UNIVERSITY OF CALIFORNIA
SANTA CRUZ

OBJECTIVELY CHARACTERIZED LINEAR MODEL OF STROKE INDUCED JOINT SYNERGIES IN RELATION TO CLINICAL MEASURES

A thesis submitted in partial satisfaction of the
requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER ENGINEERING

by

Aimen Hamid Al-Refai

September 2012

The Thesis of Aimen Hamid Al-Refai
is approved:

Professor Jacob Rosen, Chair

Professor Gabrial Elkaim

Professor Don Wiberg

Tyrus Miller
Vice Provost and Dean of Graduate Studies

Table of Contents

List of Figures v

List of Tables vii

Abstract viii

Dedication x

Acknowledgments xi

I 1

1 Introduction 2

2 Previous Work 6

3 The Model 15

II 20

4 Experimental Setup 21
- 4.1 Subjects . . . 21
- 4.2 System . . . 23
  - 4.2.1 Hardware . . . 23
  - 4.2.2 Software . . . 24
- 4.3 Protocol . . . 25

5 Results 29
- 5.0.1 SM Score vs. MFM Score . . . 30
- 5.0.2 Synergy Matrix Data Density . . . 31
- 5.0.3 Sensitivity . . . 34

III **39**

6 Analysis and Discussion **40**

7 Conclusion **49**

A Code **51**
A.1 Bodybuilder Code 51
A.2 MATLAB Code 67

B Phone Screening Worksheet **94**

C Protocol **96**

D MFM Worksheet **110**

E MFM and SMS Data **112**

Bibliography **114**

List of Figures

2.1 Experimental setup for stroke survivor with right-side hemiparesis. FT sensors at fingers, arm, trunk, bottom of seat, and feet. 7
2.2 Wearable RUPERT exoskeleton for arm with hemiparesis. 8
2.3 (a) The speech interface. (b) The speech transcriptions manager for manual corrections. 9
2.4 The left hand is moved on the lower surface by the examiner to one of the four target locations, according to a visual image on the computer screen. The subject is asked to move the tested right hand to a point just above the left hand. Data is captured of the proprioceptive positioning of the hands. 10
2.5 Configuration of the Stroke-Upper Limb Activity Monitor (A) The electrogoniometer positions (B) The accelerometer position 11
2.6 The motion suit worn by the subjects consists of the inertial measurement units (IMU) and a wearable computer. In the experiments, only one IMU is used. . . 12
2.7 The sensor setup and orientation of the axes of the accelerometers. 13

3.1 The diagonal has a 45 degree straight line representing theta equal to the same theta, i.e. a slope of positive one. 19

4.1 Experimental setup for stroke survivor with right-side hemiparesis. (a) The infra-red reflective maker. (b) The VICON camera. Notice the infra-red broadcast rings around the lens apeture. (c) The subject, the data collection volume, and one of the researchers. 24
4.2 Z is the axis of rotation. Z_1 is shoulder inner/outer rotation, Z_2 is shoulder extension/flexion, Z_3 is shoulder abd/adduction, Z_4 is elbow flexion/extension, Z_5 is wrist extension/flexion, Z_6 is wrist radial/ulnar deviation, Z_7 is wrist supination/pronation 25

4.3 (a) 1=jugular notch where clavicles meet the sternum, 2=xiphoid process of the sternum, 3=acromio-clavicular joint, 4=lateral lower belly of the deltoid (asymmetric with 5 and 6), 5=distal end of triceps outer head (asymmetric with 4 and 6), 6=distal end of the long head of the bicep brachii (asymmetric with 4 and 5), 7=lateral humerus epicondyle approximating the elbow joint axis, 8=insertion point of the anconeus, 9=thumb side of the radial styloid (symmetrical with 10 to form an axis through center of wrist), 10=little finger side of ulnar styloid (symmetrical with 9 to form an axis through center of wrist) (b)A=7th cervical vertebra, B=10th thoracic vertebra, C=medial humerus epicondyle approximating the elbow joint axis 26
4.4 Experimental setup for stroke survivor with left-side hemiparesis. Pictured in (a) is the standard start position. Pictured in (b) is the subject-specific arm model. 26
4.5 All image sequences show start and stop positions with actions in-between. a, c, e, g, i, k, m are synergy matrix motions while b, d, f, h, j, l, n are Fugl-Meyer assessment actions. The synergy experiment motions were developed to best match the MFM motions selected from the full FM assessment to ensure valid comparison between SMS and FM scores. 28

5.1 Y-axis is the SM scores. Subjects 6, 13, 23 have standout data issues. Subsequently, Subject 6 was left out of calculations. 30
5.2 Equivalence between healthy SM mean score and MFM mean score. Image produced by stattools.net. 32
5.3 Nonparameter analysis with Wilcoxon signed rank test on healthy SM mean score and MFM mean score. Image produced by stattools.net. 32
5.4 ANOVA box plots. (a) The SM stroke means compared to MFM stroke mean. (b) The SM healthy means compared to MFM healthy mean. 33
5.5 Stroke SM score and MFM score ICC comparison. Image produced by stattools.net. 34
5.6 (a) The ideal input with max range per joint and no synergies. (b) Averaging all the healthy subjects into one gold standard SM times the max range per joint. . 35
5.7 (a) Stroke subject 16's SM. (b) The gold standard SM with the scale adjusted to match stroke subject 16. 36
5.8 Subject 16's residuals show strong synergies in joints 1, 3, 5, 7. 37
5.9 (a) An increase in synergies is detected. (b) A refinement of synergies is detected. 38

6.1 Scoring positions (a) = 0 (b) = 1(c) = 1 or 2? Depends. 44
6.2 (a) Show subject 7's residuals. (b) Shows subject 22 residuals. $(b) - (a) = (c)$ the residuals of b-a 47
6.3 This graph displays the residuals of subjects' SM minus the gold standard healthy SM. The resulting residuals show healthy to deleterious synergies. Subjects 1, 3, 7, 22 are close on the boarder. Meaning the synergies show a restrictive by marginally healthy movement. Closer inspection of the actual synergies are needed. 48

List of Tables

4.1 Biometric Data of Experimental Subjects ($n = 22$) 23

5.1 Descriptive Statistics ($n = 22$) 30
5.2 Joint Movements 37

6.1 FM Movements 43

Abstract

Objectively Characterized Linear Model of Stroke Induced Joint Synergies in Relation to Clinical Measures

by

Aimen Hamid Al-Refai

Subjective kinematic motion analysis done by a trained physical therapist using standard assessment such as the Fugl-Meyer assessment (FM) and/or the Wolf Motor Function test (Wolf), have a limited ability to objectively characterize post stroke movement of subjects with a hemiparetic limb or demonstrate fine change over time. The goal of the research was to take the characterized synergies linear matrix model and find an application with stroke survivors. Twenty-two participants were tested for motor impairment using a modified FM (MFM) assessment. Motion capture data was collected using the Vicon motion capture system. This data was processed using Vicon BodyBuilder and MATLAB to construct a stroke subject's Synergy Matrix (SM) from which was developed the Synergy Matrix Score (SMS). This score is compared with the MFM score in terms of efficacy of accepted assessment standards, precise quantitative data, sufficient sensitivity to characterize change over time, and usefulness for clinical reporting. The results show that the SMS provides comparable accurate assessment of the subjects post stroke motion with a high level of sensitivity and greater description of shoulder, elbow, and wrist information in the form of a seven by seven matrix. The SM scoring scale $(0,\infty)$ also has the ability to quantify change over time at a finer resolution than the MFM range of 0-2 (33% banding) or the Wolf range of 0-4 (25% banding). Results also show that the SMS objectively generates

a relevant score comparable to the MFM while maintaining a greater resolution of detail and density of information allowing the physical therapist to better target therapies as well as spend more time working with patients and less time spent on documentation. Adding the SMS to a control system on a therapeutic robot will further automate the SMS into a realtime feedback loop as applied, for example, to stroke prognosis of the targeted therapy.

To my wife and daughter,

Darcy and Yasmine,

Thank you for your patience and understanding.

I love you very much.

Acknowledgments

Thank you to all the stroke survivors for their altruistic participation, Matt Simkins for his synergistic collaboration, my committee, and the undergrad crew: Arial, Zach, Aigee, Farhad, Calvin, and Jenny. A special thank you to Professor Nancy Byl and Professor Jacob Rosen for their mentorship.

Part I

Chapter 1

Introduction

Stroke is a common disabling condition that affects approximately 800,000 individuals per year in the U. S. alone, with around 400,000 affected individuals surviving with some form of motor and/or cognitive deficit [2]. This is the leading cause of disability in the U. S. [20]. A motor deficit impairs the ability to move the stroke-affected limb or side, otherwise known as hemiparesis. Due to this limitation, survivors experience, diminished coordination, decreased autonomy, and decreased quality of life [25]. An aspect of diminished coordination is deleterious joint synergies [4]. "Joint synergies" is synonymous with "muscle synergies" that are the result of stroke damage in humans. "Joint synergy" has been used in the literature as having both positive connotations and negative connotations on motor function. A "efficacious" synergy implies that a healthy individual can move multiple joints in a coordinated way such that complex motion, such as running or throwing, is performed in a fluid and precise way. A "deleterious" synergy implies that the individual performs involuntary motions that hinder fluid and precise movements. Upper-limb synergies have certain stereotypical synergies that affect

the movements of the trunk, scapula, shoulder, elbow, wrist, and hand [4]. This study considers only the shoulder, elbow, and wrist joints. Thus, the human arm is modeled as a seven degrees of freedom (DOF) manipulator.

It is known that, by working intensely with a trained clinician, those suffering from hemiparesis following stroke can often regain partial use of the affected limb [5]. Suitable assessment tools were developed for clinicians to quantify the level of hemiparesis and the efficacy of rehabilitation. Two such tools are the Fugl-Meyer Upper Extremity Assessment (FM) [14] and the Wolf Motor Function Test (Wolf) [35], both are considered the gold standard of post stroke assessment tools. However, both Wolf and FM have limited objective ability to characterize movement [22, 34] because, as assessment measures, they are used in intervention research and are focused on task completion or clinician ratings of movement rather than objective models of stroke survivor movement deficits. Thus Wolf and FM results are limited in specific, precise, and quantitative data that effectively distinguishes remediation of deficits versus the development of compensatory movement therapies. The functional significance of a stroke survivor's ability to complete meaningful tasks should not be undermined, yet these types of outcome measures do not provide information regarding specific movement capabilities or provide targeted therapeutic goals [34].

Wolf is a common task-based outcome assessment tool that has become one standard measure in research investigations of upper-extremity rehabilitation interventions such as constraint-induced therapy (CIT) [23]. Wolf incorporates gross- and fine-motor components for all joints in a variety of functional tasks such as reaching for a can, picking up a pencil, or folding a towel. The instructions for each task emphasize speed of completion and all tasks are

videotaped for subsequent rating of the stroke survivor's functional ability. Functional ability is rated on a 5-point ordinal scale that incorporates task completion and generalizations regarding movements made in synergy. Wolf also includes two strength measures but these are reported less in the scientific literature. Wolf has established reliability as a stroke assessment and research tool [31, 19, 10, 13].

FM is another common assessment tool used in stroke rehabilitation. In addition to evaluating some basic movement tasks or task components (e.g., gripping a can or ball, holding a pencil with a two-point pinch), the FM assessment also evaluates more basic movement capacities foundational to task performance on a 3-point ordinal scale. For example, subjects are instructed to produce isolated shoulder movements while maintaining elbow extension during which an evaluator rates movement capacity (See Fig. 4.5). Other scored criteria include the presence of reflexes, tremor, dysmetria, and speed of movement. FM has established validity and reliability as a research tool [11, 27, 29]. Together, Wolf and FM assessments provide valuable information regarding motor performance and motor impairment after stroke, yet they do not objectively yield precise quantitative data on movement synergies and lack sufficient sensitivity to characterize changes over time as stroke survivors progress with therapy.

The purpose of this study is to quantify functionality of a stroke survivor such that the outcome measures provide an accurate description of the stroke survivor's movement synergies, provide information to the physical therapists to assist in designing an effective therapy protocol, and show sufficient sensitivity to characterize changes over time as the stroke survivors are reassessed as their therapy progresses by using the Synergy Matrix (SM).

SM is a seven by seven covariance matrix that characterizes the synergies of a subject,

provides depth of information about the subject's motion synergies, and has sufficient granularity to display changes in a subject's progress over time. An SM score (SMS) was developed to corroborate efficacy compared to the FM score. The SMS will describe comparable information to the FM or Wolf assessments while providing a method of tracking change over time. Eventually the SM and SMS will be applied to a robot exoskeleton for stroke rehabilitation while providing real time feedback of stroke rehabilitation prognosis of the targeted therapy.

Chapter 2

Previous Work

Telemedicine uses telecommunication, and information gathering technologies to provide clinical healthcare at a distance. An important part of telemedicine is the use of advanced diagnostic methods. These diagnostic methods include use of telemedical devices such as robotic exoskeletons, pacemakers, and insulin pumps. The dominant approach to automation of assessments has been to use telemedical devices to quantitatively measure a physiological output and to determine its relationship to functional ability. Mazzoleni et al. developed a seated robotic apparatus through an iterative design process with clinicians, stroke patients, and engineers [24]. The robotic apparatus used eight 6-axis force and torque (FT) sensors designed to measure FT applied by the subject during the performance of six activities of daily living (ADL) [18]. The system restricts subjects to isometric movements and makes use of forward kinematic models to determine subject-generated FT. Mazzoleni et al. found that the novel FT assessment tool to be a good comparison to the FM scores when analyzing the FM hand and hand-grasping motions subsections. Mazzoleni's research did not discuss how the support of

the rigid infrastructure would change the results with the stroke subjects. Since the research was isometric, there was also no focus on arm or joint movement as the research platform in Fig. 2.1 suggests.

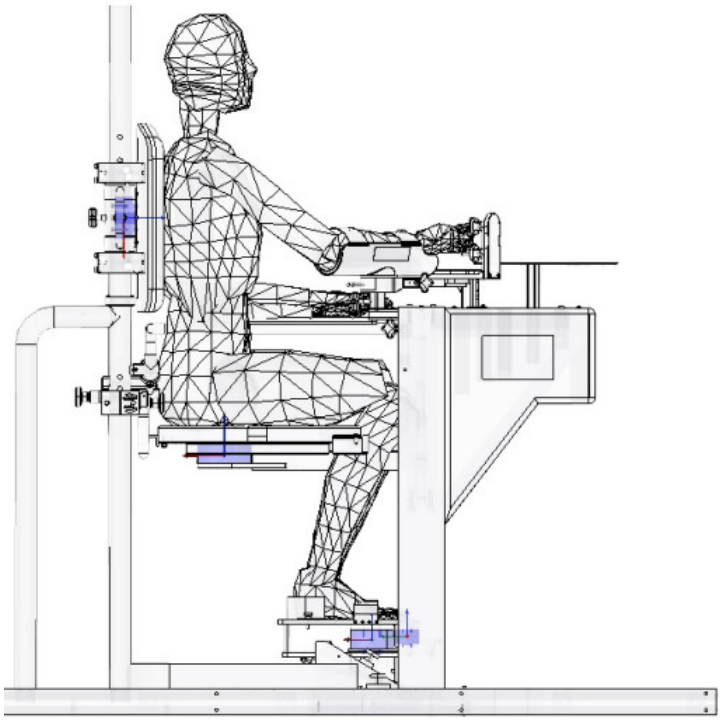


Figure 2.1: Experimental setup for stroke survivor with right-side hemiparesis. FT sensors at fingers, arm, trunk, bottom of seat, and feet.

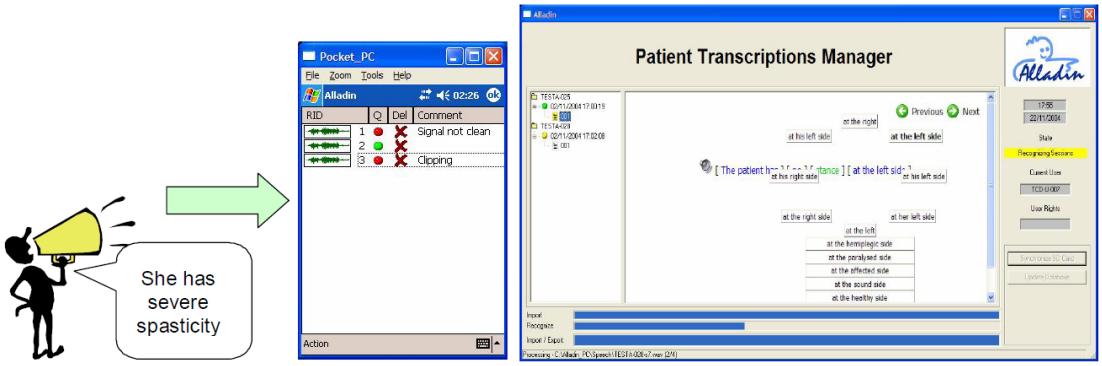
Balasubramanian et al. developed a 5-DOF wearable upper-extremity rehabilitation exoskeleton robot called RUPERT to provide clinical assessment measures. This exoskeleton was designed to be more sensitive than many clinical measures which use ordinal scales for scoring and to be used to track a subject's recovery over time [3]. RUPERT provides movement kinematic information in the form of joint angles, force and torque from the pressure exerted by the subject wearing the device on RUPERT's pneumatic actuators. From this information

performance metrics such as amount of assistance, motion smoothness, and movement synergy were measured and discussed. Balasubramanian et al. found that their newly proposed smoothness metric was a good way to describe movement synergies. The smoother the movement, the fewer the number of coupled sub movements or deleterious synergies, however, their research did not discuss which synergies increased or decreased smoothness, nor provide a comparison with existing stroke assessment tools like the Wolf or FM scores. Fig. 2.2 shows the RUPERT exoskeleton system. Cinkelj and Van Dijck furthered Mazzolenis work by describing the RU-

Figure 2.2: Wearable RUPERT exoskeleton for arm with hemiparesis.

PERT exoskeletons software and by evaluating posterior probability profiles of the RUPERT system [6, 9]. The software required a clinician to close the loop by adding clinical analysis of the subject's movement during the assessment. The software would automatically convert the clinician's comments by converting the clinician's speech, looking for FM keywords, and matching the clinician generated FM score to the task that the clinician asked the subject to perform. Cinkelj and Van Dijck were able to show correlations between the quantitative FT

measures and a modified FM score while using speech recognition to do the clinician generated stroke assessment documentation. Cinkelj and Van Dijck's research doesn't discuss how efficacious to the modified FM score was the total system nor did they discuss the clinician's time commitment to documentation while running the complete system. Fig. 2.3 shows the RUPERT system speech and graphical user interface (GUI). Van Dijck et al's work only focused on acute to 180 days after stroke but did show, using class posterior probabilities and dimentionality reduction, that the RUPERT generated data is efficacious when compared with FM scores. Cinkelj and Van Dijck's research also supports the idea of automated assessment of data collected from stroke subjects and its use in providing effective, timely feed back to the clinician during stroke treatment/therapy sessions.



(a) (b)

Figure 2.3: (a) The speech interface. (b) The speech transcriptions manager for manual corrections.

An alternative approach to robot-aided assessments is to use simpler off-the-shelf, sensor-based systems. One feature that is assessed in post-stroke subjects is their relative levels of proprioception in the affected limb. Leibowitz et al. focused on this issue by developing a

system where 5mm transmitters are worn on the hands in a 10 x 10 x 10 cm magnetic field. Sensors would track the transmitted three dimentional position (x, y, z) as the subject moved their hands to target locations as requested by the researcher and indicated by a screen [21]. This system measured proprioception with more accuracy and resolution than standard up/down tests [1, 8] currently administered in proprioception assessment. Leibowitz et al's research provides further evidence that simple technology can produce a novel automated approach for measuring upper-limb proprioception deficits following stroke as shown in Fig. 2.4

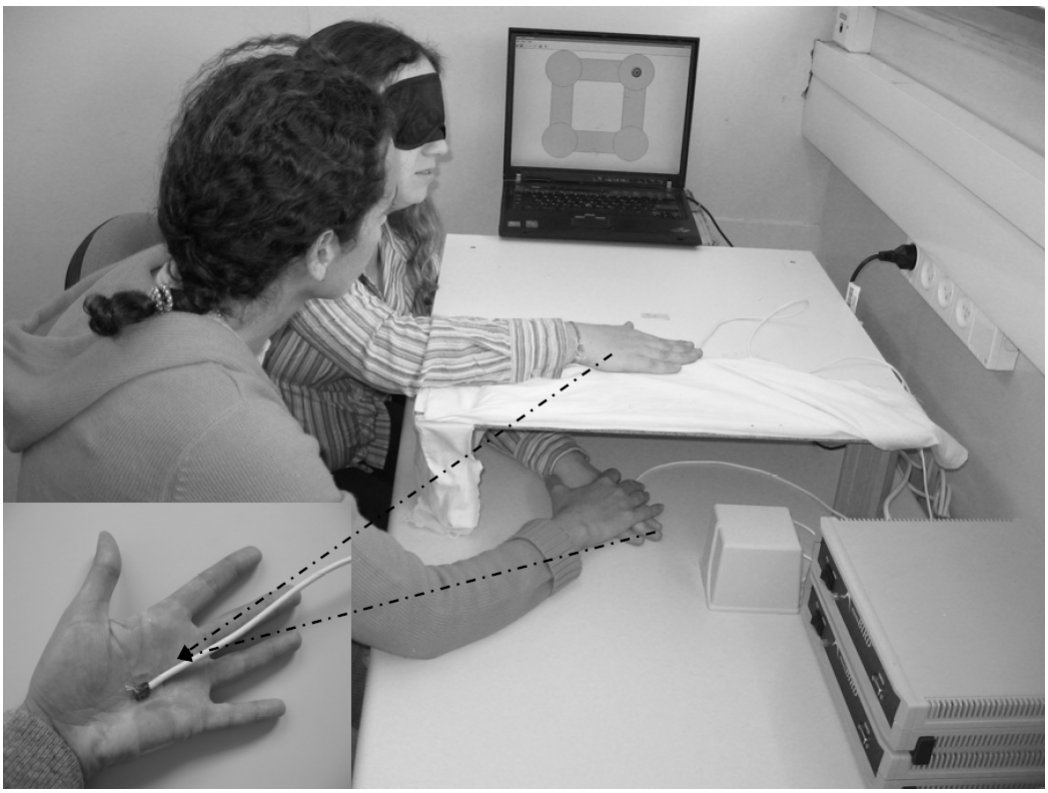


Figure 2.4: The left hand is moved on the lower surface by the examiner to one of the four target locations, according to a visual image on the computer screen. The subject is asked to move the tested right hand to a point just above the left hand. Data is captured of the proprioceptive positioning of the hands.

Bonato et al. affixed tri-axial accelerometers to a subjects arm [17] as shown in Fig. 2.5. The goal of their study was to determine if different functional impairment lev-

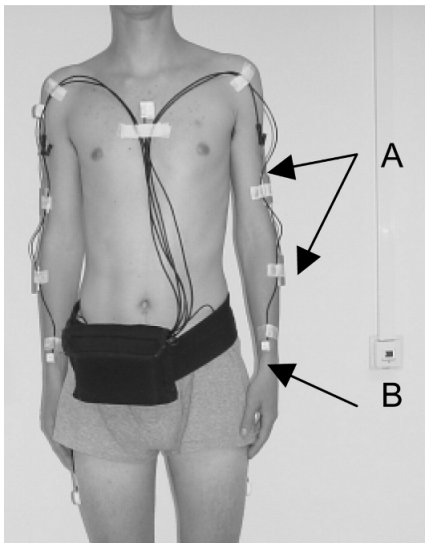


Figure 2.5: Configuration of the Stroke-Upper Limb Activity Monitor (A) The electrogoniometer positions (B) The accelerometer position

els in the characteristic motor patterns could be distinguished in the accelerometer data. The Bonato team's research gathered data using the FM and Wolf assessments and then compared accelerometer readings collected from the subjects with their standard FM and Wolf scores. The results show, that as a tool for measuring usage and movement, the accelerometers embedded in the system called Vitaport ambulatory digital recorder (Temec BV, The Netherlands) was successful with a significance of $p < 0.05$ when looking at the hand, forearm, and upper arm motions. Their research results also supported their goal to show that wearable sensors have the potential to capture characteristics of motor patterns associated with motor impairment and functional limitations. Bonato et al. analyzed their data using linear first, second, and third order root mean square (RMS) value of the accelerometer data. The results affirm that the linear

model RMS values are a valid tool to assess the post stroke physical impairments as compared to the standard Wolf and FM assessment scores. The resolution of the data was not discussed nor was the algorithm of scoring. The movement patterns discussed in the paper were patterns of linear graphs as was used in this paper to develop the SM.

The Wade et al. approach was to use off-the-shelf wearable sensors and performance sensors to estimate the timing of seven tasks from the Wolf assessment with the goal of automating post stroke assessment tests [33] as is shown in Fig. 2.6. Their results show that, once the

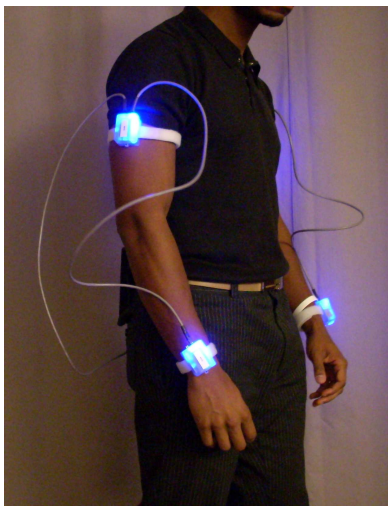


Figure 2.6: The motion suit worn by the subjects consists of the inertial measurement units (IMU) and a wearable computer. In the experiments, only one IMU is used.

data was corrected for human reaction time (approximately 0.22 hundredths of a second) on the stopwatch of the clinician during the Wolf assessment, the data's RMS error was approximately one second. This error for subjects with low functional abilities (and longer movement times) was of less significance than for those who moved normally. The researchers had problems with how best to interpret the Wolf instruction such that an automated system can deal with unique

actions. Hence the modified set of instructions, the researchers initially used for their experiment. The research did confirm that the study of automated assessment systems is a necessary issue that must be addressed in telemedicine.

Hester et al. used the Vitaport 3 (Temec BV, The Netherlands), similar to Bonato et al, to collect accelerometer data [16] as shown in Fig. 2.7. The data from the torso and arms was

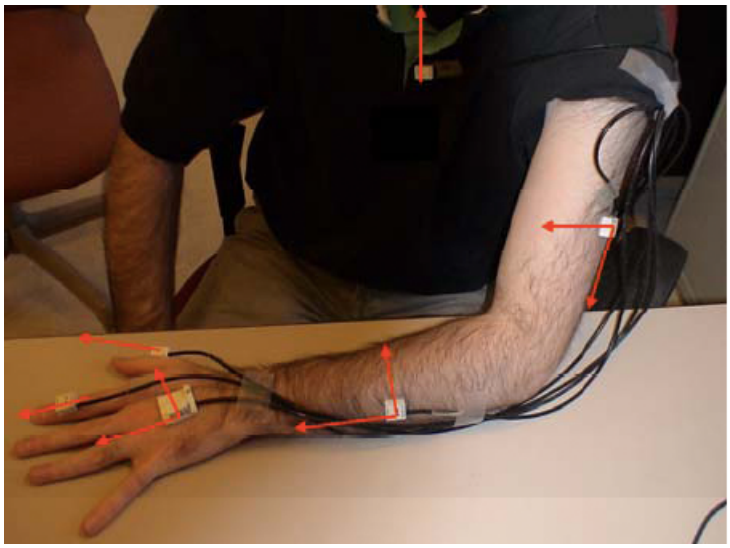


Figure 2.7: The sensor setup and orientation of the axes of the accelerometers.

correlated to Chedokee-McMaster (CM) [26], FM, and Wolf scores using a linear regression technique. Their results show that linear regression models have predicted two clinical scores within 10% of the average scores. The research also found that compensatory torso movement lowered the generated scores and that third order jerk (smoothness coefficient) was significantly related to improved clinical scores, (i.e. smooth movement, better clinical scores).

None of the pervious work approached the solution of motion assessment in terms of

robotic standard Denavit-Hartenberg (D-H) parameters' use of joint angles nor use of matrix equations to generate a model as has been explored in this study.

Chapter 3

The Model

This study used seven of the twenty-two total motions available in the FM to create a modified FM (MFM) of arm motions to independently assess and compare to each stroke survivor's clinically derived FM score. These selected arm motions closely or exactly match the joint focused motions developed for the experiment. Fig. 4.5 provides the side-by-side comparison of FM and MFM motions. The reason for this modification is to minimize the difference between the requested articulation of each joint, this allowing for discussion on the success or failure of the actual SMS data to correlate to the clinically derived FM score.

Deleterious joint synergies are characterized by the involuntary co-activation of joints. As such, if an individual attempts to move one, and only one joint, then the existence of a joint synergy among the other joints implies that some or all of the other joints in the arm will also move in response to the intended joint movement. If it is assumed that *joint one* is arbitrarily designated as the single joint intentionally moved, then the response of the voluntary and the other six involuntary joints activated during the intended movement can be represented as the

column vector, $\vec{R}_1$,

$$\vec{R}_1 = [f_{11}(x_1)\ f_{21}(x_1)\ f_{31}(x_1)\ f_{41}(x_1)\ f_{51}(x_1)\ f_{61}(x_1)\ f_{71}(x_1)]^T \tag{3.1}$$

where $f_{11}(x_1)$ denotes the response of *joint one* to an attempted motion (or as a function) of *joint one*, $f_{12}(x_1)$ denotes the response of *joint two* as a function of *joint one*, $f_{13}(x_1)$ denotes the response of *joint three* as a function of *joint one*, and so on up to *joint seven*. Likewise, the same procedure is performed on the next joint to generate $\vec{R}_2$.

Physically, column vectors, $\vec{R}_j$, are generated experimentally by asking the subject to move a single joint in an isolated movement and then simultaneously record the response of the other six joints. Likewise, the same procedure is performed for joint two, joint three, and so on. After carrying out this procedure for all seven joints the resulting functions can be expressed as the following matrix:

$$\hat{\vec{y}} = |\vec{R}_1\ \vec{R}_2\ \vec{R}_3\ \vec{R}_4\ \vec{R}_5\ \vec{R}_6\ \vec{R}_7| = \tag{3.2}$$

$$\begin{vmatrix}
f_{11}(x_1) & f_{12}(x_2) & f_{13}(x_3) & f_{14}(x_4) & f_{15}(x_5) & f_{16}(x_6) & f_{17}(x_7) \\
f_{21}(x_1) & f_{22}(x_2) & f_{23}(x_3) & f_{24}(x_4) & f_{25}(x_5) & f_{26}(x_6) & f_{27}(x_7) \\
f_{31}(x_1) & f_{32}(x_2) & f_{33}(x_3) & f_{34}(x_4) & f_{35}(x_5) & f_{36}(x_6) & f_{37}(x_7) \\
f_{41}(x_1) & f_{42}(x_2) & f_{43}(x_3) & f_{44}(x_4) & f_{45}(x_5) & f_{46}(x_6) & f_{47}(x_7) \\
f_{51}(x_1) & f_{52}(x_2) & f_{53}(x_3) & f_{54}(x_4) & f_{55}(x_5) & f_{56}(x_6) & f_{57}(x_7) \\
f_{61}(x_1) & f_{62}(x_2) & f_{63}(x_3) & f_{64}(x_4) & f_{65}(x_5) & f_{66}(x_6) & f_{67}(x_7) \\
f_{71}(x_1) & f_{72}(x_2) & f_{73}(x_3) & f_{74}(x_4) & f_{75}(x_5) & f_{76}(x_6) & f_{77}(x_7)
\end{vmatrix}$$

The elements of column vector $\hat{\vec{y}}$ in (3.2) describes the response angle as a function of the seven input angles, including the joint that is being moved. The relationship between the joint that a subject attempts to move compared to the way that joint actually moves is difficult,

and perhaps impossible to know. In order to simplify the model, it is assumed that the response angle for the joint being intentionally moved, is equal to the desired angle. Therefore, for $i = j$, $f_{ij}(x_j) = 1$, where i is the i^{th} row and j is the j^{th} column. If such a matrix could accurately represent joint synergies then one could predict the arm trajectories of a paretic arm using $\hat{\vec{y}}$ given any combination of desired input angles. Assume that the starting angles of a given reaching movement all begin at zero, and that these functions can be approximated by a linear polynomial model which sufficiently characterizes the synergy, then

$$\hat{\vec{y}} = \sum_{k=0}^{n} \vec{A}(k)\vec{x}^{k} \tag{3.3}$$

where n is the order of the polynomial model and $A(k)$ is the matrix of coefficients of the k^{th} ordered term. The zero order term A(0) includes the start position angles which are arbitrarily assigned. If it is assumed that the start and end positions are approximately the same then $A(0)$ is not especially important for these purposes, and is thus ignored. Because the output of the desired joint is assumed to equal the desired rotation, the 1^{st} order matrix $A(1)$ always has ones on the diagonal. Higher order terms, $A(2)$ through $A(n)$, will have zeros on the diagonal. For the purposes of this paper, let's consider the linear model only. Linear $(n = 1)$, quadratic $(n = 2)$, and cubic $(n = 3)$ model fits are further evaluated in Simkins' paper [30]. In this case the linear model is

$$\hat{\vec{y}}_{linear} = \vec{A}(1)\vec{x} + \vec{A}(0) = \tag{3.4}$$

$$\vec{A}(1)\vec{x} = \left|\begin{array}{ccccccc||c} 1 & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} & a_{17} & x_1 \\ a_{21} & 1 & a_{23} & a_{24} & a_{25} & a_{26} & a_{27} & x_2 \\ a_{31} & a_{32} & 1 & a_{34} & a_{35} & a_{36} & a_{37} & x_3 \\ a_{41} & a_{42} & a_{43} & 1 & a_{45} & a_{46} & a_{47} & x_4 \\ a_{51} & a_{52} & a_{53} & a_{54} & 1 & a_{56} & a_{57} & x_5 \\ a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & 1 & a_{67} & x_6 \\ a_{71} & a_{72} & a_{73} & a_{74} & a_{75} & a_{76} & 1 & x_7 \end{array}\right| \tag{3.5}$$

where $A(1)$ gives the linear coefficients for the synergistic joint responses and $A(0)$ denotes the start and end angles. The resulting matrix in (3.5) is sometimes referred to as a covariance matrix and a plot of the slope values is referred to as an interactions plot. Fig. 3.1 shows the plots of each joint of focus versus the response of the other joints. For this paper, the matrix is called the synergy matrix (SM) because it describes the interactions of human joints which are discussed in medical literature as synergies [4].

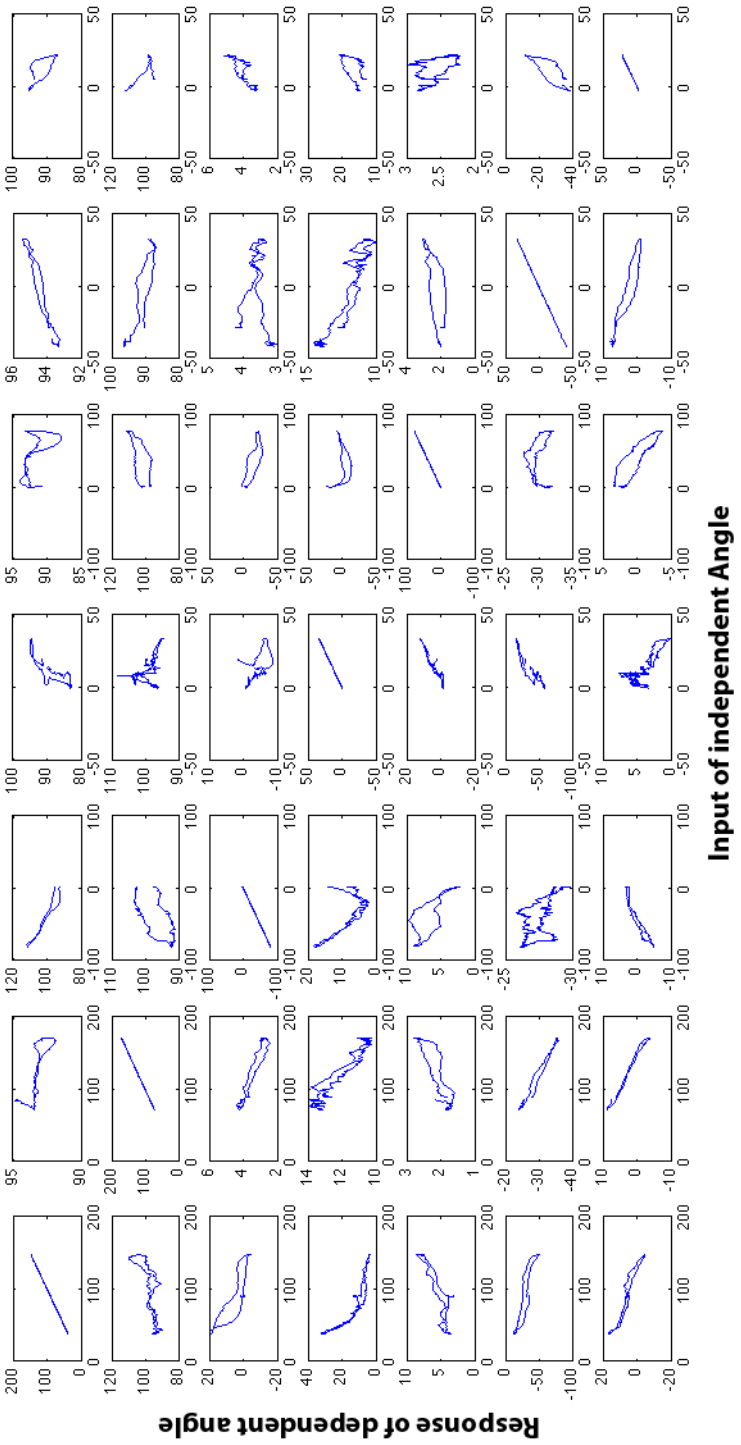


Figure 3.1: The diagonal has a 45 degree straight line representing theta equal to the same theta, i.e. a slope of positive one.

Part II

Chapter 4

Experimental Setup

4.1 Subjects

The research was approved by the University of California, Santa Cruz, Internal Review Board. All subjects provided written consent prior to study participation. Gender was not anticipated to be a factor and was not controlled for in the data collected. Twenty-three subjects participated in this study. Twelve healthy control subjects and 11 stroke affected subjects. Of the 12 healthy male control subjects, five were age-matched because age was a possible confounding factor. The age-matched ages ranged from 61 to 77 years old. The remaining seven ages ranged from 19 to 45 years old. The control subjects had no history of strokes and were neurologically intact. Control candidates with arm and/or shoulder injuries or neurological damage were excluded. Healthy control subjects performed arm motions using their dominant arm.

Eleven hemiparetic subjects participated in the study, six female and five male, ages

ranging from 54 to 82 years old. All stroke survivors were in their chronic phase of recovery. The number of years since their most recent stroke ranged from 1 year to 15 years. All stroke subjects were screened with a phone interview prior to being scheduled for testing to ensure they met the minimum set of requirements to participate in this study. Candidates were rejected if their impairment was too mild; if they had fully recovered; if the impairment was caused by an injury other than a stroke,(i.e. head trauma); if the stroke was less than two months old; if the candidate has no range of motion or less than ten degrees for arc in their shoulder, elbow, and wrist. The reason for this last requirement is that this study uses the FM as a comparison standard to corroborate efficacy of the SM. The FM requires the candidate to have the ability to straighten the arm (See Fig. 4.5(b)) as well as the ability to maintain the desired start position (See Fig. 4.5(f)). If the start position cannot be executed by the candidate, then the FM assessment cannot be done [14]. Therefore, if the candidate could not complete the FM assessment the candidate's results could not be compared to an existing standard and the candidate was ineligible to take part in this experiment.

The resulting sample population was intended to have as diverse a range of hemipareses as possible while ensuring that subjects had enough range-of-motion in the subject's affected arm to generate meaningful motion capture data. In other words, the stroke affects were severe enough that the subject's arm demonstrated measurable joint synergies. Table 4.1 has statistical biometric data of experimental subjects. See Appendix B for the phone screening worksheet.

Table 4.1: Biometric Data of Experimental Subjects ($n = 22$)

Health Age (years)	$\bar{x} = 48.5(\pm 22.4)$
Stroke Age (years)	$\bar{x} = 65.5(\pm 8.4)$
Time since stroke (years)	$\bar{x} = 7.6(\pm 5.3)$
Side of Infarct	3 LCVA 8 RCVA
Gender	6 females 16 males

LCVA: left cerebrovascular accident.
RCVA: right cerebrovascular accident.

4.2 System

4.2.1 Hardware

The Vicon MX motion capture system [32] was used to record infra-red reflective marker locations in a calibrated target volume. Ten ceiling mounted MX cameras pointed at a calibrated target volume centered on the subject's arm. Fig.4.1 shows the hardware and the experimental volume. The calibration wand set up the reference (x, y, z) origin from which was derived the standard D-H parameters for the seven degrees of freedom (7-DOF). Fourteen markers in total were attached to the subjects right or left arm and torso. Ten arm markers were attached to bare skin and four markers attached to a shirt that was taped snug to the body as depicted in Fig. 4.3(a), 4.3(b) and Fig. 4.4. The sampling rate of the cameras was set to 100 Hz. The marker placement is particularly sensitive due to the needs of the Nexus' model template applied to the raw data. Following the template's desired positions ensures accurate results. The subjects were seated on a metal chair with a backrest and no arm rests. The subject's torso was held in place with duct tape that wrapped around the subject's abdomen and backrest of the chair. With respect to the taping of the torso, it is important to note that results from [7] suggest

(a) (b) (c)

Figure 4.1: Experimental setup for stroke survivor with right-side hemiparesis. (a) The infra-red reflective maker. (b) The VICON camera. Notice the infra-red broadcast rings around the lens apeture. (c) The subject, the data collection volume, and one of the researchers.

that there is no strong relationship between the amount of trunk use and functional performance of the subsequent requested action. Thus, the purpose of limiting torso lateral movement was to help with isolating joint movement and core stability. All arm motions started and ended from the same position as depicted in Fig. 4.4.

4.2.2 Software

Once the frame of reference was set up during calibration, all markers' positions were relative to that origin. The VICON Nexus program does not use standard D-H parameters when constructing the x, y, z orientation of the markers during data collection. Therefore, post processing of the data was required to modify the current orientation of the x, y, z frame to the desired D-H orientation of the x, y, z, frame where z always points along the axis of rotation. Fig. 4.2 shows the desired D-H frame orientation. This was accomplished by creating custom

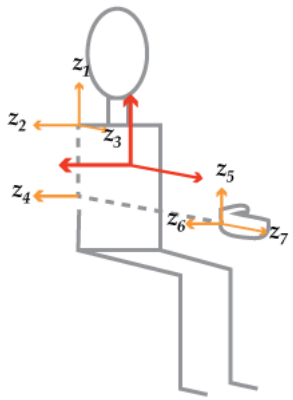


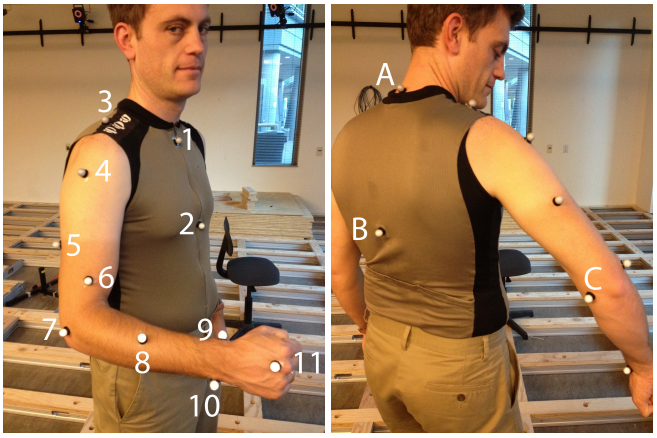
Figure 4.2: Z is the axis of rotation. Z_1 is shoulder inner/outer rotation, Z_2 is shoulder extension/flexion, Z_3 is shoulder abd/adduction, Z_4 is elbow flexion/extension, Z_5 is wrist extension/flexion, Z_6 is wrist radial/ulnar deviation, Z_7 is wrist supination/pronation

code written in Vicon Bodybuilder. In Bodybuilder, the calibration origin was adjusted by using fixed Euler angles to match the torso origin with the same z-pattern as the shoulder. The resulting joint angles were calculated from the solution of the fixed Euler angles, (4.1). Joint angle information was then further processed using various MATLAB scripts to generate the SM scores and results section graphs and table data. (See Appendix A for Bodybuilder and MATLAB code.)

$$R_{xzy} = \begin{vmatrix} c2c3 & -s2 & c2s3 \\ c1s2c3+s1s3 & c1c2 & c1s2s3-s1c3 \\ s1s2c3-c1s3 & s1c2 & s1s2s3+c1c3 \end{vmatrix} \tag{4.1}$$

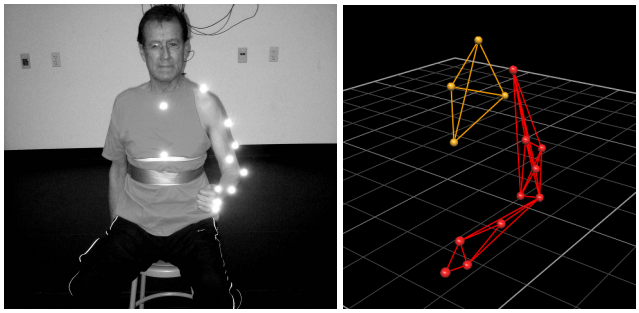
4.3 Protocol

The experiment took approximately 60 to 90 minutes for each subject. The actual data collection took approximately 15 minutes. Control and hemiplegic subjects performed 21



(a) (b)

Figure 4.3: (a) 1=jugular notch where clavicles meet the sternum, 2=xiphoid process of the sternum, 3=acromio-clavicular joint, 4=lateral lower belly of the deltoid (asymmetric with 5 and 6), 5=distal end of triceps outer head (asymmetric with 4 and 6), 6=distal end of the long head of the bicep brachii (asymmetric with 4 and 5), 7=lateral humerus epicondyle approximating the elbow joint axis, 8=insertion point of the anconeus, 9=thumb side of the radial styloid (symmetrical with 10 to form an axis through center of wrist), 10=little finger side of ulnar styloid (symmetrical with 9 to form an axis through center of wrist) (b)A=7th cervical vertebra, B=10th thoracic vertebra, C=medial humerus epicondyle approximating the elbow joint axis



(a) (b)

Figure 4.4: Experimental setup for stroke survivor with left-side hemiparesis. Pictured in (a) is the standard start position. Pictured in (b) is the subject-specific arm model.

arm movements in total. Using the start position depicted in Fig. 4.4, subjects were asked to perform arm movements that focused on one joint at a time, until all seven joints were moved. Every iteration through the seven joint movements is considered a set. All subjects completed at least three sets. All subjects also performed a modified Fugl-Meyer assessment (MFM) which has seven arm motions as well. The MFM assessment was conducted by the author as trained by Nancy Byl, PhD, PT, FAPTA, Professor and Chair, Department of Physical Therapy and Rehabilitation Science at UCSF. The purpose of the modification to the FM will be discussed in the Data Analysis section that follows. The performed actions for the experiment and the MFM are shown in Fig. 4.5 and summarized in Table 5.2. See Appendix C for the protocol and Appendix D for the MFM worksheet.

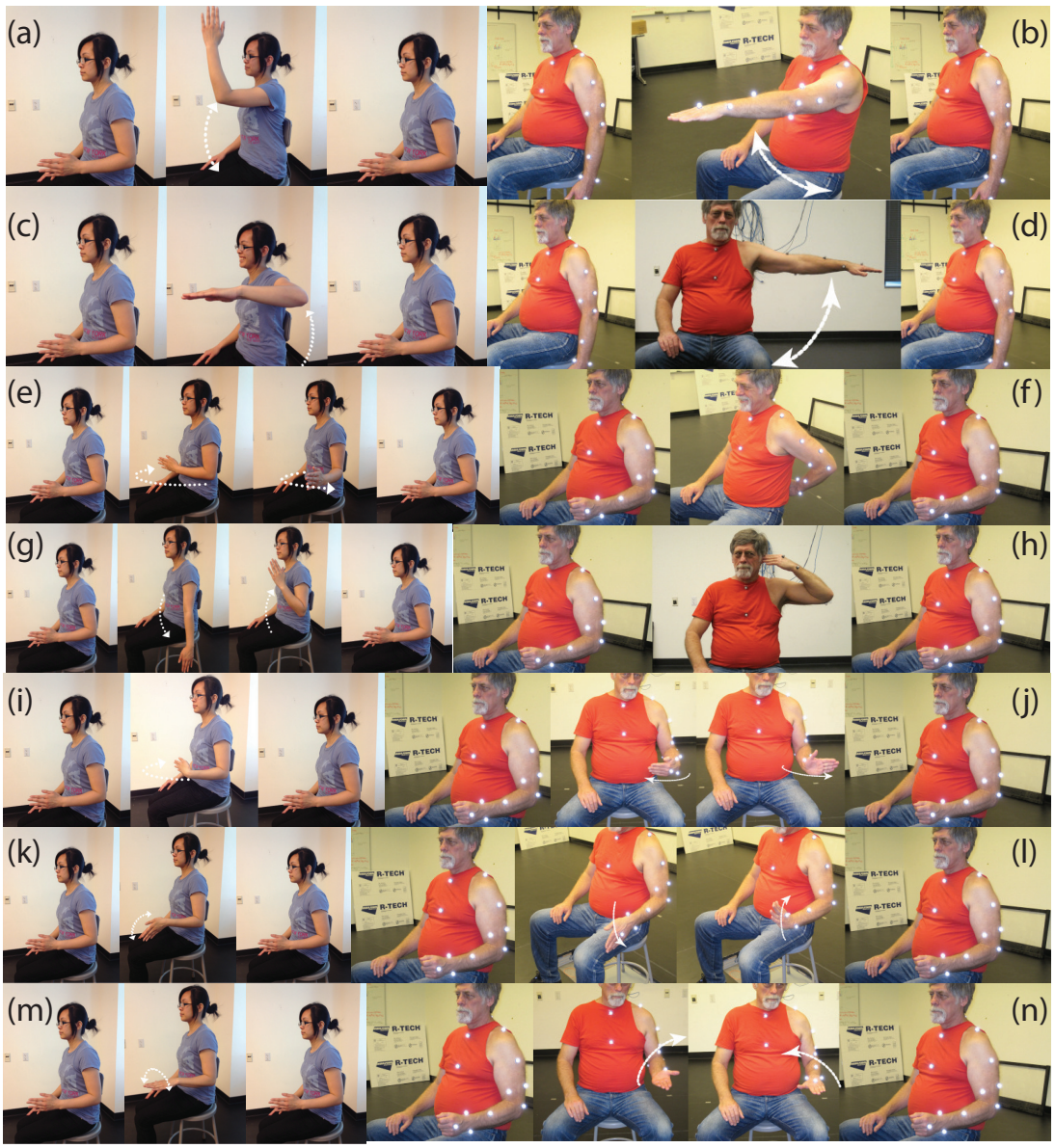


Figure 4.5: All image sequences show start and stop positions with actions in-between. a, c, e, g, i, k, m are synergy matrix motions while b, d, f, h, j, l, n are Fugl-Meyer assessment actions. The synergy experiment motions were developed to best match the MFM motions selected from the full FM assessment to ensure valid comparison between SMS and FM scores.

Chapter 5

Results

All of Subject 6's data, Subject 13's trial 1, and Subject 23's trial 2 data were left out of the calculations due to occlusions of the markers causing subsequent data loss (blank columns) which reported uncharacteristically calculated values as evidenced by the box plot in Fig. 5.1. Hence, the population number is $n = 22$. Interpolation of the data points was not done nor any filtering to close the gaps if data points were lost by noise or occlusions. A typical single action data set takes approximately five seconds to execute. At 100 Hz, that is 5000 columns of data points containing joint angles. Twenty-one actions times three repetitions times 22 subjects is approximately seven million data points. With a data density of 100 Hz, filtering was not necessary. Table 5.1 has descriptive general statistics for healthy and stroke subjects. The MFM score is 0-14 where 14 is the top score indicating excellent healthy movement and 0 is indicating poor unhealthy movement. With a small population of 22 and three trials or repetitions done, a normal distribution is less certain since the standard minimum population for statistics is greater than 30 [12]. Nevertheless, normal distribution is assumed and the results

are reassured by the good fit shown in Table 5.1. See Appendix E for MFM and SMS data.

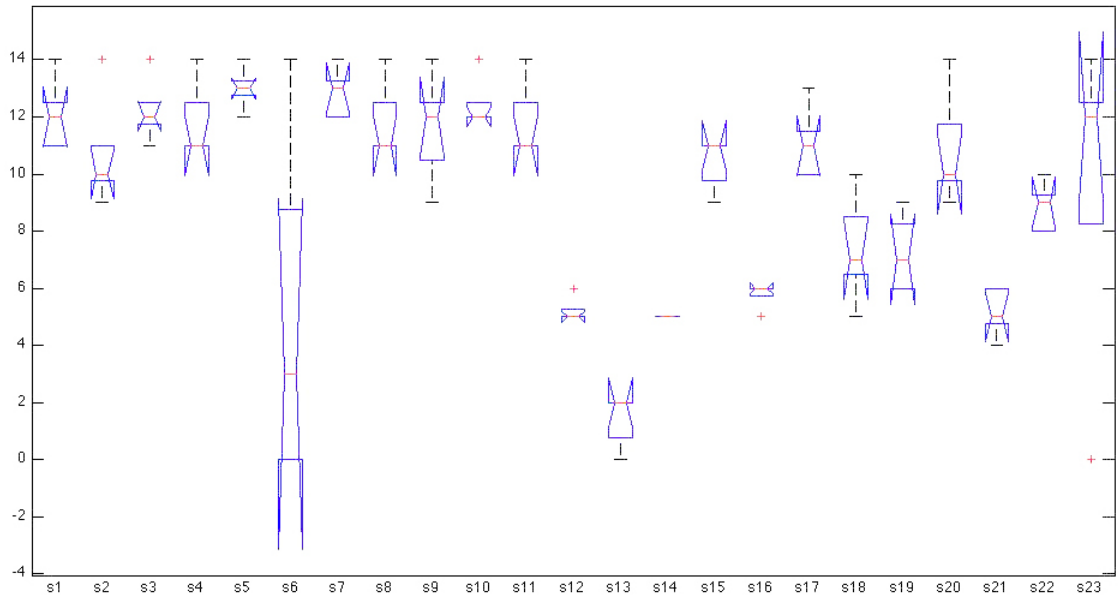


Figure 5.1: Y-axis is the SM scores. Subjects 6, 13, 23 have standout data issues. Subsequently, Subject 6 was left out of calculations.

Table 5.1: Descriptive Statistics ($n = 22$)

	Mean	Variance	STDERR
Healthy			
MFM Score	14	1.08002e-12	3.1109e-07
SM Score	11.6364	0.854545	0.278722
Stroke			
MFM Score	7.90909	15.6909	1.19434
SM Score	7	7.4	0.8202

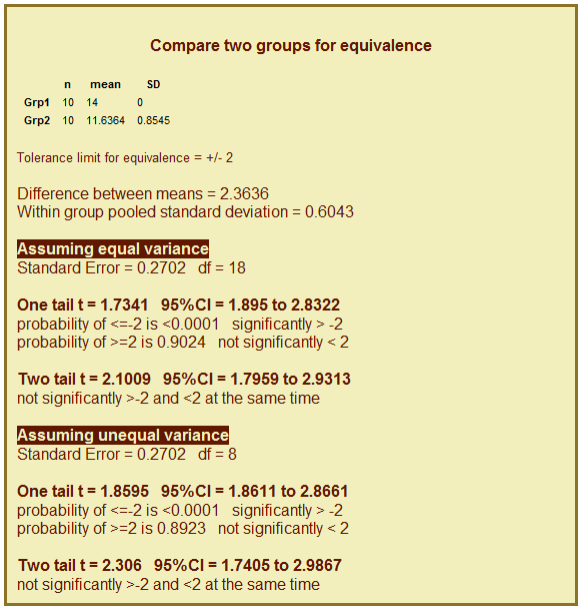
5.0.1 SM Score vs. MFM Score

The MFM scores of all the healthy subjects, as expected, are 14 (maximum score). A comparison for equivalence between the healthy SM and MFM mean scores did not produce

a significant value to show equivalence with a p-value of 0.9024 as summarized in Fig. 5.2. Since the healthy subjects' MFM scores were not normally distributed, a comparison between the healthy SM and MFM scores was done using the Wilcoxon signed rank (paired sample) test. The results show that the null hypothesis; "MFM scores do not match SM scores", was rejected with a p-value of less than 0.001 as Fig. 5.3 shows. ANOVA was used to corroborate the result and weakly supported the Interclass Correlation Coefficient (ICC) results with a p-value of less than 0.2872. As shown in Fig. 5.4(b), it is evident that the SM scores do track the MFM scores with a plus or minus of two and a half points for the healthy subjects, which suggests the subjective hysteresis of clinical assessment of healthy subjects is more generous than of the stroke subjects. This point will be addressed in more detail in the Discussion. As a study in contrast, the stroke data was normally distributed. Thus, it was analyzed using ICC. The results show an almost perfect agreement between SM stroke scores and MFM stroke scores with an ICC equal to 0.8227 as shown in Fig. 5.5. ANOVA was used to corroborate the result and strongly supported the ICC results with a p-value of 0.9034. As shown in Fig. 5.4(a), the SM scores do track closer than the healthy SM scores with the MFM scores at less than one point variance.

5.0.2 Synergy Matrix Data Density

The output, $\hat{\vec{y}}_{linear}$, is the product of SM times the range of each joint as demonstrated in equation (3.4). The resulting 7x7 matrix has joint angles that express joint synergies characterized by the SM. Fig. 5.6(a) illustrates the ideal case where the resulting bar graph of the SM has only values on the diagonal and zeros everywhere else. The larger the bar, the greater



Compare two groups for equivalence

	n	mean	SD
Grp1	10	14	0
Grp2	10	11.6364	0.8545

Tolerance limit for equivalence = +/- 2

Difference between means = 2.3636
Within group pooled standard deviation = 0.6043

Assuming equal variance
Standard Error = 0.2702 df = 18

One tail t = 1.7341 95%CI = 1.895 to 2.8322
probability of <=-2 is <0.0001 significantly > -2
probability of >=2 is 0.9024 not significantly < 2

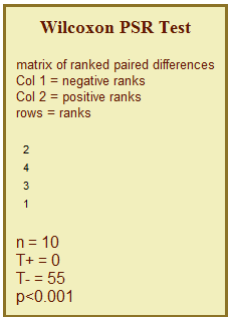
Two tail t = 2.1009 95%CI = 1.7959 to 2.9313
not significantly >-2 and <2 at the same time

Assuming unequal variance
Standard Error = 0.2702 df = 8

One tail t = 1.8595 95%CI = 1.8611 to 2.8661
probability of <=-2 is <0.0001 significantly > -2
probability of >=2 is 0.8923 not significantly < 2

Two tail t = 2.306 95%CI = 1.7405 to 2.9867
not significantly >-2 and <2 at the same time

Figure 5.2: Equivalence between healthy SM mean score and MFM mean score. Image produced by stattools.net.



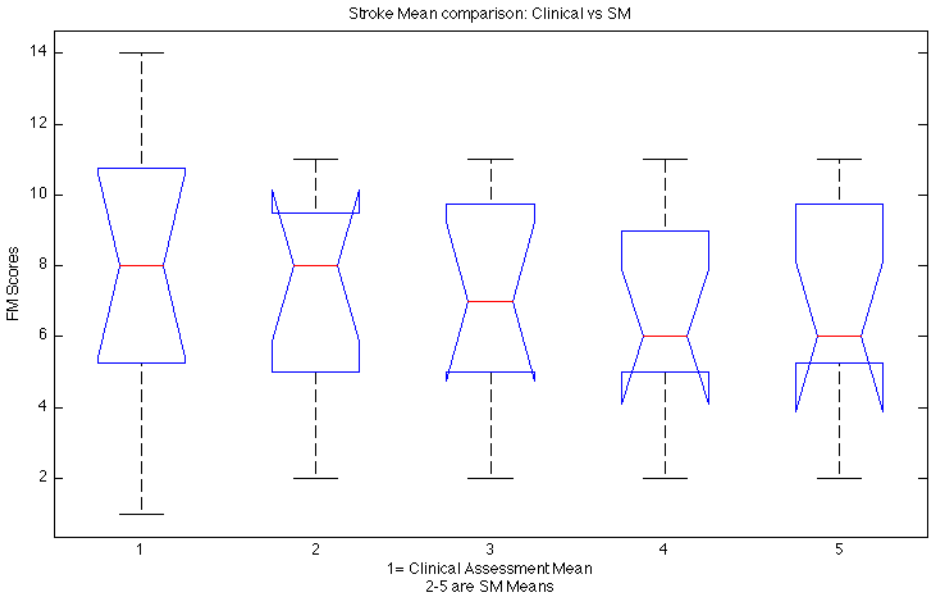
Wilcoxon PSR Test

matrix of ranked paired differences
Col 1 = negative ranks
Col 2 = positive ranks
rows = ranks

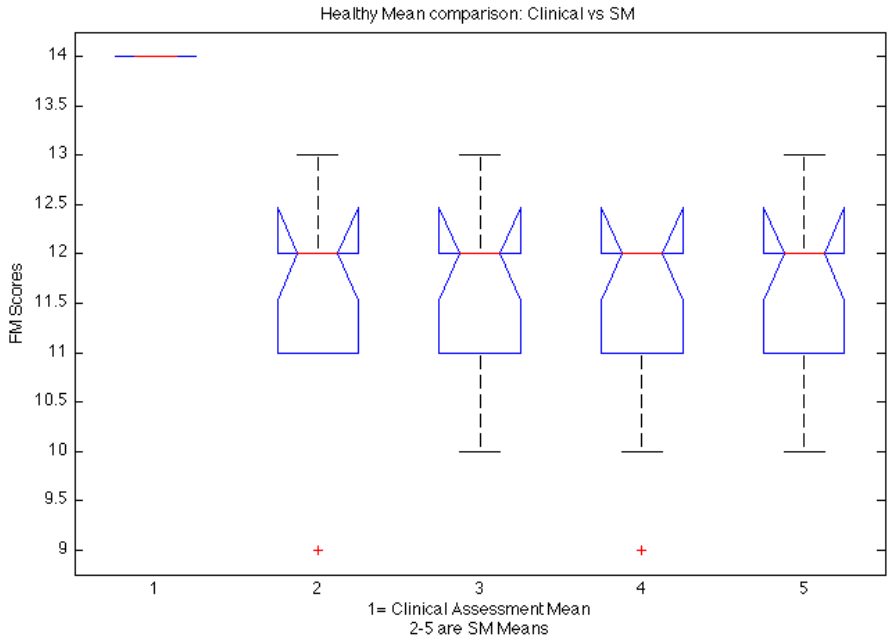
2
4
3
1

n = 10
T+ = 0
T- = 55
p<0.001

Figure 5.3: Nonparameter analysis with Wilcoxon signed rank test on healthy SM mean score and MFM mean score. Image produced by stattools.net.

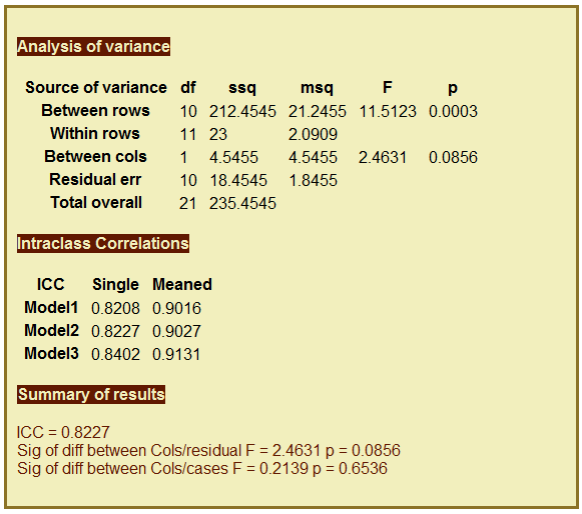


(a)



(b)

Figure 5.4: ANOVA box plots. (a) The SM stroke means compared to MFM stroke mean. (b) The SM healthy means compared to MFM healthy mean.



Analysis of variance

Source of variance	df	ssq	msq	F	p
Between rows	10	212.4545	21.2455	11.5123	0.0003
Within rows	11	23	2.0909		
Between cols	1	4.5455	4.5455	2.4631	0.0856
Residual err	10	18.4545	1.8455		
Total overall	21	235.4545			

Intraclass Correlations

ICC	Single	Meaned
Model1	0.8208	0.9016
Model2	0.8227	0.9027
Model3	0.8402	0.9131

Summary of results

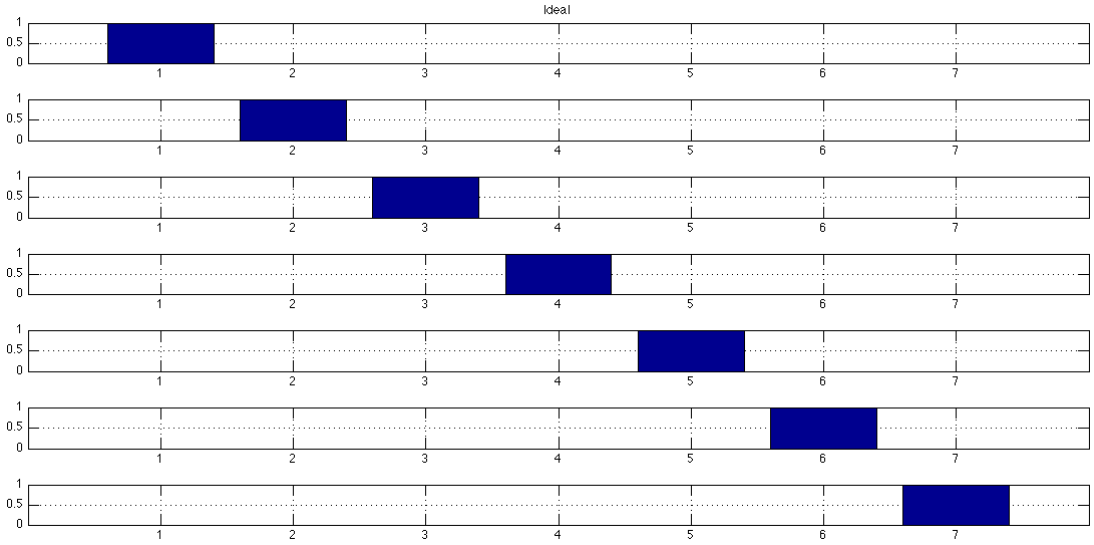
ICC = 0.8227
Sig of diff between Cols/residual F = 2.4631 p = 0.0856
Sig of diff between Cols/cases F = 0.2139 p = 0.6536

Figure 5.5: Stroke SM score and MFM score ICC comparison. Image produced by stattools.net.

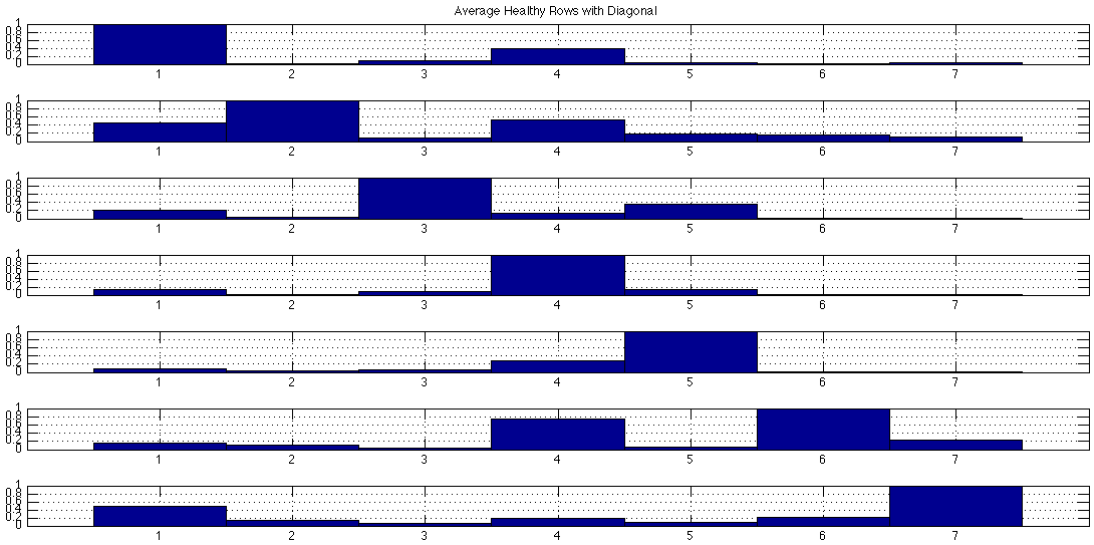
the joint movement. Averaging all the healthy subjects' SM into a representative mean healthy "golden standard" SM results in Fig. 5.6(b), which illustrates the typical healthy synergies allowing humans to move with ease and grace. Fig. 5.7 shows a stroke subjects' deleterious SM as compared to the gold standard efficacious SM. The scale of the gold standard SM is set to the max value of the stroke subjects' SM in order to better compare each image. Subtracting Fig. 5.7(a) from Fig. 5.7(b) leaves the residuals. These residuals in Fig. 5.8 are only those synergies that are deleterious to healthy movement synergies. Note the y-axis values exceed one as normalization was done with the max value of range for the specific joint (row major). Any number above one has deleterious synergies. The x-axis specifies the joint of focus.

5.0.3 Sensitivity

The SM provides sensitivity to change over time as well as detail to distinguish differences. The results show in general that all subjects tended to score 1.5 points lower on sub-

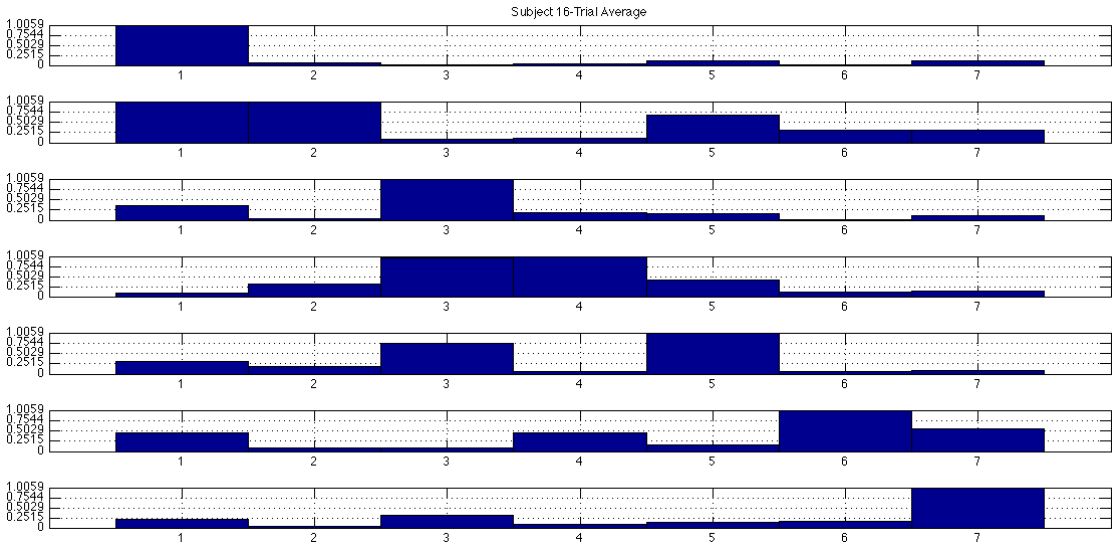


(a)

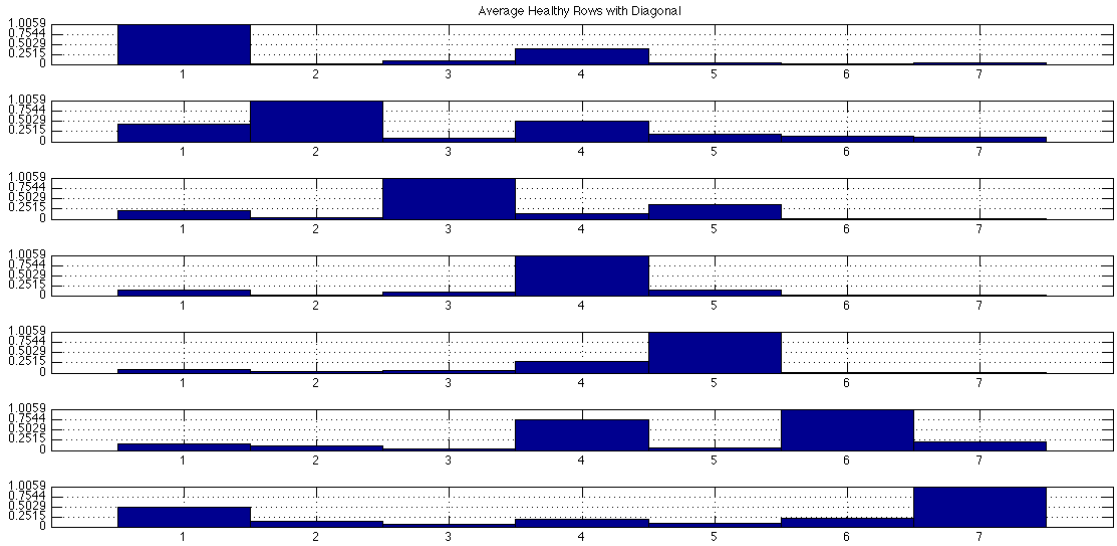


(b)

Figure 5.6: (a) The ideal input with max range per joint and no synergies. (b) Averaging all the healthy subjects into one gold standard SM times the max range per joint.



(a)



(b)

Figure 5.7: (a) Stroke subject 16's SM. (b) The gold standard SM with the scale adjusted to match stroke subject 16.

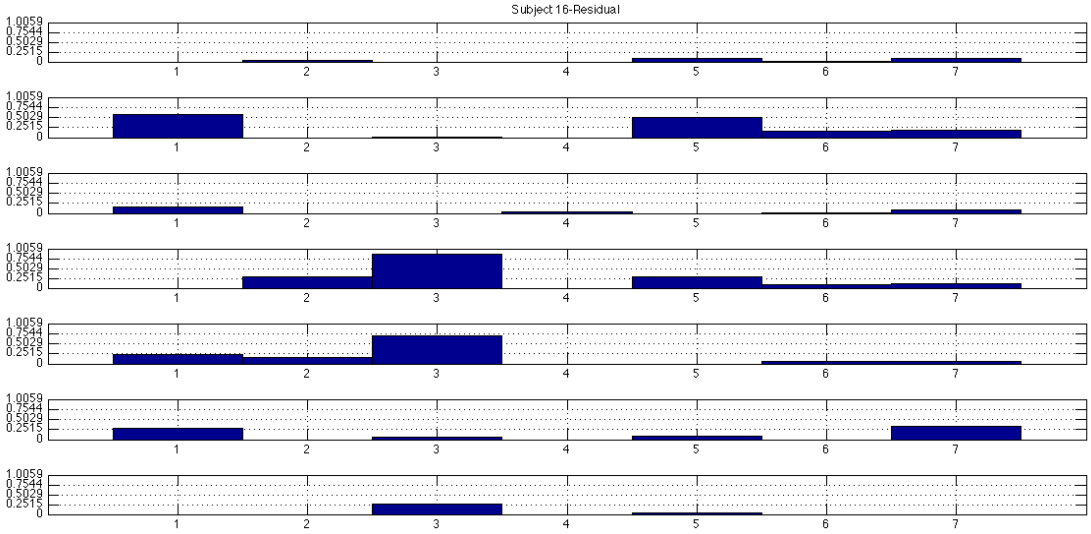


Figure 5.8: Subject 16's residuals show strong synergies in joints 1, 3, 5, 7.

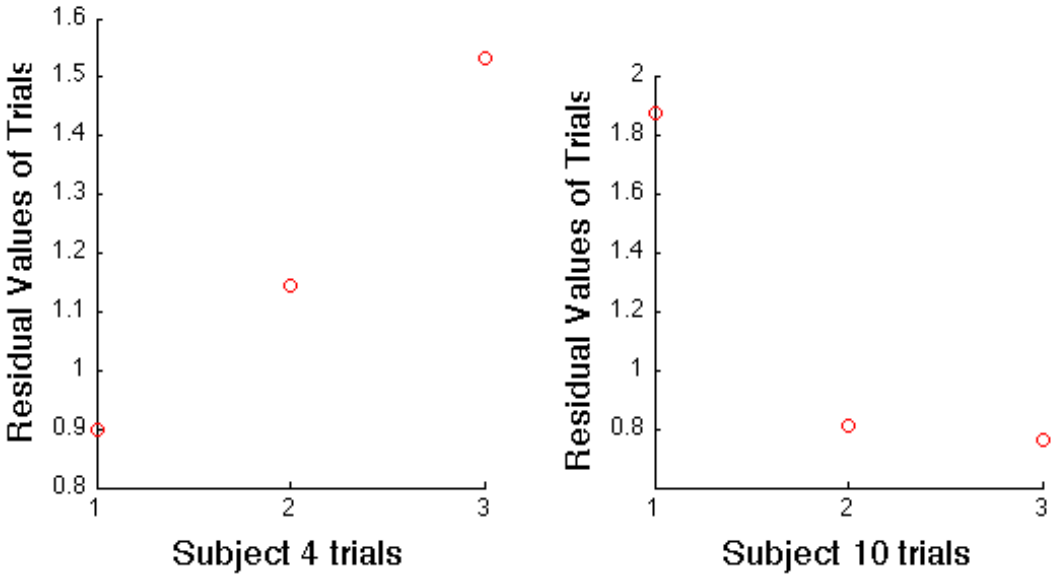
Table 5.2: Joint Movements

Joint	Joint No.	Req. Movement	Range (degrees)
Elbow	1	Extension, flexion	160
Wrist	2	Pronation, supination	90
Shoulder	3	Flexion	90
Shoulder	4	Inner/outer Rotation	130
Shoulder	5	Abduction	90
Wrist	6	Flexion	90
Wrist	7	Ulnar deviation	55

130 adjusted from 160 due to instruction of stopping at torso contact.

55 adjusted from 90 due to only asking for ulnar deviation

sequent trials than the initial SM trail score. The residuals were calculated to show how much performance decays between each trial completed by each subject. Subject 4's SM scores are 12, 11, and 11 for trials 1, 2, and 3 respectively. The residuals express the difference between trials 2 and 3 even though they scored an 11. Subject 10's SM scores remained a consistent SM score of 12 through all the trials, yet the residuals show an improvement over time. Fig 5.9 demonstrates this point.



(a) (b)

Figure 5.9: (a) An increase in synergies is detected. (b) A refinement of synergies is detected.

Part III

Chapter 6

Analysis and Discussion

Statistical analysis was done using Statistics Toolbox from MATLAB. Nonparametric analysis was used for all data not normally distributed such as the Wilcoxon signed rank (paired sample) test for comparing the healthy SM and MFM scores. The normally distributed SM and MFM stroke scores were compared by ICC, models (1, 2, 3). A repeated measures tool, one-way ANOVA, was performed to corroborate and examine a subject's intra-trial SM and MFM scores in order to see change over time. For all analyses the chosen values for the criterion alpha level was 0.05, power was 0.84, and difference divided by standard deviation was 0.65. These values allowed for generalization of results when using a small population of 22 non-random subjects.

The SM score is a number created to corroborate efficacy of the SM as clinical assessment tool. Therefore a natural comparison to an existing clinical standard, such as the FM score, is required. This paper has shown that an objectively collected data set and calculated SM scores closely match the subjective MFM scores. Furthermore, any standard clinical as-

sessment, such as Wolf, can be matched as well. Supporting this statement is the method of calculation. In this paper the calculation is an algorithm, called the SM rules engine, which was created to match the assessment tool desired - in this case the MFM. The SM rules engine for the MFM is as follows in pseudocode, Algorithm 1.

Algorithm 1 can be written to output matching assessment scores by calculating the range of each joint for the FM, or range of time for Wolf, or a combination from the data. The range is then divided by the ordinal scale of the assessment being used. In the case of FM the ordinal scale is three and by adding two more *if clauses* per joint the range can be divided by five as is the case with Wolf. This crude system was found to fall within the 95% CI. However, there are those actions that don't easily map as in the case of elbow flexion, Fig. 4.5(g) and Fig. 4.5(h). Thus, a way to fine tune for the specific assessment is needed.

To fine tune the SM rules engine containing the MFM algorithm, *if clauses* echoed synergistic conditions expressed by the data. Hypothetically, a healthy person has a stroke with a loss of degree of freedom in the wrist. When wrist movement is lost, elbow flexion and shoulder abd/adduction synergies dramatically increase to maintain the workspace of the hand. The algorithm then must reflect this synergistic conditions expressed by the data as a hysteresis which further bounds the joint actions logic to produce more exact results. Brunnstrom, et al. summarized that there are typical patterns of synergistic responses with loss of degree of freedoms [4]. Fig. 5.8 supports this finding as empirically observed in the stroke subjects of this experiment.

Data is as precise as the hardware that collected it and the software's limitation of precision. That being said, The SM score which was made to emulate MFM scores tended to be

Result: MFM algorithm generates 0, 1, 2 scores

initialization;

while *not at end of this document* **do**

read joint and joint range;

if *joint equals elbow* **then**

if *(absolute(joint range) <= (Max static ROM value of elbow*1/3)) or (absolute(joint range of shoulder rotation) > (Max static ROM value of shoulder *5/16))* **then**

MFM score = 0;

end

if *(absolute(joint range) > (Max static ROM value of elbow*1/3)) and (absolute(joint range) <= (Max static ROM value of joint*2/3)) and (absolute(Max static ROM value of shoulder)<=(Max static ROM value of shoulder*5/16)* **then**

MFM score = 1;

end

if *(absolute(joint range) > (Max static ROM value of elbow*2/3)) and (absolute(joint range of shoulder rotation)<=(Max static ROM value of shoulder *5/16))* **then**

MFM score = 2;

end

end

if *joint equals shoulder abduction* **then**

three sub if statements as above changing the Max static ROM value

end

if *joint equals shoulder inner rotation* **then**

three sub if statements as above changing the Max static ROM value

end

. . .

end

Algorithm 1: This is the general SM rules engine for the MFM scoring.

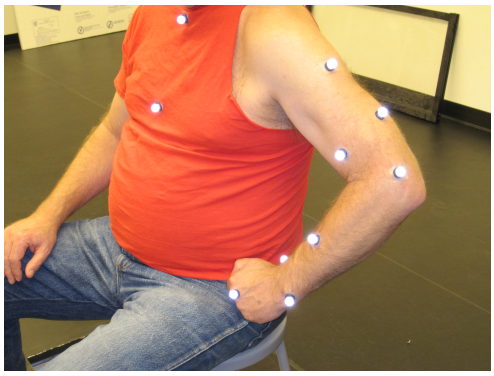
2.5 points lower for healthy subjects versus within 1 point lower for with stroke subjects. It is argued that this discrepancy is from the clinician's ability and experience to decide if the MFM score is 0, 1, or 2. For example, Fig. 6.1 shows a subject attempting to move the back of the hand to the small of the back as way of showing range of motion for inner shoulder rotation. Table 6.1 contains the scoring instructions to the clinician based on the requested action. In Fig. 6.1(c), the subject could score a 1 or 2 depending on the experience of a clinician. The data calculation being objectively exact tended to be less generous with this score. Even with a hystereses that fine tuned the values from 0 to 1 and 1 to 2, the SM scores tended to be lower. Interestingly, the stroke MFM score closely matched the SM scores implying that greater care was taken with the stroke assessments by the clinician. Further implying that the SM score is more reliable than the MFM score.

Table 6.1: FM Movements

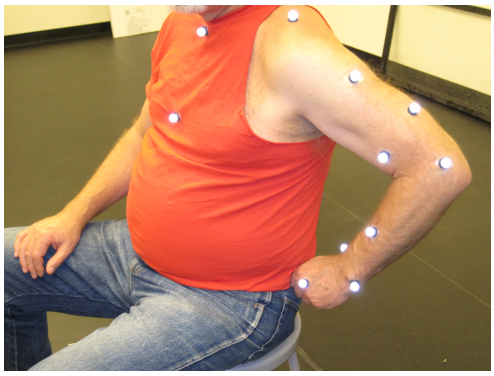
Scoring
0 = Hand does not move posterior to the frontal plane at the location of the anterior superior iliac spine.
1 = Hand does move posterior to the frontal plane at the location of the anterior superior iliac spine.
2 = Hand is placed on the small of the back with equivalent movement quality as the unaffected side.

Referenced from [29]

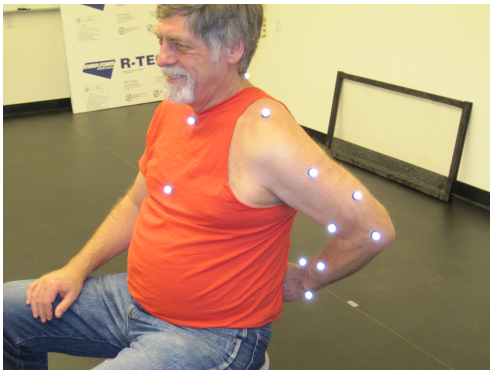
Due to the richness of data available from the SM, any scale can be developed. For example Fig. 6.3 is based on the residuals. Fig. 6.3 shows the difference between healthy subject's SM minus the gold standard SM as well as the stroke subject's SM minus the gold standard SM. Clearly evident is the difference between healthy and stroke. However the border



(a)



(b)



(c)

Figure 6.1: Scoring positions (a) = 0 (b) = 1(c) = 1 or 2? Depends.

cases are the most interesting to examine. Focusing on the difference between healthy subject 7 and stroke subject 22, both show total residual scores of 2.546 and 2.606 (SM scores for subjects 7 and 22 are 13 and 9), respectively. Focusing further on the individual synergies, the clinician can see the individual synergies and how they relate with one another in Fig. 6.2(a) and Fig. 6.2(b). This is where a user friendly interface can enhance the information being presented as suggested by Fig. 6.2. The residual matrix of numbers are assigned a color. The higher the number the more red the color (roygbiv – 1 to 0). Now a clinician can see that Fig. 6.2(c) shows the subject has strong synergies around shoulder abduction, shoulder flexion and wrist flex, and wrist ulnar deviation. Using Table 5.2 to decode the column numbers and with some future training on the squares relationships, a clinician can see with the colors that the wrist is held in max ROM in flexion and supination causing recruitment of the shoulder for motion when attempting to do wrist ulnar deviation.

The research is not without its limitations. The model described by (3.5) makes several important assumptions. First and foremost, this is a kinematic model. As such the model does not include arm dynamics, gravity, forces, or torques. Second, representing the human nervous system and limb anatomy as a geometric function might appear overly simplistic. Ideally, a more complete model based on first principals would likely include factors relating to, at a minimum, the nervous system and biomechanics.

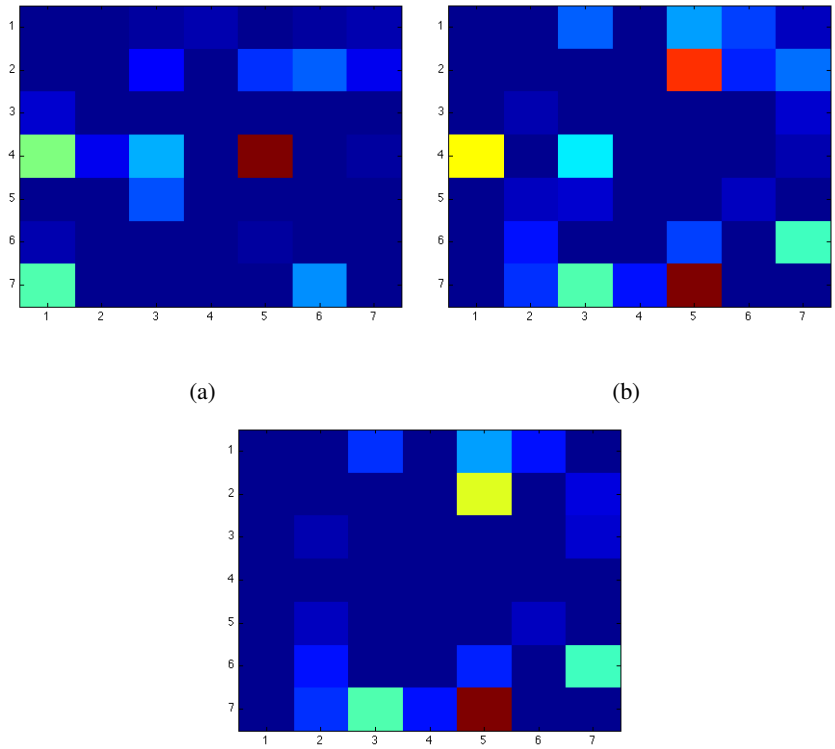
Third, (3.5) assumes that the same joint relationships apply regardless of the start or end angle. Strictly speaking, the models developed in this paper relate to the start and end positions depicted in Fig. 4.4.

Fourth, (3.5) assumes that there exists only one-way interactions between joints and

that a greater number of interactions do not exist. To take a physical example, one-way interactions assume that the simultaneous rotation of the wrist and elbow result in the same shoulder response as the combined effect of moving the wrist and elbow by themselves. While interactions greater than one are certainly possible, evaluation of such interactions are very difficult to measure in practice. Stroke survivors often have speech or cognitive effects. Communicating the need for such subjects to move two or more joints simultaneously while holding all other joints fixed was deemed too confusing. Therefore, in the strictest sense this model is most reliably applied to discrete, singular joint movements and it might apply to more complex multi-joint movements such as general reaching.

Fifth, (3.5) assumes that a single synergistic interaction is the same in the forward rotation direction as in the reverse direction, or that it is reversible. Stated another way, the model assumes that interactions are the same for joint flexion and extension. Note, this model does not assume that the matrices in (3.5) are symmetric. Even though the classical view of flexor and extensor synergies does not make this distinction, matrices described by (3.5) were generally not symmetric, i.e. interaction $(i, j) \neq (j, i)$ for $i \neq j$. Assumption one, two, and three are indeterminate using the foregoing protocol. However, the procedure does allow for an evaluation of reversibility. Quantifying reversibility is important because a largely irreversible interaction will result in a poor model fit no matter what model is used or how small the noise is. If irreversibility diminishes the fit enough, then separate functions in the forward and reverse directions would be required. By definition, a joint interaction is non-reversible if the flexion path differs from the extension path. This is never the case for linear interactions, but it is possible for nonlinear interactions.

Lastly, subjects represented only a subset of the stroke population that have some voluntary control and motor function of the stroke affected limb. In particular the ability to extend the elbow. Difficulty extending the elbow after stroke is common and this limitation is clinically observed as part of a flexor synergy pattern that produces concurrent flexion motions, and which also often impairs the stroke subject's ability to control individual joints [36]. Filtering for elbow extension commonly eliminates 20% of the stroke population [15]. A larger sample size would increase external validity by allowing for more generalizations to be made from this research to a population presenting varying degrees of motor impairment.



(a) (b)

(c)

Figure 6.2: (a) Show subject 7's residuals. (b) Shows subject 22 residuals. $(b) - (a) = (c)$ the residuals of b-a

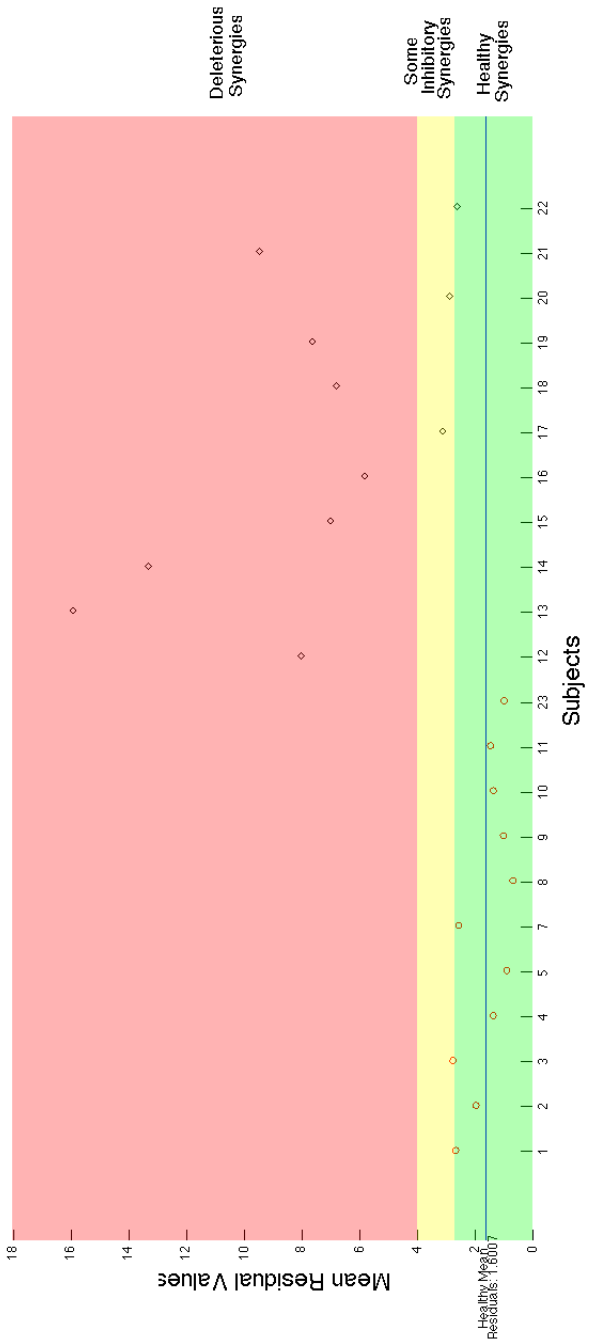


Figure 6.3: This graph displays the residuals of subjects' SM minus the gold standard healthy SM. The resulting residuals show healthy to deleterious synergies. Subjects 1, 3, 7, 22 are close on the boarder. Meaning the synergies show a restrictive by marginally healthy movement. Closer inspection of the actual synergies are needed.

Chapter 7

Conclusion

The goal of the research was to take the characterized synergies linear matrix model of a human arm and find an application with stroke survivors. The research successfully demonstrated an objective way to collect subject data, a novel description of the collected data in a functional matrix model, and a solution that addresses the deficiencies of the standard methods, such as measures of change and prognosis of recovery. By describing stroke synergies as a matrix, it allows for the use of more powerful mathematic tools. Applying these tools for categorizing subjects based on their movements was demonstrated as compared to a modified FM assessment. The SM could be useful in further studies that seek to find more subtle affects of rehabilitation or brain injury, such as physical therapy strategies for symmetric motion. The Vicon motion capture system used for this research is too expensive to use for most clinical settings, however, relatively high quality low cost position sensing devices are now commercially available such as the Microsoft Kinect which could be used to cost effectively gather similar data in a clinical setting.

Such systems could conceivably be used in conjunction with this model to provide finer-grained, more quantitative measures of synergy with better repeatability and reproducibility. Use of these tools and assessment strategies with stroke survivors would dramatically reduce clinical assessment and documentation times and provide increased high quality movement data to allow physical therapists to better focus rehabilitation therapies.

Finally, the methods described in this paper may allow for new types of algorithms for use in robotic physical therapy (RPT), such as an upper limb powered exoskeleton [28]. By the nature of their design, rehabilitation robots typically require high-precision position and force sensing. Implementing the synergy matrix model into the robotic system might allow for continuous monitoring of progress by tracking synergy reduction in stroke survivors. Additionally, the matrix might allow for new types of therapy whereby the robot tracks progress and modifies movement training in a way that targets a patients individual needs.

Appendix A

Code

A.1 Bodybuilder Code

```
{*UCSC Bionics Lab stroke synergy model*}

{*Use only with Model_UpperLimbRight.MP parameters and Model_UpperLimb.MKR or Model_UpperLimb_PiG.MKR*}

{*This file is supplied to illustrate the normal operation of BodyLanguage.
Oxford Metrics and Vicon Motion Systems accept no responsibility for its
correct operation*}

{*
Run first the model on a static trial. It calculates the local coordinates of the markers with respect to the correspondent
cluster of markers
*}
{*----------------------------------------------------------------------------------------------------------*}


{*START OF MACRO SECTION*}

Macro AxesVis(segment,axislength)
{* This macro creates segment axes for display purposes*}
segment#o={0,0,0}*segment
segment#1={axislength,0,0}*segment
segment#2={0,axislength,0}*segment
segment#3={0,0,axislength}*segment
output (segment#o,segment#1,segment#2,segment#3)
endmacro

{*----------------------------------------------------------------*}

macro REPLACE4(p1,p2,p3,p4)
{*Replaces any point missing from set of four fixed in a segment*}

{*SECTION FOR INITIALISATION OF VIRTUAL POINTS*}
{*REPLACE4*}
s123 = [p2,p1-p2,p2-p3]
p4V1 = Average(p4/s123)*s123
s124 = [p2,p1-p2,p2-p4]
p3V1 = Average(p3/s124)*s124
s134 = [p3,p1-p3,p3-p4]
p2V1 = Average(p2/s134)*s134
s234 = [p3,p2-p3,p3-p4]
p1V1 = Average(p1/s234)*s234

{*SECTION FOR SPECIFICATION OF VIRTUAL POINTS*}
p1 = p1 ? p1V1
p2 = p2 ? p2V1
p3 = p3 ? p3V1
p3 = p3 ? p3V1
p4 = p4 ? p4V1

output(p1,p2,p3,p4)

endmacro

{*----------------------------------------------------------------*}

macro calibratePoint(point,segment)
{*this macro caluclates the coordinates of a point (input argument) locally to a segment (input argument) and then stores
them in the parameter file of the subject*}

$%#point= point/segment                {*local coordinates calculation. See the operator '/' on the user manual: this
operator just operates the coordinate transformation from global to local*}

$%#point#X=$%#point(1)                 {*split the local coordinates in three; in order to have BB and WS behaving the same
way*}
```

```
$%#point#Y=$%#point(2)
$%#point#Z=$%#point(3)

param($%#point#X)                    {*Store the local coordinates into the parameter file. See the 'param' command on
the user manual*}
param($%#point#Y)                    {*it is worth noting that this macro does not write the local virtual point on the c3d
file; it just uses it for further needs, such as writing the coordinates on the mp file*}
param($%#point#Z)

param($%#point)                      {*For compatibility between BB and PiM*}

endmacro
{*-------------------------------------------------------------------------------------------------*}

macro reconstructPoint(P1,label)
{*This macro reads the local coordinates of the point from the .mp file and then recreates the calibrated virtual point in
the global space*}
                                                                                {*recreation of the calibrated point
P1 (input argument), locally to the label (input argument) segment*}
P1 = $%#P1*label                                                                {*coordinate
transformation from local to global*}
OUTPUT(P1)                                                                      {*this is the point
we want to write on the C3D file*}

endmacro

{*-------------------------------------------------------------------------------------------------*}

macro RotXZY(child, parent, joint)

{*
  Calculating the rotation matrix through which parent to child moves through three rotations
   sequence around the fixed axes xyz of the parent: it is just to go back to the matrix
   I will need for the numerical calculation of the angles that interest me
*}

joint#angles=<child,parent,xyz>
joint#alpha=joint#angles(1)
joint#beta=joint#angles(2)
joint#gamma=joint#angles(3)

{* Construct the numerical matrix from the values of the angles alpha, beta, gamma *}
joint#Rxyz1={cos(joint#beta)*cos(joint#gamma),sin(joint#alpha)*sin(joint#beta)*(cos(joint#gamma))-
(cos(joint#alpha)*sin(joint#gamma)),(cos(joint#alpha)*sin(joint#beta)*cos(joint#gamma))+
(sin(joint#alpha)*sin(joint#gamma))}
joint#Rxyz2={cos(joint#beta)*sin(joint#gamma),(sin(joint#alpha)*sin(joint#beta)*sin(joint#gamma))+
(cos(joint#alpha)*cos(joint#gamma)),(cos(joint#alpha)*sin(joint#beta)*sin(joint#gamma))-
(sin(joint#alpha)*cos(joint#gamma))}
joint#Rxyz3={-sin(joint#beta),sin(joint#alpha)*cos(joint#beta),cos(joint#alpha)*cos(joint#beta)}

{*
  At this point I can attach any type of numeric parameters to the array that I just
   found. For the sequence XZY the matrix will have the parametric seq. form:
  Rxzy = [c2c3       -s2          c2s3
            c1s2c3+s1s3       c1c2       c1s2s3-s1c3
            s1s2c3-c1s3        s1c2       s1s2s3+c1c3]

Nota: implementation of the procedure Z1-2 are my teta2;
                        X1-2 are my teta1;
                        Y1-2 are my teta3;
*}


z1 = asin(-(joint#Rxyz1(2)))
```

```
sx = joint#Rxyz3(2)/cos(z1)
cx = joint#Rxyz2(2)/cos(z1)
X1 = atan2(sx,cx)

sy = joint#Rxyz1(3)/cos(z1)
cy = joint#Rxyz1(1)/cos(z1)
y1 = atan2(sy,cy)

IF (z1>=0)
   z2 = 180 - z1
ELSE
   z2 = -180 - z1
ENDIF

sx2 = joint#Rxyz3(2)/cos(z2)
cx2 = joint#Rxyz2(2)/cos(z2)
x2 = atan2(sx2,cx2)

sy2 = joint#Rxyz1(3)/cos(z2)
cy2 = joint#Rxyz1(1)/cos(z2)
y2 = atan2(sy2,cy2)


IF ((-90<=z1) AND (z1<=90))
        x = x1
        y = y1
        z = z1
ELSE
        x = x2
        y = y2
        z = z2
ENDIF

joint#XZY = <x,z,y>
output(joint#XZY)

endmacro

{*-------------------------------------------------------------------------------------------------*}
{*-------------------------------------------------------------------------------------------------*}
{*------------------------------------ END OF MACRO SECTION ---------------------------------------*}
{*-------------------------------------------------------------------------------------------------*}
{*-------------------------------------------------------------------------------------------------*}

optionalPoints(RMEP)

replace4(RELB,RUPA,RUPB,RUPC)
replace4(C7,T10,CLAV,STRN)

{* Static part of the model *}

IF $STATIC == 1 THEN
                {* Create a technical reference frame associated with the markers ELB,UPA,UPB,UPC *}
        RUPTECH = [RELB,RUPA-RELB,RUPB-RELB,xyz]


                {* Calibrate the medial epycondile with respect to the cluster of markers formed by ELB,UPA,UPB,UPC
*}
        calibratePoint(RMEP,RUPTECH)


                {* Calculate the location of the centre of the glenohumeral joint and calibrate it with respect to the
UpperArm *}
```

```
		{* Define a dummy thorax segment to locate the glenohumeral joint centre *}
		{* Origin: STRN; Z axis: from STRN to CLAV; X axis: perpendicular to the plane formed by CLAV, STRN, C7
towards the right; Y axis follows *}
	thoraxGH = [STRN,CLAV-STRN,STRN-C7,zxy]

		{* Introduce the scale factor as researched by Ingram Murray *}
		{* Murray IA, Determining upper limb kinematics and dynamics during everyday tasks. PhD Thesis.
Ch6. University of Newcastle upon Tyne; 1999 *}
	GHscaleFactor = {0.0261,-0.0126,-0.1115}

		{* RIGHT SIDE *}
		{* Elbow joint centre calculation *}
	REJC = (RELB+RMEP)/2
	rightArmScalar = DIST(RSHO,REJC)
		{* Calculate the size of the vector from the Acromioclavicular Joint to the centre of rotation of the
gleno-humeral joint *}
	rightShoulderVector = rightArmScalar*GHscaleFactor

		{* Express the RSHO marker in the thoraxGH local coordinates *}
	%RSHO = RSHO/thoraxGH
		{* Locate the glenohumeral joint centre: it is the point at the tip of the vector rightShoulderVector with
orgin in %RSHO *}
	%RGH = %RSHO+rightShoulderVector
		{* Express the glenohumeral joint centre in global coordinates *}
	RGH = %RGH*thoraxGH
	output(RGH)
		{* Calibrate RGH with respect to the upperArm *}
	calibratePoint(RGH,RUPTECH)


		{* Code added on 2-17-12 by Matt Simkins to reference shoulder to global *}
		{* This creates a shoulder frame that must be be set up during the static calibration. *}
		{* Note, this is not the static cal from Nexus. The calibration position is seated in a chair with the elbow
bent at 90 deg. *}
	dumShoulderO = {0 + 300, 0, 0}
	dumShoulderX = {0 + 300 + 100, 0, 0}
	dumShoulderY = {0 + 300, 0 + 100, 0}
	dumShoulderZ = {0 + 300, 0, 0 + 100}
	calibratePoint(dumShoulderO,RUPTECH)
	calibratePoint(dumShoulderX,RUPTECH)
	calibratePoint(dumShoulderY,RUPTECH)
	calibratePoint(dumShoulderZ,RUPTECH)


ENDIF

IF $STATIC == 0 THEN
		{* Recreate the technical reference frames used to calibrate the points in the static *}
	RUPTECH = [RELB,RUPA-RELB,RUPB-RELB,xyz]


		{* Reconstruct the calibrated points *}
	reconstructPoint(RMEP,RUPTECH)
	reconstructPoint(RGH,RUPTECH)


		{* Thorax segment definition *}
		{* Note: all the segment are defined according to Cutti et al, Soft tissue artifact assessment in humeral
axial rotation, G&P 21 (2005), 341 - 349 *}
		{* Note: the directions of the axes for segment definition in static pose are referred to the subject
standing in anatomical position, with the hand palms directed forwards *}

	firstDefLine=(CLAV+C7)/2-(T10+STRN)/2
	Thorax = [CLAV,firstDefLine,STRN-T10,yxz]
```

```
	AxesVis(Thorax,100)

	{*--------------------------------------------*}
	{*RIGHT SIDE*}
	{*--------------------------------------------*}

	REJC = (RELB+RMEP)/2
	RWJC = (RWRB+RWRA)/2
	output(REJC,RWJC)

		{* Segment definitions *}
		{* Upper Arm *}
		{* Origin: RGH; Y axis: from REJC to RGH; X axis: perpendicular to the plane formed by RGH, REJC, RWJC
directed lateral; Z axis follows *}
	rightUpperArm = [RGH,RGH-REJC,RWJC-REJC,yxz]
	AxesVis(rightUpperArm,100)

		{* ForeArm *}
		{* Origin: REJC; Y axis: from RWJC to REJC; Z axis: perpendicular to the plane formed by REJC, RWRA,
RWRB directed posteriorly; X axis follows *}
	rightForeArm = [REJC,REJC-RWJC,RWRA-RWRB,yzx]
	AxesVis(rightForeArm,100)

		{* Hand *}
		{* Origin: RWJC; Y axis: from RFIN to RWJC; Z axis: perpendicular to the plane formed by RFIN, RWRA,
RWRB directed posteriorly; X axis follows *}
	rightHand = [RWJC,RWJC-RFIN,RWRA-RWRB,yzx]
	AxesVis(rightHand,100)


		{* Elbow *}
		{* Euler angles sequence: XZ'Y". Flexion-extension, ab-adduction, internal-external rotation *}
	RotXZY(rightForeArm, rightUpperArm, aRElbAngles)

		{* Wrist *}
		{* Euler angles sequence: XZ'Y". Flexion-extension, ab-adduction, internal-external rotation *}
	RotXZY(rightHand,rightForeArm,aRWristAngles)

		{* Joint angles calculation *}
		{* Shoulder *}
		{* Code was added by Matt Simkins on 2-17-12. *}
		{* Rebuild the shoulder frame that was created during the static calibration. *}
	reconstructPoint(dumShoulderO,RUPTECH)
	reconstructPoint(dumShoulderX,RUPTECH)
	reconstructPoint(dumShoulderY,RUPTECH)
	reconstructPoint(dumShoulderZ,RUPTECH)
	RupTechTwo = [dumShoulderO, dumShoulderX - dumShoulderO, dumShoulderZ - dumShoulderO, xyz]
	AxesVis(RupTechTwo,100)
	globe = [{0 +300, 0, 0},{1,0,0},{0,0,1},xyz] {* This orientation is consistent with the shoulder *}
	AxesVis(globe,100)
	RShoAnglesXYZ = -<globe,RupTechTwo,xyz>   {* Note, original model used child: "rightUpperArm" and parent:
"Thorax" *}
	output(RShoAnglesXYZ)
	RotXZY(RupTechTwo,globe,aRShoAngles)

ENDIF
```

```
{*UCSC Bionics Lab stroke synergy model*}

{*Use only with Model_UpperLimbRight.MP parameters and Model_UpperLimb.MKR or Model_UpperLimb_PiG.MKR*}

{*This file is supplied to illustrate the normal operation of BodyLanguage.
Oxford Metrics and Vicon Motion Systems accept no responsibility for its
correct operation*}

{*
Run first the model on a static trial. It calculates the local coordinates of the markers with respect to the correspondent
cluster of markers
*}
{*----------------------------------------------------------------------------------------------------------*}


{*START OF MACRO SECTION*}

Macro AxesVis(segment,axislength)
{* This macro creates segment axes for display purposes*}
segment#o={0,0,0}*segment
segment#1={axislength,0,0}*segment
segment#2={0,axislength,0}*segment
segment#3={0,0,axislength}*segment
output (segment#o,segment#1,segment#2,segment#3)
endmacro

{*----------------------------------------------------------------*}

macro REPLACE4(p1,p2,p3,p4)
{*Replaces any point missing from set of four fixed in a segment*}

{*SECTION FOR INITIALISATION OF VIRTUAL POINTS*}
{*REPLACE4*}
s123 = [p2,p1-p2,p2-p3]
p4V1 = Average(p4/s123)*s123
s124 = [p2,p1-p2,p2-p4]
p3V1 = Average(p3/s124)*s124
s134 = [p3,p1-p3,p3-p4]
p2V1 = Average(p2/s134)*s134
s234 = [p3,p2-p3,p3-p4]
p1V1 = Average(p1/s234)*s234

{*SECTION FOR SPECIFICATION OF VIRTUAL POINTS*}
p1 = p1 ? p1V1
p2 = p2 ? p2V1
p3 = p3 ? p3V1
p3 = p3 ? p3V1
p4 = p4 ? p4V1

output(p1,p2,p3,p4)

endmacro

{*----------------------------------------------------------------*}

macro calibratePoint(point,segment)
{*this macro caluclates the coordinates of a point (input argument) locally to a segment (input argument) and then stores
them in the parameter file of the subject*}

$%#point= point/segment                {*local coordinates calculation. See the operator '/' on the user manual: this
operator just operates the coordinate transformation from global to local*}

$%#point#X=$%#point(1)                 {*split the local coordinates in three; in order to have BB and WS behaving the same
way*}
```

```
$%#point#Y=$%#point(2)
$%#point#Z=$%#point(3)

param($%#point#X)                    {*Store the local coordinates into the parameter file. See the 'param' command on
the user manual*}
param($%#point#Y)                    {*it is worth noting that this macro does not write the local virtual point on the c3d
file; it just uses it for further needs, such as writing the coordinates on the mp file*}
param($%#point#Z)

param($%#point)                      {*For compatibility between BB and PiM*}

endmacro
{*-------------------------------------------------------------------------------------------------*}

macro reconstructPoint(P1,label)
{*This macro reads the local coordinates of the point from the .mp file and then recreates the calibrated virtual point in
the global space*}
                                                                                {*recreation of the calibrated point
P1 (input argument), locally to the label (input argument) segment*}
P1 = $%#P1*label                                                                {*coordinate
transformation from local to global*}
OUTPUT(P1)                                                                      {*this is the point
we want to write on the C3D file*}

endmacro

{*-------------------------------------------------------------------------------------------------*}

macro RotXZY(child, parent, joint)

{*
  Calculating the rotation matrix through which parent to child moves through three rotations
   sequence around the fixed axes xyz of the parent: it is just to go back to the matrix
   I will need for the numerical calculation of the angles that interest me
*}

joint#angles=<child,parent,xyz>
joint#alpha=joint#angles(1)
joint#beta=joint#angles(2)
joint#gamma=joint#angles(3)

{* Construct the numerical matrix from the values of the angles alpha, beta, gamma *}
joint#Rxyz1={cos(joint#beta)*cos(joint#gamma),sin(joint#alpha)*sin(joint#beta)*(cos(joint#gamma))-
(cos(joint#alpha)*sin(joint#gamma)),(cos(joint#alpha)*sin(joint#beta)*cos(joint#gamma))+
(sin(joint#alpha)*sin(joint#gamma))}
joint#Rxyz2={cos(joint#beta)*sin(joint#gamma),(sin(joint#alpha)*sin(joint#beta)*sin(joint#gamma))+
(cos(joint#alpha)*cos(joint#gamma)),(cos(joint#alpha)*sin(joint#beta)*sin(joint#gamma))-
(sin(joint#alpha)*cos(joint#gamma))}
joint#Rxyz3={-sin(joint#beta),sin(joint#alpha)*cos(joint#beta),cos(joint#alpha)*cos(joint#beta)}

{*
  At this point I can attach any type of numeric parameters to the array that I just
   found. For the sequence XZY the matrix will have the parametric seq. form:
  Rxzy = [c2c3       -s2          c2s3
            c1s2c3+s1s3       c1c2       c1s2s3-s1c3
            s1s2c3-c1s3        s1c2       s1s2s3+c1c3]

Nota: nell'implementazione della procedura Z1-2 sono il mio teta2;
                          X1-2 sono il mio teta1;
                          Y1-2 sono il mio teta3;
*}


z1 = asin(-(joint#Rxyz1(2)))
```

```
sx = joint#Rxyz3(2)/cos(z1)
cx = joint#Rxyz2(2)/cos(z1)
X1 = atan2(sx,cx)

sy = joint#Rxyz1(3)/cos(z1)
cy = joint#Rxyz1(1)/cos(z1)
y1 = atan2(sy,cy)

IF (z1>=0)
   z2 = 180 - z1
ELSE
   z2 = -180 - z1
ENDIF

sx2 = joint#Rxyz3(2)/cos(z2)
cx2 = joint#Rxyz2(2)/cos(z2)
x2 = atan2(sx2,cx2)

sy2 = joint#Rxyz1(3)/cos(z2)
cy2 = joint#Rxyz1(1)/cos(z2)
y2 = atan2(sy2,cy2)


IF ((-90<=z1) AND (z1<=90))
        x = x1
        y = y1
        z = z1
ELSE
        x = x2
        y = y2
        z = z2
ENDIF

joint#XZY = <x,z,y>
output(joint#XZY)

endmacro

{*-------------------------------------------------------------------------------------------------*}
{*-------------------------------------------------------------------------------------------------*}
{*------------------------------------ END OF MACRO SECTION ---------------------------------------*}
{*-------------------------------------------------------------------------------------------------*}
{*-------------------------------------------------------------------------------------------------*}

optionalPoints(RMEP)

replace4(RELB,RUPA,RUPB,RUPC)
replace4(C7,T10,CLAV,STRN)

{* Static part of the model *}

IF $STATIC == 1 THEN
                {* Create a technical reference frame associated with the markers ELB,UPA,UPB,UPC *}
        RUPTECH = [RELB,RUPA-RELB,RUPB-RELB,xyz]


                {* Calibrate the medial epycondile with respect to the cluster of markers formed by ELB,UPA,UPB,UPC
*}
        calibratePoint(RMEP,RUPTECH)


                {* Calculate the location of the centre of the glenohumeral joint and calibrate it with respect to the
UpperArm *}
```

```
		{* Define a dummy thorax segment to locate the glenohumeral joint centre *}
		{* Origin: STRN; Z axis: from STRN to CLAV; X axis: perpendicular to the plane formed by CLAV, STRN, C7
towards the right; Y axis follows *}
	thoraxGH = [STRN,CLAV-STRN,STRN-C7,zxy]

		{* Introduce the scale factor as researched by Ingram Murray *}
		{* Murray IA, Determining upper limb kinematics and dynamics during everyday tasks. PhD Thesis.
Ch6. University of Newcastle upon Tyne; 1999 *}
	GHscaleFactor = {0.0261,-0.0126,-0.1115}

		{* RIGHT SIDE *}
		{* Elbow joint centre calculation *}
	REJC = (RELB+RMEP)/2
	rightArmScalar = DIST(RSHO,REJC)
		{* Calculate the size of the vector from the Acromioclavicular Joint to the centre of rotation of the
gleno-humeral joint *}
	rightShoulderVector = rightArmScalar*GHscaleFactor

		{* Express the RSHO marker in the thoraxGH local coordinates *}
	%RSHO = RSHO/thoraxGH
		{* Locate the glenohumeral joint centre: it is the point at the tip of the vector rightShoulderVector with
orgin in %RSHO *}
	%RGH = %RSHO+rightShoulderVector
		{* Express the glenohumeral joint centre in global coordinates *}
	RGH = %RGH*thoraxGH
	output(RGH)
		{* Calibrate RGH with respect to the upperArm *}
	calibratePoint(RGH,RUPTECH)


		{* Code added on 2-17-12 by Matt Simkins to reference shoulder to global *}
		{* This creates a shoulder frame that must be be set up during the static calibration. *}
		{* Note, this is not the static cal from Nexus. The calibration position is seated in a chair with the elbow
bent at 90 deg. *}
	dumShoulderO = {0 + 300, 0, 0}
	dumShoulderX = {0 + 300 + 100, 0, 0}
	dumShoulderY = {0 + 300, 0 + 100, 0}
	dumShoulderZ = {0 + 300, 0, 0 + 100}
	calibratePoint(dumShoulderO,RUPTECH)
	calibratePoint(dumShoulderX,RUPTECH)
	calibratePoint(dumShoulderY,RUPTECH)
	calibratePoint(dumShoulderZ,RUPTECH)


ENDIF

IF $STATIC == 0 THEN
		{* Recreate the technical reference frames used to calibrate the points in the static *}
	RUPTECH = [RELB,RUPA-RELB,RUPB-RELB,xyz]


		{* Reconstruct the calibrated points *}
	reconstructPoint(RMEP,RUPTECH)
	reconstructPoint(RGH,RUPTECH)


		{* Thorax segment definition *}
		{* Note: all the segment are defined according to Cutti et al, Soft tissue artifact assessment in humeral
axial rotation, G&P 21 (2005), 341 - 349 *}
		{* Note: the directions of the axes for segment definition in static pose are referred to the subject
standing in anatomical position, with the hand palms directed forwards *}

	firstDefLine=(CLAV+C7)/2-(T10+STRN)/2
	Thorax = [CLAV,firstDefLine,STRN-T10,yxz]
```

```
	AxesVis(Thorax,100)

	{*--------------------------------------------*}
	{*RIGHT SIDE*}
	{*--------------------------------------------*}

	REJC = (RELB+RMEP)/2
	RWJC = (RWRB+RWRA)/2
	output(REJC,RWJC)

		{* Segment definitions *}
		{* Upper Arm *}
		{* Origin: RGH; Y axis: from REJC to RGH; X axis: perpendicular to the plane formed by RGH, REJC, RWJC
directed lateral; Z axis follows *}
	rightUpperArm = [RGH,RGH-REJC,RWJC-REJC,yxz]
	AxesVis(rightUpperArm,100)

		{* ForeArm *}
		{* Origin: REJC; Y axis: from RWJC to REJC; Z axis: perpendicular to the plane formed by REJC, RWRA,
RWRB directed posteriorly; X axis follows *}
	rightForeArm = [REJC,REJC-RWJC,RWRA-RWRB,yzx]
	AxesVis(rightForeArm,100)

		{* Hand *}
		{* Origin: RWJC; Y axis: from RFIN to RWJC; Z axis: perpendicular to the plane formed by RFIN, RWRA,
RWRB directed posteriorly; X axis follows *}
	rightHand = [RWJC,RWJC-RFIN,RWRA-RWRB,yzx]
	AxesVis(rightHand,100)


		{* Elbow *}
		{* Euler angles sequence: XZ'Y". Flexion-extension, ab-adduction, internal-external rotation *}
	RotXZY(rightForeArm, rightUpperArm, aRElbAngles)

		{* Wrist *}
		{* Euler angles sequence: XZ'Y". Flexion-extension, ab-adduction, internal-external rotation *}
	RotXZY(rightHand,rightForeArm,aRWristAngles)

		{* Joint angles calculation *}
		{* Shoulder *}
		{* Code was added by Matt Simkins on 2-17-12. *}
		{* Rebuild the shoulder frame that was created during the static calibration. *}
	reconstructPoint(dumShoulderO,RUPTECH)
	reconstructPoint(dumShoulderX,RUPTECH)
	reconstructPoint(dumShoulderY,RUPTECH)
	reconstructPoint(dumShoulderZ,RUPTECH)
	RupTechTwo = [dumShoulderO, dumShoulderX - dumShoulderO, dumShoulderZ - dumShoulderO, xyz]
	AxesVis(RupTechTwo,100)
	globe = [{0 +300, 0, 0},{1,0,0},{0,0,1},xyz] {* This orientation is consistent with the shoulder *}
	AxesVis(globe,100)
	RShoAnglesXYZ = -<globe,RupTechTwo,xyz>   {* Note, original model used child: "rightUpperArm" and parent:
"Thorax" *}
	output(RShoAnglesXYZ)
	RotXZY(RupTechTwo,globe,aRShoAngles)

ENDIF{*UCSC Bionics Lab stroke synergy left arm model*}


{*
Run first the model on a static trial. It calculates the local coordinates of the markers with respect to the correspondent
cluster of markers
*}
{*--------------------------------------------------------------------------------------------------*}
```

```
{*START OF MACRO SECTION*}

Macro AxesVis(segment,axislength)
{* This macro creates segment axes for display purposes*}
segment#o={0,0,0}*segment
segment#1={axislength,0,0}*segment
segment#2={0,axislength,0}*segment
segment#3={0,0,axislength}*segment
output (segment#o,segment#1,segment#2,segment#3)
endmacro

{*-------------------------------------------------------------------*}

macro REPLACE4(p1,p2,p3,p4)
{*Replaces any point missing from set of four fixed in a segment*}

{*SECTION FOR INITIALISATION OF VIRTUAL POINTS*}
{*REPLACE4*}
s123 = [p2,p1-p2,p2-p3]
p4V1 = Average(p4/s123)*s123
s124 = [p2,p1-p2,p2-p4]
p3V1 = Average(p3/s124)*s124
s134 = [p3,p1-p3,p3-p4]
p2V1 = Average(p2/s134)*s134
s234 = [p3,p2-p3,p3-p4]
p1V1 = Average(p1/s234)*s234

{*SECTION FOR SPECIFICATION OF VIRTUAL POINTS*}
p1 = p1 ? p1V1
p2 = p2 ? p2V1
p3 = p3 ? p3V1
p3 = p3 ? p3V1
p4 = p4 ? p4V1

output(p1,p2,p3,p4)

endmacro

{*-------------------------------------------------------------------*}

macro calibratePoint(point,segment)
{*this macro caluclates the coordinates of a point (input argument) locally to a segment (input argument) and then stores them in the parameter file of the subject*}

$%#point= point/segment          {*local coordinates calculation. See the operator '/' on the user manual: this operator just operates the coordinate transformation from global to local*}

$%#point#X=$%#point(1)           {*split the local coordinates in three; in order to have BB and WS behaving the same way*}
$%#point#Y=$%#point(2)
$%#point#Z=$%#point(3)

param($%#point#X)                {*Store the local coordinates into the parameter file. See the 'param' command on the user manual*}
param($%#point#Y)                {*it is worth noting that this macro does not write the local virtual point on the c3d file; it just uses it for further needs, such as writing the coordinates on the mp file*}
param($%#point#Z)

param($%#point)                  {*For compatibility between BB and PiM*}

endmacro
{*-------------------------------------------------------------------------------------------------*}
```

```
macro reconstructPoint(P1,label)
{*This macro reads the local coordinates of the point from the .mp file and then recreates the calibrated virtual point in
the global space*}
                                                                                {*recreation of the calibrated point
P1 (input argument), locally to the label (input argument) segment*}
P1 = $%#P1*label                                                                {*coordinate
transformation from local to global*}
OUTPUT(P1)                                                                      {*this is the point
we want to write on the C3D file*}

endmacro

{*-------------------------------------------------------------------------------------------------------*}

macro RotXZY(child, parent, joint)

{*
  Calcolo la matrice di rotazione attraverso cui parent si porta su child attraverso tre rotazioni
  con sequenza xyz intorno agli ASSI FISSI del parent: lo si fa al solo scopo di risalire alla matrice
  numerica che mi servirà per il calcolo degli angoli che mi interessano
*}

joint#angles=<child,parent,xyz>
joint#alpha=joint#angles(1)
joint#beta=joint#angles(2)
joint#gamma=joint#angles(3)

{* Costruisco la matrice numerica a partire dai valori degli angoli alfa,beta,gamma *}
joint#Rxyz1={cos(joint#beta)*cos(joint#gamma),sin(joint#alpha)*sin(joint#beta)*(cos(joint#gamma))-
(cos(joint#alpha)*sin(joint#gamma)),(cos(joint#alpha)*sin(joint#beta)*cos(joint#gamma))+
(sin(joint#alpha)*sin(joint#gamma))}
joint#Rxyz2={cos(joint#beta)*sin(joint#gamma),(sin(joint#alpha)*sin(joint#beta)*sin(joint#gamma))+
(cos(joint#alpha)*cos(joint#gamma)),(cos(joint#alpha)*sin(joint#beta)*sin(joint#gamma))-
(sin(joint#alpha)*cos(joint#gamma))}
joint#Rxyz3={-sin(joint#beta),sin(joint#alpha)*cos(joint#beta),cos(joint#alpha)*cos(joint#beta)}

{*
  A questo punto posso associare qualsiasi tipo di parametrizzazione alla matrice numerica che ho appena
  trovato. Per la sequenza XZY la matrice parametrica avrà la seg. forma:
  Rxzy = [c2c3       -s2         c2s3
          c1s2c3+s1s3      c1c2      c1s2s3-s1c3
          s1s2c3-c1s3       s1c2      s1s2s3+c1c3]

Nota: nell'implementazione della procedura Z1-2 sono il mio teta2;
                        X1-2 sono il mio teta1;
                        Y1-2 sono il mio teta3;
*}


z1 = asin(-(joint#Rxyz1(2)))

sx = joint#Rxyz3(2)/cos(z1)
cx = joint#Rxyz2(2)/cos(z1)
X1 = atan2(sx,cx)

sy = joint#Rxyz1(3)/cos(z1)
cy = joint#Rxyz1(1)/cos(z1)
y1 = atan2(sy,cy)

IF (z1>=0)
   z2 = 180 - z1
ELSE
   z2 = -180 - z1
ENDIF
```

```
sx2 = joint#Rxyz3(2)/cos(z2)
cx2 = joint#Rxyz2(2)/cos(z2)
x2 = atan2(sx2,cx2)

sy2 = joint#Rxyz1(3)/cos(z2)
cy2 = joint#Rxyz1(1)/cos(z2)
y2 = atan2(sy2,cy2)


IF ((-90<=z1) AND (z1<=90))
        x = x1
        y = y1
        z = z1
ELSE
        x = x2
        y = y2
        z = z2
ENDIF

joint#XZY = <x,z,y>
output(joint#XZY)

endmacro

{*-------------------------------------------------------------------------------------------------*}
{*-------------------------------------------------------------------------------------------------*}
{*------------------------------------ END OF MACRO SECTION ---------------------------------------*}
{*-------------------------------------------------------------------------------------------------*}
{*-------------------------------------------------------------------------------------------------*}

optionalPoints(LMEP)


replace4(LELB,LUPA,LUPB,LUPC)
replace4(C7,T10,CLAV,STRN)

{* Static part of the model *}

IF $STATIC == 1 THEN
                {* Create a technical reference frame associated with the markers ELB,UPA,UPB,UPC *}

        LUPTECH = [LELB,LUPA-LELB,LUPB-LELB,xyz]

                {* Calibrate the medial epycondile with respect to the cluster of markers formed by ELB,UPA,UPB,UPC
*}

        calibratePoint(LMEP,LUPTECH)

                {* Calculate the location of the centre of the glenohumeral joint and calibrate it with respect to the
UpperArm *}
                {* Define a dummy thorax segment to locate the glenohumeral joint centre *}
                {* Origin: STRN; Z axis: from STRN to CLAV; X axis: perpendicular to the plane formed by CLAV, STRN, C7
towards the right; Y axis follows *}
        thoraxGH = [STRN,CLAV-STRN,STRN-C7,zxy]

                {* Introduce the scale factor as researched by Ingram Murray *}
                {* Murray IA, Determining upper limb kinematics and dynamics during everyday tasks. PhD Thesis.
Ch6. University of Newcastle upon Tyne; 1999 *}
        GHscaleFactor = {0.0261,-0.0126,-0.1115}
```

```
		{*LEFT SIDE*}
		{* For description of the command lines, please refer to the right side *}
	LEJC = (LELB+LMEP)/2
	leftArmScalar = DIST(LSHO,LEJC)
		{* Calculate the size of the vector from the Acromioclavicular Joint to the centre of rotation of the
gleno-humeral joint *}
	leftShoulderVector = leftArmScalar*GHscaleFactor
		{* Sign change needed for the first component of the left shoulder vector, because the x axis of the
thoraxGH segment is oriented towards the right *}
	leftShoulderVector = {-leftShoulderVector(1),leftShoulderVector(2),leftShoulderVector(3)}

	%LSHO = LSHO/thoraxGH
	%LGH = %LSHO+leftShoulderVector
	LGH = %LGH*thoraxGH
	output(LGH)
	calibratePoint(LGH,LUPTECH)



		{* Code added on 2-18-12 by Matt Simkins to refence shoulder to global *}
		{* This creates a shoulder frame that must be be set up during the static calibration. *}
		{* Note, this is not the static cal from Nexus. The calibration position is seated in a chair with the elbow
bent at 90 deg. *}
	dumShoulderO = {0 + 300, 0, 0}
	dumShoulderX = {0 + 300 + 100, 0, 0}
	dumShoulderY = {0 + 300, 0 + 100, 0}
	dumShoulderZ = {0 + 300, 0, 0 + 100}
	calibratePoint(dumShoulderO,LUPTECH)
	calibratePoint(dumShoulderX,LUPTECH)
	calibratePoint(dumShoulderY,LUPTECH)
	calibratePoint(dumShoulderZ,LUPTECH)

ENDIF

IF $STATIC == 0 THEN
		{* Recreate the technical reference frames used to calibrate the points in the static *}

	LUPTECH = [LELB,LUPA-LELB,LUPB-LELB,xyz]

		{* Reconstruct the calibrated points *}

	reconstructPoint(LMEP,LUPTECH)
	reconstructPoint(LGH,LUPTECH)

		{* Thorax segment definition *}
		{* Note: all the segment are defined according to Cutti et al, Soft tissue artefact assessment in humeral
axial rotation, G&P 21 (2005), 341 - 349 *}
		{* Note: the directions of the axes for segment definition in static pose are referred to the subject
standing in anatomical position, with the hand palms directed forwards *}

	firstDefLine=(CLAV+C7)/2-(T10+STRN)/2
	Thorax = [CLAV,firstDefLine,STRN-T10,yxz]
	AxesVis(Thorax,100)

	{*---------------------------------------------*}
	{*LEFT SIDE*}
	{*---------------------------------------------*}

	LEJC = (LELB+LMEP)/2
	LWJC = (LWRB+LWRA)/2
	output(LEJC,LWJC)

		{* Segment definitions *}
		{* Note: the directions of the axes are the same as the right side, NOT symmetrical! *}
```

```
		{* Upper Arm *}
	leftUpperArm = [LGH,LGH-LEJC,LWJC-LEJC,yxz]
	AxesVis(leftUpperArm,100)

		{* ForeArm *}
	leftForeArm = [LEJC,LEJC-LWJC,LWRB-LWRA,yzx]
	AxesVis(leftForeArm,100)

		{* Hand *}
	leftHand = [LWJC,LWJC-LFIN,LWRB-LWRA,yzx]
	AxesVis(leftHand,100)

		{* Joint Angles calculation *}
		{* Shoulder *}

		{* Elbow *}
	RotXZY(leftForeArm,leftUpperArm,aLElbAngles)

		{* Wrist *}
	RotXZY(leftHand,leftForeArm,aLWristAngles)


		{* Joint angles calculation *}
		{* Shoulder *}
		{* Code was added by Matt Simkins on 2-18-12. *}
		{* Rebuild the shoulder frame that was created during the static calibration. *}
	reconstructPoint(dumShoulderO,LUPTECH)
	reconstructPoint(dumShoulderX,LUPTECH)
	reconstructPoint(dumShoulderY,LUPTECH)
	reconstructPoint(dumShoulderZ,LUPTECH)
	LupTechTwo = [dumShoulderO, dumShoulderX - dumShoulderO, dumShoulderZ - dumShoulderO, xyz]
	AxesVis(LupTechTwo,100)
	globe = [{0 +300, 0, 0},{1,0,0},{0,0,1},xyz] {* This orientation is consistent with the shoulder *}
	AxesVis(globe,100)
	LShoAnglesXYZ = -<globe,LupTechTwo,xyz>    {* Note, original model used child: "lefttUpperArm" and parent:
"Thorax" *}
	output(LShoAnglesXYZ)
	RotXZY(LupTechTwo,globe,aLShoAngles)



ENDIF
```

A.2 MATLAB Code

```
% top level processing script which call all subsequent Matlab custom functions included below
clear;
clc;
group1 = {'s1' 's2' 's3' 's4' 's5' 's6' 's7' 's8' 's9' 's10' ...
          's11' 's12' 's13' 's14' 's15' 's16' 's17' 's18' 's19'...
          's20' 's21' 's22' 's23'}' ; % Cell array of strings
group2 = {'s1' 's2' 's3' 's4' 's5' 's7' 's8' 's9' 's10' 's11' 's23'}' ; % Cell array of strings
group3 = {'s12' 's13' 's14' 's15' 's16' 's17' 's18' 's19' 's20' 's21' 's22'}' ; % Cell array of
strings
colorizeData([160 0 0 0 0 0 0;...
              0 90 0 0 0 0 0;...
              0 0 90 0 0 0 0;...
              0 0 0 130 0 0 0;...
              0 0 0 0 90 0 0;...
              0 0 0 0 0 90 0;...
              0 0 0 0 0 0 55])
AmeanTotal = [];
%%
%
%  subject 1 age matched
%
data = '/Users/<usrname>/Desktop/research/SynJointData/Subject 1 Age Match/1st_Order_Synergy';
A1_tem = xlsread(data, 'Full Range1');
A1= [A1_tem(:,1) A1_tem(:,2) A1_tem(:,3) A1_tem(:,4) A1_tem(:,5) A1_tem(:,6) A1_tem(:,7)];
A2_tem = xlsread(data, 'Full Range2');
A2= [A2_tem(:,1) A2_tem(:,2) A2_tem(:,3) A2_tem(:,4) A2_tem(:,5) A2_tem(:,6) A2_tem(:,7)];
A3_tem = xlsread(data, 'Full Range3');
A3= [A3_tem(:,1) A3_tem(:,2) A3_tem(:,3) A3_tem(:,4) A3_tem(:,5) A3_tem(:,6) A3_tem(:,7)];
AmeanTotal = A1(:,:)+A2(:,:)+A3(:,:);
Amean = (A1(:,:)+A2(:,:)+A3(:,:))./3;
[ xInputMatrix1 xInputMatrix2 xInputMatrix3 xInputMatrixAve xInputThetas1 xInputThetas2
xInputThetas3 xInputThetasAve ] = xInputMatrixMaker('/Users/<usrname>/Desktop/research/
SynJointData/Subject 1 Age Match/');

[ FMS1_s1_syn_ideal FMScore1_s1_syn_ideal ] = FMScore( abs(A1*xInputThetas1) )
[ FMS1_s1_syn_raw FMScore1_s1_syn_raw ] = FMScore( abs(A1*xInputMatrix1) )
[ FMS1_s1_raw FMScore1_s1_raw ] = FMScore( xInputMatrix1 )
%colorizeData(abs(A1*xInputThetas1))

[ FMS2_s1_syn_ideal FMScore2_s1_syn_ideal ] = FMScore( abs(A2*xInputThetas2) )
[ FMS2_s1_syn_raw FMScore2_s1_syn_raw ] = FMScore( abs(A2*xInputMatrix2) )
[ FMS2_s1_raw FMScore2_s1_raw ] = FMScore( xInputMatrix2 )


[ FMS3_s1_syn_ideal FMScore3_s1_syn_ideal ] = FMScore( abs(A3*xInputThetas3) )
[ FMS3_s1_syn_raw FMScore3_s1_syn_raw ] = FMScore( abs(A3*xInputMatrix3) )
[ FMS3_s1_raw FMScore3_s1_raw ] = FMScore( xInputMatrix3 )


[ FMSave_s1_syn_ideal FMScoreAve_s1_syn_ideal ] = FMScore( abs(Amean*xInputThetasAve) )
[ FMSave_s1_syn_raw FMScoreAve_s1_syn_raw ] = FMScore( abs(Amean*xInputMatrixAve) )
[ FMSave_s1_raw FMScoreAve_s1_raw ] = FMScore( xInputMatrixAve )


%%
%
%  subject 2
%
data = '/Users/<usrname>/Desktop/research/SynJointData/Subject 2/1st_Order_Synergy';
A1_tem = xlsread(data, 'Full Range1');
A1= [A1_tem(:,1) A1_tem(:,2) A1_tem(:,3) A1_tem(:,4) A1_tem(:,5) A1_tem(:,6) A1_tem(:,7)];
A2_tem = xlsread(data, 'Full Range2');
A2= [A2_tem(:,1) A2_tem(:,2) A2_tem(:,3) A2_tem(:,4) A2_tem(:,5) A2_tem(:,6) A2_tem(:,7)];
A3_tem = xlsread(data, 'Full Range3');
```

```
A3= [A3_tem(:,1) A3_tem(:,2) A3_tem(:,3) A3_tem(:,4) A3_tem(:,5) A3_tem(:,6) A3_tem(:,7)];
AmeanTotal = AmeanTotal(:,:)+A1(:,:)+A2(:,:)+A3(:,:);
Amean = (A1(:,:)+A2(:,:)+A3(:,:))./3;
[ xInputMatrix1 xInputMatrix2 xInputMatrix3 xInputMatrixAve xInputThetas1 xInputThetas2
xInputThetas3 xInputThetasAve ] = xInputMatrixMaker('/Users/<usrname>/Desktop/research/
SynJointData/Subject 2/');

[ FMS1_s2_syn_ideal FMScore1_s2_syn_ideal ] = FMScore( abs(A1*xInputThetas1) )
[ FMS1_s2_syn_raw FMScore1_s2_syn_raw ] = FMScore(abs(A1*xInputMatrix1))
[ FMS1_s2_raw FMScore1_s2_raw ] = FMScore( xInputMatrix1 )
%colorizeData(abs(A1*xInputThetas1))

[ FMS2_s2_syn_ideal FMScore2_s2_syn_ideal ] = FMScore( abs(A2*xInputThetas2) )
[ FMS2_s2_syn_raw FMScore2_s2_syn_raw ] = FMScore( abs(A2*xInputMatrix2) )
[ FMS2_s2_raw FMScore2_s2_raw ] = FMScore( xInputMatrix2 )

[ FMS3_s2_syn_ideal FMScore3_s2_syn_ideal ] = FMScore( abs(A3*xInputThetas3) )
[ FMS3_s2_syn_raw FMScore3_s2_syn_raw ] = FMScore( abs(A3*xInputMatrix3) )
[ FMS3_s2_raw FMScore3_s2_raw ] = FMScore( xInputMatrix3 )

[ FMSave_s2_syn_ideal FMScoreAve_s2_syn_ideal ] = FMScore( abs(Amean*xInputThetasAve) )
[ FMSave_s2_syn_raw FMScoreAve_s2_syn_raw ] = FMScore( abs(Amean*xInputMatrixAve) )
[ FMSave_s2_raw FMScoreAve_s2_raw ] = FMScore( xInputMatrixAve )


%%
%
%  subject 3 age matched
%
data = '/Users/<usrname>/Desktop/research/SynJointData/Subject 3 Age Match/1st_Order_Synergy';
A1_tem = xlsread(data, 'Full Range1');
A1= [A1_tem(:,1) A1_tem(:,2) A1_tem(:,3) A1_tem(:,4) A1_tem(:,5) A1_tem(:,6) A1_tem(:,7)];
A2_tem = xlsread(data, 'Full Range2');
A2= [A2_tem(:,1) A2_tem(:,2) A2_tem(:,3) A2_tem(:,4) A2_tem(:,5) A2_tem(:,6) A2_tem(:,7)];
A3_tem = xlsread(data, 'Full Range3');
A3= [A3_tem(:,1) A3_tem(:,2) A3_tem(:,3) A3_tem(:,4) A3_tem(:,5) A3_tem(:,6) A3_tem(:,7)];
AmeanTotal = AmeanTotal(:,:)+A1(:,:)+A2(:,:)+A3(:,:);
Amean = (A1(:,:)+A2(:,:)+A3(:,:))./3;
[ xInputMatrix1 xInputMatrix2 xInputMatrix3 xInputMatrixAve xInputThetas1 xInputThetas2
xInputThetas3 xInputThetasAve ] = xInputMatrixMaker('/Users/<usrname>/Desktop/research/
SynJointData/Subject 3 Age Match/');

[ FMS1_s3_syn_ideal FMScore1_s3_syn_ideal ] = FMScore( abs(A1*xInputThetas1) )
[ FMS1_s3_syn_raw FMScore1_s3_syn_raw ] = FMScore(abs(A1*xInputMatrix1))
[ FMS1_s3_raw FMScore1_s3_raw ] = FMScore( xInputMatrix1 )
%colorizeData(abs(A1*xInputThetas1))

[ FMS2_s3_syn_ideal FMScore2_s3_syn_ideal ] = FMScore( abs(A2*xInputThetas2) )
[ FMS2_s3_syn_raw FMScore2_s3_syn_raw ] = FMScore( abs(A2*xInputMatrix2) )
[ FMS2_s3_raw FMScore2_s3_raw ] = FMScore( xInputMatrix2 )

[ FMS3_s3_syn_ideal FMScore3_s3_syn_ideal ] = FMScore( abs(A3*xInputThetas3) )
[ FMS3_s3_syn_raw FMScore3_s3_syn_raw ] = FMScore( abs(A3*xInputMatrix3) )
[ FMS3_s3_raw FMScore3_s3_raw ] = FMScore( xInputMatrix3 )

[ FMSave_s3_syn_ideal FMScoreAve_s3_syn_ideal ] = FMScore( abs(Amean*xInputThetasAve) )
[ FMSave_s3_syn_raw FMScoreAve_s3_syn_raw ] = FMScore( abs(Amean*xInputMatrixAve) )
[ FMSave_s3_raw FMScoreAve_s3_raw ] = FMScore( xInputMatrixAve )


%%
%
%  subject 4 age matched
%
```

```
data = '/Users/<usrname>/Desktop/research/SynJointData/Subject 4 Age Match/1st_Order_Synergy';
A1_tem = xlsread(data, 'Full Range1');
A1= [A1_tem(:,1) A1_tem(:,2) A1_tem(:,3) A1_tem(:,4) A1_tem(:,5) A1_tem(:,6) A1_tem(:,7)];
A2_tem = xlsread(data, 'Full Range2');
A2= [A2_tem(:,1) A2_tem(:,2) A2_tem(:,3) A2_tem(:,4) A2_tem(:,5) A2_tem(:,6) A2_tem(:,7)];
A3_tem = xlsread(data, 'Full Range3');
A3= [A3_tem(:,1) A3_tem(:,2) A3_tem(:,3) A3_tem(:,4) A3_tem(:,5) A3_tem(:,6) A3_tem(:,7)];
Amean = (A1(:,:)+A2(:,:)+A3(:,:))./3;
AmeanTotal = AmeanTotal(:,:)+A1(:,:)+A2(:,:)+A3(:,:);
[ xInputMatrix1 xInputMatrix2 xInputMatrix3 xInputMatrixAve xInputThetas1 xInputThetas2
xInputThetas3 xInputThetasAve ] = xInputMatrixMaker('/Users/<usrname>/Desktop/research/
SynJointData/Subject 4 Age Match/');

[ FMS1_s4_syn_ideal FMScore1_s4_syn_ideal ] = FMScore( abs(A1*xInputThetas1) )
[ FMS1_s4_syn_raw FMScore1_s4_syn_raw ] = FMScore(abs(A1*xInputMatrix1))
[ FMS1_s4_raw FMScore1_s4_raw ] = FMScore( xInputMatrix1 )
%colorizeData(abs(A1*xInputThetas1))

[ FMS2_s4_syn_ideal FMScore2_s4_syn_ideal ] = FMScore( abs(A2*xInputThetas2) )
[ FMS2_s4_syn_raw FMScore2_s4_syn_raw ] = FMScore( abs(A2*xInputMatrix2) )
[ FMS2_s4_raw FMScore2_s4_raw ] = FMScore( xInputMatrix2 )

[ FMS3_s4_syn_ideal FMScore3_s4_syn_ideal ] = FMScore( abs(A3*xInputThetas3) )
[ FMS3_s4_syn_raw FMScore3_s4_syn_raw ] = FMScore( abs(A3*xInputMatrix3) )
[ FMS3_s4_raw FMScore3_s4_raw ] = FMScore( xInputMatrix3 )

[ FMSave_s4_syn_ideal FMScoreAve_s4_syn_ideal ] = FMScore( abs(Amean*xInputThetasAve) )
[ FMSave_s4_syn_raw FMScoreAve_s4_syn_raw ] = FMScore( abs(Amean*xInputMatrixAve) )
[ FMSave_s4_raw FMScoreAve_s4_raw ] = FMScore( xInputMatrixAve )

%%
%
%  subject 5 age matched
%
data = '/Users/<usrname>/Desktop/research/SynJointData/Subject 5 Age Match/1st_Order_Synergy';
A1_tem = xlsread(data, 'Full Range1');
A1= [A1_tem(:,1) A1_tem(:,2) A1_tem(:,3) A1_tem(:,4) A1_tem(:,5) A1_tem(:,6) A1_tem(:,7)];
A2_tem = xlsread(data, 'Full Range2');
A2= [A2_tem(:,1) A2_tem(:,2) A2_tem(:,3) A2_tem(:,4) A2_tem(:,5) A2_tem(:,6) A2_tem(:,7)];
A3_tem = xlsread(data, 'Full Range3');
A3= [A3_tem(:,1) A3_tem(:,2) A3_tem(:,3) A3_tem(:,4) A3_tem(:,5) A3_tem(:,6) A3_tem(:,7)];
AmeanTotal = AmeanTotal(:,:)+A1(:,:)+A2(:,:)+A3(:,:);
Amean = (A1(:,:)+A2(:,:)+A3(:,:))./3;
[ xInputMatrix1 xInputMatrix2 xInputMatrix3 xInputMatrixAve xInputThetas1 xInputThetas2
xInputThetas3 xInputThetasAve ] = xInputMatrixMaker('/Users/<usrname>/Desktop/research/
SynJointData/Subject 5 Age Match/');

[ FMS1_s5_syn_ideal FMScore1_s5_syn_ideal ] = FMScore( abs(A1*xInputThetas1) )
[ FMS1_s5_syn_raw FMScore1_s5_syn_raw ] = FMScore(abs(A1*xInputMatrix1))
[ FMS1_s5_raw FMScore1_s5_raw ] = FMScore( xInputMatrix1 )
%colorizeData(abs(A1*xInputThetas1))

[ FMS2_s5_syn_ideal FMScore2_s5_syn_ideal ] = FMScore( abs(A2*xInputThetas2) )
[ FMS2_s5_syn_raw FMScore2_s5_syn_raw ] = FMScore( abs(A2*xInputMatrix2) )
[ FMS2_s5_raw FMScore2_s5_raw ] = FMScore( xInputMatrix2 )

[ FMS3_s5_syn_ideal FMScore3_s5_syn_ideal ] = FMScore( abs(A3*xInputThetas3) )
[ FMS3_s5_syn_raw FMScore3_s5_syn_raw ] = FMScore( abs(A3*xInputMatrix3) )
[ FMS3_s5_raw FMScore3_s5_raw ] = FMScore( xInputMatrix3 )

[ FMSave_s5_syn_ideal FMScoreAve_s5_syn_ideal ] = FMScore( abs(Amean*xInputThetasAve) )
[ FMSave_s5_syn_raw FMScoreAve_s5_syn_raw ] = FMScore( abs(Amean*xInputMatrixAve) )
[ FMSave_s5_raw FMScoreAve_s5_raw ] = FMScore( xInputMatrixAve )
```

```
%%
%
%  subject 6
%
data = '/Users/<usrname>/Desktop/research/SynJointData/Subject 6/1st_Order_Synergy';
A1_tem = xlsread(data, 'Full Range1');
A1= [A1_tem(:,1) A1_tem(:,2) A1_tem(:,3) A1_tem(:,4) A1_tem(:,5) A1_tem(:,6) A1_tem(:,7)];
A2_tem = xlsread(data, 'Full Range2');
A2= [A2_tem(:,1) A2_tem(:,2) A2_tem(:,3) A2_tem(:,4) A2_tem(:,5) A2_tem(:,6) A2_tem(:,7)];
A3_tem = xlsread(data, 'Full Range3');
A3= [A3_tem(:,1) A3_tem(:,2) A3_tem(:,3) A3_tem(:,4) A3_tem(:,5) A3_tem(:,6) A3_tem(:,7)];
%AmeanTotal = AmeanTotal(:,:)+A1(:,:)+A2(:,:)+A3(:,:); not used bad data
Amean = (A1(:,:)+A2(:,:)+A3(:,:))./3;
[ xInputMatrix1 xInputMatrix2 xInputMatrix3 xInputMatrixAve xInputThetas1 xInputThetas2
xInputThetas3 xInputThetasAve ] = xInputMatrixMaker('/Users/<usrname>/Desktop/research/
SynJointData/Subject 6/');

[ FMS1_s6_syn_ideal FMScore1_s6_syn_ideal ] = FMScore( abs(A1*xInputThetas1) )
[ FMS1_s6_syn_raw FMScore1_s6_syn_raw ] = FMScore(abs(A1*xInputMatrix1))
[ FMS1_s6_raw FMScore1_s6_raw ] = FMScore( xInputMatrix1 )
%colorizeData(abs(A1*xInputThetas1))

[ FMS2_s6_syn_ideal FMScore2_s6_syn_ideal ] = FMScore( abs(A2*xInputThetas2) )
[ FMS2_s6_syn_raw FMScore2_s6_syn_raw ] = FMScore( abs(A2*xInputMatrix2) )
[ FMS2_s6_raw FMScore2_s6_raw ] = FMScore( xInputMatrix2 )

[ FMS3_s6_syn_ideal FMScore3_s6_syn_ideal ] = FMScore( abs(A3*xInputThetas3) )
[ FMS3_s6_syn_raw FMScore3_s6_syn_raw ] = FMScore( abs(A3*xInputMatrix3) )
[ FMS3_s6_raw FMScore3_s6_raw ] = FMScore( xInputMatrix3 )

[ FMSave_s6_syn_ideal FMScoreAve_s6_syn_ideal ] = FMScore( abs(Amean*xInputThetasAve) )
[ FMSave_s6_syn_raw FMScoreAve_s6_syn_raw ] = FMScore( abs(Amean*xInputMatrixAve) )
[ FMSave_s6_raw FMScoreAve_s6_raw ] = FMScore( xInputMatrixAve )

%%
%
%  subject 7
%
data = '/Users/<usrname>/Desktop/research/SynJointData/Subject 7/1st_Order_Synergy';
A1_tem = xlsread(data, 'Full Range1');
A1= [A1_tem(:,1) A1_tem(:,2) A1_tem(:,3) A1_tem(:,4) A1_tem(:,5) A1_tem(:,6) A1_tem(:,7)];
A2_tem = xlsread(data, 'Full Range2');
A2= [A2_tem(:,1) A2_tem(:,2) A2_tem(:,3) A2_tem(:,4) A2_tem(:,5) A2_tem(:,6) A2_tem(:,7)];
A3_tem = xlsread(data, 'Full Range3');
A3= [A3_tem(:,1) A3_tem(:,2) A3_tem(:,3) A3_tem(:,4) A3_tem(:,5) A3_tem(:,6) A3_tem(:,7)];
AmeanTotal = AmeanTotal(:,:)+A1(:,:)+A2(:,:)+A3(:,:);
Amean = (A1(:,:)+A2(:,:)+A3(:,:))./3;
[ xInputMatrix1 xInputMatrix2 xInputMatrix3 xInputMatrixAve xInputThetas1 xInputThetas2
xInputThetas3 xInputThetasAve ] = xInputMatrixMaker('/Users/<usrname>/Desktop/research/
SynJointData/Subject 7/');

[ FMS1_s7_syn_ideal FMScore1_s7_syn_ideal ] = FMScore( abs(A1*xInputThetas1) )
[ FMS1_s7_syn_raw FMScore1_s7_syn_raw ] = FMScore(abs(A1*xInputMatrix1))
[ FMS1_s7_raw FMScore1_s7_raw ] = FMScore( xInputMatrix1 )
colorizeData(abs(A1*xInputThetas1))

[ FMS2_s7_syn_ideal FMScore2_s7_syn_ideal ] = FMScore( abs(A2*xInputThetas2) )
[ FMS2_s7_syn_raw FMScore2_s7_syn_raw ] = FMScore( abs(A2*xInputMatrix2) )
[ FMS2_s7_raw FMScore2_s7_raw ] = FMScore( xInputMatrix2 )

[ FMS3_s7_syn_ideal FMScore3_s7_syn_ideal ] = FMScore( abs(A3*xInputThetas3) )
[ FMS3_s7_syn_raw FMScore3_s7_syn_raw ] = FMScore( abs(A3*xInputMatrix3) )
[ FMS3_s7_raw FMScore3_s7_raw ] = FMScore( xInputMatrix3 )
```

```
[ FMSave_s7_syn_ideal FMScoreAve_s7_syn_ideal ] = FMScore( abs(Amean*xInputThetasAve) )
[ FMSave_s7_syn_raw FMScoreAve_s7_syn_raw ] = FMScore( abs(Amean*xInputMatrixAve) )
[ FMSave_s7_raw FMScoreAve_s7_raw ] = FMScore( xInputMatrixAve )

%%
%
%  subject 8
%
data = '/Users/<usrname>/Desktop/research/SynJointData/Subject 8/1st_Order_Synergy';
A1_tem = xlsread(data, 'Full Range1');
A1= [A1_tem(:,1) A1_tem(:,2) A1_tem(:,3) A1_tem(:,4) A1_tem(:,5) A1_tem(:,6) A1_tem(:,7)];
A2_tem = xlsread(data, 'Full Range2');
A2= [A2_tem(:,1) A2_tem(:,2) A2_tem(:,3) A2_tem(:,4) A2_tem(:,5) A2_tem(:,6) A2_tem(:,7)];
A3_tem = xlsread(data, 'Full Range3');
A3= [A3_tem(:,1) A3_tem(:,2) A3_tem(:,3) A3_tem(:,4) A3_tem(:,5) A3_tem(:,6) A3_tem(:,7)];
AmeanTotal = AmeanTotal(:,:)+A1(:,:)+A2(:,:)+A3(:,:);
Amean = (A1(:,:)+A2(:,:)+A3(:,:))./3;
[ xInputMatrix1 xInputMatrix2 xInputMatrix3 xInputMatrixAve xInputThetas1 xInputThetas2
xInputThetas3 xInputThetasAve ] = xInputMatrixMaker('/Users/<usrname>/Desktop/research/
SynJointData/Subject 8/');

[ FMS1_s8_syn_ideal FMScore1_s8_syn_ideal ] = FMScore( abs(A1*xInputThetas1) )
[ FMS1_s8_syn_raw FMScore1_s8_syn_raw ] = FMScore(abs(A1*xInputMatrix1))
[ FMS1_s8_raw FMScore1_s8_raw ] = FMScore( xInputMatrix1 )
%colorizeData(abs(A1*xInputThetas1))

[ FMS2_s8_syn_ideal FMScore2_s8_syn_ideal ] = FMScore( abs(A2*xInputThetas2) )
[ FMS2_s8_syn_raw FMScore2_s8_syn_raw ] = FMScore( abs(A2*xInputMatrix2) )
[ FMS2_s8_raw FMScore2_s8_raw ] = FMScore( xInputMatrix2 )

[ FMS3_s8_syn_ideal FMScore3_s8_syn_ideal ] = FMScore( abs(A3*xInputThetas3) )
[ FMS3_s8_syn_raw FMScore3_s8_syn_raw ] = FMScore( abs(A3*xInputMatrix3) )
[ FMS3_s8_raw FMScore3_s8_raw ] = FMScore( xInputMatrix3 )

[ FMSave_s8_syn_ideal FMScoreAve_s8_syn_ideal ] = FMScore( abs(Amean*xInputThetasAve) )
[ FMSave_s8_syn_raw FMScoreAve_s8_syn_raw ] = FMScore( abs(Amean*xInputMatrixAve) )
[ FMSave_s8_raw FMScoreAve_s8_raw ] = FMScore( xInputMatrixAve )

%%
%
%  subject 9
%
data = '/Users/<usrname>/Desktop/research/SynJointData/Subject 9/1st_Order_Synergy';
A1_tem = xlsread(data, 'Full Range1');
A1= [A1_tem(:,1) A1_tem(:,2) A1_tem(:,3) A1_tem(:,4) A1_tem(:,5) A1_tem(:,6) A1_tem(:,7)];
A2_tem = xlsread(data, 'Full Range2');
A2= [A2_tem(:,1) A2_tem(:,2) A2_tem(:,3) A2_tem(:,4) A2_tem(:,5) A2_tem(:,6) A2_tem(:,7)];
A3_tem = xlsread(data, 'Full Range3');
A3= [A3_tem(:,1) A3_tem(:,2) A3_tem(:,3) A3_tem(:,4) A3_tem(:,5) A3_tem(:,6) A3_tem(:,7)];
AmeanTotal = AmeanTotal(:,:)+A1(:,:)+A2(:,:)+A3(:,:);
Amean = (A1(:,:)+A2(:,:)+A3(:,:))./3;
[ xInputMatrix1 xInputMatrix2 xInputMatrix3 xInputMatrixAve xInputThetas1 xInputThetas2
xInputThetas3 xInputThetasAve ] = xInputMatrixMaker('/Users/<usrname>/Desktop/research/
SynJointData/Subject 9/');

[ FMS1_s9_syn_ideal FMScore1_s9_syn_ideal ] = FMScore( abs(A1*xInputThetas1) )
[ FMS1_s9_syn_raw FMScore1_s9_syn_raw ] = FMScore(abs(A1*xInputMatrix1))
[ FMS1_s9_raw FMScore1_s9_raw ] = FMScore( xInputMatrix1 )
%colorizeData(abs(A1*xInputThetas1))

[ FMS2_s9_syn_ideal FMScore2_s9_syn_ideal ] = FMScore( abs(A2*xInputThetas2) )
[ FMS2_s9_syn_raw FMScore2_s9_syn_raw ] = FMScore( abs(A2*xInputMatrix2) )
[ FMS2_s9_raw FMScore2_s9_raw ] = FMScore( xInputMatrix2 )
```

```
[ FMS3_s9_syn_ideal FMScore3_s9_syn_ideal ] = FMScore( abs(A3*xInputThetas3) )
[ FMS3_s9_syn_raw FMScore3_s9_syn_raw ] = FMScore( abs(A3*xInputMatrix3) )
[ FMS3_s9_raw FMScore3_s9_raw ] = FMScore( xInputMatrix3 )

[ FMSave_s9_syn_ideal FMScoreAve_s9_syn_ideal ] = FMScore( abs(Amean*xInputThetasAve) )
[ FMSave_s9_syn_raw FMScoreAve_s9_syn_raw ] = FMScore( abs(Amean*xInputMatrixAve) )
[ FMSave_s9_raw FMScoreAve_s9_raw ] = FMScore( xInputMatrixAve )

%%
%
%  subject 10
%
data = '/Users/<usrname>/Desktop/research/SynJointData/Subject 10/1st_Order_Synergy';
A1_tem = xlsread(data, 'Full Range1');
A1= [A1_tem(:,1) A1_tem(:,2) A1_tem(:,3) A1_tem(:,4) A1_tem(:,5) A1_tem(:,6) A1_tem(:,7)];
A2_tem = xlsread(data, 'Full Range2');
A2= [A2_tem(:,1) A2_tem(:,2) A2_tem(:,3) A2_tem(:,4) A2_tem(:,5) A2_tem(:,6) A2_tem(:,7)];
A3_tem = xlsread(data, 'Full Range3');
A3= [A3_tem(:,1) A3_tem(:,2) A3_tem(:,3) A3_tem(:,4) A3_tem(:,5) A3_tem(:,6) A3_tem(:,7)];
AmeanTotal = AmeanTotal(:,:)+A1(:,:)+A2(:,:)+A3(:,:);
Amean = (A1(:,:)+A2(:,:)+A3(:,:))./3;
[ xInputMatrix1 xInputMatrix2 xInputMatrix3 xInputMatrixAve xInputThetas1 xInputThetas2
xInputThetas3 xInputThetasAve ] = xInputMatrixMaker('/Users/<usrname>/Desktop/research/
SynJointData/Subject 10/');

[ FMS1_s10_syn_ideal FMScore1_s10_syn_ideal ] = FMScore( abs(A1*xInputThetas1) )
[ FMS1_s10_syn_raw FMScore1_s10_syn_raw ] = FMScore(abs(A1*xInputMatrix1))
[ FMS1_s10_raw FMScore1_s10_raw ] = FMScore( xInputMatrix1 )
%colorizeData(abs(A1*xInputThetas1))

[ FMS2_s10_syn_ideal FMScore2_s10_syn_ideal ] = FMScore( abs(A2*xInputThetas2) )
[ FMS2_s10_syn_raw FMScore2_s10_syn_raw ] = FMScore( abs(A2*xInputMatrix2) )
[ FMS2_s10_raw FMScore2_s10_raw ] = FMScore( xInputMatrix2 )

[ FMS3_s10_syn_ideal FMScore3_s10_syn_ideal ] = FMScore( abs(A3*xInputThetas3) )
[ FMS3_s10_syn_raw FMScore3_s10_syn_raw ] = FMScore( abs(A3*xInputMatrix3) )
[ FMS3_s10_raw FMScore3_s10_raw ] = FMScore( xInputMatrix3 )

[ FMSave_s10_syn_ideal FMScoreAve_s10_syn_ideal ] = FMScore( abs(Amean*xInputThetasAve) )
[ FMSave_s10_syn_raw FMScoreAve_s10_syn_raw ] = FMScore( abs(Amean*xInputMatrixAve) )
[ FMSave_s10_raw FMScoreAve_s10_raw ] = FMScore( xInputMatrixAve )

%%
%
%  subject 11
%
data = '/Users/<usrname>/Desktop/research/SynJointData/Subject 11/1st_Order_Synergy';
A1_tem = xlsread(data, 'Full Range1');
A1= [A1_tem(:,1) A1_tem(:,2) A1_tem(:,3) A1_tem(:,4) A1_tem(:,5) A1_tem(:,6) A1_tem(:,7)];
A2_tem = xlsread(data, 'Full Range2');
A2= [A2_tem(:,1) A2_tem(:,2) A2_tem(:,3) A2_tem(:,4) A2_tem(:,5) A2_tem(:,6) A2_tem(:,7)];
A3_tem = xlsread(data, 'Full Range3');
A3= [A3_tem(:,1) A3_tem(:,2) A3_tem(:,3) A3_tem(:,4) A3_tem(:,5) A3_tem(:,6) A3_tem(:,7)];
AmeanTotal = AmeanTotal(:,:)+A1(:,:)+A2(:,:)+A3(:,:);
Amean = (A1(:,:)+A2(:,:)+A3(:,:))./3;
[ xInputMatrix1 xInputMatrix2 xInputMatrix3 xInputMatrixAve xInputThetas1 xInputThetas2
xInputThetas3 xInputThetasAve ] = xInputMatrixMaker('/Users/<usrname>/Desktop/research/
SynJointData/Subject 11/');

[ FMS1_s11_syn_ideal FMScore1_s11_syn_ideal ] = FMScore( abs(A1*xInputThetas1) )
[ FMS1_s11_syn_raw FMScore1_s11_syn_raw ] = FMScore(abs(A1*xInputMatrix1))
[ FMS1_s1_raw FMScore1_s11_raw ] = FMScore( xInputMatrix1 )
%colorizeData(abs(A1*xInputThetas1))
```

```
[ FMS2_s11_syn_ideal FMScore2_s11_syn_ideal ] = FMScore( abs(A2*xInputThetas2) )
[ FMS2_s11_syn_raw FMScore2_s11_syn_raw ] = FMScore( abs(A2*xInputMatrix2) )
[ FMS2_s11_raw FMScore2_s11_raw ] = FMScore( xInputMatrix2 )

[ FMS3_s11_syn_ideal FMScore3_s11_syn_ideal ] = FMScore( abs(A3*xInputThetas3) )
[ FMS3_s11_syn_raw FMScore3_s11_syn_raw ] = FMScore( abs(A3*xInputMatrix3) )
[ FMS3_s11_raw FMScore3_s11_raw ] = FMScore( xInputMatrix3 )

[ FMSave_s11_syn_ideal FMScoreAve_s11_syn_ideal ] = FMScore( abs(Amean*xInputThetasAve) )
[ FMSave_s11_syn_raw FMScoreAve_s11_syn_raw ] = FMScore( abs(Amean*xInputMatrixAve) )
[ FMSave_s11_raw FMScoreAve_s11_raw ] = FMScore( xInputMatrixAve )

%%
%
%  subject 23
%
data = '/Users/<usrname>/Desktop/research/SynJointData/Subject 23 Age Match/1st_Order_Synergy';
A1_tem = xlsread(data, 'Full Range1');
A1= [A1_tem(:,1) A1_tem(:,2) A1_tem(:,3) A1_tem(:,4) A1_tem(:,5) A1_tem(:,6) A1_tem(:,7)];
A2_tem = xlsread(data, 'Full Range2');
A2= [A2_tem(:,1) A2_tem(:,2) A2_tem(:,3) A2_tem(:,4) A2_tem(:,5) A2_tem(:,6) A2_tem(:,7)];
A3_tem = xlsread(data, 'Full Range3');
A3= [A3_tem(:,1) A3_tem(:,2) A3_tem(:,3) A3_tem(:,4) A3_tem(:,5) A3_tem(:,6) A3_tem(:,7)];
AmeanTotal = AmeanTotal(:,:)+A1(:,:)+A3(:,:); %A2 has missing data
Amean = (A1(:,:)+A3(:,:))./2;
[ xInputMatrix1 xInputMatrix2 xInputMatrix3 xInputMatrixAve xInputThetas1 xInputThetas2
xInputThetas3 xInputThetasAve ] = xInputMatrixMaker('/Users/<usrname>/Desktop/research/
SynJointData/Subject 23 Age Match/');

[ FMS1_s23_syn_ideal FMScore1_s23_syn_ideal ] = FMScore( abs(A1*xInputThetas1) )
[ FMS1_s23_syn_raw FMScore1_s23_syn_raw ] = FMScore(abs(A1*xInputMatrix1))
[ FMS1_s23_raw FMScore1_s23_raw ] = FMScore( xInputMatrix1 )
%colorizeData(abs(A1*xInputThetas1))

[ FMS2_s23_syn_ideal FMScore2_s23_syn_ideal ] = FMScore( abs(A2*xInputThetas2) )
[ FMS2_s23_syn_raw FMScore2_s23_syn_raw ] = FMScore( abs(A2*xInputMatrix2) )
[ FMS2_s23_raw FMScore2_s23_raw ] = FMScore( xInputMatrix2 )

[ FMS3_s23_syn_ideal FMScore3_s23_syn_ideal ] = FMScore( abs(A3*xInputThetas3) )
[ FMS3_s23_syn_raw FMScore3_s23_syn_raw ] = FMScore( abs(A3*xInputMatrix3) )
[ FMS3_s23_raw FMScore3_s23_raw ] = FMScore( xInputMatrix3 )

[ FMSave_s23_syn_ideal FMScoreAve_s23_syn_ideal ] = FMScore( abs(Amean*xInputThetasAve) )
[ FMSave_s23_syn_raw FMScoreAve_s23_syn_raw ] = FMScore( abs(Amean*xInputMatrixAve) )
[ FMSave_s23_raw FMScoreAve_s23_raw ] = FMScore( xInputMatrixAve )

%%
AmeanTotal
AmeanTotal = AmeanTotal(:,:)./32
%%
colorizeData(abs(AmeanTotal)*([160 0 0 0 0 0 0;...
              0 90 0 0 0 0 0;...
              0 0 90 0 0 0 0;...
              0 0 0 130 0 0 0;...
              0 0 0 0 90 0 0;...
              0 0 0 0 0 90 0;...
              0 0 0 0 0 0 55]))

%%
%
%  subject 12
%
data = '/Users/<usrname>/Desktop/research/SynJointData/Subject 12 - Stroke/1st_Order_Synergy';
```

```
A1_tem = xlsread(data, 'Full Range1');
A1= [A1_tem(:,1) A1_tem(:,2) A1_tem(:,3) A1_tem(:,4) A1_tem(:,5) A1_tem(:,6) A1_tem(:,7)];
A2_tem = xlsread(data, 'Full Range2');
A2= [A2_tem(:,1) A2_tem(:,2) A2_tem(:,3) A2_tem(:,4) A2_tem(:,5) A2_tem(:,6) A2_tem(:,7)];
A3_tem = xlsread(data, 'Full Range3');
A3= [A3_tem(:,1) A3_tem(:,2) A3_tem(:,3) A3_tem(:,4) A3_tem(:,5) A3_tem(:,6) A3_tem(:,7)];
Amean = (A1(:,:)+A2(:,:)+A3(:,:))./3;
[ xInputMatrix1 xInputMatrix2 xInputMatrix3 xInputMatrixAve xInputThetas1 xInputThetas2
xInputThetas3 xInputThetasAve ] = xInputMatrixMaker('/Users/<usrname>/Desktop/research/
SynJointData/Subject 12 - Stroke/');

[ FMS1_s12_syn_ideal FMScore1_s12_syn_ideal ] = FMScore( abs(A1*xInputThetas1) )
[ FMS1_s12_syn_raw FMScore1_s12_syn_raw ] = FMScore(abs(A1*xInputMatrix1))
[ FMS1_s12_raw FMScore1_s12_raw ] = FMScore( xInputMatrix1 )
%colorizeData(abs(A1*xInputThetas1))

[ FMS2_s12_syn_ideal FMScore2_s12_syn_ideal ] = FMScore( abs(A2*xInputThetas2) )
[ FMS2_s12_syn_raw FMScore2_s12_syn_raw ] = FMScore( abs(A2*xInputMatrix2) )
[ FMS2_s12_raw FMScore2_s12_raw ] = FMScore( xInputMatrix2 )

[ FMS3_s12_syn_ideal FMScore3_s12_syn_ideal ] = FMScore( abs(A3*xInputThetas3) )
[ FMS3_s12_syn_raw FMScore3_s12_syn_raw ] = FMScore( abs(A3*xInputMatrix3) )
[ FMS3_s12_raw FMScore3_s12_raw ] = FMScore( xInputMatrix3 )

[ FMSave_s12_syn_ideal FMScoreAve_s12_syn_ideal ] = FMScore( abs(Amean*xInputThetasAve) )
[ FMSave_s12_syn_raw FMScoreAve_s12_syn_raw ] = FMScore( abs(Amean*xInputMatrixAve) )
[ FMSave_s12_raw FMScoreAve_s12_raw ] = FMScore( xInputMatrixAve )

%%
%
%  subject 13
%
data = '/Users/<usrname>/Desktop/research/SynJointData/Subject 13 - Stroke/1st_Order_Synergy';
A1_tem = xlsread(data, 'Full Range1');
A1= [A1_tem(:,1) A1_tem(:,2) A1_tem(:,3) A1_tem(:,4) A1_tem(:,5) A1_tem(:,6) A1_tem(:,7)];
A2_tem = xlsread(data, 'Full Range2');
A2= [A2_tem(:,1) A2_tem(:,2) A2_tem(:,3) A2_tem(:,4) A2_tem(:,5) A2_tem(:,6) A2_tem(:,7)];
A3_tem = xlsread(data, 'Full Range3');
A3= [A3_tem(:,1) A3_tem(:,2) A3_tem(:,3) A3_tem(:,4) A3_tem(:,5) A3_tem(:,6) A3_tem(:,7)];
Amean = (A2(:,:)+A3(:,:))./2;%A1 has missing data
[ xInputMatrix1 xInputMatrix2 xInputMatrix3 xInputMatrixAve xInputThetas1 xInputThetas2
xInputThetas3 xInputThetasAve ] = xInputMatrixMaker('/Users/<usrname>/Desktop/research/
SynJointData/Subject 13 - Stroke/');

[ FMS1_s13_syn_ideal FMScore1_s13_syn_ideal ] = FMScore( abs(A1*xInputThetas1) )
[ FMS1_s13_syn_raw FMScore1_s13_syn_raw ] = FMScore(abs(A1*xInputMatrix1))
[ FMS1_s13_raw FMScore1_s13_raw ] = FMScore( xInputMatrix1 )
%colorizeData(abs(A1*xInputThetas1))

[ FMS2_s13_syn_ideal FMScore2_s13_syn_ideal ] = FMScore( abs(A2*xInputThetas2) )
[ FMS2_s13_syn_raw FMScore2_s13_syn_raw ] = FMScore( abs(A2*xInputMatrix2) )
[ FMS2_s13_raw FMScore2_s13_raw ] = FMScore( xInputMatrix2 )

[ FMS3_s13_syn_ideal FMScore3_s13_syn_ideal ] = FMScore( abs(A3*xInputThetas3) )
[ FMS3_s13_syn_raw FMScore3_s13_syn_raw ] = FMScore( abs(A3*xInputMatrix3) )
[ FMS3_s13_raw FMScore3_s13_raw ] = FMScore( xInputMatrix3 )

[ FMSave_s13_syn_ideal FMScoreAve_s13_syn_ideal ] = FMScore( abs(Amean*xInputThetasAve) )
[ FMSave_s13_syn_raw FMScoreAve_s13_syn_raw ] = FMScore( abs(Amean*xInputMatrixAve) )
[ FMSave_s13_raw FMScoreAve_s13_raw ] = FMScore( xInputMatrixAve )

%%
%
%  subject 14
```

```
%
data = '/Users/<usrname>/Desktop/research/SynJointData/Subject 14 - Stroke/1st_Order_Synergy';
A1_tem = xlsread(data, 'Full Range1');
A1= [A1_tem(:,1) A1_tem(:,2) A1_tem(:,3) A1_tem(:,4) A1_tem(:,5) A1_tem(:,6) A1_tem(:,7)];
A2_tem = xlsread(data, 'Full Range2');
A2= [A2_tem(:,1) A2_tem(:,2) A2_tem(:,3) A2_tem(:,4) A2_tem(:,5) A2_tem(:,6) A2_tem(:,7)];
A3_tem = xlsread(data, 'Full Range3');
A3= [A3_tem(:,1) A3_tem(:,2) A3_tem(:,3) A3_tem(:,4) A3_tem(:,5) A3_tem(:,6) A3_tem(:,7)];
Amean = (A1(:,:)+A2(:,:)+A3(:,:))./3;
[ xInputMatrix1 xInputMatrix2 xInputMatrix3 xInputMatrixAve xInputThetas1 xInputThetas2
xInputThetas3 xInputThetasAve ] = xInputMatrixMaker('/Users/<usrname>/Desktop/research/
SynJointData/Subject 14 - Stroke/');

[ FMS1_s14_syn_ideal FMScore1_s14_syn_ideal ] = FMScore( abs(A1*xInputThetas1) )
[ FMS1_s14_syn_raw FMScore1_s14_syn_raw ] = FMScore(abs(A1*xInputMatrix1))
[ FMS1_s14_raw FMScore1_s14_raw ] = FMScore( xInputMatrix1 )
colorizeData(abs(A1*xInputThetas1))

[ FMS2_s14_syn_ideal FMScore2_s14_syn_ideal ] = FMScore( abs(A2*xInputThetas2) )
[ FMS2_s14_syn_raw FMScore2_s14_syn_raw ] = FMScore( abs(A2*xInputMatrix2) )
[ FMS2_s14_raw FMScore2_s14_raw ] = FMScore( xInputMatrix2 )

[ FMS3_s14_syn_ideal FMScore3_s14_syn_ideal ] = FMScore( abs(A3*xInputThetas3) )
[ FMS3_s14_syn_raw FMScore3_s14_syn_raw ] = FMScore( abs(A3*xInputMatrix3) )
[ FMS3_s14_raw FMScore3_s14_raw ] = FMScore( xInputMatrix3 )

[ FMSave_s14_syn_ideal FMScoreAve_s14_syn_ideal ] = FMScore( abs(Amean*xInputThetasAve) )
[ FMSave_s14_syn_raw FMScoreAve_s14_syn_raw ] = FMScore( abs(Amean*xInputMatrixAve) )
[ FMSave_s14_raw FMScoreAve_s14_raw ] = FMScore( xInputMatrixAve )

%%
%
%  subject 15
%
data = '/Users/<usrname>/Desktop/research/SynJointData/Subject 15 - Stroke/1st_Order_Synergy';
A1_tem = xlsread(data, 'Full Range1');
A1= [A1_tem(:,1) A1_tem(:,2) A1_tem(:,3) A1_tem(:,4) A1_tem(:,5) A1_tem(:,6) A1_tem(:,7)];
A2_tem = xlsread(data, 'Full Range2');
A2= [A2_tem(:,1) A2_tem(:,2) A2_tem(:,3) A2_tem(:,4) A2_tem(:,5) A2_tem(:,6) A2_tem(:,7)];
A3_tem = xlsread(data, 'Full Range3');
A3= [A3_tem(:,1) A3_tem(:,2) A3_tem(:,3) A3_tem(:,4) A3_tem(:,5) A3_tem(:,6) A3_tem(:,7)];
Amean = (A1(:,:)+A2(:,:)+A3(:,:))./3;
[ xInputMatrix1 xInputMatrix2 xInputMatrix3 xInputMatrixAve xInputThetas1 xInputThetas2
xInputThetas3 xInputThetasAve ] = xInputMatrixMaker('/Users/<usrname>/Desktop/research/
SynJointData/Subject 15 - Stroke/');

[ FMS1_s15_syn_ideal FMScore1_s15_syn_ideal ] = FMScore( abs(A1*xInputThetas1) )
[ FMS1_s15_syn_raw FMScore1_s15_syn_raw ] = FMScore(abs(A1*xInputMatrix1))
[ FMS1_s15_raw FMScore1_s15_raw ] = FMScore( xInputMatrix1 )
%colorizeData(abs(A1*xInputThetas1))

[ FMS2_s15_syn_ideal FMScore2_s15_syn_ideal ] = FMScore( abs(A2*xInputThetas2) )
[ FMS2_s15_syn_raw FMScore2_s15_syn_raw ] = FMScore( abs(A2*xInputMatrix2) )
[ FMS2_s15_raw FMScore2_s15_raw ] = FMScore( xInputMatrix2 )

[ FMS3_s15_syn_ideal FMScore3_s15_syn_ideal ] = FMScore( abs(A3*xInputThetas3) )
[ FMS3_s15_syn_raw FMScore3_s15_syn_raw ] = FMScore( abs(A3*xInputMatrix3) )
[ FMS3_s15_raw FMScore3_s15_raw ] = FMScore( xInputMatrix3 )

[ FMSave_s15_syn_ideal FMScoreAve_s15_syn_ideal ] = FMScore( abs(Amean*xInputThetasAve) )
[ FMSave_s15_syn_raw FMScoreAve_s15_syn_raw ] = FMScore( abs(Amean*xInputMatrixAve) )
[ FMSave_s15_raw FMScoreAve_s15_raw ] = FMScore( xInputMatrixAve )
%%
%
```

```
%  subject 16
%
data = '/Users/<usrname>/Desktop/research/SynJointData/Subject 16 - Stroke/1st_Order_Synergy';
A1_tem = xlsread(data, 'Full Range1');
A1= [A1_tem(:,1) A1_tem(:,2) A1_tem(:,3) A1_tem(:,4) A1_tem(:,5) A1_tem(:,6) A1_tem(:,7)];
A2_tem = xlsread(data, 'Full Range2');
A2= [A2_tem(:,1) A2_tem(:,2) A2_tem(:,3) A2_tem(:,4) A2_tem(:,5) A2_tem(:,6) A2_tem(:,7)];
A3_tem = xlsread(data, 'Full Range3');
A3= [A3_tem(:,1) A3_tem(:,2) A3_tem(:,3) A3_tem(:,4) A3_tem(:,5) A3_tem(:,6) A3_tem(:,7)];
Amean = (A1(:,:)+A2(:,:)+A3(:,:))./3;
[ xInputMatrix1 xInputMatrix2 xInputMatrix3 xInputMatrixAve xInputThetas1 xInputThetas2
xInputThetas3 xInputThetasAve ] = xInputMatrixMaker('/Users/<usrname>/Desktop/research/
SynJointData/Subject 16 - Stroke/');

[ FMS1_s16_syn_ideal FMScore1_s16_syn_ideal ] = FMScore( abs(A1*xInputThetas1) )
[ FMS1_s16_syn_raw FMScore1_s16_syn_raw ] = FMScore(abs(A1*xInputMatrix1))
[ FMS1_s16_raw FMScore1_s16_raw ] = FMScore( xInputMatrix1 )
%colorizeData(abs(A1*xInputThetas1))

[ FMS2_s16_syn_ideal FMScore2_s16_syn_ideal ] = FMScore( abs(A2*xInputThetas2) )
[ FMS2_s16_syn_raw FMScore2_s16_syn_raw ] = FMScore( abs(A2*xInputMatrix2) )
[ FMS2_s16_raw FMScore2_s16_raw ] = FMScore( xInputMatrix2 )

[ FMS3_s16_syn_ideal FMScore3_s16_syn_ideal ] = FMScore( abs(A3*xInputThetas3) )
[ FMS3_s16_syn_raw FMScore3_s16_syn_raw ] = FMScore( abs(A3*xInputMatrix3) )
[ FMS3_s16_raw FMScore3_s16_raw ] = FMScore( xInputMatrix3 )

[ FMSave_s16_syn_ideal FMScoreAve_s16_syn_ideal ] = FMScore( abs(Amean*xInputThetasAve) )
[ FMSave_s16_syn_raw FMScoreAve_s16_syn_raw ] = FMScore( abs(Amean*xInputMatrixAve) )
[ FMSave_s16_raw FMScoreAve_s16_raw ] = FMScore( xInputMatrixAve )
%%
%
%  subject 17
%
data = '/Users/<usrname>/Desktop/research/SynJointData/Subject 17 - Stroke/1st_Order_Synergy';
A1_tem = xlsread(data, 'Full Range1');
A1= [A1_tem(:,1) A1_tem(:,2) A1_tem(:,3) A1_tem(:,4) A1_tem(:,5) A1_tem(:,6) A1_tem(:,7)];
A2_tem = xlsread(data, 'Full Range2');
A2= [A2_tem(:,1) A2_tem(:,2) A2_tem(:,3) A2_tem(:,4) A2_tem(:,5) A2_tem(:,6) A2_tem(:,7)];
A3_tem = xlsread(data, 'Full Range3');
A3= [A3_tem(:,1) A3_tem(:,2) A3_tem(:,3) A3_tem(:,4) A3_tem(:,5) A3_tem(:,6) A3_tem(:,7)];
Amean = (A1(:,:)+A2(:,:)+A3(:,:))./3;
[ xInputMatrix1 xInputMatrix2 xInputMatrix3 xInputMatrixAve xInputThetas1 xInputThetas2
xInputThetas3 xInputThetasAve ] = xInputMatrixMaker('/Users/<usrname>/Desktop/research/
SynJointData/Subject 17 - Stroke/');

[ FMS1_s17_syn_ideal FMScore1_s17_syn_ideal ] = FMScore( abs(A1*xInputThetas1) )
[ FMS1_s17_syn_raw FMScore1_s17_syn_raw ] = FMScore(abs(A1*xInputMatrix1))
[ FMS1_s17_raw FMScore1_s17_raw ] = FMScore( xInputMatrix1 )
%colorizeData(abs(A1*xInputThetas1))

[ FMS2_s17_syn_ideal FMScore2_s17_syn_ideal ] = FMScore( abs(A2*xInputThetas2) )
[ FMS2_s17_syn_raw FMScore2_s17_syn_raw ] = FMScore( abs(A2*xInputMatrix2) )
[ FMS2_s17_raw FMScore2_s17_raw ] = FMScore( xInputMatrix2 )

[ FMS3_s17_syn_ideal FMScore3_s17_syn_ideal ] = FMScore( abs(A3*xInputThetas3) )
[ FMS3_s17_syn_raw FMScore3_s17_syn_raw ] = FMScore( abs(A3*xInputMatrix3) )
[ FMS3_s17_raw FMScore3_s17_raw ] = FMScore( xInputMatrix3 )

[ FMSave_s17_syn_ideal FMScoreAve_s17_syn_ideal ] = FMScore( abs(Amean*xInputThetasAve) )
[ FMSave_s17_syn_raw FMScoreAve_s17_syn_raw ] = FMScore( abs(Amean*xInputMatrixAve) )
[ FMSave_s17_raw FMScoreAve_s17_raw ] = FMScore( xInputMatrixAve )
%%
%
```

```
%  subject 18
%
data = '/Users/<usrname>/Desktop/research/SynJointData/Subject 18 - Stroke/1st_Order_Synergy';
A1_tem = xlsread(data, 'Full Range1');
A1= [A1_tem(:,1) A1_tem(:,2) A1_tem(:,3) A1_tem(:,4) A1_tem(:,5) A1_tem(:,6) A1_tem(:,7)];
A2_tem = xlsread(data, 'Full Range2');
A2= [A2_tem(:,1) A2_tem(:,2) A2_tem(:,3) A2_tem(:,4) A2_tem(:,5) A2_tem(:,6) A2_tem(:,7)];
A3_tem = xlsread(data, 'Full Range3');
A3= [A3_tem(:,1) A3_tem(:,2) A3_tem(:,3) A3_tem(:,4) A3_tem(:,5) A3_tem(:,6) A3_tem(:,7)];
Amean = (A1(:,:)+A2(:,:)+A3(:,:))./3;
[ xInputMatrix1 xInputMatrix2 xInputMatrix3 xInputMatrixAve xInputThetas1 xInputThetas2
xInputThetas3 xInputThetasAve ] = xInputMatrixMaker('/Users/<usrname>/Desktop/research/
SynJointData/Subject 18 - Stroke/');

[ FMS1_s18_syn_ideal FMScore1_s18_syn_ideal ] = FMScore( abs(A1*xInputThetas1) )
[ FMS1_s18_syn_raw FMScore1_s18_syn_raw ] = FMScore(abs(A1*xInputMatrix1))
[ FMS1_s18_raw FMScore1_s18_raw ] = FMScore( xInputMatrix1 )
%colorizeData(abs(A1*xInputThetas1))

[ FMS2_s18_syn_ideal FMScore2_s18_syn_ideal ] = FMScore( abs(A2*xInputThetas2) )
[ FMS2_s18_syn_raw FMScore2_s18_syn_raw ] = FMScore( abs(A2*xInputMatrix2) )
[ FMS2_s18_raw FMScore2_s18_raw ] = FMScore( xInputMatrix2 )

[ FMS3_s18_syn_ideal FMScore3_s18_syn_ideal ] = FMScore( abs(A3*xInputThetas3) )
[ FMS3_s18_syn_raw FMScore3_s18_syn_raw ] = FMScore( abs(A3*xInputMatrix3) )
[ FMS3_s18_raw FMScore3_s18_raw ] = FMScore( xInputMatrix3 )

[ FMSave_s18_syn_ideal FMScoreAve_s18_syn_ideal ] = FMScore( abs(Amean*xInputThetasAve) )
[ FMSave_s18_syn_raw FMScoreAve_s18_syn_raw ] = FMScore( abs(Amean*xInputMatrixAve) )
[ FMSave_s18_raw FMScoreAve_s18_raw ] = FMScore( xInputMatrixAve )

%%
%
%  subject 19
%
data = '/Users/<usrname>/Desktop/research/SynJointData/Subject 19 - Stroke/1st_Order_Synergy';
A1_tem = xlsread(data, 'Full Range1');
A1= [A1_tem(:,1) A1_tem(:,2) A1_tem(:,3) A1_tem(:,4) A1_tem(:,5) A1_tem(:,6) A1_tem(:,7)];
A2_tem = xlsread(data, 'Full Range2');
A2= [A2_tem(:,1) A2_tem(:,2) A2_tem(:,3) A2_tem(:,4) A2_tem(:,5) A2_tem(:,6) A2_tem(:,7)];
A3_tem = xlsread(data, 'Full Range3');
A3= [A3_tem(:,1) A3_tem(:,2) A3_tem(:,3) A3_tem(:,4) A3_tem(:,5) A3_tem(:,6) A3_tem(:,7)];
Amean = (A1(:,:)+A2(:,:)+A3(:,:))./3;
[ xInputMatrix1 xInputMatrix2 xInputMatrix3 xInputMatrixAve xInputThetas1 xInputThetas2
xInputThetas3 xInputThetasAve ] = xInputMatrixMaker('/Users/<usrname>/Desktop/research/
SynJointData/Subject 19 - Stroke/');

[ FMS1_s19_syn_ideal FMScore1_s19_syn_ideal ] = FMScore( abs(A1*xInputThetas1) )
[ FMS1_s19_syn_raw FMScore1_s19_syn_raw ] = FMScore(abs(A1*xInputMatrix1))
[ FMS1_s19_raw FMScore1_s19_raw ] = FMScore( xInputMatrix1 )
%colorizeData(abs(A1*xInputThetas1))

[ FMS2_s19_syn_ideal FMScore2_s19_syn_ideal ] = FMScore( abs(A2*xInputThetas2) )
[ FMS2_s19_syn_raw FMScore2_s19_syn_raw ] = FMScore( abs(A2*xInputMatrix2) )
[ FMS2_s19_raw FMScore2_s19_raw ] = FMScore( xInputMatrix2 )

[ FMS3_s19_syn_ideal FMScore3_s19_syn_ideal ] = FMScore( abs(A3*xInputThetas3) )
[ FMS3_s19_syn_raw FMScore3_s19_syn_raw ] = FMScore( abs(A3*xInputMatrix3) )
[ FMS3_s19_raw FMScore3_s19_raw ] = FMScore( xInputMatrix3 )

[ FMSave_s19_syn_ideal FMScoreAve_s19_syn_ideal ] = FMScore( abs(Amean*xInputThetasAve) )
[ FMSave_s19_syn_raw FMScoreAve_s19_syn_raw ] = FMScore( abs(Amean*xInputMatrixAve) )
[ FMSave_s19_raw FMScoreAve_s19_raw ] = FMScore( xInputMatrixAve )
```

```
%%
%
%  subject 20
%
data = '/Users/<usrname>/Desktop/research/SynJointData/Subject 20 - Stroke/1st_Order_Synergy';
A1_tem = xlsread(data, 'Full Range1');
A1= [A1_tem(:,1) A1_tem(:,2) A1_tem(:,3) A1_tem(:,4) A1_tem(:,5) A1_tem(:,6) A1_tem(:,7)];
A2_tem = xlsread(data, 'Full Range2');
A2= [A2_tem(:,1) A2_tem(:,2) A2_tem(:,3) A2_tem(:,4) A2_tem(:,5) A2_tem(:,6) A2_tem(:,7)];
A3_tem = xlsread(data, 'Full Range3');
A3= [A3_tem(:,1) A3_tem(:,2) A3_tem(:,3) A3_tem(:,4) A3_tem(:,5) A3_tem(:,6) A3_tem(:,7)];
Amean = (A1(:,:)+A2(:,:)+A3(:,:))./3;
[ xInputMatrix1 xInputMatrix2 xInputMatrix3 xInputMatrixAve xInputThetas1 xInputThetas2
xInputThetas3 xInputThetasAve ] = xInputMatrixMaker('/Users/<usrname>/Desktop/research/
SynJointData/Subject 20 - Stroke/');

[ FMS1_s20_syn_ideal FMScore1_s20_syn_ideal ] = FMScore( abs(A1*xInputThetas1) )
[ FMS1_s20_syn_raw FMScore1_s20_syn_raw ] = FMScore(abs(A1*xInputMatrix1))
[ FMS1_s20_raw FMScore1_s20_raw ] = FMScore( xInputMatrix1 )
%colorizeData(abs(A1*xInputThetas1))

[ FMS2_s20_syn_ideal FMScore2_s20_syn_ideal ] = FMScore( abs(A2*xInputThetas2) )
[ FMS2_s20_syn_raw FMScore2_s20_syn_raw ] = FMScore( abs(A2*xInputMatrix2) )
[ FMS2_s20_raw FMScore2_s20_raw ] = FMScore( xInputMatrix2 )

[ FMS3_s20_syn_ideal FMScore3_s20_syn_ideal ] = FMScore( abs(A3*xInputThetas3) )
[ FMS3_s20_syn_raw FMScore3_s20_syn_raw ] = FMScore( abs(A3*xInputMatrix3) )
[ FMS3_s20_raw FMScore3_s20_raw ] = FMScore( xInputMatrix3 )

[ FMSave_s20_syn_ideal FMScoreAve_s20_syn_ideal ] = FMScore( abs(Amean*xInputThetasAve) )
[ FMSave_s20_syn_raw FMScoreAve_s20_syn_raw ] = FMScore( abs(Amean*xInputMatrixAve) )
[ FMSave_s20_raw FMScoreAve_s20_raw ] = FMScore( xInputMatrixAve )

%%
%
%  subject 21
%
data = '/Users/<usrname>/Desktop/research/SynJointData/Subject 21 - Stroke/1st_Order_Synergy';
A1_tem = xlsread(data, 'Full Range1');
A1= [A1_tem(:,1) A1_tem(:,2) A1_tem(:,3) A1_tem(:,4) A1_tem(:,5) A1_tem(:,6) A1_tem(:,7)];
A2_tem = xlsread(data, 'Full Range2');
A2= [A2_tem(:,1) A2_tem(:,2) A2_tem(:,3) A2_tem(:,4) A2_tem(:,5) A2_tem(:,6) A2_tem(:,7)];
A3_tem = xlsread(data, 'Full Range3');
A3= [A3_tem(:,1) A3_tem(:,2) A3_tem(:,3) A3_tem(:,4) A3_tem(:,5) A3_tem(:,6) A3_tem(:,7)];
Amean = (A1(:,:)+A2(:,:)+A3(:,:))./3;
[ xInputMatrix1 xInputMatrix2 xInputMatrix3 xInputMatrixAve xInputThetas1 xInputThetas2
xInputThetas3 xInputThetasAve ] = xInputMatrixMaker('/Users/<usrname>/Desktop/research/
SynJointData/Subject 21 - Stroke/');

[ FMS1_s21_syn_ideal FMScore1_s21_syn_ideal ] = FMScore( abs(A1*xInputThetas1) )
[ FMS1_s21_syn_raw FMScore1_s21_syn_raw ] = FMScore(abs(A1*xInputMatrix1))
[ FMS1_s21_raw FMScore1_s21_raw ] = FMScore( xInputMatrix1 )
%colorizeData(abs(A1*xInputThetas1))

[ FMS2_s21_syn_ideal FMScore2_s21_syn_ideal ] = FMScore( abs(A2*xInputThetas2) )
[ FMS2_s21_syn_raw FMScore2_s21_syn_raw ] = FMScore( abs(A2*xInputMatrix2) )
[ FMS2_s21_raw FMScore2_s21_raw ] = FMScore( xInputMatrix2 )

[ FMS3_s21_syn_ideal FMScore3_s21_syn_ideal ] = FMScore( abs(A3*xInputThetas3) )
[ FMS3_s21_syn_raw FMScore3_s21_syn_raw ] = FMScore( abs(A3*xInputMatrix3) )
[ FMS3_s21_raw FMScore3_s21_raw ] = FMScore( xInputMatrix3 )

[ FMSave_s21_syn_ideal FMScoreAve_s21_syn_ideal ] = FMScore( abs(Amean*xInputThetasAve) )
[ FMSave_s21_syn_raw FMScoreAve_s21_syn_raw ] = FMScore( abs(Amean*xInputMatrixAve) )
```

```
[ FMSave_s21_raw FMScoreAve_s21_raw ] = FMScore( xInputMatrixAve )

%%
%
%  subject 22
%
data = '/Users/<usrname>/Desktop/research/SynJointData/Subject 22 - Stroke/1st_Order_Synergy';
A1_tem = xlsread(data, 'Full Range1');
A1= [A1_tem(:,1) A1_tem(:,2) A1_tem(:,3) A1_tem(:,4) A1_tem(:,5) A1_tem(:,6) A1_tem(:,7)];
A2_tem = xlsread(data, 'Full Range2');
A2= [A2_tem(:,1) A2_tem(:,2) A2_tem(:,3) A2_tem(:,4) A2_tem(:,5) A2_tem(:,6) A2_tem(:,7)];
A3_tem = xlsread(data, 'Full Range3');
A3= [A3_tem(:,1) A3_tem(:,2) A3_tem(:,3) A3_tem(:,4) A3_tem(:,5) A3_tem(:,6) A3_tem(:,7)];
Amean = (A1(:,:)+A2(:,:)+A3(:,:))./3;
[ xInputMatrix1 xInputMatrix2 xInputMatrix3 xInputMatrixAve xInputThetas1 xInputThetas2
xInputThetas3 xInputThetasAve ] = xInputMatrixMaker('/Users/<usrname>/Desktop/research/
SynJointData/Subject 22 - Stroke/');

[ FMS1_s22_syn_ideal FMScore1_s22_syn_ideal ] = FMScore( abs(A1*xInputThetas1) )
[ FMS1_s22_syn_raw FMScore1_s22_syn_raw ] = FMScore(abs(A1*xInputMatrix1))
[ FMS1_s22_raw FMScore1_s22_raw ] = FMScore( xInputMatrix1 )
colorizeData(abs(A1*xInputThetas1))

[ FMS2_s22_syn_ideal FMScore2_s22_syn_ideal ] = FMScore( abs(A2*xInputThetas2) )
[ FMS2_s22_syn_raw FMScore2_s22_syn_raw ] = FMScore( abs(A2*xInputMatrix2) )
[ FMS2_s22_raw FMScore2_s22_raw ] = FMScore( xInputMatrix2 )

[ FMS3_s22_syn_ideal FMScore3_s22_syn_ideal ] = FMScore( abs(A3*xInputThetas3) )
[ FMS3_s22_syn_raw FMScore3_s22_syn_raw ] = FMScore( abs(A3*xInputMatrix3) )
[ FMS3_s22_raw FMScore3_s22_raw ] = FMScore( xInputMatrix3 )

[ FMSave_s22_syn_ideal FMScoreAve_s22_syn_ideal ] = FMScore( abs(Amean*xInputThetasAve) )
[ FMSave_s22_syn_raw FMScoreAve_s22_syn_raw ] = FMScore( abs(Amean*xInputMatrixAve) )
[ FMSave_s22_raw FMScoreAve_s22_raw ] = FMScore( xInputMatrixAve )

%%


RESULTS_syn_ideal = [14 FMScore1_s1_syn_ideal FMScore2_s1_syn_ideal FMScore3_s1_syn_ideal
FMScoreAve_s1_syn_ideal;
           14 FMScore1_s2_syn_ideal FMScore2_s2_syn_ideal FMScore3_s2_syn_ideal
FMScoreAve_s2_syn_ideal;
           14 FMScore1_s3_syn_ideal FMScore2_s3_syn_ideal FMScore3_s3_syn_ideal
FMScoreAve_s3_syn_ideal;
           14 FMScore1_s4_syn_ideal FMScore2_s4_syn_ideal FMScore3_s4_syn_ideal
FMScoreAve_s4_syn_ideal;
           14 FMScore1_s5_syn_ideal FMScore2_s5_syn_ideal FMScore3_s5_syn_ideal
FMScoreAve_s5_syn_ideal;
           14 FMScore1_s6_syn_ideal FMScore2_s6_syn_ideal FMScore3_s6_syn_ideal
FMScoreAve_s6_syn_ideal;
           14 FMScore1_s7_syn_ideal FMScore2_s7_syn_ideal FMScore3_s7_syn_ideal
FMScoreAve_s7_syn_ideal;
           14 FMScore1_s8_syn_ideal FMScore2_s8_syn_ideal FMScore3_s8_syn_ideal
FMScoreAve_s8_syn_ideal;
           14 FMScore1_s9_syn_ideal FMScore2_s9_syn_ideal FMScore3_s9_syn_ideal
FMScoreAve_s9_syn_ideal;
           14 FMScore1_s10_syn_ideal FMScore2_s10_syn_ideal FMScore3_s10_syn_ideal
FMScoreAve_s10_syn_ideal;
           14 FMScore1_s11_syn_ideal FMScore2_s11_syn_ideal FMScore3_s11_syn_ideal
FMScoreAve_s11_syn_ideal;
           6 FMScore1_s12_syn_ideal FMScore2_s12_syn_ideal FMScore3_s12_syn_ideal
FMScoreAve_s12_syn_ideal;
           1 2 FMScore2_s13_syn_ideal FMScore3_s13_syn_ideal FMScoreAve_s13_syn_ideal;
```

```
          5 FMScore1_s14_syn_ideal FMScore2_s14_syn_ideal FMScore3_s14_syn_ideal
FMScoreAve_s14_syn_ideal;
          11 FMScore1_s15_syn_ideal FMScore2_s15_syn_ideal FMScore3_s15_syn_ideal
FMScoreAve_s15_syn_ideal;
          6 FMScore1_s16_syn_ideal FMScore2_s16_syn_ideal FMScore3_s16_syn_ideal
FMScoreAve_s16_syn_ideal;
          13 FMScore1_s17_syn_ideal FMScore2_s17_syn_ideal FMScore3_s17_syn_ideal
FMScoreAve_s17_syn_ideal;
          10 FMScore1_s18_syn_ideal FMScore2_s18_syn_ideal FMScore3_s18_syn_ideal
FMScoreAve_s18_syn_ideal;
          9 FMScore1_s19_syn_ideal FMScore2_s19_syn_ideal FMScore3_s19_syn_ideal
FMScoreAve_s19_syn_ideal;
          14 FMScore1_s20_syn_ideal FMScore2_s20_syn_ideal FMScore3_s20_syn_ideal
FMScoreAve_s20_syn_ideal;
          4 FMScore1_s21_syn_ideal FMScore2_s21_syn_ideal FMScore3_s21_syn_ideal
FMScoreAve_s21_syn_ideal;
          8 FMScore1_s22_syn_ideal FMScore2_s22_syn_ideal FMScore3_s22_syn_ideal
FMScoreAve_s22_syn_ideal;
          14 FMScore1_s23_syn_ideal FMScore2_s23_syn_ideal FMScore3_s23_syn_ideal
FMScoreAve_s23_syn_ideal]

RESULTS_syn_ideal_h = [14 FMScore1_s1_syn_ideal FMScore2_s1_syn_ideal FMScore3_s1_syn_ideal
FMScoreAve_s1_syn_ideal;
                       14 FMScore1_s2_syn_ideal FMScore2_s2_syn_ideal FMScore3_s2_syn_ideal
FMScoreAve_s2_syn_ideal;
                       14 FMScore1_s3_syn_ideal FMScore2_s3_syn_ideal FMScore3_s3_syn_ideal
FMScoreAve_s3_syn_ideal;
                       14 FMScore1_s4_syn_ideal FMScore2_s4_syn_ideal FMScore3_s4_syn_ideal
FMScoreAve_s4_syn_ideal;
                       14 FMScore1_s5_syn_ideal FMScore2_s5_syn_ideal FMScore3_s5_syn_ideal
FMScoreAve_s5_syn_ideal;
                       14 FMScore1_s7_syn_ideal FMScore2_s7_syn_ideal FMScore3_s7_syn_ideal
FMScoreAve_s7_syn_ideal;
                       14 FMScore1_s8_syn_ideal FMScore2_s8_syn_ideal FMScore3_s8_syn_ideal
FMScoreAve_s8_syn_ideal;
                       14 FMScore1_s9_syn_ideal FMScore2_s9_syn_ideal FMScore3_s9_syn_ideal
FMScoreAve_s9_syn_ideal;
                       14 FMScore1_s10_syn_ideal FMScore2_s10_syn_ideal FMScore3_s10_syn_ideal
FMScoreAve_s10_syn_ideal;
                       14 FMScore1_s11_syn_ideal FMScore2_s11_syn_ideal FMScore3_s11_syn_ideal
FMScoreAve_s11_syn_ideal;
                       14 FMScore1_s23_syn_ideal 12 FMScore3_s23_syn_ideal
FMScoreAve_s23_syn_ideal];

RESULTS_syn_ideal_s = [6 FMScore1_s12_syn_ideal FMScore2_s12_syn_ideal FMScore3_s12_syn_ideal
FMScoreAve_s12_syn_ideal;
                       1 2 FMScore2_s13_syn_ideal FMScore3_s13_syn_ideal
FMScoreAve_s13_syn_ideal;
                       5 FMScore1_s14_syn_ideal FMScore2_s14_syn_ideal FMScore3_s14_syn_ideal
FMScoreAve_s14_syn_ideal;
                       11 FMScore1_s15_syn_ideal FMScore2_s15_syn_ideal FMScore3_s15_syn_ideal
FMScoreAve_s15_syn_ideal;
                       6 FMScore1_s16_syn_ideal FMScore2_s16_syn_ideal FMScore3_s16_syn_ideal
FMScoreAve_s16_syn_ideal;
                       13 FMScore1_s17_syn_ideal FMScore2_s17_syn_ideal FMScore3_s17_syn_ideal
FMScoreAve_s17_syn_ideal;
                       10 FMScore1_s18_syn_ideal FMScore2_s18_syn_ideal FMScore3_s18_syn_ideal
FMScoreAve_s18_syn_ideal;
                       9 FMScore1_s19_syn_ideal FMScore2_s19_syn_ideal FMScore3_s19_syn_ideal
FMScoreAve_s19_syn_ideal;
                       14 FMScore1_s20_syn_ideal FMScore2_s20_syn_ideal FMScore3_s20_syn_ideal
FMScoreAve_s20_syn_ideal;
                       4 FMScore1_s21_syn_ideal FMScore2_s21_syn_ideal FMScore3_s21_syn_ideal
FMScoreAve_s21_syn_ideal;
```

```
                    8 FMScore1_s22_syn_ideal FMScore2_s22_syn_ideal FMScore3_s22_syn_ideal
FMScoreAve_s22_syn_ideal];

RESULTS_syn_raw = [14 FMScore1_s1_syn_raw FMScore2_s1_syn_raw FMScore3_s1_syn_raw
FMScoreAve_s1_syn_raw;
          14 FMScore1_s2_syn_raw FMScore2_s2_syn_raw FMScore3_s2_syn_raw FMScoreAve_s2_syn_raw;
          14 FMScore1_s3_syn_raw FMScore2_s3_syn_raw FMScore3_s3_syn_raw FMScoreAve_s3_syn_raw;
          14 FMScore1_s4_syn_raw FMScore2_s4_syn_raw FMScore3_s4_syn_raw FMScoreAve_s4_syn_raw;
          14 FMScore1_s5_syn_raw FMScore2_s5_syn_raw FMScore3_s5_syn_raw FMScoreAve_s5_syn_raw;
          14 FMScore1_s6_syn_raw FMScore2_s6_syn_raw FMScore3_s6_syn_raw FMScoreAve_s6_syn_raw;
          14 FMScore1_s7_syn_raw FMScore2_s7_syn_raw FMScore3_s7_syn_raw FMScoreAve_s7_syn_raw;
          14 FMScore1_s8_syn_raw FMScore2_s8_syn_raw FMScore3_s8_syn_raw FMScoreAve_s8_syn_raw;
          14 FMScore1_s9_syn_raw FMScore2_s9_syn_raw FMScore3_s9_syn_raw FMScoreAve_s9_syn_raw;
          14 FMScore1_s10_syn_raw FMScore2_s10_syn_raw FMScore3_s10_syn_raw
FMScoreAve_s10_syn_raw;
          14 FMScore1_s11_syn_raw FMScore2_s11_syn_raw FMScore3_s11_syn_raw
FMScoreAve_s11_syn_raw;
          6 FMScore1_s12_syn_raw FMScore2_s12_syn_raw FMScore3_s12_syn_raw
FMScoreAve_s12_syn_raw;
          1 FMScore1_s13_syn_raw FMScore2_s13_syn_raw FMScore3_s13_syn_raw
FMScoreAve_s13_syn_raw;
          5 FMScore1_s14_syn_raw FMScore2_s14_syn_raw FMScore3_s14_syn_raw
FMScoreAve_s14_syn_raw;
          11 FMScore1_s15_syn_raw FMScore2_s15_syn_raw FMScore3_s15_syn_raw
FMScoreAve_s15_syn_raw;
          6 FMScore1_s16_syn_raw FMScore2_s16_syn_raw FMScore3_s16_syn_raw
FMScoreAve_s16_syn_raw;
          13 FMScore1_s17_syn_raw FMScore2_s17_syn_raw FMScore3_s17_syn_raw
FMScoreAve_s17_syn_raw;
          10 FMScore1_s18_syn_raw FMScore2_s18_syn_raw FMScore3_s18_syn_raw
FMScoreAve_s18_syn_raw;
          9 FMScore1_s19_syn_raw FMScore2_s19_syn_raw FMScore3_s19_syn_raw
FMScoreAve_s19_syn_raw;
          14 FMScore1_s20_syn_raw FMScore2_s20_syn_raw FMScore3_s20_syn_raw
FMScoreAve_s20_syn_raw;
          4 FMScore1_s21_syn_raw FMScore2_s21_syn_raw FMScore3_s21_syn_raw
FMScoreAve_s21_syn_raw;
          8 FMScore1_s22_syn_raw FMScore2_s22_syn_raw FMScore3_s22_syn_raw
FMScoreAve_s22_syn_raw;
          14 FMScore1_s23_syn_raw FMScore2_s23_syn_raw FMScore3_s23_syn_raw
FMScoreAve_s23_syn_raw]

RESULTS_raw = [14 FMScore1_s1_raw FMScore2_s1_raw FMScore3_s1_raw FMScoreAve_s1_raw;
          14 FMScore1_s2_raw FMScore2_s2_raw FMScore3_s2_raw FMScoreAve_s2_raw;
          14 FMScore1_s3_raw FMScore2_s3_raw FMScore3_s3_raw FMScoreAve_s3_raw;
          14 FMScore1_s4_raw FMScore2_s4_raw FMScore3_s4_raw FMScoreAve_s4_raw;
          14 FMScore1_s5_raw FMScore2_s5_raw FMScore3_s5_raw FMScoreAve_s5_raw;
          14 FMScore1_s6_raw FMScore2_s6_raw FMScore3_s6_raw FMScoreAve_s6_raw;
          14 FMScore1_s7_raw FMScore2_s7_raw FMScore3_s7_raw FMScoreAve_s7_raw;
          14 FMScore1_s8_raw FMScore2_s8_raw FMScore3_s8_raw FMScoreAve_s8_raw;
          14 FMScore1_s9_raw FMScore2_s9_raw FMScore3_s9_raw FMScoreAve_s9_raw;
          14 FMScore1_s10_raw FMScore2_s10_raw FMScore3_s10_raw FMScoreAve_s10_raw;
          14 FMScore1_s11_raw FMScore2_s11_raw FMScore3_s11_raw FMScoreAve_s11_raw;
          6 FMScore1_s12_raw FMScore2_s12_raw FMScore3_s12_raw FMScoreAve_s12_raw;
          1 FMScore1_s13_raw FMScore2_s13_raw FMScore3_s13_raw FMScoreAve_s13_raw;
          5 FMScore1_s14_raw FMScore2_s14_raw FMScore3_s14_raw FMScoreAve_s14_raw;
          11 FMScore1_s15_raw FMScore2_s15_raw FMScore3_s15_raw FMScoreAve_s15_raw;
          6 FMScore1_s16_raw FMScore2_s16_raw FMScore3_s16_raw FMScoreAve_s16_raw;
          13 FMScore1_s17_raw FMScore2_s17_raw FMScore3_s17_raw FMScoreAve_s17_raw;
          10 FMScore1_s18_raw FMScore2_s18_raw FMScore3_s18_raw FMScoreAve_s18_raw;
          9 FMScore1_s19_raw FMScore2_s19_raw FMScore3_s19_raw FMScoreAve_s19_raw;
          14 FMScore1_s20_raw FMScore2_s20_raw FMScore3_s20_raw FMScoreAve_s20_raw;
          4 FMScore1_s21_raw FMScore2_s21_raw FMScore3_s21_raw FMScoreAve_s21_raw;
          8 FMScore1_s22_raw FMScore2_s22_raw FMScore3_s22_raw FMScoreAve_s22_raw;
```

```
          14 FMScore1_s23_raw FMScore2_s23_raw FMScore3_s23_raw FMScoreAve_s23_raw]

%%

statsPerRow = [ mean([FMScore1_s1_syn_ideal, FMScore2_s1_syn_ideal, FMScore3_s1_syn_ideal])
std([FMScore1_s1_syn_ideal, FMScore2_s1_syn_ideal, FMScore3_s1_syn_ideal])
std([FMScore1_s1_syn_ideal, FMScore2_s1_syn_ideal, FMScore3_s1_syn_ideal])/sqrt(3);

mean([FMScore1_s2_syn_ideal, FMScore2_s2_syn_ideal, FMScore3_s2_syn_ideal])
std([FMScore1_s2_syn_ideal, FMScore2_s2_syn_ideal, FMScore3_s2_syn_ideal])
std([FMScore1_s2_syn_ideal, FMScore2_s2_syn_ideal, FMScore3_s2_syn_ideal])/sqrt(3);

mean([FMScore1_s3_syn_ideal, FMScore2_s3_syn_ideal, FMScore3_s3_syn_ideal])
std([FMScore1_s3_syn_ideal, FMScore2_s3_syn_ideal, FMScore3_s3_syn_ideal])
std([FMScore1_s3_syn_ideal, FMScore2_s3_syn_ideal, FMScore3_s3_syn_ideal])/sqrt(3);

mean([FMScore1_s4_syn_ideal, FMScore2_s4_syn_ideal, FMScore3_s4_syn_ideal])
std([FMScore1_s4_syn_ideal, FMScore2_s4_syn_ideal, FMScore3_s4_syn_ideal])
std([FMScore1_s4_syn_ideal, FMScore2_s4_syn_ideal, FMScore3_s4_syn_ideal])/sqrt(3);

mean([FMScore1_s5_syn_ideal, FMScore2_s5_syn_ideal, FMScore3_s5_syn_ideal])
std([FMScore1_s5_syn_ideal, FMScore2_s5_syn_ideal, FMScore3_s5_syn_ideal])
std([FMScore1_s5_syn_ideal, FMScore2_s5_syn_ideal, FMScore3_s5_syn_ideal])/sqrt(3);

mean([FMScore1_s6_syn_ideal, FMScore2_s6_syn_ideal, FMScore3_s6_syn_ideal])
std([FMScore1_s6_syn_ideal, FMScore2_s6_syn_ideal, FMScore3_s6_syn_ideal])
std([FMScore1_s6_syn_ideal, FMScore2_s6_syn_ideal, FMScore3_s6_syn_ideal])/sqrt(3);

mean([FMScore1_s7_syn_ideal, FMScore2_s7_syn_ideal, FMScore3_s7_syn_ideal])
std([FMScore1_s7_syn_ideal, FMScore2_s7_syn_ideal, FMScore3_s7_syn_ideal])
std([FMScore1_s7_syn_ideal, FMScore2_s7_syn_ideal, FMScore3_s7_syn_ideal])/sqrt(3);

mean([FMScore1_s8_syn_ideal, FMScore2_s8_syn_ideal, FMScore3_s8_syn_ideal])
std([FMScore1_s8_syn_ideal, FMScore2_s8_syn_ideal, FMScore3_s8_syn_ideal])
std([FMScore1_s8_syn_ideal, FMScore2_s8_syn_ideal, FMScore3_s8_syn_ideal])/sqrt(3);

mean([FMScore1_s9_syn_ideal, FMScore2_s9_syn_ideal, FMScore3_s9_syn_ideal])
std([FMScore1_s9_syn_ideal, FMScore2_s9_syn_ideal, FMScore3_s9_syn_ideal])
std([FMScore1_s9_syn_ideal, FMScore2_s9_syn_ideal, FMScore3_s9_syn_ideal])/sqrt(3);

mean([FMScore1_s10_syn_ideal, FMScore2_s10_syn_ideal, FMScore3_s10_syn_ideal])
std([FMScore1_s10_syn_ideal, FMScore2_s10_syn_ideal, FMScore3_s10_syn_ideal])
std([FMScore1_s10_syn_ideal, FMScore2_s10_syn_ideal, FMScore3_s10_syn_ideal])/sqrt(3);

mean([FMScore1_s11_syn_ideal, FMScore2_s11_syn_ideal, FMScore3_s11_syn_ideal])
std([FMScore1_s11_syn_ideal, FMScore2_s11_syn_ideal, FMScore3_s11_syn_ideal])
std([FMScore1_s11_syn_ideal, FMScore2_s11_syn_ideal, FMScore3_s11_syn_ideal])/sqrt(3);

mean([FMScore1_s23_syn_ideal, FMScore2_s23_syn_ideal, FMScore3_s23_syn_ideal])
std([FMScore1_s23_syn_ideal, FMScore2_s23_syn_ideal, FMScore3_s23_syn_ideal])
std([FMScore1_s23_syn_ideal, FMScore2_s23_syn_ideal, FMScore3_s23_syn_ideal])/sqrt(3)]

statsPerCol =[ mean([FMScore1_s1_syn_ideal, FMScore1_s2_syn_ideal, FMScore1_s3_syn_ideal...
    FMScore1_s4_syn_ideal, FMScore1_s5_syn_ideal, FMScore1_s6_syn_ideal...
    FMScore1_s7_syn_ideal, FMScore1_s8_syn_ideal, FMScore1_s9_syn_ideal...
    FMScore1_s10_syn_ideal, FMScore1_s11_syn_ideal, FMScore1_s23_syn_ideal])...
    std([FMScore1_s1_syn_ideal, FMScore1_s2_syn_ideal, FMScore1_s3_syn_ideal...
    FMScore1_s4_syn_ideal, FMScore1_s5_syn_ideal, FMScore1_s6_syn_ideal...
    FMScore1_s7_syn_ideal, FMScore1_s8_syn_ideal, FMScore1_s9_syn_ideal...
    FMScore1_s10_syn_ideal, FMScore1_s11_syn_ideal, FMScore1_s23_syn_ideal])...
    std([FMScore1_s1_syn_ideal, FMScore1_s2_syn_ideal, FMScore1_s3_syn_ideal...
    FMScore1_s4_syn_ideal, FMScore1_s5_syn_ideal, FMScore1_s6_syn_ideal...
    FMScore1_s7_syn_ideal, FMScore1_s8_syn_ideal, FMScore1_s9_syn_ideal...
```

```
    FMScore1_s10_syn_ideal, FMScore1_s11_syn_ideal, FMScore1_s23_syn_ideal])/sqrt(12);
    mean([FMScore2_s1_syn_ideal, FMScore2_s2_syn_ideal, FMScore2_s3_syn_ideal...
    FMScore2_s4_syn_ideal, FMScore2_s5_syn_ideal, FMScore2_s6_syn_ideal...
    FMScore2_s7_syn_ideal, FMScore2_s8_syn_ideal, FMScore2_s9_syn_ideal...
    FMScore2_s10_syn_ideal, FMScore2_s11_syn_ideal, FMScore2_s23_syn_ideal])...
    std([FMScore2_s1_syn_ideal, FMScore2_s2_syn_ideal, FMScore2_s3_syn_ideal...
    FMScore2_s4_syn_ideal, FMScore2_s5_syn_ideal, FMScore2_s6_syn_ideal...
    FMScore2_s7_syn_ideal, FMScore2_s8_syn_ideal, FMScore2_s9_syn_ideal...
    FMScore2_s10_syn_ideal, FMScore2_s11_syn_ideal, FMScore2_s23_syn_ideal])...
    std([FMScore2_s1_syn_ideal, FMScore2_s2_syn_ideal, FMScore2_s3_syn_ideal...
    FMScore2_s4_syn_ideal, FMScore2_s5_syn_ideal, FMScore2_s6_syn_ideal...
    FMScore2_s7_syn_ideal, FMScore2_s8_syn_ideal, FMScore2_s9_syn_ideal...
    FMScore2_s10_syn_ideal, FMScore2_s11_syn_ideal, FMScore2_s23_syn_ideal])/sqrt(12);
    mean([FMScore3_s1_syn_ideal, FMScore3_s2_syn_ideal, FMScore3_s3_syn_ideal...
    FMScore3_s4_syn_ideal, FMScore3_s5_syn_ideal, FMScore3_s6_syn_ideal...
    FMScore3_s7_syn_ideal, FMScore3_s8_syn_ideal, FMScore3_s9_syn_ideal...
    FMScore3_s10_syn_ideal, FMScore3_s11_syn_ideal, FMScore3_s23_syn_ideal])...
    std([FMScore3_s1_syn_ideal, FMScore3_s2_syn_ideal, FMScore3_s3_syn_ideal...
    FMScore3_s4_syn_ideal, FMScore3_s5_syn_ideal, FMScore3_s6_syn_ideal...
    FMScore3_s7_syn_ideal, FMScore3_s8_syn_ideal, FMScore3_s9_syn_ideal...
    FMScore3_s10_syn_ideal, FMScore3_s11_syn_ideal, FMScore3_s23_syn_ideal])...
    std([FMScore3_s1_syn_ideal, FMScore3_s2_syn_ideal, FMScore3_s3_syn_ideal...
    FMScore3_s4_syn_ideal, FMScore3_s5_syn_ideal, FMScore3_s6_syn_ideal...
    FMScore3_s7_syn_ideal, FMScore3_s8_syn_ideal, FMScore3_s9_syn_ideal...
    FMScore3_s10_syn_ideal, FMScore3_s11_syn_ideal, FMScore3_s23_syn_ideal])/sqrt(12)]

HealthyTotalStats =[ mean([FMScore1_s1_syn_ideal, FMScore1_s2_syn_ideal, FMScore1_s3_syn_ideal...
    FMScore1_s4_syn_ideal, FMScore1_s5_syn_ideal...
    FMScore1_s7_syn_ideal, FMScore1_s8_syn_ideal, FMScore1_s9_syn_ideal...
    FMScore1_s10_syn_ideal, FMScore1_s11_syn_ideal, FMScore1_s23_syn_ideal...
    FMScore2_s1_syn_ideal, FMScore2_s2_syn_ideal, FMScore2_s3_syn_ideal...
    FMScore2_s4_syn_ideal, FMScore2_s5_syn_ideal...
    FMScore2_s7_syn_ideal, FMScore2_s8_syn_ideal, FMScore2_s9_syn_ideal...
    FMScore2_s10_syn_ideal, FMScore2_s11_syn_ideal...
    FMScore3_s1_syn_ideal, FMScore3_s2_syn_ideal, FMScore3_s3_syn_ideal...
    FMScore3_s4_syn_ideal, FMScore3_s5_syn_ideal...
    FMScore3_s7_syn_ideal, FMScore3_s8_syn_ideal, FMScore3_s9_syn_ideal...
    FMScore3_s10_syn_ideal, FMScore3_s11_syn_ideal, FMScore3_s23_syn_ideal])...
    std([FMScore1_s1_syn_ideal, FMScore1_s2_syn_ideal, FMScore1_s3_syn_ideal...
    FMScore1_s4_syn_ideal, FMScore1_s5_syn_ideal...
    FMScore1_s7_syn_ideal, FMScore1_s8_syn_ideal, FMScore1_s9_syn_ideal...
    FMScore1_s10_syn_ideal, FMScore1_s11_syn_ideal, FMScore1_s23_syn_ideal...
    FMScore2_s1_syn_ideal, FMScore2_s2_syn_ideal, FMScore2_s3_syn_ideal...
    FMScore2_s4_syn_ideal, FMScore2_s5_syn_ideal...
    FMScore2_s7_syn_ideal, FMScore2_s8_syn_ideal, FMScore2_s9_syn_ideal...
    FMScore2_s10_syn_ideal, FMScore2_s11_syn_ideal...
    FMScore3_s1_syn_ideal, FMScore3_s2_syn_ideal, FMScore3_s3_syn_ideal...
    FMScore3_s4_syn_ideal, FMScore3_s5_syn_ideal...
    FMScore3_s7_syn_ideal, FMScore3_s8_syn_ideal, FMScore3_s9_syn_ideal...
    FMScore3_s10_syn_ideal, FMScore3_s11_syn_ideal, FMScore3_s23_syn_ideal])...
    std([FMScore1_s1_syn_ideal, FMScore1_s2_syn_ideal, FMScore1_s3_syn_ideal...
    FMScore1_s4_syn_ideal, FMScore1_s5_syn_ideal...
    FMScore1_s7_syn_ideal, FMScore1_s8_syn_ideal, FMScore1_s9_syn_ideal...
    FMScore1_s10_syn_ideal, FMScore1_s11_syn_ideal, FMScore1_s23_syn_ideal...
    FMScore2_s1_syn_ideal, FMScore2_s2_syn_ideal, FMScore2_s3_syn_ideal...
    FMScore2_s4_syn_ideal, FMScore2_s5_syn_ideal...
    FMScore2_s7_syn_ideal, FMScore2_s8_syn_ideal, FMScore2_s9_syn_ideal...
    FMScore2_s10_syn_ideal, FMScore2_s11_syn_ideal...
    FMScore3_s1_syn_ideal, FMScore3_s2_syn_ideal, FMScore3_s3_syn_ideal...
    FMScore3_s4_syn_ideal, FMScore3_s5_syn_ideal...
    FMScore3_s7_syn_ideal, FMScore3_s8_syn_ideal, FMScore3_s9_syn_ideal...
    FMScore3_s10_syn_ideal, FMScore3_s11_syn_ideal, FMScore3_s23_syn_ideal])/sqrt(11)]
```

```
HealthyoObjSubStats = [mean([HealthyTotalStats(1,1),14]) std([HealthyTotalStats(1,1),14])
std([HealthyTotalStats(1,1),14])/sqrt(2)]

StrokeTotalStats =[ mean([FMScore1_s12_syn_ideal, FMScore1_s14_syn_ideal...
    FMScore1_s15_syn_ideal, FMScore1_s16_syn_ideal, FMScore1_s17_syn_ideal...
    FMScore1_s18_syn_ideal, FMScore1_s19_syn_ideal, FMScore1_s20_syn_ideal...
    FMScore1_s21_syn_ideal, FMScore1_s22_syn_ideal...
    FMScore2_s12_syn_ideal, FMScore2_s13_syn_ideal, FMScore2_s14_syn_ideal...
    FMScore2_s15_syn_ideal, FMScore2_s16_syn_ideal, FMScore2_s17_syn_ideal...
    FMScore2_s18_syn_ideal, FMScore2_s19_syn_ideal, FMScore2_s20_syn_ideal...
    FMScore2_s21_syn_ideal, FMScore2_s22_syn_ideal...
    FMScore3_s12_syn_ideal, FMScore3_s13_syn_ideal, FMScore3_s14_syn_ideal...
    FMScore3_s15_syn_ideal, FMScore3_s16_syn_ideal, FMScore3_s17_syn_ideal...
    FMScore3_s18_syn_ideal, FMScore3_s19_syn_ideal, FMScore3_s20_syn_ideal...
    FMScore3_s21_syn_ideal, FMScore3_s22_syn_ideal])...
    std([FMScore1_s12_syn_ideal, FMScore1_s14_syn_ideal...
    FMScore1_s15_syn_ideal, FMScore1_s16_syn_ideal, FMScore1_s17_syn_ideal...
    FMScore1_s18_syn_ideal, FMScore1_s19_syn_ideal, FMScore1_s20_syn_ideal...
    FMScore1_s21_syn_ideal, FMScore1_s22_syn_ideal...
    FMScore2_s12_syn_ideal, FMScore2_s13_syn_ideal, FMScore2_s14_syn_ideal...
    FMScore2_s15_syn_ideal, FMScore2_s16_syn_ideal, FMScore2_s17_syn_ideal...
    FMScore2_s18_syn_ideal, FMScore2_s19_syn_ideal, FMScore2_s20_syn_ideal...
    FMScore2_s21_syn_ideal, FMScore2_s22_syn_ideal...
    FMScore3_s12_syn_ideal, FMScore3_s13_syn_ideal, FMScore3_s14_syn_ideal...
    FMScore3_s15_syn_ideal, FMScore3_s16_syn_ideal, FMScore3_s17_syn_ideal...
    FMScore3_s18_syn_ideal, FMScore3_s19_syn_ideal, FMScore3_s20_syn_ideal...
    FMScore3_s21_syn_ideal, FMScore3_s22_syn_ideal])...
    std([FMScore1_s12_syn_ideal, FMScore1_s14_syn_ideal...
    FMScore1_s15_syn_ideal, FMScore1_s16_syn_ideal, FMScore1_s17_syn_ideal...
    FMScore1_s18_syn_ideal, FMScore1_s19_syn_ideal, FMScore1_s20_syn_ideal...
    FMScore1_s21_syn_ideal, FMScore1_s22_syn_ideal...
    FMScore2_s12_syn_ideal, FMScore2_s13_syn_ideal, FMScore2_s14_syn_ideal...
    FMScore2_s15_syn_ideal, FMScore2_s16_syn_ideal, FMScore2_s17_syn_ideal...
    FMScore2_s18_syn_ideal, FMScore2_s19_syn_ideal, FMScore2_s20_syn_ideal...
    FMScore2_s21_syn_ideal, FMScore2_s22_syn_ideal...
    FMScore3_s12_syn_ideal, FMScore3_s13_syn_ideal, FMScore3_s14_syn_ideal...
    FMScore3_s15_syn_ideal, FMScore3_s16_syn_ideal, FMScore3_s17_syn_ideal...
    FMScore3_s18_syn_ideal, FMScore3_s19_syn_ideal, FMScore3_s20_syn_ideal...
    FMScore3_s21_syn_ideal, FMScore3_s22_syn_ideal])/sqrt(11)]

objSubStrokeStats = [mean([StrokeTotalStats(1,1),7.9090]) std([StrokeTotalStats(1,1),7.9090])
std([StrokeTotalStats(1,1),7.9090])/sqrt(2)]
%%
% figure(1)
% X = [RESULTS_syn_ideal(:,2) RESULTS_syn_ideal(:,3) RESULTS_syn_ideal(:,4) RESULTS_syn_ideal(:,
5)]
% Y = [RESULTS_syn_ideal(:,1) RESULTS_syn_ideal(:,1) RESULTS_syn_ideal(:,1) RESULTS_syn_ideal(:,
1)]
% E = mean(std(X))*ones(size(X))
% errorbar(X,Y,E)
%
% %%
% gplotmatrix(X,Y,group1,'','xos')
% %%
% figure(2)
% plot(X,Y,'.','markersize',10)
% k = convhull(X,Y);
% hold on, plot(X(k),Y(k),'-r'), hold off
% grid on
% %%
% figure(3)
% plotmatrix(X,Y, 'o')
% %%
% % figure(4)
```

```
% % [m,s,v,sem,mci,pci] = grpstats(X,Y,group1,{'mean','std','var','sem','meanci','predci'})
% %
% % set(gca,'xtick',1:n,'xticklabel',g)
% % title('95% prediction intervals for mean weight by year')
%
% %%
% figure(5)
% boxplot(RESULTS_syn_ideal','notch','on','labels',group1)
% %%
% figure(6)
% plot(X(:,1),Y(:,1),'*')
% lsline
% %%
% figure(7)
% [bootstat,bootsam] = bootstrp(1000,@corr,X,Y);
% se = std(bootstat)'
% hist(bootstat,30)
% set(get(gca,'Children'),'FaceColor',[.8 .8 1])
% %%
% figure(8)
% %ci = bootci(5000,@corr,X,Y)
% %m = bootstrp(1000,@mean,X,Y);
% [fi,xi] = ksdensity(bootstat);
% plot(xi,fi);
%
% %%
% figure(9)
% stats = bootstrp(100,@(X)[mean(X) std(X)],Y)
% [fi,xi] = ksdensity(stats(:,1));
% subplot(2,1,1),plot(xi,fi);
% subplot(2,1,2),plot(stats(:,1),stats(:,2),'*')
% %%
% b = regress(Y(:,1),X);
% Yfit = X*b;
% resid = Y(:,1) - Yfit
% se = std(bootstrp(...
%          1000,@(bootr)regress(yfit+bootr,X),resid))
% %%
% figure(10)
 %[p1,t1,st1] = anova1(RESULTS_syn_ideal',group1')
 RESULTS_syn_ideal_h
 [p2,t2,st2] = anova1(RESULTS_syn_ideal_h)
 RESULTS_syn_ideal_s
 [p3,t3,st3] = anova1(RESULTS_syn_ideal_s)

% %%
%  figure(11)
%  [c2,m2,h2,nms2] = multcompare(st2,'display','on')
%  figure(12)
%  [c3,m3,h3,nms3] = multcompare(st3,'display','on')

%
% %%
% stdev = std(RESULTS_syn_ideal')'
% Stderr = stdev./sqrt(5)
% varnc = var(RESULTS_syn_ideal')'
% ave = mean(RESULTS_syn_ideal')'
% stdev = std(RESULTS_raw')'
% Stderr = stdev./sqrt(5)
% varnc = var(RESULTS_raw')'
% ave = mean(RESULTS_raw')'
%
% %%
% [center, U, obj_fcn] = fcm(RESULTS_syn_ideal, 2);
```

```
% maxU = max(U);
% index1 = find(U(1, :) == maxU);
% index2 = find(U(2, :) == maxU);
% line(RESULTS_syn_ideal(index1, 1), RESULTS_syn_ideal(index1, 2), 'linestyle',...
% 'none','marker', 'o','color','g');
% line(RESULTS_syn_ideal(index2,1),RESULTS_syn_ideal(index2,2),'linestyle',...
% 'none','marker', 'x','color','r');
% hold on
% plot(center(1,1),center(1,2),'ko','markersize',15,'LineWidth',2)
% plot(center(2,1),center(2,2),'kx','markersize',15,'LineWidth',2)
%
% %%
% figure(11)
% idx4 = kmeans(RESULTS_syn_ideal,2,'distance','city');
% [silh3,h] = silhouette(RESULTS_syn_ideal,idx4,'city');
% set(get(gca,'Children'),'FaceColor',[.8 .8 1])
% xlabel('Silhouette Value')
% ylabel('Cluster')
%
% figure(12)
% idx5 = kmeans(RESULTS_syn_ideal,2, 'dist','city', 'display','iter');
% [silh4,h] = silhouette(RESULTS_syn_ideal,idx5,'city');
% set(get(gca,'Children'),'FaceColor',[.8 .8 1])
% xlabel('Silhouette Value')
% ylabel('Cluster')
%
% mean(silh3)
% mean(silh4)
% [idx5,cent4,sumdist] = kmeans(RESULTS_syn_ideal,2,'dist','city','display','final','replicates',
2);
% sum(sumdist)
% % %%
% % figure(13)
% % opts = statset('Display','final');
% %
% % [idx,ctrs] = kmeans(RESULTS_syn_ideal,2,...
% %                     'Distance','city',...
% %                     'Replicates',2,...
% %                     'Options',opts)
% %
% % plot(RESULTS_syn_ideal(idx==1,8),RESULTS_syn_ideal(idx==1,8),'r.','MarkerSize',12)
% % hold on
% % plot(RESULTS_syn_ideal(idx==2,8),RESULTS_syn_ideal(idx==2,8),'b.','MarkerSize',12)
% % plot(ctrs(:,1),ctrs(:,2),ctrs(:,3),ctrs(:,4),ctrs(:,5),'kx',...
% %      'MarkerSize',12,'LineWidth',2)
% % plot(ctrs(:,1),ctrs(:,2),ctrs(:,3),ctrs(:,4),ctrs(:,5),'ko',...
% %      'MarkerSize',12,'LineWidth',2)
% % legend('Cluster 1','Cluster 2','Centroids',...
% %        'Location','NW')
% % %%
% % d1 = mahal(Y,X) % Mahalanobis
% % d2 = sum((Y-repmat(mean(X),4,1)).^2, 2) % Squared Euclidean
% %
% % scatter(X(:,1))
% % hold on
% % scatter(Y(:,1),23,d1,'*','LineWidth',2)
% % hb = colorbar;
% % ylabel(hb,'Mahalanobis Distance')
% % legend('X','Y','Location','NW' )
%
% %%
% y = pdist(RESULTS_syn_ideal,'seuclidean')
% z = linkage(y, 'average')
% i = inconsistent(z)
```

```
% dendrogram(z)
% c = cophenet(z,y)
% T = cluster(z,'cutoff',c)


function [ FMS FMScore ] = FMScore( Y )
%   Creates Fugl-Meyer Score
%   take input range and create a FM score
%   based on ROM of specific Joint

FMS = [FMTransRules(Y(:,1),'elb'),FMTransRules(Y(:,2),'elb'),FMTransRules(Y(:,
3),'elb'),FMTransRules(Y(:,4),'elb'),FMTransRules(Y(:,5),'elb'),FMTransRules(Y(:,
6),'elb'),FMTransRules(Y(:,7),'elb');
       FMTransRules(Y(:,1),'wsp'),FMTransRules(Y(:,2),'wsp'),FMTransRules(Y(:,
3),'wsp'),FMTransRules(Y(:,4),'wsp'),FMTransRules(Y(:,5),'wsp'),FMTransRules(Y(:,
6),'wsp'),FMTransRules(Y(:,7),'wsp');
       FMTransRules(Y(:,1),'sfx'),FMTransRules(Y(:,2),'sfx'),FMTransRules(Y(:,
3),'sfx'),FMTransRules(Y(:,4),'sfx'),FMTransRules(Y(:,5),'sfx'),FMTransRules(Y(:,
6),'sfx'),FMTransRules(Y(:,7),'sfx');
       FMTransRules(Y(:,1),'srt'),FMTransRules(Y(:,2),'srt'),FMTransRules(Y(:,
3),'srt'),FMTransRules(Y(:,4),'srt'),FMTransRules(Y(:,5),'srt'),FMTransRules(Y(:,
6),'srt'),FMTransRules(Y(:,7),'srt');
       FMTransRules(Y(:,1),'saa'),FMTransRules(Y(:,2),'saa'),FMTransRules(Y(:,
3),'saa'),FMTransRules(Y(:,4),'saa'),FMTransRules(Y(:,5),'saa'),FMTransRules(Y(:,
6),'saa'),FMTransRules(Y(:,7),'saa');
       FMTransRules(Y(:,1),'wfx'),FMTransRules(Y(:,2),'wfx'),FMTransRules(Y(:,
3),'wfx'),FMTransRules(Y(:,4),'wfx'),FMTransRules(Y(:,5),'wfx'),FMTransRules(Y(:,
6),'wfx'),FMTransRules(Y(:,7),'wfx');
       FMTransRules(Y(:,1),'wud'),FMTransRules(Y(:,2),'wud'),FMTransRules(Y(:,
3),'wud'),FMTransRules(Y(:,4),'wud'),FMTransRules(Y(:,5),'wud'),FMTransRules(Y(:,
6),'wud'),FMTransRules(Y(:,7),'wud')];

FMScore = trace(FMS);
end


function [ FMS ] = FMTransRules( range, joint )
%translates Range to Fugl-Meyer Score
%   take input range and create a FM score
%   based on ROM of specific angle

ROM1 = 160; %elbow 160
ROM2 = 90;  %wrist sup pro
ROM3 = 90;  %shoulder ext flex
ROM4 = 130; %shoulder rot 160
ROM5 = 90;  %shoulder add abd
ROM6 = 90;  %wrist ext flex
ROM7 = 55;  %wrist ulna devia  65

FMS = 0;

    if(strcmp(joint,'elb'))
        if(abs(range(1,1)) <= (ROM1*1/3) || abs(range(5,1)) > (ROM5*5/16))
            FMS = 0;
        end

        if((abs(range(1,1)) > (ROM1*1/3)) && (abs(range(1,1)) <= (ROM1*2/3)) && abs(range(5,1))
<= (ROM5*5/16))
            FMS = 1;
        end

        if(abs(range(1,1)) > (ROM1*2/3) && abs(range(5,1)) <= (ROM5*5/16))
            FMS = 2;
        end
```

```
    end

    if(strcmp(joint,'wsp'))
        if(abs(range(2,1)) <= (ROM2*1/3) || abs(range(1,1)) > (ROM1*5/16) || abs(range(5,1)) >
(ROM5*5/16))
            FMS = 0;
        end
        if((abs(range(2,1)) > (ROM2*1/3)) && (abs(range(2,1)) <= (ROM2*2/3)) && abs(range(1,1))
<= (ROM1*5/16) && abs(range(5,1)) <= (ROM5*5/16))
            FMS = 1;
        end
        if(abs(range(2,1)) > (ROM2*2/3) && abs(range(1,1)) <= (ROM1*5/16) && abs(range(5,1)) <=
(ROM5*5/16))
            FMS = 2;
        end
    end

    if(strcmp(joint,'sfx'))
        if(abs(range(3,1)) <= (ROM3*1/3) || abs(range(1,1)) > (ROM1*5/16) || abs(range(5,1)) >
(ROM5*5/16))
            FMS = 0;
        end
        if((abs(range(3,1)) > (ROM3*1/3)) && (abs(range(3,1)) <= (ROM3*2/3)) && abs(range(1,1))
<= (ROM1*5/16) && abs(range(5,1)) <= (ROM5*5/16))
            FMS = 1;
        end

        if(abs(range(3,1)) > (ROM3*2/3) && abs(range(1,1)) <= (ROM1*5/16) && abs(range(5,1)) <=
(ROM5*5/16))
            FMS = 2;
        end
    end

    if(strcmp(joint,'srt'))
        if(abs(range(4,1)) <= (ROM4*1/3) || abs(range(1,1)) > (ROM1*5/16) || abs(range(5,1)) >
(ROM5*5/16))
            FMS = 0;
        end

        if((abs(range(4,1)) > (ROM4*1/3)) && (abs(range(4,1)) <= (ROM4*2/3)) && abs(range(1,1))
<= (ROM1*5/16) && abs(range(5,1)) <= (ROM5*5/16))
            FMS = 1;
        end

        if(abs(range(4,1)) > (ROM4*2/3) && abs(range(1,1)) <= (ROM1*5/16) && abs(range(5,1)) <=
(ROM5*5/16))
            FMS = 2;
        end
    end

    if(strcmp(joint,'saa'))
        if(abs(range(5,1)) <= (ROM5*1/3) || abs(range(1,1)) > (ROM1*5/16))
            FMS = 0;
        end

        if((abs(range(5,1)) > (ROM5*1/3)) && (abs(range(5,1)) <= (ROM5*2/3)) && abs(range(1,1))
<= (ROM1*5/16))
            FMS = 1;
        end

        if(abs(range(5,1)) > (ROM5*2/3) && abs(range(1,1)) <= (ROM1*5/16))
            FMS = 2;
        end
    end
```

```
    if(strcmp(joint,'wfx'))
        if(abs(range(6,1)) <= (ROM6*1/3) || abs(range(1,1)) > (ROM1*5/16) || abs(range(5,1)) >
(ROM5*5/16))
            FMS = 0;
        end

        if((abs(range(6,1)) > (ROM6*1/3)) && (abs(range(6,1)) <= (ROM6*2/3)) && abs(range(1,1))
<= (ROM1*5/16) && abs(range(5,1)) <= (ROM5*5/16))
            FMS = 1;
        end

        if(abs(range(6,1)) > (ROM6*2/3) && abs(range(1,1)) <= (ROM1*5/16) && abs(range(5,1)) <=
(ROM5*5/16))
            FMS = 2;
        end
    end

    if(strcmp(joint,'wud'))
        if(abs(range(7,1)) <= (ROM7*1/3) || abs(range(1,1)) > (ROM1*5/16))
            FMS = 0;
        end

        if((abs(range(7,1)) > (ROM7*1/3)) && (abs(range(7,1)) <= (ROM7*2/3)) && abs(range(1,1))
<= (ROM1*5/16))
            FMS = 1;
        end

        if(abs(range(7,1)) > (ROM7*2/3) && abs(range(1,1)) <= (ROM1*5/16))
            FMS = 2;
        end
    end


end

function [ xCol ] = xInputColmMaker( fileToRead )
%Read the file and create range of motion for each desired joint

% Import the file
T = importdata(fileToRead);

% Break out columns associated with rom.
ROM1 = range(T.data(:,3)); %ElbAnglesXZY:X  -- elbow flex
%ROM2 = range(T.data(:,4)); %ElbAnglesXZY:Y  -- not used
ROM3 = range(T.data(:,5)); %ElbAnglesXZY:Z  -- wrist sup

ROM4 = range(T.data(:,6)); %ShoAnglesXZY:X  -- shoulder flex
ROM5 = range(T.data(:,7)); %ShoAnglesXZY:Y  -- shoulder Rot
ROM6 = range(T.data(:,8)); %ShoAnglesXZY:Z  -- shoulder ABD

ROM7 = range(T.data(:,9)); %WristAnglesXZY:X -- wrist flex
ROM8 = range(T.data(:,10)); %WristAnglesXZY:Y -- wrist ulna dev
ROM9 = range(T.data(:,11)); %WristAnglesXZY:Z -- not used

xCol = [ROM1; ROM3; ROM4; ROM5; ROM6; ROM7; ROM8];


end


function [ xInputMatrix1 xInputMatrix2 xInputMatrix3 xInputMatrixAve xInputThetas1 xInputThetas2
xInputThetas3 xInputThetasAve ] = xInputMatrixMaker(SubjDirectory)
%%
```

```
%
%   Processing data for generating ROM for each subject
%   Trail1 = 01.txt
%   Trial2= 08.txt
%   Trial3= 015.txt
%   Shoulder Flexion
SEFROM1 = xInputColmMaker(strcat(SubjDirectory,'0 1.txt'));
SEFROM2 = xInputColmMaker(strcat(SubjDirectory,'0 8.txt'));
SEFROM3 = xInputColmMaker(strcat(SubjDirectory,'0 15.txt'));

SEFAVEROM = [mean([SEFROM1(1,1),SEFROM2(1,1),SEFROM3(1,1)]);
             mean([SEFROM1(2,1),SEFROM2(2,1),SEFROM3(2,1)]);
             mean([SEFROM1(3,1),SEFROM2(3,1),SEFROM3(3,1)]);
             mean([SEFROM1(4,1),SEFROM2(4,1),SEFROM3(4,1)]);
             mean([SEFROM1(5,1),SEFROM2(5,1),SEFROM3(5,1)]);
             mean([SEFROM1(6,1),SEFROM2(6,1),SEFROM3(6,1)]);
             mean([SEFROM1(7,1),SEFROM2(7,1),SEFROM3(7,1)])];


%   Trail1 = 02.txt
%   Trial2= 09.txt
%   Trial3= 016.txt
%   Shoulder ABD
SAAROM1 = xInputColmMaker(strcat(SubjDirectory,'0 2.txt'));
SAAROM2 = xInputColmMaker(strcat(SubjDirectory,'0 9.txt'));
SAAROM3 = xInputColmMaker(strcat(SubjDirectory,'0 16.txt'));

SAAAVEROM = [mean([SAAROM1(1,1),SAAROM2(1,1),SAAROM3(1,1)]);
             mean([SAAROM1(2,1),SAAROM2(2,1),SAAROM3(2,1)]);
             mean([SAAROM1(3,1),SAAROM2(3,1),SAAROM3(3,1)]);
             mean([SAAROM1(4,1),SAAROM2(4,1),SAAROM3(4,1)]);
             mean([SAAROM1(5,1),SAAROM2(5,1),SAAROM3(5,1)]);
             mean([SAAROM1(6,1),SAAROM2(6,1),SAAROM3(6,1)]);
             mean([SAAROM1(7,1),SAAROM2(7,1),SAAROM3(7,1)])];


%   Trail1 = 03.txt
%   Trial2= 010.txt
%   Trial3= 017.txt
%   Shoulder Rotation
SRTROM1 = xInputColmMaker(strcat(SubjDirectory,'0 3.txt'));
SRTROM2 = xInputColmMaker(strcat(SubjDirectory,'0 10.txt'));
SRTROM3 = xInputColmMaker(strcat(SubjDirectory,'0 17.txt'));

SRTAVEROM = [mean([SRTROM1(1,1),SRTROM2(1,1),SRTROM3(1,1)]);
             mean([SRTROM1(2,1),SRTROM2(2,1),SRTROM3(2,1)]);
             mean([SRTROM1(3,1),SRTROM2(3,1),SRTROM3(3,1)]);
             mean([SRTROM1(4,1),SRTROM2(4,1),SRTROM3(4,1)]);
             mean([SRTROM1(5,1),SRTROM2(5,1),SRTROM3(5,1)]);
             mean([SRTROM1(6,1),SRTROM2(6,1),SRTROM3(6,1)]);
             mean([SRTROM1(7,1),SRTROM2(7,1),SRTROM3(7,1)])];


%   Trail1 = 04.txt
%   Trial2= 011.txt
%   Trial3= 018.txt
%   Elbow Ext Flex
EROM1 = xInputColmMaker(strcat(SubjDirectory,'0 4.txt'));
EROM2 = xInputColmMaker(strcat(SubjDirectory,'0 11.txt'));
EROM3 = xInputColmMaker(strcat(SubjDirectory,'0 18.txt'));

EAVEROM = [mean([EROM1(1,1),EROM2(1,1),EROM3(1,1)]);
             mean([EROM1(2,1),EROM2(2,1),EROM3(2,1)]);
             mean([EROM1(3,1),EROM2(3,1),EROM3(3,1)]);
```

```
            mean([EROM1(4,1),EROM2(4,1),EROM3(4,1)]);
            mean([EROM1(5,1),EROM2(5,1),EROM3(5,1)]);
            mean([EROM1(6,1),EROM2(6,1),EROM3(6,1)]);
            mean([EROM1(7,1),EROM2(7,1),EROM3(7,1)])];

%   Trail1 = 05.txt
%   Trial2= 012.txt
%   Trial3= 019.txt
%   Wrist Pronation
WPROM1 = xInputColmMaker(strcat(SubjDirectory,'0 5.txt'));
WPROM2 = xInputColmMaker(strcat(SubjDirectory,'0 12.txt'));
WPROM3 = xInputColmMaker(strcat(SubjDirectory,'0 19.txt'));

WPAVEROM = [mean([WPROM1(1,1),WPROM2(1,1),WPROM3(1,1)]);
            mean([WPROM1(2,1),WPROM2(2,1),WPROM3(2,1)]);
            mean([WPROM1(3,1),WPROM2(3,1),WPROM3(3,1)]);
            mean([WPROM1(4,1),WPROM2(4,1),WPROM3(4,1)]);
            mean([WPROM1(5,1),WPROM2(5,1),WPROM3(5,1)]);
            mean([WPROM1(6,1),WPROM2(6,1),WPROM3(6,1)]);
            mean([WPROM1(7,1),WPROM2(7,1),WPROM3(7,1)])];

%   Trail1 = 06.txt
%   Trial2= 013.txt
%   Trial3= 020.txt
%   Wrist Flexion
WFROM1 = xInputColmMaker(strcat(SubjDirectory,'0 6.txt'));
WFROM2 = xInputColmMaker(strcat(SubjDirectory,'0 13.txt'));
WFROM3 = xInputColmMaker(strcat(SubjDirectory,'0 20.txt'));

WFAVEROM = [mean([WFROM1(1,1),WFROM2(1,1),WFROM3(1,1)]);
            mean([WFROM1(2,1),WFROM2(2,1),WFROM3(2,1)]);
            mean([WFROM1(3,1),WFROM2(3,1),WFROM3(3,1)]);
            mean([WFROM1(4,1),WFROM2(4,1),WFROM3(4,1)]);
            mean([WFROM1(5,1),WFROM2(5,1),WFROM3(5,1)]);
            mean([WFROM1(6,1),WFROM2(6,1),WFROM3(6,1)]);
            mean([WFROM1(7,1),WFROM2(7,1),WFROM3(7,1)])];

%   Trail1 = 07.txt
%   Trial2= 014.txt
%   Trial3= 021.txt
%   Wrist Ulnar Deviation
WUDROM1 = xInputColmMaker(strcat(SubjDirectory,'0 7.txt'));
WUDROM2 = xInputColmMaker(strcat(SubjDirectory,'0 14.txt'));
WUDROM3 = xInputColmMaker(strcat(SubjDirectory,'0 21.txt'));

WUDAVEROM = [mean([WUDROM1(1,1),WUDROM2(1,1),WUDROM3(1,1)]);
            mean([WUDROM1(2,1),WUDROM2(2,1),WUDROM3(2,1)]);
            mean([WUDROM1(3,1),WUDROM2(3,1),WUDROM3(3,1)]);
            mean([WUDROM1(4,1),WUDROM2(4,1),WUDROM3(4,1)]);
            mean([WUDROM1(5,1),WUDROM2(5,1),WUDROM3(5,1)]);
            mean([WUDROM1(6,1),WUDROM2(6,1),WUDROM3(6,1)]);
            mean([WUDROM1(7,1),WUDROM2(7,1),WUDROM3(7,1)])];


xInputMatrix1 = [EROM1 WPROM1 SEFROM1 SRTROM1 SAAROM1 WFROM1 WUDROM1];
xInputMatrix2 = [EROM2 WPROM2 SEFROM2 SRTROM2 SAAROM2 WFROM2 WUDROM2];
xInputMatrix3 = [EROM3 WPROM3 SEFROM3 SRTROM3 SAAROM3 WFROM3 WUDROM3];
xInputMatrixAve = [EAVEROM WPAVEROM SEFAVEROM SRTAVEROM SAAAVEROM WFAVEROM WUDAVEROM];


xDiag=diag(xInputMatrix1);
xInputThetas1 =  [xDiag(1,1) 0 0 0 0 0;
                  0 xDiag(2,1) 0 0 0 0;
                  0 0 xDiag(3,1) 0 0 0;
```

```
                0 0 0 xDiag(4,1) 0 0 0;
                0 0 0 0 xDiag(5,1) 0 0;
                0 0 0 0 0 xDiag(6,1) 0;
                0 0 0 0 0 0 xDiag(7,1)];
xDiag=diag(xInputMatrix2);
xInputThetas2 =  [xDiag(1,1) 0 0 0 0 0 0;
                0 xDiag(2,1) 0 0 0 0 0;
                0 0 xDiag(3,1) 0 0 0 0;
                0 0 0 xDiag(4,1) 0 0 0;
                0 0 0 0 xDiag(5,1) 0 0;
                0 0 0 0 0 xDiag(6,1) 0;
                0 0 0 0 0 0 xDiag(7,1)];
xDiag=diag(xInputMatrix3);
xInputThetas3 =  [xDiag(1,1) 0 0 0 0 0 0;
                0 xDiag(2,1) 0 0 0 0 0;
                0 0 xDiag(3,1) 0 0 0 0;
                0 0 0 xDiag(4,1) 0 0 0;
                0 0 0 0 xDiag(5,1) 0 0;
                0 0 0 0 0 xDiag(6,1) 0;
                0 0 0 0 0 0 xDiag(7,1)];
xDiag=diag(xInputMatrixAve);
xInputThetasAve =  [xDiag(1,1) 0 0 0 0 0 0;
                0 xDiag(2,1) 0 0 0 0 0;
                0 0 xDiag(3,1) 0 0 0 0;
                0 0 0 xDiag(4,1) 0 0 0;
                0 0 0 0 xDiag(5,1) 0 0;
                0 0 0 0 0 xDiag(6,1) 0;
                0 0 0 0 0 0 xDiag(7,1)];
end
```

Appendix B

Phone Screening Worksheet

Interview for Participation In:
Kinematics Of Upper-Limb Joint Synergies For Hemiplegia

Research being conducted by the Bionics Laboratory at the University of California, Santa Cruz
Sponsored by: Jacob Rosen, PhD
Investigator: Matt Simkins (PhD candidate), Aimen AL-Refai (MS candidate)

The following questions will be asked over the phone, in person, or by email. Responses will not be documented during the interview. However, if the candidates are determined to be suitable subjects they will complete a hard copy questionnaire at the start of their first scheduled session. The following format is in question “Q” response “R” format.

1. Q: What is your age?
R: If the candidate's is under the age of 18 they are ineligibly.

Question 1 will be asked for all candidates. The following questions do not apply to healthy candidates.

2. Q: Was your impairment caused by a stroke?
R: If the impairment was caused by an injury other than a stroke they are ineligible.

3. Q: Do you still have noticeable impairment in your left or right arm?
R: If impairment is too mild, or the candidate has fully recovered they are ineligible.

4. Q: How long ago did you have your stroke?
R: If less than 2 months they are ineligible.

5. Q: Can you bend the elbow to of your affected arm 90 degrees without support and keep it there?
R: If the subject can not perform this task they are ineligibly.

6. Q: Can you reach up and touch your ear?
R: If not the candidate is ineligible.

7. Q: Can you move your wrist up and down?
R: If the subject can not move there hand up and down they might be ineligible pending further review.
8. Q: Can you open and close your hand? Can you grasp objects?
R: If they can not open their hand, or particularity if they can not close it, they might be ineligible.

9. Q: Can you sit in a chair with your affected arm pointed straight down at your side?
R: If no the subject might be ineligible.

10. Q: Can you raise your arm in front of you at 90 degrees with your thumb pointing toward the ceiling and your elbow straight?
R: If not, the subject might be ineligible pending further review.

11. Q: Have you undergone any procedures involving Botox injections in your affected arm?
R: If yes the candidate might be ineligible pending further review.

12. R: At the conclusion of the interview the candidate will be told that they will be contacted regarding their eligible. If there is doubt about one of the responses, particularly questions 7 through 9 the investigator will research the question and provide a timely response to the candidate. If the candidate is deemed a suitable subject they will be scheduled for a session and logistical considerations will be discussed. The interview is then concluded with a statement of appreciation for the candidate or subject taking interest in this research.

Appendix C

Protocol

University of California, Santa Cruz	Investigational Protocol	Document No.: N/A
	Kinematics Of Upper-Limb Joint Synergies For Hemiplegia	Revision: 3 Effective: 4/24/12

Purpose:

The purpose of this research is to capture the synergistic movements of the paretic arms of hemiplegic stroke survivors using a motion-capture system. This data will be used to populate a "synergy matrix". A matrix representation allows for a thorough mathematical treatment of these complicated arm motions.

Scope:

The scope is further limited to subject preparation, discrete joint movements, and recording of data. A video taped clinical assessment may be conducted under a separate protocol herein referred to as an "assessment protocol". The requirements of the assessment protocol are within the scope of IRB51734 but are outside the scope of this protocol.

Referenced Documents:

Fugl-Meyer Assessment

Responsibilities:

Dr. Jacob Rosen, the faculty sponsor, and Matt Simkins, the PI, have ultimate responsibility for this all activities described in IRB51734. The PI will execute all activities pursuant to this protocol. The PI can be contacted at 530-204-7050. Activities described as part of the assessment protocol will be executed and evaluated by the parties described in Section 1 of IRB51734.

Subjects:

Subjects are screened prior to all sessions. Subject screening is performed in accordance to the screening procedure outlined in IRB51734. Gender and handedness are not criteria. The study will include 10 stroke survivors and 5 healthy individuals over the age of 18.

Equipment:

- Vicon Motion Capture System
- Reflective markers with double-sided adhesive
- A wrist hand brace suitable for wrist stabilization
- Stool or chair with no upper backrest
- Metronome

Background:

Most clinical assessments are subjective in nature and do not lend themselves to rapid, accurate measurements of joint angles. A motion capture system will allow for more sophisticated mathematical analysis of the movement of stroke.

This experiment is divided into a discrete joint movement section and a rhythmic section. One of the motivations for the rhythmic section is to determine knee frequencies. The lowest frequency of 2 Hz was based on the work of D. Sternad et al, see “Interaction of rhythmic and discrete pattern generators in single-joint movments”. Sternad used 2 Hz for the elbow. Given that the shoulder joint might not move as quickly as the elbow shoulder flexion only requires a minimum frequency of 1.5 Hz.

1 Nexus Procedure

1.1 IMPORTANT: Ensure that all Vicon software other than Nexus are closed. In particular, be sure that the program “Bodybuilder” is closed.

1.2 The first part of this section is assumed to be done concurrently with the General Procedure section.

1.3 Turn on camera control box and computer and open Nexus. If a software license error occurs, ensure that the license dongle is installed.

1.4 Open the “Synergy Experiment” project and create a new subject. Ensure that the subject number assigned in Nexus is consistent with the number assigned for the circle drawing and consent forms. Create “session 1” and make that session current.

1.5 Ensure that all of the cameras are active and assign a Vicon Skeleton Template, “.vst”.

For the right arm use:

C:\Program Files\Vicon\Nexus\ModelTemplates\Model_UpperLimbRight.vst

For the left arm use:

C:\ProgramFiles\Vicon\Nexus\ModelTemplates\Model_UpperLimbLeft.vst

After selecting the template you will be prompted to name the .vsk file. Accept the default by selecting OK.

1.6 Static Capture

1.6.1 After the markers are attached to the subject have the subject sit in the neutral position with the subject's upper arm pointing downward with the elbow bent at 90°. Use the default file name that begins with “SubjectXXCalXXX” When ready click start and stop. The capture session for calibration need only last a second.

1.6.2 IMPORTANT: the calibration section must be done with care! In the subject pane run

the pipeline "core processing". The white marker dots should appear. If the markers do not appear do not continue and contact the PI.

1.6.3 Manually label the markers and save.

1.7 Dynamic Trial

1.7.1 The tab with the clapboard is for dynamic capture. After the static capture is complete use only the dynamic tab. Make sure that "auto-incrementing" and "automatic overwriting" are checked.

1.7.2 Note: If the high speed camera is installed and a path is not defined the system will not permit dynamic data collection. In this event either set the path or disconnect the camera.

1.7.3 Name the first trial name "0". Succeeding trials are automatically numbered 0 1, 0 2 , … 0 n. The second number *must* correspond to the numbered action in the Data Sheets section. If the numbering gets mangled due to a repeated trial correct the numbering and note the deviation.

1.7.4 Dynamic trials are captured by pressing the start/stop button. Note, the cameras go off line automatically at the end of each capture. Be sure to click the "go live" button each time a dynamic capture starts.

1.8 Post Processing

1.8.1 For every dynamic trial run the "Reconstruct and Label" pipeline. The pipeline should include the following processing:

- Core Processing (required)
- Check and rename markers as needed (required)
- Apply Woltring filtering routine as needed
- Save trial (required)

2 Rythmic Data Analysis Using Open Office or Excel

2.1 Analogous instructions apply for Microsoft Excel.

2.2 Open the .xls files with open office Calc.

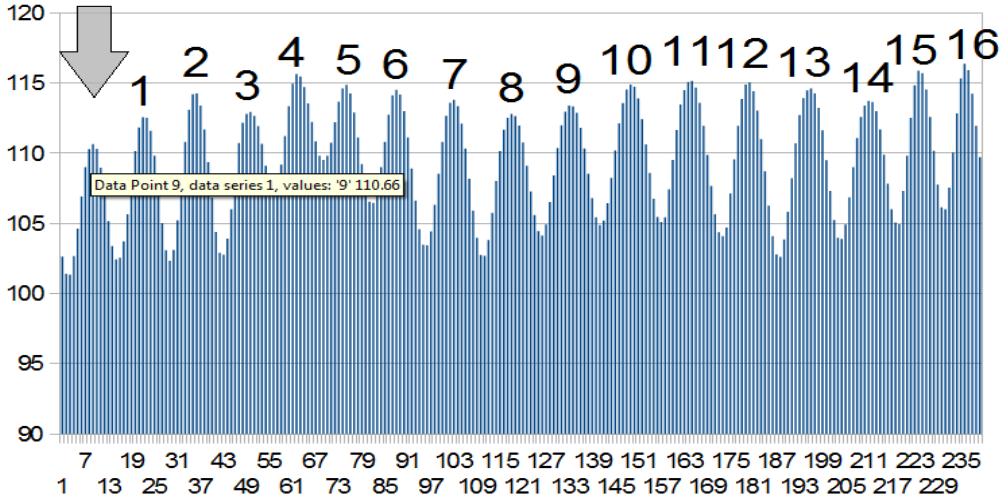
2.3 While opening file, under “Separated By” check the box next to “Tab”.

2.4 Select the column of interest by clicking on the associated letter.

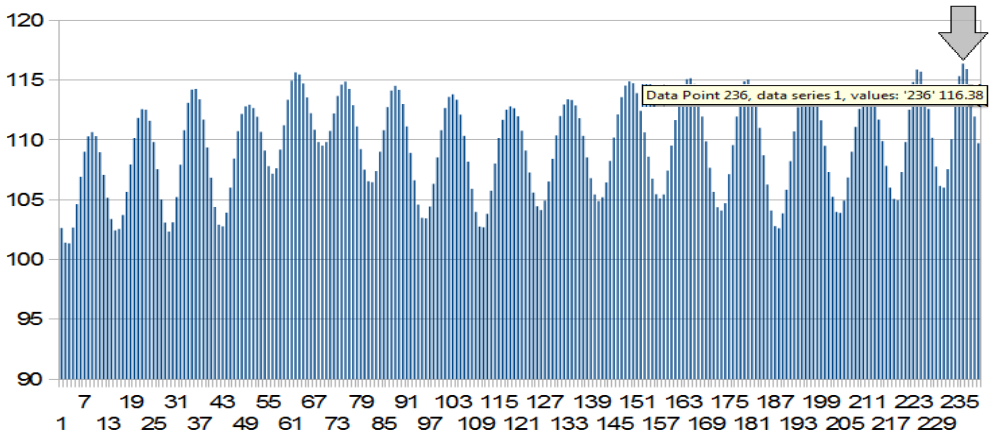
2.5 Click on the chart button (icon depicting a pie or bar chart).

2.6 Click “Finish” and inspect the chart as follows

2.7 Determine frequency. Select the left most peak by dragging the cursor over that portion of the graph. Record the data point number, in this case it is 9. Count the number of peaks to the right of the first, in this case it is 16.



2.8 Next select the right most peak and record the data point number. In this case it is 236.

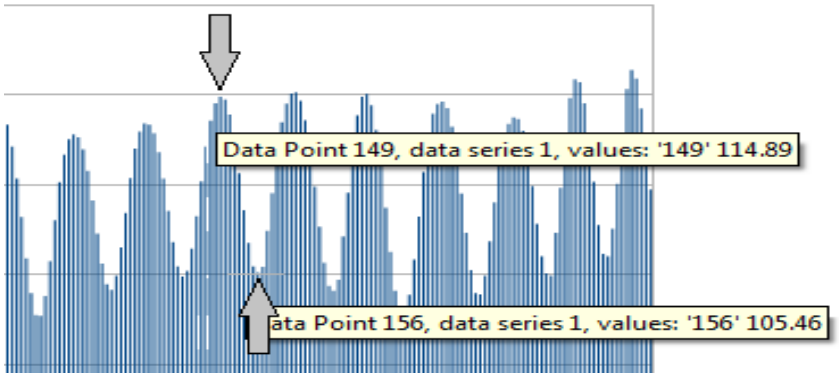


2.9 Finally, calculate the frequency using the following formula:

Frequency = [100 * (Number of Peaks)]/[(second data point number)-(first data point number)]

2.10 Calculate amplitude by selecting 3 representative cycles. A good "representative" sample should include amplitudes that are typical of the wave. A good selection is three consecutive samples that have peaks and troughs that are approximately at the same heights. The average is calculated from the 3 measurements.

2.11 For each cycle move the cursor over the peak and trough of the cycle. The magnitude is the last number in the information box. In the figure below the peak is 115 and 105 (with rounding).



2.12 The average amplitude is calculated as follows:

Av. Mag. = [(Peak 1) + (Peak 2) + (Peak 3) – (Trough 1) – (Trough 2) – (Trough 3)] / 3

2.13 For each subject use the table below to determine which file and column to evaluate.

File	Column	Frequency	Amplitude	File	Column	Frequency	Amplitude
0 29.txt	C		X	0 42.txt	C	X	X
0 30.txt	C		X	0 43.txt	C	X	X
0 31.txt	C	X	X	0 44.txt	I		X
0 32.txt	C	X	X	0 45.txt	I		X
0 33.txt	C	X	X	0 46.txt	I	X	X
0 34.txt	F		X	0 47.txt	I	X	X
0 35.txt	F		X	0 48.txt	I	X	X
0 36.txt	F	X	X	0 49.txt	C		X
0 37.txt	F	X	X	0 50.txt	C		X
0 38.txt	F	X	X	0 51.txt	C	X	X
0 39.txt	C		X	0 52.txt	C	X	X
0 40.txt	C		X	0 53.txt	C	X	X
0 41.txt	C	X	X				X

3 Bodybuilder

3.1 IMPORTANT: Ensure that all Vicon software other than Bodybuilder are closed. In particular, be sure that the program "Nexus" is closed.

3.2 Ensure that the following files are inserted into the session folder that contains the .c3d files being processed. Open Bodybuilder and associate (the icon showing a profile of an open hand) the 3 files are associated with the capture session being processed.

3.2.1 Right Arm

- Parameter File: Model_Right_Arm_Synergy.mp
- Model File: Model_Right_Arm_Synergy.mod
- Marker File: Model_Right_Arm_Synergy.mkr

3.2.2 Left Arm

- Parameter File: Model_Left_Arm_Synergy.mp
- Model File: Model_Left_Arm_Synergy.mod
- Marker File: Model_Left_Arm_Synergy.mkr

3.3 Open the file called Cal-XXX. Click on the "static" box and ensure that it is checked. Click the run button.

3.4 Close static and open file 0.c3d. Run the model and save.

3.5 Running model and writing output angles to file.

3.5.1 Open the session trial .c3d file being processed.

3.5.2 Associate the synergy .mp, .mod, and .mkr files.

3.5.3 Run the model (the icon with a gear in it).

3.5.4 Ensure that the model was correctly run[1].

3.5.5 Click on the window that graphically displays the markers. Select file → Write ASCII File.

3.5.6 Choose a destination folder and click a[R/L]elbAnglesXZY, a[R/L]ShoAngles XZY, and a[R/L]WristAnglesXZY and click OK.

3.5.7 Save and close trial.

3.5.8 Repeat this subsection until all of the files have been processed. Files are named 0.c3d, 0 1.c3d, 0 2.c3d, . . . 0 65.c3d stroke survivors and to 0.c3d to 0 53.c3d for healthy

1 This can be done by clicking on the "joints and axes" with graph checked and clicking on the output angle file name. A graph of the 3 Euler angles must appear Alternatively, if the model was correctly run additional "virtual markers" will appear under the torso of the model.

subjects.

4 MatLab:

4.1 Matlab processing is outside the scope of this protocol.

5 General Procedure:

5.1 All subjects must complete appropriate consent forms prior to continuing.

5.2 Assign subject number and prepare circle drawing sheet.

5.3 Attach markers using double-sided tape as pictured to the affected side. For otherwise healthy subjects, attach markers to the subjects dominant side.

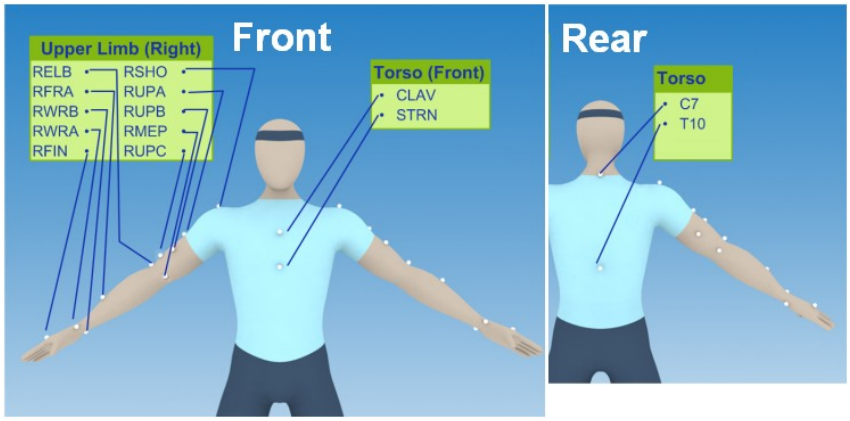


Illustration 1: Marker placement depicted for right arm.

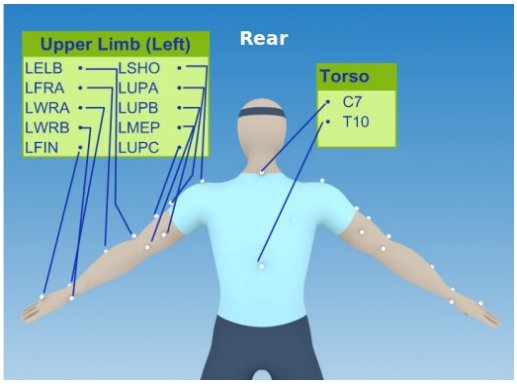


Illustration 2: Marker placement for left arm.

5.4 Stroke survivor only - Perform a video taped Fugl-Meyer assessment. While the assessment is being performed the motion-capture operator should prepare the system per the setup instructions in the previous secion.

5.5 Seat subject in chair with left arm facing the windows of Room 201 or building E2 (South).

5.6 Have subject draw a circle on paper starting from 12:00 position going counter clockwise. The drawing surface must be held at shoulder height at a distance roughly at the mid forearm if the subject was pointing forward.

5.7 Continue in according to the Data Sheets section. Arm motions are depicted below.

Illustration 3: Shoulder Flexion

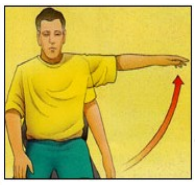


Illustration 4: Shoulder Abduction

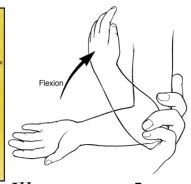


Illustration 5: Elbow Flexion

Illustration 6: Sholder Rotation

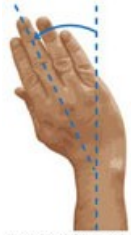


Illustration 7:

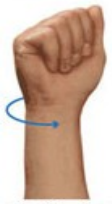


Illustration 8:

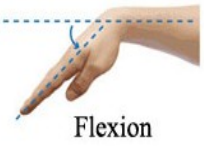


Illustration 9:

Data Sheets:

File Name	Joint	Comments
0	Circle Drawing	Counter clockwise starting at 12:00 position, approximately same radius as drawing.
0 1	Shoulder Flexion	Full Range. Not to exceed chest height (no scapular motion)
0 2	Shoulder Abduction	Full Range.
0 3	Shoulder Rotation	Full Range. Internal rotation to external and back to neutral position
0 4	Elbow Extension-Flexion	Full Range. Extension to flexion and back to neutral position
0 5	Wrist Pronation	Full Range. Pronate to neutral position
0 6	Wrist Fexion	Full Range.
0 7	Wrist Ulnar Dev.	Full Range. Ulnar deviation then back to neutral position.
0 8	Shoulder Flexion	Full Range. Not to exceed chest height (no scapular motion)
0 9	Shoulder Abduction	Full Range.
0 10	Shoulder Rotation	Full Range. Internal rotation to external and back to neutral position
0 11	Elbow Extension-Flexion	Full Range. Extension to flexion and back to neutral position
0 12	Wrist Pronation	Full Range. Pronate to neutral position
0 13	Wrist Fexion	Full Range.
0 14	Wrist Ulnar Dev.	Full Range. Ulnar deviation then back to neutral position.
0 15	Shoulder Flexion	Full Range. Not to exceed chest height (no scapular motion)
0 16	Shoulder Abduction	Full Range.
0 17	Shoulder Rotation	Full Range. Internal rotation to external and back to neutral position
0 18	Elbow Extension-Flexion	Full Range. Extension to flexion and back to neutral position
0 19	Wrist Pronation	Full Range. Pronate to neutral position
0 20	Wrist Fexion	Full Range.
0 21	Wrist Ulnar Deviation	Full Range. Ulnar deviation then back to neutral position.
0 22	Shoulder Flexion	Half of what was *achieved*. Not to exceed chest height (no scapular motion)
0 23	Shoulder Abduction	Half of what was *achieved*.
0 24	Shoulder Rotation	Half of what was *achieved*. Internal to external and back to neutral position
0 25	Elbow Extension-Flexion	Half of what was *achieved*. Extension to flexion and back to neutral position

0 26	Wrist Pronation	Half of what was *achieved*. Pronate to neutral position
0 27	Wrist Fexion	Half of what was *achieved*.
0 28	Wrist Ulnar Deviation	Half of what was *achieved*. Ulnar deviation then back to neutral position.
Subjects to use sinusoidal, rhythmic motion with a minimum of 5 cycles.		
0 29	Elbow Flexion	Metronome = 40 bpm, no subdivisions, full range, flexion and back to neutral.
0 30	Elbow Flexion	Metronome = 100 bpm, no subdivisions, full range, flexion and back to neutral.
0 31	Elbow Flexion	Full speed, full range
0 32	Elbow Flexion	Full speed, half range
0 33	Elbow Flexion	Full speed, tremor
0 34	Shoulder Flexion	Metronome = 40 bpm, no subdivisions, full range, flexion and back to neutral.
0 35	Shoulder Flexion	Metronome = 100 bpm, no subdivisions, full range, flexion and back to neutral.
0 36	Shoulder Flexion	Full speed, full range
0 37	Shoulder Flexion	Full speed, half range
0 38	Shoulder Flexion	Full speed, tremor
0 39	Elbow Flexion	Metronome = 40 bpm, no subdivisions, full range, flexion and back to neutral.
0 40	Elbow Flexion	Metronome = 100 bpm, no subdivisions, full range, flexion and back to neutral.
0 41	Elbow Flexion	Full speed, full range
0 42	Elbow Flexion	Full speed, half range
0 43	Elbow Flexion	Full speed, tremor
0 44	Wrist Fexion	Metronome = 40 bpm, no subdivisions, full range, flexion and back to neutral.
0 45	Wrist Fexion	Metronome = 100 bpm, no subdivisions, full range, flexion and back to neutral.
0 46	Wrist Fexion	Full speed, full range
0 47	Wrist Fexion	Full speed, half range
0 48	Wrist Flexion	Full speed, tremor
0 49	Elbow Flexion	Metronome = 40 bpm, no subdivisions, full range, flexion and back to neutral.
0 50	Elbow Flexion	Metronome = 100 bpm, no subdivisions, full range, flexion and back to neutral.
0 51	Elbow Flexion	Full speed, full range
0 52	Elbow Flexion	Full speed, half range
0 53	Elbow Flexion	Full speed, tremor

From here on is for stroke survivors only. Subject must wear a wrist brace on their affected side.		
0 54	Shoulder Flexion	With wrist stabilization. Full range, not to exceed chest height
0 55	Shoulder Abduction	With wrist stabilization, full range
0 56	Shoulder Rotation	With wrist stabilization, internal rotation to external and back to neutral position.
0 57	Elbow Extension-Flexion	With wrist stabilization, full Range. Extension to flexion and back to neutral.
0 58	Shoulder Flexion	With wrist stabilization. Full range, not to exceed chest height
0 59	Shoulder Abduction	With wrist stabilization, full range.
0 60	Shoulder Rotation	With wrist stabilization, internal rotation to external and back to neutral position.
0 61	Elbow Extension-Flexion	With wrist stabilization, full Range. Extension to flexion and back to neutral.
0 62	Shoulder Flexion	With wrist stabilization. Full range, not to exceed chest height
0 63	Shoulder Abduction	With wrist stabilization, full range
0 64	Shoulder Rotation	With wrist stabilization, internal rotation to external and back to neutral position.
0 65	Elbow Extension-Flexion	With wrist stabilization, full Range. Extension to flexion and back to neutral.

Bodybuilder Output Names

Right Arm:

Shoulder Abduction	RshoAnglesXZY:Z	(+)
Shoulder Flexion	RshoAnglesXZY:X	(-)
Shoulder Inner Rotation	RshoAnglesXZY:Y	(+)
Elbow Flexion	RelbAnglesXZY:X	(+)
Pronation	RelbAnglesXZY:Z	(+)
Wrist Flexion	RwristAngleXZY:X	(+)
Ulnar Deviation	RwristAngleXZY:Y	(-)

Left Arm:

Shoulder Abduction	LshoAnglesXZY:Z	(-)
Shoulder Flexion	LshoAnglesXZY:X	(+)
Shoulder Inner Rotation	LshoAnglesXZY:Y	(-)
Elbow Flexion	LelbAnglesXZY:X	(+)
Pronation	LelbAnglesXZY:Z	(-)
Wrist Flexion	LwristAngleXZY:X	(+)
Ulnar Deviation	LwristAngleXZY:Y	(+)

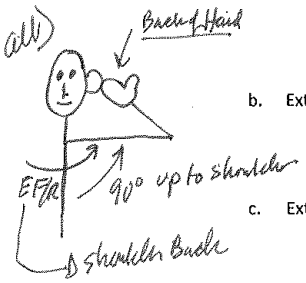
Appendix D

MFM Worksheet

Modified Fugl-Meyer Assessment

Subject:_____ Number:_____

1) Dynamic Movement within Flexion Synergy



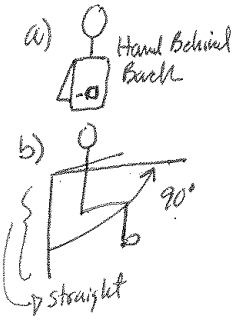
a. Abduction of GH joint (AB)
   - i. 0 No GH joint AB
   - ii. 1 Partial GH joint AB but less than 90
   - iii. 2 Equal to or greater than 90 GH joint AB

b. External Rotation (ER)
   - i. 0 No EF
   - ii. 1 Partial but less ER than unaffected side
   - iii. 2 Equal to or greater ER than unaffected side

c. External Flexion (EF)
   - i. 0 No EF
   - ii. 1 Partial but less EF than unaffected side
   - iii. 2 Equal to or greater EF than unaffected side

d. Supination (Sup)
   - i. 0 No Sup
   - ii. 1 Partial but less Sup than unaffected side
   - iii. 2 Equal to or greater Sup than unaffected side

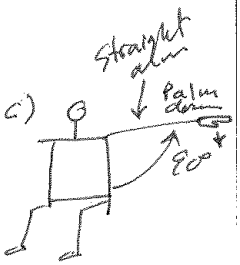
2) Movement Mixing Flexion and Extension Synergies



a. Internal Rotations GH
   - i. 0 Hand does not move posterior to the frontal plane
   - ii. 1 Hand does move posterior to the plane
   - iii. 2 Hand is placed on the small of the back with equivalent to unaffected

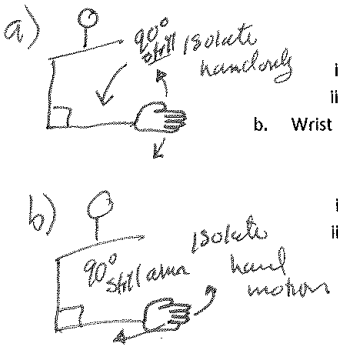
b. Shoulder Flexion
   - i. 0 Subject cannot achieve the position or elbow flexion occurs during test
   - ii. 1 Elbow flexion occurs during shoulder flexion onset or shoulder sub 90
   - iii. 2 Elbow fully extended and shoulder flexes to 90 equivalent to unaffected

c. Shoulder Abduction
   - i. 0 Subject cannot achieve the position or elbow flexion occurs during test
   - ii. 1 Elbow flexion occurs during shoulder flexion onset or shoulder sub 90
   - iii. 2 Elbow fully extended and shoulder flexes to 90 equivalent to unaffected



d. Pronation/Supination
   - i. 0 Subject cannot achieve test position even supported at proximal to the elbow by the clinician or EF/EE, GH joint AB, or IR/ER occurs on onset of test
   - ii. 1 EF/EE, GH AB, or IR/ER occurs on onset of test or can do sup/pro w/ hlp
   - iii. 2 Holds position and sup/pro equivalent to unaffected

3) Movements with little or no synergy dependence



a. Radius/Ulner deviation
   - i. 0 Subject cannot achieve test position even supported at proximal to the elbow by the clinician or EF/EE, GH joint AB, or IR/ER occurs on onset of test
   - ii. 1 EF/EE, GH AB, or IR/ER occurs on onset of test or can do sup/pro w/ hlp
   - iii. 2 Holds position and sup/pro equivalent to unaffected

b. Wrist Extension/Flexion
   - i. 0 Subject cannot achieve test position even supported at proximal to the elbow by the clinician or EF/EE, GH joint AB, or IR/ER occurs on onset of test
   - ii. 1 EF/EE, GH AB, or IR/ER occurs on onset of test or can do sup/pro w/ hlp
   - iii. 2 Holds position and sup/pro equivalent to unaffected

Appendix E

MFM and SMS Data

Subject #	MFM Score	SMS Trial 1	SMS Trial 2	SMS Trial 3	SMS Trial Mean
Healthy					
1	14	12	11	11	12
2	14	9	10	10	10
3	14	11	12	12	12
4	14	12	11	11	11
5	14	13	13	12	13
6	14	0	7	3	0
7	14	13	12	12	13
8 age matched	14	11	11	12	11
9 age matched	14	11	12	9	12
10 age matched	14	12	12	12	12
11 age matched	14	11	11	12	11
23 age matched	14	12	0	12	11
Stroke					
12	6	5	5	5	5
13	1	2	2	2	2
14	5	5	5	5	5
15	11	11	11	9	10
16	6	6	6	5	6
17	13	10	11	10	11
18	10	8	7	5	7
19	9	8	7	6	6
20	14	10	9	11	10
21	4	5	5	6	6
22	8	8	10	9	9

Bibliography

[1] *Clinical examination*. Mosby, London, 1 edition, 2004.

[2] *Circulation*, 125(1):e2–e220, 2012.

[3] S. Balasubramanian, R. Wei, R. Herman, and J. He. Robot-measured performance metrics in stroke rehabilitation. *Complex Medical Engineering*, pages 1–6, 2009.

[4] Signe Brunnstrom. *Movement Therapy in Hemiplegia: A Neurophysiological Approach*. Harper and Row, New York: Medical Dept., Harper and Row, 1970.

[5] James H. Cauraugh and Jeffery J. Summers. Neural plasticity and bilateral movements: A rehabilitation approach for chronic stroke. *Progress in Neurobiology*, 75:309–320, 2005.

[6] J. Cinkelj, M. Mihelj, D. Bacciu, M. Jurak, E. Guglielmelli, A. Toth, J. D. LaFonteyne, J.-L. Vershelde, S. Mazzoleni, J. V. Vaerenbergh, S. de Ruijter, and M. Munih. Assessment of stroke patients by wholebody isometric force-torque measurements ii: software design of the alladin diagnostic device. *EMBC*, 2005.

[7] Stacy Fritz Crystal L. Massie and Matthew P. Malcolm. Elbow extension predicts motor impairment and performance after stroke. *Rehabilitation Research and Practice*, 2011(1):23–30, 2011.

[8] R. L. DeGowen, P. R. Jochimsen, and E. O. Theilen. *DeGowin and DeGowin's bedside diagnostic examination*. McMillan, 5 edition, 1987.

[9] G. V. Dijck, J. V. Vaerenbergh, and M. V. Hulle. Posterior probability profiles for the automated assessment of the recovery of stroke patients. *AAAI*, pages 347–353, 2007.

[10] D.M.Morris, G. Uswatte, J. E. Crago, E.W. Cook, and E. Taub. The reliability of the wolf motor function test for assessing upper extremity function after stroke. *Archives of Physical Medicine and Rehabilitation*, 82(6):750–755, 2001.

[11] P. W. Duncan, M. Propst, and S. G. Nelson. Reliability of the fugl-meyer assessment of sensorimotor recovery following cerebrovascular accident. *Physical Therapy*, 63(10):1606–1610, 1983.

[12] John E. Freund. *Modern Elementary Statistics*. Prentice Hall Inc., 7 edition, 1988.

[13] S. L. Fritz, S. Blanton, G. Uswatte, E. Taub, and S. L. Wolf. Minimal detectable change scores for the wolf motor function test. *Neurorehabilitation and Neural Repair*, 23(7):662–667, 2009.

[14] AR Fugl-Meyer, L Jaasko, I Leyman, S Olsson, and S Steglind. The post-stroke hemiplegic patient. 1. a method for evaluation of physical performance. *Scandinavian journal of rehabilitation medicine*, 7(1), 1975.

[15] S. Hakkennes and J. L. Keating. Constraint-induced movement therapy following stroke: A systematic review of randomized controlled trials. *Australian Journal of Physiotherapy*, 51(4):221–231, 2005.

[16] T. Hester, R. Hughes, D. Sherrill, B. Knorr, M. Akay, J. Stein, and P. Bonato. Using wearable sensors to measure motor abilities following stroke. Proc. of the Int. Workshop on Wearable and Implantable Body Sensor Networks, 2006.

[17] B. Knorr, R. Hughes, J. Stein D. Sherrill, M. Akay, and P. Bonato. Quantitative measures of functional upper limb movement in persons after stroke. *IEEE EMBS, Neural Engineering*, pages 252–255, 2005.

[18] K. Krapp. *Activities of Daily Living Evaluation*. Gale Group, Inc., 2007.

[19] A. Kunkel, B. Kopp, and G. Müller et al. Constraint-induced movement therapy for motor recovery in chronic stroke patients. *Archives of Physical Medicine and Rehabilitation*, 80(6):624–628, 1999.

[20] G. Kwakkel, B. Kollen, and H. Krebs. Effects of robot-assisted therapy on upper limb recovery after stroke: A systematic review. *Neurorehanil Neural Repair*, 111(22):117–119, 2008.

[21] N. Leibowitz, N. Levy, S. Weingarten, Y. Grinberg, A. Karniel, Y. Sacher, C. Serfaty, and N. Soroker. Automated measurement of proprioception following stroke. *Disability and Rehabilitation*, 30(24):1829–1836, 2008.

[22] M. F. Levin, J. A. Kleim, and S. L. Wolf. What do motor recovery and compensation mean in patients following stroke? *Neurorehabilitation and Neural Repair*, 23(4):313–319, 2009.

[23] J. Liepert, W.H.R. Miltner, H. Bauder, M. Sommer, C. Dettmers, E. Taub, and C. Weiller. Motor cortex plasticity during constraint-induced movement therapy in stroke patients. *Neuroscience Letters*, 250(1):5 – 8, 1998.

[24] S. Mazzoleni, A. Toth, M. Munih, J. V. Vaerenbergh, G. Cavallo, P. Dario S. Micera, and E. Guglielmelli. Whole-body isometric force-torque measurements for functional assessment in neurorehabilitation: platform design, development and verification. *Journal of NeuroEngineering and Rehabilitation*, 6(38):1–15, Oct 2009.

[25] S.M. Michaelsen, S. Jacobs, A. Roby-Brami, and M.F. Levin. Compensation for distal impairments of grasping in adults with hemiparesis. *Experimental Brain Research*, 157(2):162–173, 2004.

[26] R. Perry. Chedoke-mcmaster stroke assessment development, validation and administration manual. *Physiotherapy*, 82(3):211–357, March 1996.

[27] T. Platz, C. Pinkowski, F. van Wijck, I. N. H. Kim, P. di Bella, and G. Johnson. Reliability and validity of arm function assessment with standardized guidelines for the fugl-meyer test, action research arm test and box and block test: a multicentre study. *Clinical Rehabilitation*, 19(4):404–411, 2005.

[28] Jacob Rosen and Joel C. Perry. Upper limb powered exoskeleton. *International Journal of Humanoid Robotics*, 4(3):529–548, 2007.

[29] J. Sanford, J. Moreland, L. R. Swanson, P. W. Stratford, and C. Gowland. Reliability of the fugl-meyer assessment for testing motor performance in patients following stroke. *Physical Therapy*, 73(7):447–454, 1993.

[30] Matt Simkins. Upper limb joint space modeling of stroke induced synergies using isolated and voluntary arm perturbations. September 2012.

[31] E. Taub, G. Uswatte, and R. Pidikiti. Constraint-induced movement therapy: a new family of techniques with broad application to physical rehabilitation—a clinical review. *Journal of Rehabilitation Research and Development,*, 36(3):237–251, 1999.

[32] Vicon.com. Vicon motion systems.

[33] Eric Wade, Avinash Rao Parnandi, and Maja J. Mataric. Automated administration of the wolf motor function test for post stroke assessment. *IEEE*, pages 1–7, 2010.

[34] J. M. Wagner, C. E. Lang, S. A. Sahrmann, D. F. Edwards, and A. W. Dromerick. Sensorimotor impairments and reaching performance in subjects with poststroke hemiparesis during the first few months of recovery. *Physical Therapy*, 87(6):751–765, 2007.

[35] Steven L. Wolf, Pamela A. Catlin, Michael Ellis, Audrey Link Archer, Bryn Morgan, and Aimee Piacentino. Assessing wolf motor function test as outcome measure for research in patients after stroke. *Stroke*, 32(7):1635–1639, 2001.

[36] K. M. Zackowski, A. W. Dromerick, S. A. Sahrmann, W. T. Thach, and A. J. Bastian. How do strength, sensation, spasticity, and joint individuation relate to the reaching deficits of people with chronic hemiparesis? *Brian*, 127(5):1035–1046, 2004.