDPI.activation
y[131] = FDPI.fiber_length
y[132] = EDCL.activation
y[133] = EDCL.fiber_length
y[134] = EDCR.activation

y[135] = EDCR.fiber_length
y[136] = EDCM.activation
y[137] = EDCM.fiber_length
y[138] = EDCI.activation
y[139] = EDCI.fiber_length
y[140] = EDM.activation
y[141] = EDM.fiber_length
y[142] = EIP.activation
y[143] = EIP.fiber_length
y[144] = EPL.activation
y[145] = EPL.fiber_length
y[146] = EPB.activation
y[147] = EPB.fiber_length
y[148] = FPL.activation
y[149] = FPL.fiber_length
y[150] = APL.activation
y[151] = APL.fiber_length

Integrating from 0 to 0.1

Schutte1993Muscle_Deprecated::computeForce, muscle LAT2 force < 0 at time = 0.05

Schutte1993Muscle_Deprecated::computeForce, muscle SUP force < 0 at time = 0.05

Schutte1993Muscle_Deprecated::computeForce, muscle LAT2 force < 0 at time = 0.05

Schutte1993Muscle_Deprecated::computeForce, muscle EPB force < 0 at time =
0.0194975

Schutte1993Muscle_Deprecated::computeForce, muscle DELT2 force < 0 at time = 0.0199367

Schutte1993Muscle_Deprecated::computeForce, muscle TRIl at force < 0 at time = 0.0198879

Schutte1993Muscle_Deprecated::computeForce, muscle TRImed force < 0 at time = 0.0198879

Schutte1993Muscle_Deprecated::computeForce, muscle DELT2 force < 0 at time = 0.0203174

Schutte1993Muscle_Deprecated::computeForce, muscle CORB force < 0 at time = 0.0229717

Schutte1993Muscle_Deprecated::computeForce, muscle TRIl long force < 0 at time = 0.0229717

Schutte1993Muscle_Deprecated::computeForce, muscle TRIl at force < 0 at time = 0.0229717

Schutte1993Muscle_Deprecated::computeForce, muscle TRImed force < 0 at time = 0.0229717

Schutte1993Muscle_Deprecated::computeForce, muscle DELT2 force < 0 at time = 0.0233435

Schutte1993Muscle_Deprecated::computeForce, muscle FDPR force < 0 at time = 0.05

Schutte1993Muscle_Deprecated::computeForce, muscle LAT1 force < 0 at time = 0.05

Schutte1993Muscle_Deprecated::computeForce, muscle EPB force < 0 at time = 0.05

Schutte1993Muscle_Deprecated::computeForce, muscle EPB force < 0 at time = 0.0415332

Schutte1993Muscle_Deprecated::computeForce, muscle DELT2 force < 0 at time = 0.0419859

Schutte1993Muscle_Deprecated::computeForce, muscle TMAJ force < 0 at time = 0.05

Schutte1993Muscle_Deprecated::computeForce, muscle LAT2 force < 0 at time = 0.05

Schutte1993Muscle_Deprecated::computeForce, muscle EDCM force < 0 at time = 0.0473898

Schutte1993Muscle_Deprecated::computeForce, muscle DELT2 force < 0 at time = 0.04792

Schutte1993Muscle_Deprecated::computeForce, muscle LAT2 force < 0 at time = 0.0561219

Schutte1993Muscle_Deprecated::computeForce, muscle TRIl at force < 0 at time = 0.0561219

Schutte1993Muscle_Deprecated::computeForce, muscle TMAJ force < 0 at time = 0.05

Schutte1993Muscle_Deprecated::computeForce, muscle LAT2 force < 0 at time = 0.0677118

Schutte1993Muscle_Deprecated::computeForce, muscle DELT2 force < 0 at time = 0.068278

Schutte1993Muscle_Deprecated::computeForce, muscle TRImed force < 0 at time = 0.05

Schutte1993Muscle_Deprecated::computeForce, muscle LAT2 force < 0 at time = 0.05

Schutte1993Muscle_Deprecated::computeForce, muscle LAT2 force < 0 at time = 0.0811142

Schutte1993Muscle_Deprecated::computeForce, muscle TRImed force < 0 at time = 0.0811142

Schutte1993Muscle_Deprecated::computeForce, muscle LAT2 force < 0 at time = 0.05

Schutte1993Muscle_Deprecated::computeForce, muscle LAT2 force < 0 at time = 0.0877394

Rough labelled description of what each functions do:

