ESTIMATION OF KINETICS USING A THREE-DIMENSIONAL MODEL OF THE HUMAN BODY

by

Erin Hutchinson

Bachelor of Mechanical Engineering
Georgia Institute of Technology, 1992

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 1994

Signature of Author_____
Department of Mechanical Engineering
December 7, 1993

Certified by_____
Dr. Patrick Riley
Thesis Supervisor

Accepted by_____
Ain A. Sonin
Chairman, Departmental Committee on Graduate Studies

ESTIMATION OF KINETICS USING A THREE-DIMENSIONAL MODEL OF THE HUMAN BODY

by

Erin Hutchinson

Submitted to the Department of Mechanical Engineering on December 7, 1993,
in partial fulfillment of the requirements for the degree of Master of Science.

Abstract

A method of calculating three-dimensional kinetics of the human body is presented. Net joint forces and torques were calculated using inverse dynamics and an eleven segment, three dimensional, sixty-six degree of freedom model of the human body. Segment linear and angular momentum were also calculated as well as whole body linear and angular momentum. This dynamic kinetics estimator was applied to chair-rise and gait trials performed by normal human subjects to determine the role of segmental dynamics in these everyday activities. For chair-rise it was determined that static loads dominate the joint forces and torques, but dynamic effects become increasingly significant while proceeding rostrally from the lower body joints to the joints of the upper body. For gait, it was determined that segmental dynamics were relatively more important during the swing phase than the stance phase of gait, and particular care should be given to obtaining meaningful estimates for segment linear and angular accelerations. The force and torque estimator was also applied to lifting trials by an elderly normal subject, and peak forces and torques on the back were correlated to specific anatomical configurations during the lifting trial. The whole body momentum curves calculated during chair-rise were consistent between normal subjects, physiologically interpretable, and may provide insight regarding the inability of some people to rise from a chair.

Thesis Supervisor: Patrick O. Riley
Title: Lecturer of Mechanical Engineering
Technical Director, Massachusetts General Hospital Biomotion Laboratory

Acknowledgements

I would like to thank Patrick Riley, the supervisor of this thesis, for giving me the opportunity to work on this project at the MGH Biomotion Lab. I am grateful for his advice, patience and input throughout the writing of this thesis and other papers, as well as my day to day work.

I would also like to thank David Krebs for his contributions to my papers, and research in general. Thanks to Dave Barret for Superplot, with which most of the figures in thesis were created. Thanks to Jose for his help obtaining and recommending experimental data sets. Thanks to Rebecca for her help with the IEEE Conference trip and in other areas. Thanks to the other RA's, S.S. and Krishna, for their suggestions during RA meetings. A general thank you to everyone at the Biomotion Lab, for making it an enjoyable place to work.

In addition, I would like to thank my grad student friends, who have made this year and a half at MIT much more enjoyable, and who have helped me remember that there is more to life than being an MIT graduate student.

Thanks to my family for their support both during my undergraduate years at Georgia Tech and during the past year and a half at MIT.

And most of all, to Alison, thanks for everything, without your support and encouragement, none of this would have been possible.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Biomechanics, and in particular the analysis of human motion, has many applications to our everyday lives. Activities, such as walking, rising from a chair, climbing and descending stairs, or lifting a heavy box are tasks that many of us take for granted, but are challenging or even impossible for many disabled and elderly people. These tasks present demands on the muscles and joints of our bodies, which must be able to meet these demands to successfully execute a given task. The focus of many researchers is to better understand the demands of these tasks on the human body. By gaining more insight into the details of human movement and its demands on the body, we will be in a better position to help those who have trouble performing these common activities. These same motion analysis techniques can also be applied to more complex and demanding activities and used to enhance athletic performance.

At the Massachusetts General Hospital Biomotion Laboratory an 11 segment, three-dimensional, 66 degree of freedom model of the human body was created to study the dynamics and kinetics of human movement. This model uses the ground reaction force, segment kinematics, and anatomical data as inputs. The goal of this thesis was to develop a method of using these inputs to calculate the kinetics of the body segments and joints and then apply this method to analyze various activities of daily life. Specifically, the net forces and torques on the joints and the segmental and whole body linear and angular momentum were to be calculated, taking into account and quantifying the role of the body segment

dynamics. The tasks that were to be analyzed with this dynamic estimator were chair-rise, gait and lifting.

1.1 Dynamic Motion Analysis

Human motion analysis is not a new field, with studies dating back to age of Aristotle. In 1836, Wilhelm and Eduard Weber combined cadaver measurements, tests on living subjects and mathematical pendulum models to study walking and running [1]. Marey introduced chronophotography, with which successive exposures on the same photographic plate were taken, and presented findings on the body vertical displacement, energy calculations, and the locus of the center of pressure on the foot in studies from 1873 to 1895 [2], [3], [4], [5]. Braun and Fischer, often credited with the classic study of gait, obtained stroboscopic data on the end points of flashing tubes attached to the limbs of subjects in the 1890's [6]. In 1950, Bresler and Frankel studied the forces and moments in the lower extremity during level walking by using continuously lighted bulbs placed on the subjects legs in a dark room [7].

Human dynamic motion analysis uses kinematic, kinetic and anatomic data as inputs to equations of motion, which can be used to find the forces and torques on the human joints. The forces and torques found are the net reactions needed at the joints to produce motion of the model as described by the equations of motion with the appropriate inputs. These are not necessarily the actual *in vivo* joint reactions, which include muscle co-contraction to effectively increase the actual joint reactions for the purpose of added stability across the joint. The net joint reactions represent the minimum values that could occur, in the absence of muscle co-contraction.

The kinematic inputs for dynamic motion analysis are the body segment linear and angular position, velocity and acceleration data during the time for which motion is to be analyzed. The most common way to obtain this data is to use photogrammetric techniques,

which record position data based on the location of markers mounted on appropriate locations on the body. The velocities and accelerations are then obtained by successive differentiation. One problem with this approach is that numerical differentiation methods tend to amplify noise in the data with each subsequent differentiation, thereby requiring the use of smoothing and/or filtering to obtain meaningful estimates of the derivatives. Methods using goniometers to measure relative joint angles are similar to photogrammetric techniques in that position data are collected and then must be differentiated. An alternative method uses accelerometers to directly measure segment linear and angular accelerations, from which velocity and position are determined by successive integrations. Opposite to differentiation, integration tends to reduce the effect of noise, however, knowledge of the accelerometer orientation is needed to remove the effects of gravity from the acceleration data. Ladin showed that accelerometers can be used to verify the numerical estimates for acceleration which are obtained from differentiated photogrammetric position data [8].

The kinetic inputs used in dynamic motion analysis consist of measured external forces acting on the body. During gait the external force consists only of the foot-floor interaction if air resistance is neglected [9]. The foot-floor interaction is the most often measured external force and is usually accomplished with the use of one or more piezoelectric or strain guage force plates. During chair-rise there is a foot-floor reaction, but also additional reactions between the chair and the buttocks and the hands and armrests when armrests are used to assist rising. One study on chair-rise used an instrumented chair to measure the external hand-armrest interaction [10]. For the lifting of weights, there once again is a foot-floor interaction, however, the weight being lifted must also be considered as an external force acting on the body.

The anatomical inputs used in dynamic motion analysis consist of body segment parameters. These include the segment mass, center of mass location and mass moment of inertia. Anthropometry, the study of human body measurements, is the most widely used method of inertial parameter estimation. Anthropometric methods use subject

measurements as inputs into equations which calculate body segment parameters based on previously studied populations. Scaling methods, computational methods and photogrammetry are three of the more popular anthropometric methods. Of these anthropometric methods, the scaling methods are the simplest and most commonly used.

Scaling methods use combinations of the subject's weight and body segment lengths to determine inertial parameters based on averages from studies done with cadaver body segments. Using cadavers to estimates inertial parameters is not a new endeavor. In 1860, Harless studied two male cadavers using immersion techniques [11], and in 1889 Braun studied three cadavers representative of the average height and weight for a German infantryman [12]. Dempster studied eight male cadavers over 50 years old [13]. Clauser reported segment masses and centers of mass for 13 male cadavers [14]. Chandler calculated complete inertia tensors based on six male cadavers [15]. More recently, living subjects have been used to estimate segment inertial parameters. Bernstein determined scaling equations from 76 males and 76 females [16]. Jensen studied 12 Canadian boys over a three year period [17]. In 1980, McConville determined regression equations from 31 representative U.S. Air Force males [18]. Young determined regression equations for 46 U.S. females in 1983 [19].

Computational methods mathematically model body segments as combinations of geometric solids with varying density. Harless modeled the trunk as a frustrum of a cone, since he could not use immersion techniques as he did for other body segments [11]. Amar used mathematical models to estimate moments of inertia of body segments [20].

Photogrammetry uses series of photographs to make contour maps of the body, from which segment volumes can be estimated. Segment volume estimations can be used with the assumption of constant density to calculate segment masses. Jensen [17], McConville [18], and Young [19] all employed stereophotogrammetry, the use of two cameras, in determining their scaling equations.

While scaling, computational, and photogrammetric methods are used most often, other methods exist. Computerized tomography (CT) is used to directly determine tissue density based on the amount of ionizing radiation that passes through the tissue. Information regarding the physical principles, equipment, and techniques used in CT imaging can be found in Macovski's radiology text book [21]. Sjostrom [22] and Huang [23] have reported tissue densities based on CT attenuation coefficients. Similar to CT, Magnetic resonance imaging (MRI) produces anatomical images, however, a magnetic field is used in place of ionizing radiation. Woltring first suggested using MRI to estimate segment inertial properties in 1984 [24]. Mungiole [25] and Brown [26] have both compared CT and MRI imaging for the determination of body segment parameters. A third imaging technique, dual photon absorptiometry (DPA), has recently been investigated as a possible means to determine inertial parameters. This technique irradiates tissue with two distinct energy levels, and differentiates tissue types based on differential attenuation. The first study using DPA, by Gotfredson in 1984 [27], could only distinguish between bone and soft tissue. By 1989, Heymsfield differentiated soft tissue into lean body mass and fat [28].

The accuracy of the estimates of the segment inertial parameters is often questionable, since there is no exact way of measuring these parameters and the parameters vary greatly from subject to subject. Yeadon showed that regression equations often are in significant error due to differences in tissue composition and morphology between cadavers and living subjects [29]. Chandler showed that mathematical modeling has a limited ability to accurately describe segmental anatomy [15]. The CT, MRI and DPA methods tend to be more accurate, especially for subjects with pathologies. However, in addition to the radiation risks (CT and DPA), the high equipment and usage costs make these methods uneconomical to use for every subject.

1.2 Chair-Rise Literature Review

Rising from a chair is one of the most common activities of daily life, yet also one of the most demanding. Berger showed that chair-rise produces greater knee torques than gait or stair-climb [30]. Many studies have examined joint torques during chair-rise in which specific parameters are varied. Burdett [31], Rodosky [32] and Wheeler [33] showed that varying the chair height affected the lower limb ranges of motion and extension moments. Fleckenstein demonstrated the effect of initial knee flexion angle on hip extension moments [34]. Alexander [35], Arborelius [36], Miller [37], Schultz [10] and Seedhom [38] studied the effects of using the hands to facilitate chair rise. Pai and Rogers studied the effect of varying the speed of rising [39]-[41]. While most previous studies, including [10], [35]-[41] and others by Nuzik [42] and Kralj [43], have assumed sagittally-symmetric motion, the model used for chair-rise studies in our lab by Berger [30], Hutchinson [44], Ikeda [45], Riley [46] and Schenkman [47] is three-dimensional and bilateral.

Although many studies which have calculated joint forces and torques have assumed a quasi-static model, in which dynamic terms were neglected, it has been suggested that inertial loads due to segment dynamics may not be negligible [10], [36], [44]. While many previous studies have used quasi-static models to examine how changes in various parameters affect total joint forces and torques, the importance of the dynamic components of force and torque has not yet been investigated. One of the goals of this thesis was to quantify the contribution of segment dynamics to joint forces and torques in young, healthy adults and investigate the effect that increasing the speed of ascent has on the dynamic components of force and torque.

1.3 Gait Literature Review

Many studies have examined the speed of gait and how it affects joint forces and torques. Cavanagh [48], May [49] and Paul [50] showed that increases in walking speed result in higher knee and hip torques during the swing phase of gait. Patriarco calculated the net forces and torques on the lower extremity joints during full gait cycles of normals [51]. Winter showed that lower extremity joint torques during the stance phase of gait increased with walking speed [52]. Winter also reported in a study of 18 subjects that while inter-subject ankle moments were consistent, inter-subject trends in knee and hip moments were less evident during the stance phase of gait [53]. Resultant joint torques during running have also been studied during the swing phase by Cavanagh [54], Chapman and Caldwell [55], Dillman [56], and Phillips and Roberts [57] and during the stance phase by Buczek and Cavanagh [58], Mann [59], Mann and Sprague [60], Robertson [61], Scott [62], Winter [53], [63]. These studies have shown that the joint torques were higher for running than for walking.

One of the goals of this thesis was to quantify the effect of body segment dynamics on net joint forces and torques. Peak forces and torques were calculated using a quasi-static method and a dynamic method during both the swing and stance phases of gait. Previous studies have shown that changes in gait speed affect the net joint forces and torques, however, for a given trial speed the role of segmental dynamics during different phases of the gait cycle has not yet been quantified. It was hypothesized that the results presented in this thesis would suggest that the differences between using a quasi-static method and a dynamic method would be more apparent during the swing phase than the stance phase of gait, and hence the effects of segmental dynamics would be more evident during the swing phase than the stance phase of gait.

1.4 Lifting Literature Review

A 1971 study reported that almost 700,000 of the 8,000,000 people in the United States that have a back impairment, incurred their injury during a one-time lifting or exertion [64]. It has been suggested by Andersson [65], Chaffin and Park [66] and Nachemson [67] that mechanical stress is responsible for injuries to the lower back. Gagnon studied torsional, flexion/extension and lateral bending net muscular moments at the L5/S1 joint during asymmetrical lifting and lowering [68]. Gagnon's study and a sagittally symmetric lifting study by Bush-Joseph [69] have demonstrated by varying the speed of lifting that dynamic factors affect net moments at the L5/S1 joint. Kishino [70] and Kumar [71] showed that back strength decreases as the lifting velocity increases. Sandover suggested that the rate of spine loading may affect the rate of lumbar structure degeneration [72]. Smeathers and Joanes suggested that the rate of spine loading may increase the risk of injury to intervertebral joint structures [73]. Marras found that varying the trunk angular acceleration and twist caused large changes in muscle forces, but relatively small changes in back joint torques [74]. Parniapour reported that back flexion/extension torques were significantly affected by the trunk flexion angle, with the maximum isometric torque occurring at 36 degrees of flexion [75]. Schipplein showed that as the weight of the load being lifted increased, the back flexion/extension torques increased, while the knee torques decreased [76]. Using a lifting machine, Stevenson determined that variations in lift could be attributed to five underlying components: force and power generated during the pulling phase, timing of the lift, momentum during wrist changeover, and the arm stroke during the pushing phase [77].

These studies have provided evidence for the importance of using a dynamic model when investigating biomechanics of the back during lifting. The back studies presented in this thesis used a dynamic model to investigate forces and torques on the back during

lifting, with straight legs and with bent knees. The development of the whole body dynamic estimator facilitated the calculation of back kinetics, which were the first such calculations performed in our lab.

1.5 Momentum Analysis Literature Review

While joint forces and torques are often reported in biomechanical studies, segmental and whole body linear and angular momentum have also been investigated, though less frequently. The trunk angular momentum of paraplegics during wheelchair propulsion has been studied by Masse as it relates to propulsion speed [78] and chair position [79]. Gagnon has suggested that inertial forces during lifting may facilitate transfer of momentum from the body to the load [80]. Ragheb has investigated the braking of horizontal linear momentum developed during running and its effect on the muscular power of the lower extremity [81]. In a study of Olympic long jumpers, Hinrichs reported that approximately 50% of the whole body angular momentum during flight was generated during takeoff, and the other 50% was generated during the run-up [82]. Figgen simulated somersaults in diving and showed that angular momentum had the greatest influence on motion followed by vertical velocity and the take-off angle [83]. A similar simulation was done by Yeadon, who calculated whole body angular momentum during twisting somersaults performed on a trampoline [84].

As mentioned in Section 1.2, many studies have been done on joint kinetics during chair-rise. However, relatively few studies have investigated segment and whole body momentum. Pai and Rogers have studied the linear momentum of the center of mass and shown that the trunk is the major contributor to the horizontal center of mass linear momentum, while the thigh is the major vertical contributor [40]. In another study Pai examined the deviations in the horizontal linear momentum time-histories in a comparison of elderly fallers with young non-fallers [85]. Schenkman described chair-rise as a four-

phase movement: flexion momentum, momentum transfer, extension, stabilization [47]. While these studies have associated various momenta with the chair-rise movement, whole body angular momentum has not been investigated. One of the goals of this thesis was to investigate whole body linear and angular momentum for normals performing chair-rise.

Chapter 2

Methods

Previously the Massachusetts General Hospital (MGH) Biomotion Laboratory used a quasi-static estimator capable of calculating joint torques on the lower limbs. The methods described in this thesis were used to develop a whole body dynamic estimator with the ability to calculate the net joint forces and torques, and the linear and angular momentum of the body segments and the whole body. The methods chapter of this thesis will first provide background information on the kinematic and kinetic data acquisition and the MGH three-dimensional human body model, including estimation of inertial parameters. Next, the data analysis tools will be discussed: differentiation and smoothing of data, inverse dynamics, and segment and whole body momentum analysis. Then the experimental methods used to apply the dynamic estimator to chair-rise, gait and lifting will be presented, followed by a discussion of the creation of a three-link model of the human body used to simulate chair-rise.

2.1 Data Collection

As described in Section 1.1, dynamic motion analysis requires three inputs: kinematic data, kinetic data, and anatomic data. Collection of kinematic and kinetic data will be discussed first, followed by a discussion of anatomic data collection, which provided the inertial parameters used by the MGH three-dimensional human body model.

2.1.1 Kinematic and Kinetic Data Acquisition

With the exception of the hand-held weight used in the lifting trials and neglecting air-resistance, the only external force acting on the body during chair-rise, gait and lifting was the ground, acting to support the feet. With these assumptions, the kinetic data consisted only of the foot-floor interaction for chair-rise and gait, but also included the hand-held weight during lifting trials. Two Kistler® piezoelectric force plates with an accuracy of ±1% of full scale were used to determine the ground reaction force vector on each foot. This provided normal force measurement with an accuracy of ±10 N and shear force measurement with an accuracy of ±5 N. A ground reaction force vector from each of the two force plates was measured.

Kinematic data were obtained using Telemetered Rapid Automatic Computerized Kinematic (TRACK™) software [86] and four Selspot II® optoelectric cameras. Two cameras on each side of the body facilitated collection of bilateral data. Thirteen arrays consisting of three to five infrared emitting diodes (irLEDs) per array were placed on 11 body segments (with redundant left and right arrays for the trunk and pelvis) and sampled at 153 Hz. The Selspot II® cameras each located the two dimensional positions of the irLEDs, which were used by the TRACK™ software to find the three dimensional position in space. Based on the size and shape of the array and data from at least three irLEDs per array, the TRACK™ software calculated the three dimensional array position and orientation. The system accuracy was verified to be within ±1 mm for position and within ±1° for orientation in a viewing volume that measured approximately 2 m in each dimension. Riley developed a method known as "pointing", which converted the irLED array position and orientation data to body segment position and orientation data using a combination of static anatomical landmarks and kinematic data which defined mean axes of rotation for the joints [87].

2.1.2 Anatomic Data Acquisition and the MGH Model

The last type of input into the dynamic estimator was patient specific anatomic data, which relied more on estimation techniques than did the precisely measured kinematic and kinetic data. The MGH eleven segment model of the human body consisted of the right and left feet, shanks, thighs and arms; and the pelvis, trunk and head. Each segment had three translational and three rotational degrees of freedom, which translated to a total of 66 degrees of freedom for the 11 segment model. The MGH model is unique in that it uses patient specific, three-dimensional inertial parameters as inputs to a dynamic model, which uses the full six degrees of freedom for each body segment. The spine was modeled as having a single joint at the L2/L3 level. Figure 2.1 shows the MGH human body model.

The male anthropometric data were based on McConville's stereophotogrammetric study of 31 males in the U.S. Air Force [18]. The female anthropometric data were based a stereophotogrammetric study by Young of 46 U.S. females [19]. Both of these studies used photogrammetry and the assumption of constant density to create regression equations for each segment mass and principle moments of inertia about the center of mass. These studies also listed the averages for body segment inertial parameters, distances from the segment centers of mass to anatomical joint axes, and the rotation matrices from anatomic axes to the principle axes.

Because the McConville and Young models consisted of 17 segments and the MGH model consisted of 11 segments, the regression equations given in these two studies could not be directly applied. Grierson developed methods for combining the segment masses, centers of mass and moments of inertia to fit the data from the McConville and Young studies to the MGH model [88]. Grierson calculated segment masses and inertias based on body weight, height and regression equations adapted to the MGH model. Subject body segment measurements were used to to find the segment centers of mass based on scaled

combinations of the segment centers of mass averages from the McConville and Young studies.

2.2 Data Analysis

Once the kinematic, kinetic, and anatomic data were collected, the data were analyzed. A primary focus of this thesis was to develop a method of using segmental dynamics to estimate joint kinetics and segmental and whole body momentum. Prior to calculating the forces and torques and the linear and angular momentums, it was necessary to obtain the linear and angular velocities and accelerations of the body segments. Software was written that numerically differentiated the linear and angular position data and then computed joint forces and torques and segment and whole body linear and angular momentum.

2.2.1 Velocity and Acceleration Estimation

Quasi-static force and torque calculations require only kinematic position data, kinetic data, and anatomic data.. However, prior to using a dynamic estimator, segment velocities and accelerations must first be obtained. Because our experimental setup did not include accelerometers or other means of directly measuring segment accelerations, numerical differentiation was employed to obtain estimates for segment velocities and accelerations. The three-dimensional directional components of linear and angular velocity and acceleration were calculated for each segment. The basic differentiation algorithm had been implemented by Grierson [88], but further work was done as a part of this thesis to reduce the effects of noise and obtain segmental velocities and accelerations in a form that could be used by the dynamic estimator. The raw data were first filtered with a six Hz filter and then differentiated with a Lagrangian five point differentiation scheme. The five point difference equation is an extension of the first difference equation, which is given in equation (2-1).

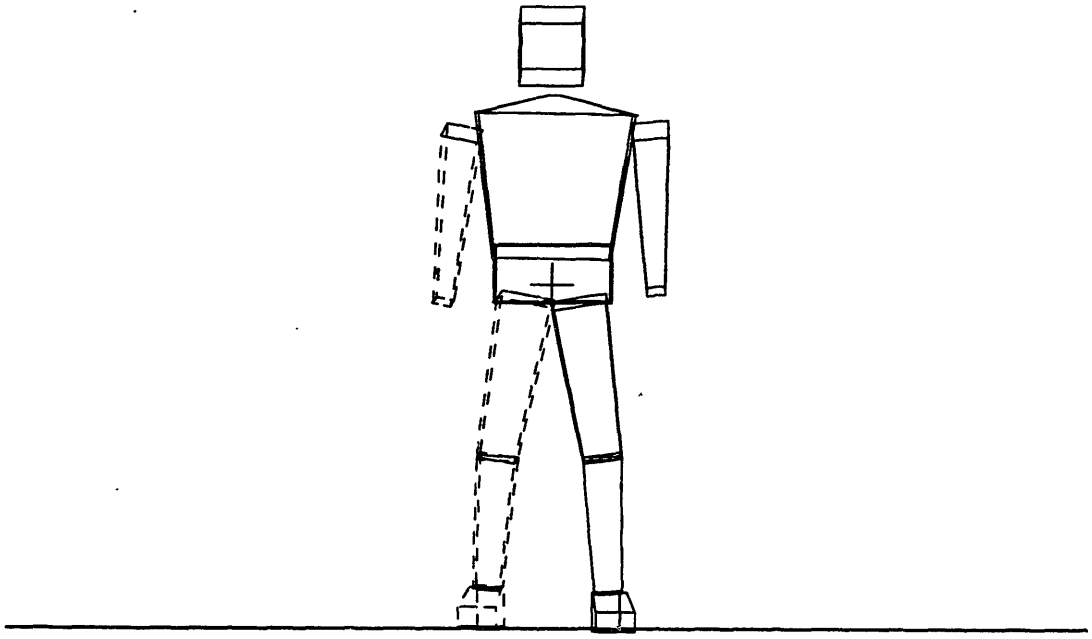

Figure 2.1 The MGH 11 segment three-dimensional, 66 D.o.F. model of the human body.

$$f'(x) = \frac{[f(x) - f(x-1)]}{T} \qquad (2\text{-}1)$$

In equation (2-1), f(x-1) and f(x) are two consecutive data points, and T is the time between the two data points. In this study, all data were sampled at 153 Hz, yielding a T of (1/153) seconds. Both linear and angular position data were differentiated with the five point difference equation in equation (2-2), given by Hildebrand [89].

$$f'(x) = \frac{1}{12*T}[f(x-2) - 8*f(x-1) + 8*f(x+1) - f(x+2)] \qquad (2\text{-}2)$$

Using five points instead of two reduced the effects of noise. The second derivative, f''(x) was found by substituting f'(x) in for f(x) in equation (2-2). Because all the data were collected prior to processing, values such as f(x+1) and f(x+2) were used, which would not have been possible with real-time processing. At the very beginning of a data set f(x-2) and f(x-1) did not exist, and at the very end of a data set f(x+1) and f(x+2) did not exist. To overcome this problem, startup and ending sequences described by Hildebrand [89] and implemented by Grierson [88] were adapted to work with the dynamic estimator. The derivative estimates were verified by comparing the results with derivatives taken using PV-WAVE® graphics software.

Due to the nature of numerical differentiation, noise in the position data was amplified when the first and second derivatives were taken. To compensate for this effect and provide more meaningful velocity and acceleration estimates, the linear and angular position data were smoothed, using median and mean smoothing routines before and after the derivatives were taken. Median smoothing was used to remove narrow spikes in the data, which were often caused by a high frequency "shake" or vibration of one of the arrays. The procedure used in median smoothing took a specified window of data points,

ordered them from lowest to highest, found the median, and replaced the original value with the median of the windowed data, as shown in equation (2-3).

$$f(x)_i = \text{median}\{f(x)_{i-n},..., f(x)_i,..., f(x)_{i+n}\} \quad \text{(for a window of 2n+1 data points)} \qquad (2\text{-}3)$$

This was effective for removing spikes because the spike data points were usually at either the high end or the low end of the window, and did not influence the resultant median. Mean smoothing was used to smooth out hard points where the derivative was discontinuous due to median smoothing or numerical differentiation. The procedure used in mean smoothing replaced a data point with an average of the data points in a window around the original data point, as shown in equation (2-4).

$$f(x)_i = \frac{1}{2n+1}(f(x)_{i-n} +... + f(x)_i + ...+ f(x)_{i+n}) \quad \text{(for a window of 2n+1 data points)} \qquad (2\text{-}4)$$

Windows for mean and median smoothing were chosen such that the amplitude of the original data was not significantly altered and smoother velocity and acceleration curves resulted. Without smoothing, spikes and hard points increased in amplitude with each subsequent differentiation.

After using the startup and ending sequences given by Hildebrand [89] and the median and mean smoothing routines, much of the data still had unsatisfactory spikes near the very beginning or the very end of a data set. To overcome this problem, a method of extending the ends of the position data trajectories prior to differentiation and smoothing was employed as described by Lesh [90]. In this method, the ends of the position data were extended using the odd reflection of the sequence. After the data were differentiated, the trajectories were returned to their original size. By using this method of extending the position data trajectories, both the numerical differentiation and the smoothing routines

were used continuously through both ends of the original position trajectory. This method of extending the curves was used in the linear and angular position data differentiation process. The C computer programs which performed the linear and angular velocity, acceleration and segment momentum calculations are listed in Appendix A.2.

2.2.2 Inverse Dynamics

TABLE 2.1

NOTATION

a_{ci}	linear acceleration vector of the center of mass of segment i
$f_{i-1,i}$	reaction force vector between segments i-1 and i at joint i-1
$f_{i,i+1}$	reaction force vector between segments i and i+1 at joint i
g	gravitational acceleration vector
h_i	angular momentum vector of segment i
H	angular momentum vector of the whole body
I_i	inertia tensor about the center of mass of segment i
m_i	mass of segment i
$N_{i-1,i}$	reaction torque vector between segments i-1 and i at joint i-1
$N_{i,i+1}$	reaction torque vector between segments i and i+1 at joint i
p_i	linear momentum vector of segment i
P	linear momentum vector of the whole body
$r_{i,cgi}$	vector from joint i to the center of mass of link i
$r_{i-1,cgi}$	vector from joint i-1 to the center of mass of link i
R_i	vector from the center of mass of segment i to a given reference point
v_{ci}	linear velocity vector of the center of mass of segment i
α_i	angular acceleration vector of segment i
ω_i	angular velocity vector of segment i

Given the kinematic, kinetic and anatomic inputs, the forces and torques acting on the joints were calculated recursively using Newton-Euler inverse dynamic analysis. Newton-Euler inverse dynamic analysis is based on Newton's Third Law of translational and rotational motion, as shown in equations (2-5) and (2-6).

$$\Sigma \text{Forces} = dp_i/dt \qquad \text{where } p_i = m_i v_i \tag{2-5}$$

$$\Sigma \text{Moments} = dh_i/dt \qquad \text{where } h_i = I_i \omega_i \tag{2-6}$$

Equation (2-5) states that the sum of the external forces acting on a rigid body is equivalent to the derivative of the linear momentum of the body with respect to time. Similarly, equation (2-6) states that the sum of external moments acting on a rigid body is equivalent to the change in the angular momentum of the body with respect to time. The body can be modeled as a collection of constant mass rigid body segments [88], and the body segment equations of motion are given by equations (2-7) and (2-8), where the right sides of the equations are the time derivatives of the segment linear and angular momentums, respectively.

$$\Sigma \text{Forces}_i = m_i a_{ci} \tag{2-7}$$

$$\Sigma \text{Moments}_i = I_i \alpha_i + (\omega_i \times I_i \omega_i) \tag{2-8}$$

Figure 2.2 shows a representative segment of the dynamic model used in this study. The external forces and moments acting on each body segment consist of a net force and a net moment reaction at both the proximal and the distal joints and a gravitational force. The Newton-Euler inverse dynamic equations for the model used in this study are given in equations (2-9) and (2-10).

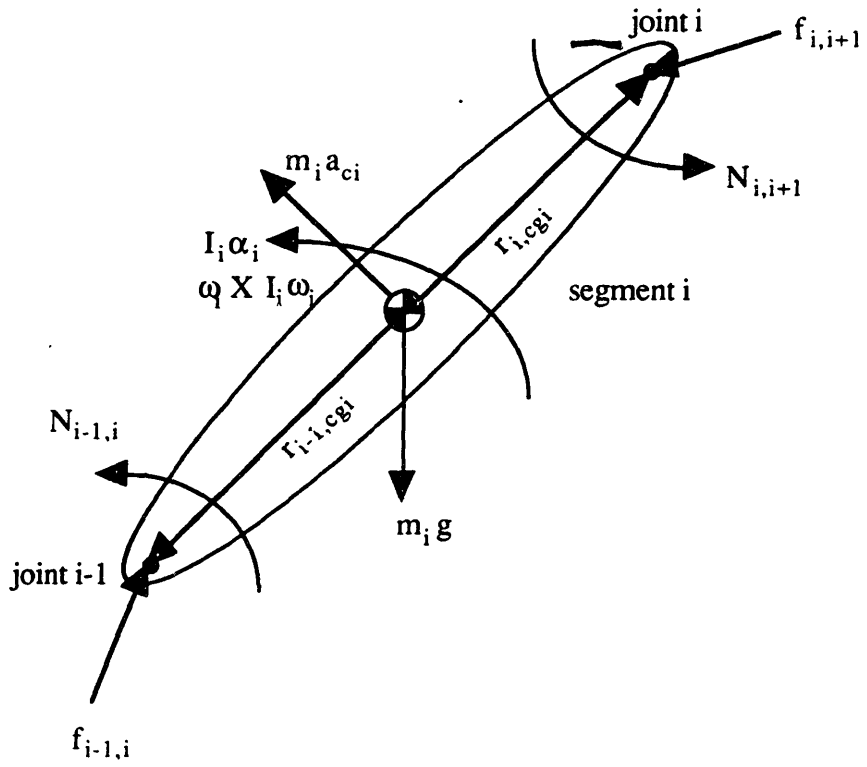

Figure 2.2 Representative Body Segment For Inverse Dynamics Model

$$f_{i-1,i} = f_{i,i+1} - m_i g + m_i a_{ci} \quad i = 1,...,11 \tag{2-9}$$

$$N_{i-1,i} = N_{i,i+1} - (r_{i,cgi} \times f_{i,i+1}) + (r_{i-1,cgi} \times f_{i-1,i}) + I_i\alpha_i + (\omega_i \times I_i\omega_i) \quad i = 1,..,11 \tag{2-10}$$

The force and torque reactions of the lower extremities were calculated as follows. The ground reaction force vector acting on the foot was determined from force plate data. The horizontal distance from the projection of the whole body center of mass on to the support surface to the center of force, termed the "gravity force moment arm" by Riley [46], facilitated the calculation of the ground reaction moment acting on the foot. Given the ground reaction force acting on the foot, the mass of the foot and the acceleration of the center of mass of the foot, the net reaction force at the ankle was calculated from equation (2-9). Similarly, given the ground reaction moment and the remainder of the terms on the right side of equation (2-10), the net reaction torque at the ankle was calculated. After the reaction force and torque at the ankle were calculated, the reaction force and torque at the knee were then calculated in a similar manner. From the reaction force and torque at the knee and the thigh kinematics, the reaction force and torque at the hip were calculated.

The upper body force and torque reactions were calculated using equations (2-9) and (2-10) and upper body kinematics. The forces and torques acting on the shoulders were calculated with no external forces or moments present at the distal ends of the arm segments. Similarly, the neck force and torque reactions were calculated with no external reaction at the distal end of the head segment. The force and torque reactions at the back joint were calculated using trunk kinematics and three distal reaction forces and torques in equations (2-9) and (2-10), due to the left and right shoulder and neck joint reactions. For each of the ten joints of the dynamic human body model, the three directional components of force and torque were calculated. The C programs which performed the net joint force and torque calculations are listed in Appendix A.3.

Because the dynamic estimator calculates the upper body kinetics from upper body kinematics, while the lower body kinetics are calculated from lower body kinematics and the ground reaction, this provided a means for checking the validity of the model. The upper body forces and torques were calculated at the back joint both from the head down to the back and from the ground up to the back. If the model were perfect, the reaction at the back joint would be the same regardless of which method was used. A comparison of these two methods is presented in the results chapter of this thesis for standing trials and chair-rise trials.

2.2.3 Momentum

Using body segment position and velocity data, the linear and angular momentum were calculated for each of the body segments and the whole body. All three of the linear and angular momentum three-dimensional vector components were calculated. The linear momentum of a given segment is given by equation (2-11), and equation (2-12) gives the angular momentum of a segment about its center of mass.

$$p_i = m_i * v_i \tag{2-11}$$

$$h_i = I_i * \omega_i \tag{2-12}$$

Once the segment linear and angular momentum were calculated, it was possible to calculate the whole body linear and angular momentum. The linear momentum of the whole body in global coordinates is the sum of all the segment linear momenta in global coordinates. Equation (2-13) was used to calculate the whole body linear momentum.

$$P = \Sigma p_i \quad (i = 1,..,11) \tag{2-13}$$

The whole body angular momentum must be calculated about a given reference point. Usually it was calculated about the body center of mass. For chair-rise, the whole body angular momentum was also calculated about a point which corresponded to the location halfway between the right and left ankle joints. This was considered meaningful because the combined ankle joint was approximately stationary during the chair-rise maneuver and represented a pivot point for the body. The whole body angular momentum about a given reference point (e.g. the body center of mass or the combined ankle joint) is given by equation (2-14), where 'R_i' is the distance from the center of mass of segment 'i' to the reference point, and 'X' represents the vector cross product operation.

$$H = \Sigma (R_i \text{ X } p_i + h_i) \quad (i = 1,..,11) \tag{2-14}$$

The C programs which performed the segmental momentum calculations are listed in Appendix A.2. The whole body momentum calculations were performed in the C program which calculated the joint forces and torques, listed in Appendix A.3.

2.3 Applications of the Dynamic Estimator

The software developed for the dynamic estimator included C programs to implement numerical differentiation and smoothing, force and torque calculations, and segment and whole body momentum calculations. Upon completion of this software, the dynamic estimator was applied to various common tasks including chair-rise, the swing and stance phases of gait, and lifting. The following sections describe the experimental protocol employed to analyze each of these tasks. Informed consent for each subject involved in these experiments was obtained prior to data collection.

In order to facilitate comparisons of different subjects performing the same tasks, several normalization techniques were employed. Forces were normalized to percent body

weight (%BW), and torques were normalized to percent body weight multiplied by meters (%BW*M) to allow comparison of the magnitudes of the force and torque curves. This normalization was done by dividing the force and torque values by the subject's body weight in newtons then multiplying by 100 percent. Whole body linear and angular momentum were normalized by dividing by the subject's mass in kilograms.

2.3.1 Chair-Rise

For the first chair-rise protocol the subject group consisted of ten normal adults, eight males and two females. The group had an average age of 30 (range 24-41) years, height of 175 (range 157-185) cm and weight of 70 (range 55-85) kg.

The subjects performing the first chair-rise protocol sat on a backless, armless chair that was adjusted to 80% of the subject's knee height. The subject's feet were positioned parallel, ten cm apart at the heels, with one foot on each force plate. The long axis of the subject's shanks were positioned in 18° flexion relative to the vertical. These constraints, as prescribed by Riley, were imposed to standardize the task and approximate normal foot placement [46]. The subjects sat with their arms folded across their chest and were instructed to rise to a standing position at their own natural speed without using their arms and then stand as still as possible. Each subject performed two chair-rise trials, and at least three seconds of data were collected for each trial.

A second protocol was employed for one of the subjects, in which the speed of ascent was varied. Using a metronome, this normal male subject (age 24, height 175 cm, weight 73 kg) performed three trials at 46 beats per minute and three trials at 92 beats per minute. In a previous study in our lab, the normal pace at which an adult rises from chair was determined to be approximately 52 beats per minute, which corresponded to an interval of 1.2 seconds between the beginning of movement and the completion of movement [46]. The command: 'ready, set, start, stand' was given in time with the metronome, such that

the subject began rising with 'start' and was erect by 'stand'. Prior to taking data, this task was practiced two or three times for both metronome speeds.

After data collection, analysis was performed to determine the importance of body segment dynamics on joint forces and torques. The contribution of segmental dynamics to net joint forces and torques was quantified by calculating the dynamic components of force and torque as percentages of the total force and torque. The dynamic force percentage was calculated with equation (2-15), and the dynamic torque percentage was calculated with equation (2-16). Both the dynamic components and the total force and torque used in equations (2-15) and (2-16) represent the magnitude of the vector composed of the three directional components of force or torque.

$$\%\text{Dynamic Force} = 100\%*(\text{dynamic component of force})/(\text{total force}) \quad (2\text{-}15)$$

$$\%\text{Dynamic Torque} = 100\%*(\text{dynamic component of torque})/(\text{total torque}) \quad (2\text{-}16)$$

These dynamic percentages were averaged over the portion of the three second trial during which motion occurred and the force and torque calculations were relevant. For the upper body joints, this consisted of the time starting with onset of forward trunk flexion and ending with stabilized stance. For the lower body joints, this consisted of the time beginning with liftoff from the chair and ending with stabilized stance. The lower body forces and torques calculated before liftoff were not relevant because the external reaction from the chair acting on the buttocks could not be measured since the experimental setup did not include a chair instrumented with force measurement capabilities. One of the advantages of calculating upper body joint forces and torques from the top down was that the upper body joint forces and torques could be calculated prior to liftoff from the chair. This was possible because the upper body force and torque calculations were independent of the reaction between the chair and the buttocks. The forces and torques calculated at the shoulders were not considered meaningful because the model did not account for the

interaction between the subject's crossed arms. As an approximation, which should be investigated further, the net effect of the two shoulder reactions acting on the trunk was considered to be legitimate even though the subject's arms were in contact with each other. Figure 2.3 shows the 11 segment model of one subject at four key points during a chair-rise trial: the onset of back flexion, liftoff from the chair, the time coinciding with several peak joint torques, and stabilized erect stance.

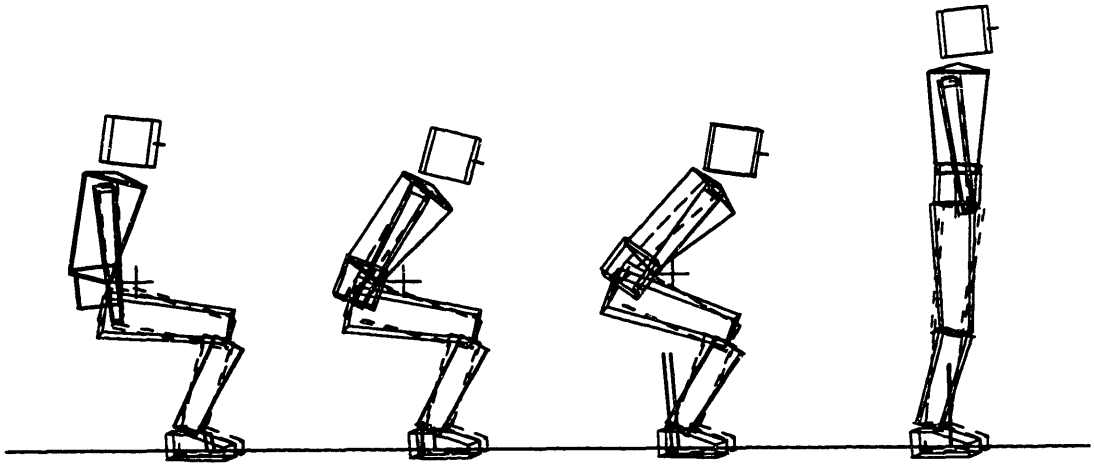

Figure 2.3 The 11 segment human body model for a representative trial performed by one subject at four key points during a chair-rise trial: the onset of back flexion, liftoff, the time coinciding with peak torques, and stabilized erect stance. The whole body center of mass is marked (+), and the ground reaction force (GRF) is also shown.

2.3.2 The Swing and Stance Phases of Gait

The subject group consisted of seven normal adults, five males and two females. The group had an average age of 27.6 (24-30) years, height of 173 (160-185) cm and weight of 69.4 (54.5-85.5) kg. Although for some subjects two trials were available, for consistency, one gait trial from each subject was used.

After performing practice trials, each subject performed two gait trials, walking to the pace of a metronome at 120 beats per minute. Krebs determine this speed to be the average pace at which normal humans naturally walk [91]. Keeping pace with the metronome required the subjects to have a heel strike coincide with each beat of the metronome. A stance phase cycle began with heel strike and ended with toeoff of the same foot, and a swing phase cycle began with toeoff and ended with heel strike of the same foot. Figure 2.4 shows the MGH 11 segment model of one subject at four key points during a chair-rise trial: the stance leg heel-strike, the swing leg toe-off, the swing leg heel-strike, and the stance leg toe-off. Time was normalized by reporting the force and torques values in percent of the swing phase cycle or the stance phase cycle.

Once this gait data were collected, the importance of segmental dynamics was assessed. Unlike the chair-rise analysis, the dynamic percentages were not averaged over the trials. The reason for this was that the curves for total force and torque frequently changed from positive to negative or vice versa at times when the dynamic force and torque components were non-negative. This caused the dynamic percentages given in equations (2-15) and (2-16) to go to infinity as the denominator, which consisted of the total force or torque, went to zero. For this reason other means of quantifying the contributions of segmental dynamics to joint kinetics were needed. This was accomplished by calculating the lower limb joint forces and torques with both the dynamic estimator and a quasi-static

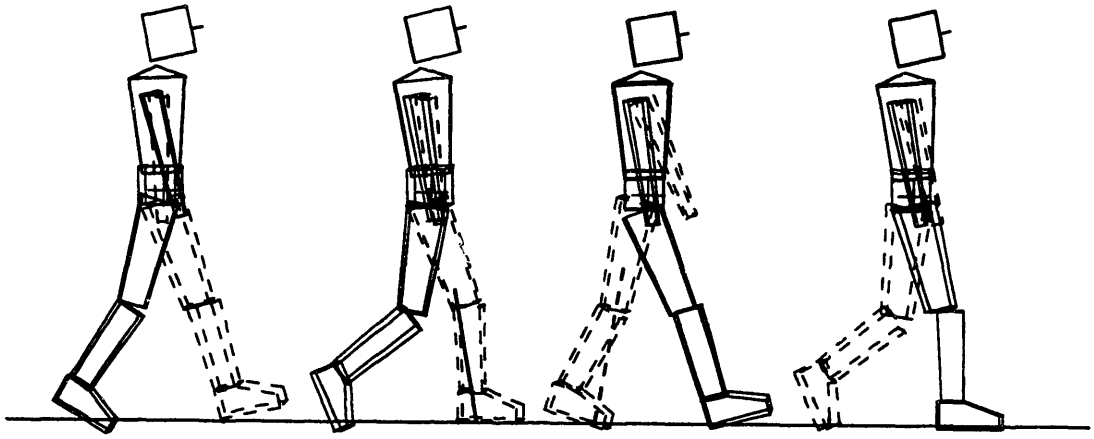

Figure 2.4 The MGH 11 segment model of one subject at four key points during a gait trial: the stance leg heel-strike, the swing leg toe-off, the swing leg heel-strike, and the stance leg toe-off. The whole body center of mass is marked (+), and the GRF is also shown.

estimator and then comparing the difference at peak values and the root mean squared error between the two curves. While equations (2-9) and (2-10) represent the dynamic equations used to calculate net joint forces and torques, equations (2-17) and (2-18) represent the comparable quasi-static equations.

$$f_{i-1,i} = f_{i,i+1} - m_i g \quad i = 1,...,11 \tag{2-17}$$

$$N_{i-1,i} = N_{i,i+1} - (r_{i,cgi} \times f_{i,i+1}) + (r_{i-1,cgi} \times f_{i-1,i}) \quad i = 1,...,11 \tag{2-18}$$

By finding the differences between the dynamic and quasi-static equations, the dynamic components of force and torque, which are given in equations (2-19) and (2-20), can be extracted.

$$f_{dyn} = m_i a_i \tag{2-19}$$

$$t_{dyn} = I_i \alpha_i + (\omega_i \times I_i \omega_i) + (r_{i-1,cgi} \times m_i a_i) \tag{2-20}$$

The dynamic component of force shown in equation (2-19) simply accounts for the force due to the linear acceleration of the segment center of mass. Similarly, the $I_i \alpha_i$ term in equation (2-20) accounts for torque which is caused by the angular acceleration of the segment. The $(\omega_i \times I_i \omega_i)$ term in equation (2-20), often referred to as the gyroscopic torque, accounts for torques that arise due to changes in the inertia tensor which occur as the relative orientation between segments change. The $(r_{i-1,cgi} \times m_i a_i)$ term in equation (2-20) is the torque that occurs due to the dynamic component of force $(m_i a_i)$. These are the dynamic components of force and torque which are neglected in studies that use quasi-static analysis.

2.3.3 Lifting

The lifting data were only collected for one subject on several different occasions. The subject was an elderly male (age: 83 years, height: 160 cm, weight: 54 kg) with an pressure instrumented femoral head, hip prosthesis.

The subject lifted a 25 pound weight in a milk crate, by handles on the sides of the crate. The 25 pound weight was modeled as an external force of 12.5 pounds acting on the distal end of each arm segment. The dynamics of the weight were not modeled.

At his own pace, the subject bent over picked up the weight, straightened his body, then set the weight up on a shelf that was about the same height as his chest. The subject performed several lifting trials with his legs straight and several with his legs bent.

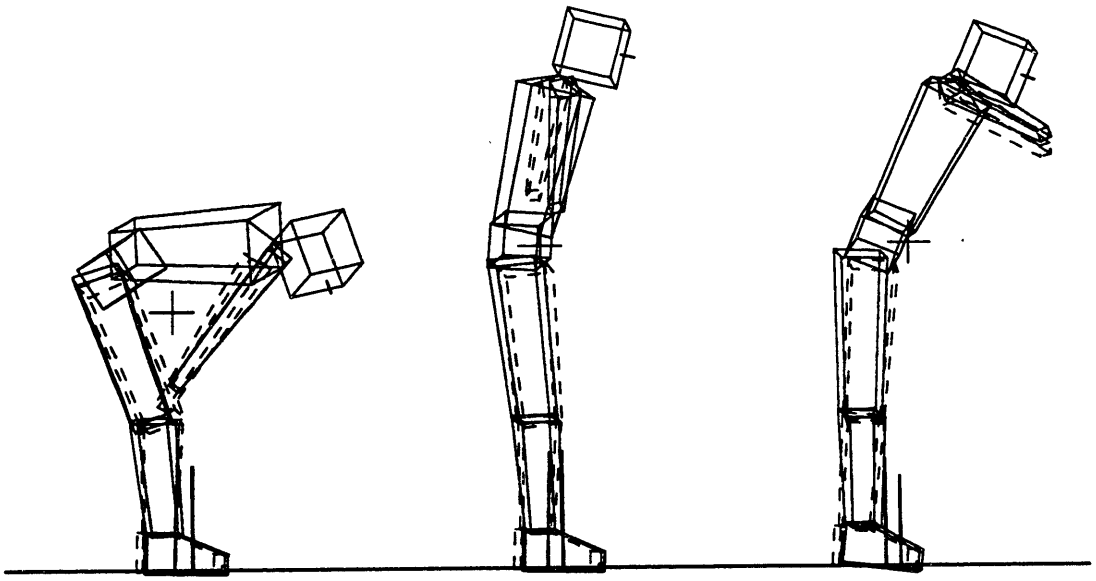

Figure 2.5 The MGH 11 segment model of the subject at three key points during a lifting trial: picking up the weight while bending over, erect stance while holding the weight, and bending forward to set the weight on a shelf. The whole body center of mass is marked by (+), and the GRF is also shown.

2.4 Simulation of Chair-Rise using a Three-Link Model of the Human Body

The human body has been modeled as a three-link system to study various human movements. Studies by Golliday [92], Hemàmi [93], Quintern [94], Wilson [95] and Kallel [96] have used three-link models consisting of a trunk and two legs to investigate biped balance. Biped locomotion has been studied with a three-link model by Hemami [97] and Hurmuzlu [98]. Another study by Hemami used a planar three-link model, similar to the one presented in this thesis, to simulate a bowing motion with a time delay to emulate the nerve signal transmission time [99]. Pai and Rogers used a three-link model to study chair-rise [38], [39].

As part of this thesis, a three-link open chain inverse dynamic model of the human body was created to model chair-rise. The motion of the linkage was constrained to the sagittal plane, and the first link was constrained by requiring the first joint to be fixed to the ground. This model had three degrees of freedom with one for each joint angle. The link inertial parameters were based on combinations of the segments used in the MGH 11 segment model. A simulation was implemented which used desired angular position, velocity and acceleration trajectories to move the three-link model from a sitting to a standing position. The simulated joint reaction torques were calculated using Lagrangian inverse dynamics and the results were compared to joint torques calculated with the MGH model for two trials performed by a human subject. Although three-link models of the human body have been used to study chair-rise before [38], [39], this simulation was useful in that it provided a comparison of a relatively simple three-link, three degree of freedom, planar model with the more complex MGH 11 segment, 66 degree of freedom, three-dimensional model. In addition, the torques obtained with the Lagrangian inverse dynamics used with the three-link model served as an alternative calculation method, with

which to check the torques obtained with Newton-Euler inverse dynamics equations used with the MGH model.

2.4.1 Inertial Parameters

The three-link model represented both shanks with link one, both thighs and the pelvis with link two, and the head, arms and trunk with link three, as shown in Figure 2.6. Joint one represented the ankle joint, joint two represented the knee joint and joint three represented the back or lumbar joint. Constraining the first joint to be fixed to the ground was assumed to be a reasonable approximation, since the ankle joint is relatively stationary during chair-rise. Absolute joint angles were used such that flexion of a joint acted to increase the joint angle, and joint angles were all equal to zero when the model was in the standing position. The model was intended to simulate chair-rise trials by a human subject in which rising was performed without the aide of the arms, which were folded across the chest.

The mass, moment of inertia, length, and center of mass of each link were estimated by using anthropometric data from a normal female subject available from the MGH Biomotion Laboratory and combining segments of the MGH 11 segment model. This allowed comparison of the simulation results with actual chair-rise trials performed by this subject.

For link one, the masses and inertias of the two shank segments were added, and the leg segment lengths and center of gravity locations were averaged. These combinations are shown in equations (2-21) - (2-24).

$$M_1 = M_{l.shank} + M_{r.shank} \tag{2-21}$$

$$L_1 = (L_{l.shank} + L_{r.shank})/2 \tag{2-22}$$

$$L_{cg1} = (L_{cg.l.leg} + L_{cg.r.leg})/2 \tag{2-23}$$

$$I_1 = I_{l.leg} + I_{r.leg} \tag{2-24}$$

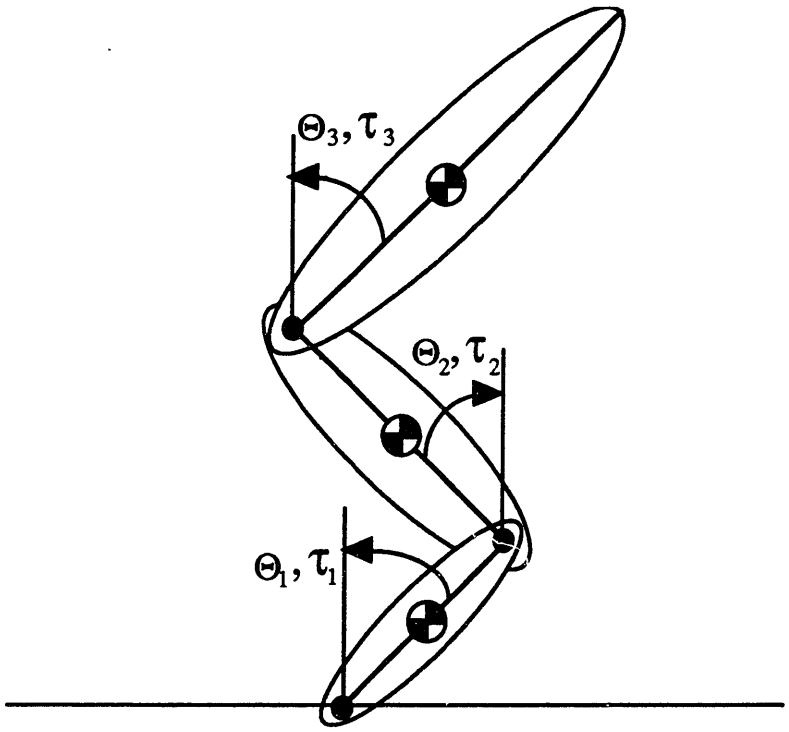

Figure 2.6 The three-link model of the human body.

For link two, the mass was obtained by combining the masses of the two thighs and the pelvis. The combined inertia was found using the inertias of the two thighs, the pelvis and the parallel axis theorem. These calculations, the link length, and the center of gravity location were calculated using equations (2-25) - (2-28).

$$M_2 = M_{l.thigh} + M_{r.thigh} + M_{pelvis} \tag{2-25}$$

$$L_2 = (L_{l.thigh} + L_{r.thigh})/2 + L_{pelvis} \tag{2-26}$$

$$L_{cg2} = (M_{l.thigh}*L_{knee\text{-}l.thigh.cg} + M_{r.thigh}*L_{knee\text{-}r.thigh.cg} + M_{pelvis}*L_{knee\text{-}pelvis.cg}])/M_2 \tag{2-27}$$

$$I_2 = I_{l.thigh} + I_{r.thigh} + I_{pelvis} + M_{l.thigh}*(L_{l.thigh.cg\text{-}cg2})^2 + M_{r.thigh}*(L_{r.thigh.cg\text{-}cg2})^2 + M_{pelvis}*(L_{pelvis.cg\text{-}cg2})^2 \tag{2-28}$$

Parameters of the two arms, the head, and the trunk were used to calculate the mass, inertia, link length, and center of gravity location of link three, using equations (2-29) - (2-32).

$$M_3 = M_{l.arm} + M_{r.arm} + M_{head} + M_{pelvis} \tag{2-29}$$

$$L_3 = L_{trunk} + L_{head} \tag{2-30}$$

$$L_{c3} = (M_{l.arm}*L_{back\text{-}l.arm.cg} + M_{r.arm}*L_{back\text{-}r.arm.cg} + M_{head}*L_{back\text{-}head.cg} + M_{trunk}*L_{back\text{-}trunk.cg})/M_3 \tag{2-31}$$

$$I_3 = I_{l.arm} + I_{r.arm} + I_{head} + I_{trunk} + M_{l.arm}*(L_{l.arm.cg\text{-}cg3})^2 + M_{r.arm}*(L_{r.arm.cg\text{-}cg2})^2 + M_{head}*(L_{head.cg\text{-}cg2})^2 + M_{trunk}*(L_{trunk.cg\text{-}cg2})^2 \tag{2-32}$$

Before these calculations were performed, every vector and matrix was first pre-multiplied by an appropriate rotation matrix so that all quantities would be in the global coordinate frame. Appendix A.4 provides a listing of the portion of a C program that was

developed to find these link parameters. A summary of the three-link parameters is shown in Table 2.2.

Table 2.2

LINK PARAMETERS

Link	Mass(Kg)	Length(m)	cg location(m)	Mom. Inertia (Kg-m^2)
1	5.91	.356	.232	.078
2	26.0	.594	.243	.340
3	25.6	.542	.209	.649

2.4.2 Lagrangian Inverse Dynamics

The Lagrangian formulation of the equations of motion was used to solve the inverse dynamics problem for the three-link model. The Lagrangian, L, is the difference between the kinetic coenergy, T*, and the potential energy, V, of the system, as shown in equation (2-33).

$$L = T^* - V \tag{2-33}$$

The differential equations of motion are obtained by differentiating the Lagrangian with respect to the generalized coordinates, Θ_1, Θ_2, Θ_3, and setting these equal to the generalized forces, Q_1, Q_2, Q_3, as shown in equation (2-34).

$$(d/dt)(dL/d\Theta'_i) - (dL/d\Theta_i) = Q_i \quad (i=1,2,3) \tag{2-34}$$

The Lagrangian dynamic equations of motion for the three-link model are shown in equations (2-35) - (2-37). A full derivation of these equations is included in Appendix B.

$$H_{11}\Theta_1'' + H_{12}\Theta_2'' + H_{13}\Theta_3'' + h_{122}\Theta_2'^2 + h_{133}\Theta_3'^2 + G_1 = Q_1 \quad (2\text{-}35)$$

$$H_{21}\Theta_1'' + H_{22}\Theta_2'' + H_{23}\Theta_3'' + h_{211}\Theta_1'^2 + h_{233}\Theta_3'^2 + G_2 = Q_2 \quad (2\text{-}36)$$

$$H_{31}\Theta_1'' + H_{32}\Theta_2'' + H_{33}\Theta_3'' + h_{311}\Theta_1'^2 + h_{322}\Theta_2'^2 + G_3 = Q_3 \quad (2\text{-}37)$$

where:

$$H_{11} = I_1 + M_1L_{c1}^2 + M_2L_1^2 + M_3L_1^2$$

$$H_{12} = (M_2L_1L_{c2} + M_3L_1L_2)\cos(\Theta_1+\Theta_2)$$

$$H_{13} = M_3L_1L_3\cos(\Theta_1-\Theta_3)$$

$$H_{21} = H_{12}$$

$$H_{22} = I_2 + M_2L_{c2}^2 + M_3L_2^2$$

$$H_{23} = M_3L_2L_{c3}\cos(\Theta_2+\Theta_3)$$

$$H_{31} = H_{13}$$

$$H_{32} = H_{23}$$

$$H_{33} = I_3 + M_3L_{c3}^2$$

$$h_{122} = -(M_2L_1L_{c2} + M_3L_1L_2)\sin(\Theta_1+\Theta_2)$$

$$h_{133} = -M_3L_1L_{c3}\sin(\Theta_1-\Theta_3)$$

$$h_{211} = h_{122}$$

$$h_{233} = -M_3L_2L_{c3}\sin(\Theta_2+\Theta_3)$$

$$h_{311} = h_{133}$$

$$h_{322} = h_{233}$$

$$G_1 = m_1gL_{c1}\sin\Theta_1 + m_2gL_1\sin\Theta_1 + m_3gL_1\sin\Theta_1$$

$$G_2 = m_2gL_{c2}\sin\Theta_2 + m_3gL_2\sin\Theta_2$$

$$G_3 = m_3gL_{c3}\sin\Theta_3$$

$Q_1 = -(\tau_1 + \tau_2)$

$Q_2 = -(\tau_2 + \tau_3)$

$Q_3 = -\tau_3$

τ_i = joint torque at joint i (actuators at joints)

$g = 9.8\ m/s^2$

The inverse dynamic equations above were implemented in a C program that calculated the joint torques as outputs given angular position, velocity and acceleration trajectories as inputs. A portion of this C program is listed in Appendix A.4. The input trajectories for angular position, velocity and acceleration at the three joints were obtained from angular position data for the ankle, knee and back joints from chair-rise trials performed by a human subject. This was the same subject whose body segment parameters were used to calculate the link parameters.

One factor that was not accounted for in the inverse dynamics equations of motion was the supporting force of the chair acting on the second link while in the seated position. To compensate for this, the torques at joints one and two were set equal to zero until liftoff from the chair was initiated. Once initiation of liftoff began, the joint torques were ramped up to their full values, so that as contact with the chair was lost, the torques went to 100% of their full values. The exact times when liftoff began and ended were obtained using forceplate data from the MGH Biomotion Laboratory. If this artificial condition were not imposed, then the initial joint torques would be equivalent to the joint torques required for a person to assume a seated position without a chair. Not only would this cause the joint torques to be very high, but it might not be quasi-statically possible for a person to assume the normal seated position without a chair. No artificial constraints were imposed on joint three.

Chapter 3

Results

In this chapter, the results of applying the dynamic estimator to the chair-rise, gait and lifting protocols described in section 2.3 are presented followed by the results of the three-link simulations of chair-rise described in section 2.4 and finally the results of the MGH model validation. To quantify the role of segmental dynamics in these activities, results of the calculations described in section 2.3 are presented. Qualitative results are provided in the form of characteristic force, torque and momentum curves.

All net joint reaction forces and torques presented in this thesis are referenced to the coordinate frame of the appropriate body segment. The convention is as follows: ankle reaction in foot coordinates, knee reaction in shank coordinates, hip reaction in thigh coordinates, back reaction in trunk coordinates, neck reaction in head coordinates, shoulder reaction in arm coordinates.

Three components of force and three components of torque were calculated for each of the ten joints of the MGH human body model. The direction conventions used for forces and torques are explained here. The axial component of force is in a direction parallel to the long axis of the body segment, with compression being positive. The lateral force component is perpendicular to the sagittal plane of the body segment, with positive forces acting to move the segment distal to the joint in the lateral direction, or leftward for the back and neck. The anterior-posterior component of force is perpendicular to the other two force components, with positive forces acting to move the segment distal to the joint in the posterior direction.

The flexion-extension torque represents the component of torque which acts about an axis perpendicular to the sagittal plane of the body segment, with positive torques acting to extend joints. The abduction-adduction torque component acts about an axis in the anterior-posterior direction, with positive torques causing abduction of the segment distal to the joint, and counterclockwise rotation about a posterior pointing axis for the back and neck. The external-internal rotation torque is the component of torque which acts about an axis parallel to the long axis of the body segment, with positive torques causing internal rotation of the segment distal to the joint, and counterclockwise rotation about an upward (distal pointing) axis for the back and neck. The force and torque sign conventions are given in Table 3.1.

TABLE 3.1

SIGN CONVENTION FOR POSITIVE JOINT FORCES AND TORQUES

	SAGITTAL PLANE	FRONTAL PLANE	TRANSVERSE PLANE
JOINT	+ Force, + Torque	+ Force, + Torque	+ Force, + Torque
Ankle	posterior, plantarflexor	lateral, valgus	compression, internal
Knee	posterior, extension	lateral, valgus	compression, internal
Hip	posterior, extension	lateral, abduction	compression, internal
Back	posterior, extension	left, ccw bend	compression, ccw twist
Neck	posterior, extension	left, ccw bend	compression, ccw twist
Shoulder	posterior, extension	lateral, abduction	compression, internal

Three components of linear and angular momentum were calculated for each of the 11 segments of the MGH human body model. The direction conventions used for linear and angular momentum are explained here. The linear momentum components are given in

global coordinates. Linear momentum in the posterior direction (+x global axis), the upward direction (+y global axis), and the leftward direction (+z global axis) is positive.

The components of angular momentum are also referenced to global coordinate frames placed at the appropriate segment origin. The knee and shank origins were referenced to the ankles, the thighs to the knees, the pelvis to the hips, the back to the back joint, the head to the neck, and the arms to the shoulders. The angular momentum component about the anterior-posterior axis is positive when the segment rotates counterclockwise about the posterior direction axis (+x global axis). The angular momentum component about the vertical axis is positive during counterclockwise rotation of the segment about the upward vertical axis (+y global axis). The angular momentum component about the lateral-medial axis is positive during counterclockwise rotation of the segment about the leftward pointing axis (+z global axis). These linear and angular momentum sign conventions are listed in Table 3.2.

TABLE 3.2

SIGN CONVENTION FOR POSITIVE LINEAR AND ANGULAR MOMENTUM

Momentum	SAGITTAL PLANE	FRONTAL PLANE	TRANSVERSE PLANE
Linear	posterior direction	leftward direction	upward direction
Angular	counterclockwise about an axis in the leftward direction	counterclockwise about an axis in the posterior direction	counterclockwise about an axis in the upward direction

3.1 Chair-Rise

Representative curves and tabulated average magnitudes for the three directional components of force and torque acting on the ten joints of the MGH model during chair-

rise are presented first. Next, the dynamic components of force and torque are reported as percentages of the total force and torque for the averages of the 20 chair-rise trials, followed by preliminary results on how variations in speed affected the dynamic percentages of force and torque in one subject. To end the section, curves for the three directional components of linear and angular momentum are given for the 11 body segments and for the whole body during a representative chair-rise trial.

3.1.1 Representative Force and Torque Curves

Due to the number of plots generated by the dynamic estimator, it was not feasible to include all the force and torque curves for all 20 chair-rise trials. The representative chair-rise force and torque curves given in this section are for a normal female subject (age: 30 years, height: 168 cm, weight: 59 kg). Table 3.3 lists the average peak magnitudes for the three components of force, and Table 3.4 lists the average peak magnitudes for the three components of torque. For the flexion-extension torques, the axial forces, and the anterior-posterior forces, the peak values were fairly consistent across subjects, with the average peak values being the same as the average peak magnitudes. The other components of force and torque tended to be smaller and more variable. For this reason, the magnitudes were averaged rather than the actual values. As mentioned previously, the net joint forces and torques are presented in this study, which do not include the effects of muscle contraction or co-contraction and represent the lower limits of the force and torque reactions which actually occur within the joint.

The representative curves of Figure 3.1 show the three components of force. Figure 3.1.a shows the right ankle, knee and hip force curves. Figure 3.1.b shows the left ankle, knee and hip force curves. Figure 3.1.c shows the back and neck force curves. Figure 3.1.d shows the torque curves for the left and right shoulder joints.

The representative curves of Figure 3.2 show the three components of torque. Figure 3.2.a shows the right ankle, knee and hip torque curves. Figure 3.2.b shows the left ankle, knee and hip torque curves. Figure 3.2.c shows the back and neck torque curves. Figure 3.2.d shows the torque curves for the left and right shoulder joints.

TABLE 3.3

AVERAGE MAGNITUDES FOR PEAK FORCES DURING CHAIR-RISE (%BW)

(mean and standard deviation, n=20)

JOINT	Anterior-Posterior	Axial	Medial-Lateral
Ankle	10.4 (4.46)	54.6 (4.71)	8.40 (3.69)
Knee	24.9 (5.80)	45.6 (4.86)	9.17 (5.21)
Hip	29.9 (4.05)	31.8 (6.44)	10.4 (6.63)
Back	18.5 (3.67)	48.5 (5.96)	2.75 (1.33)
Neck	1.63 (0.77)	7.25 (1.25)	1.43 (1.08)

TABLE 3.4

AVERAGE MAGNITUDES FOR PEAK TORQUES DURING CHAIR-RISE (%BW*M)

(mean and standard deviation, n=20)

JOINT	Flexion-Extension	Abd.-Adduction	Int.-Ext. Rotation
Ankle	3.62 (1.77)	1.28 (0.64)	0.87 (0.50)
Knee	9.09 (2.01)	3.23 (1.88)	1.03 (0.48)
Hip	7.34 (2.65)	3.75 (1.52)	2.48 (1.53)
Back	9.22 (2.13)	0.66 (0.21)	0.40 (0.29)
Neck	0.19 (0.06)	0.04 (0.03)	0.04 (0.04)

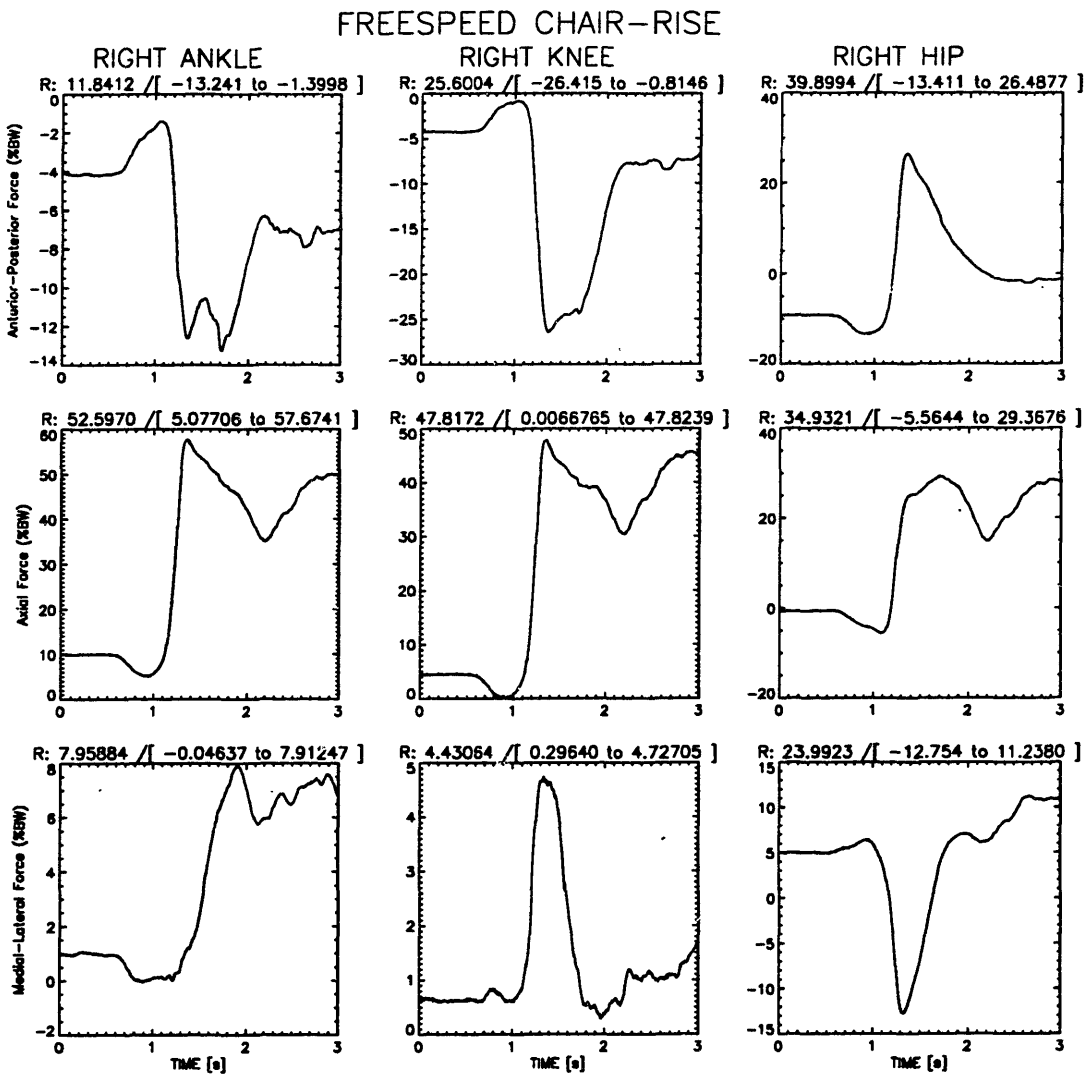

Figure 3.1.a Representative force curves for the right ankle, knee and hip during chair-rise.

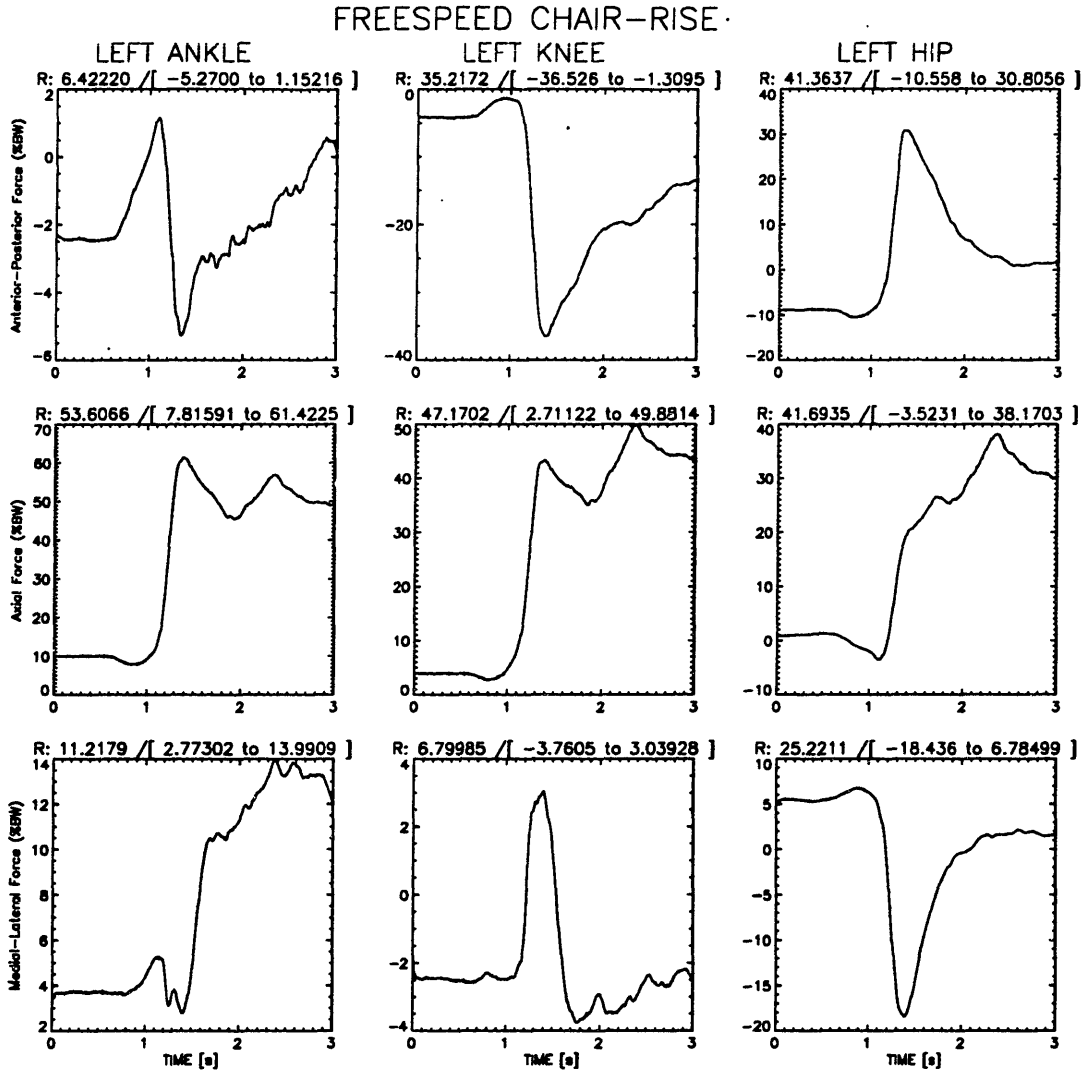

Figure 3.1.b Representative force curves for the left ankle, knee and hip during chair-rise.

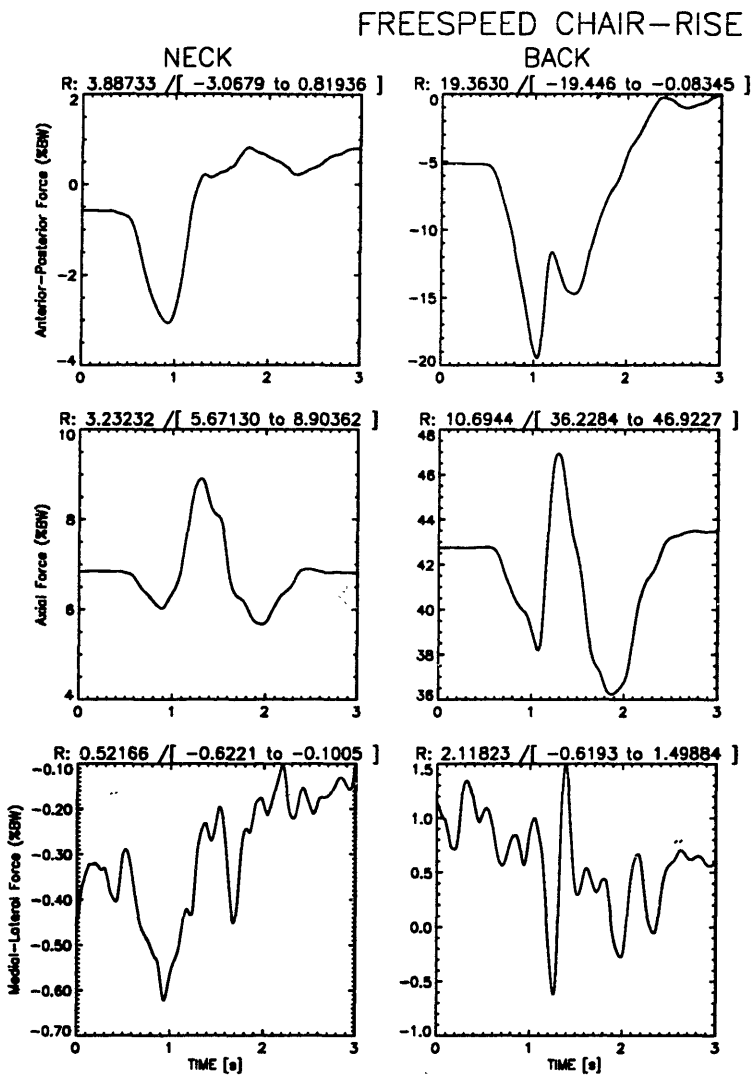

Figure 3.1.c Representative force curves for the back and neck joints during chair-rise.

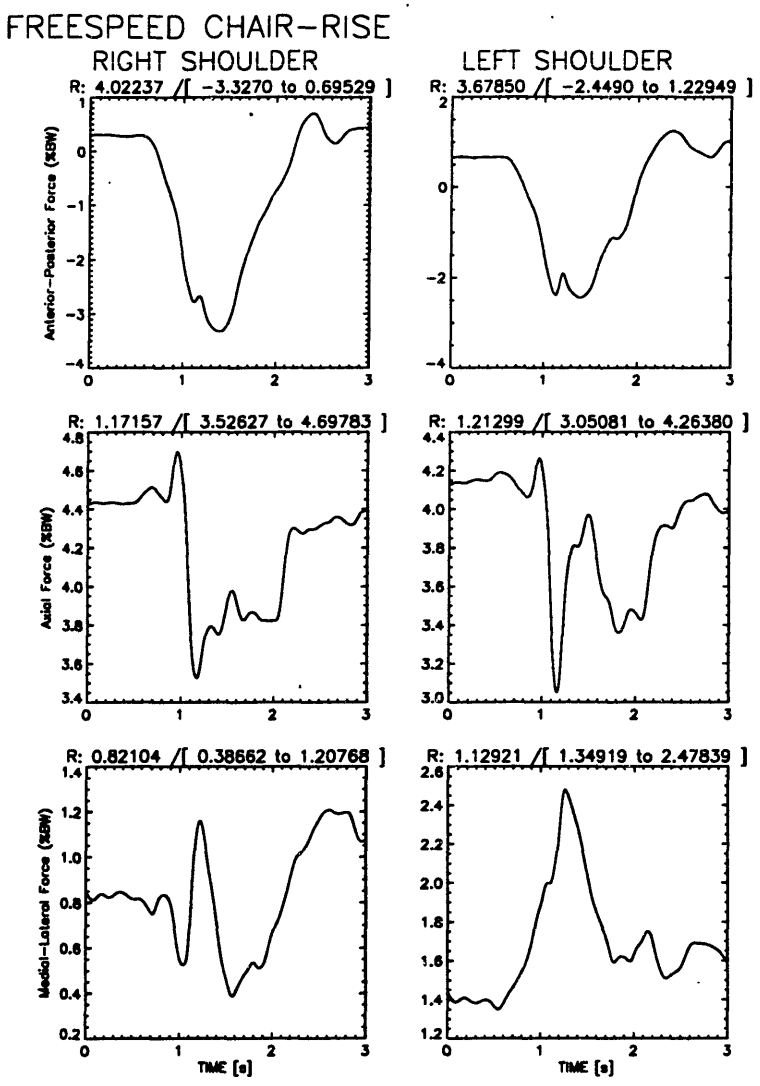

Figure 3.1.d Representative force curves for the right and left shoulders during chair-rise.

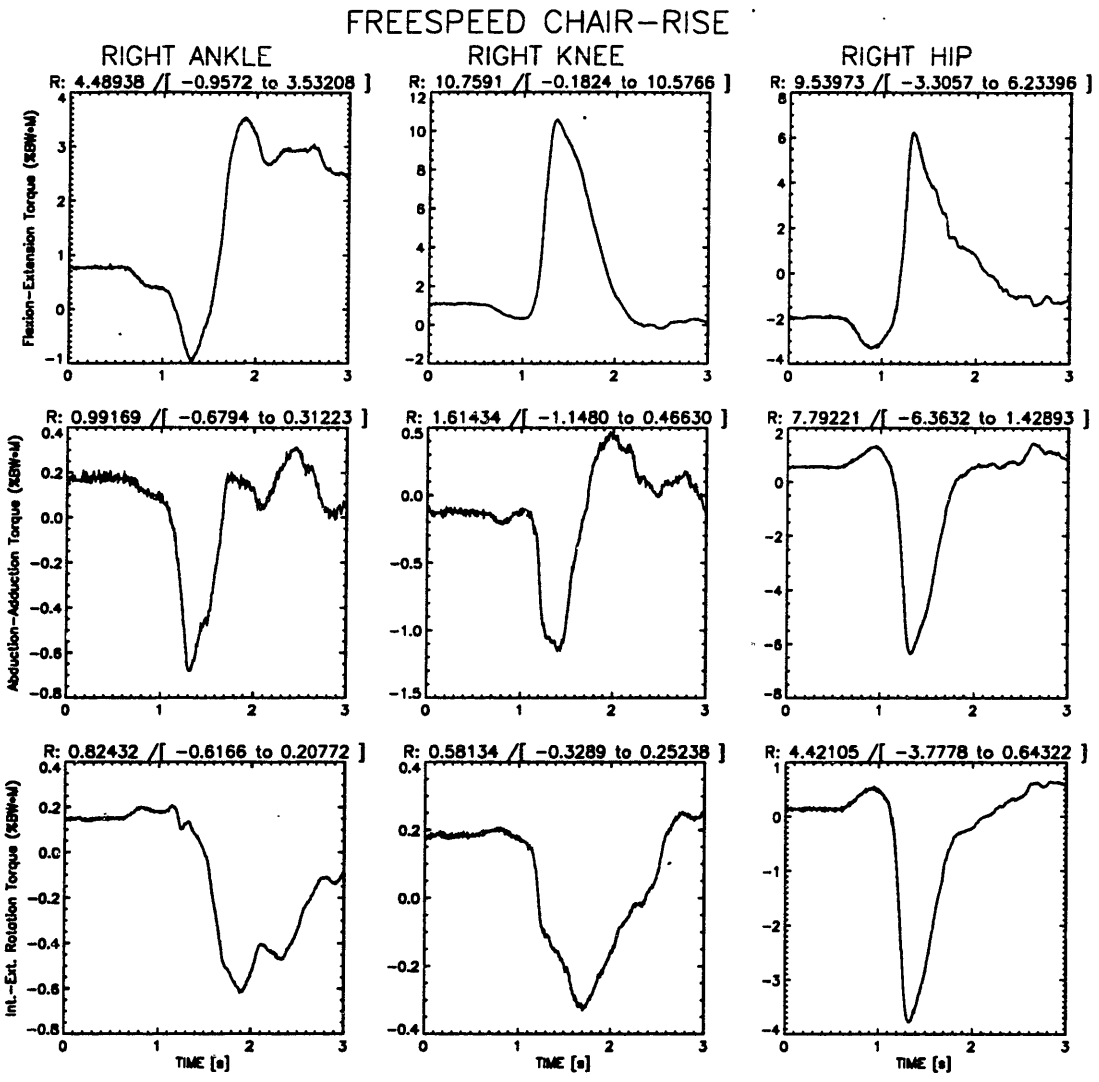

Figure 3.2.a Representative torque curves for the right ankle, knee and hip during chair-rise.

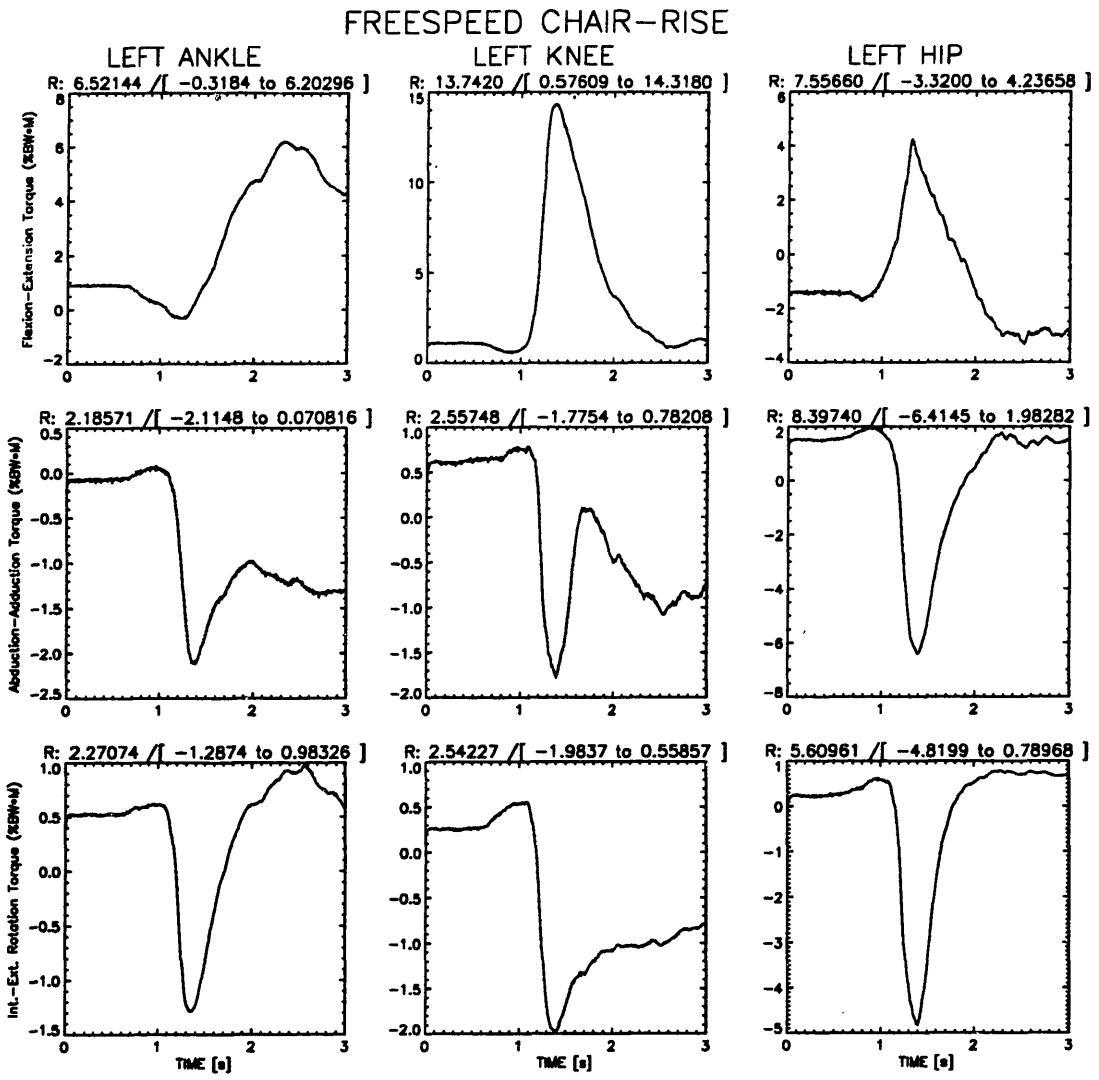

Figure 3.2.b Representative torque curves for the left ankle, knee and hip during chair-rise.

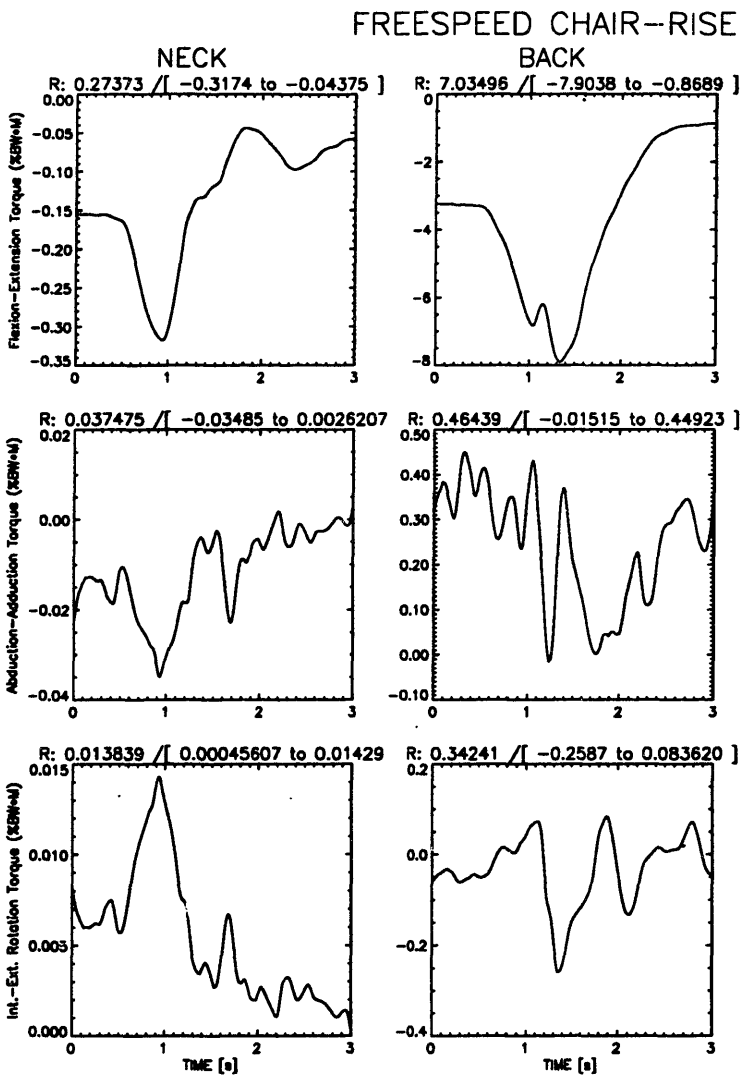

Figure 3.2.c Representative torque curves for the back and neck joints during chair-rise.

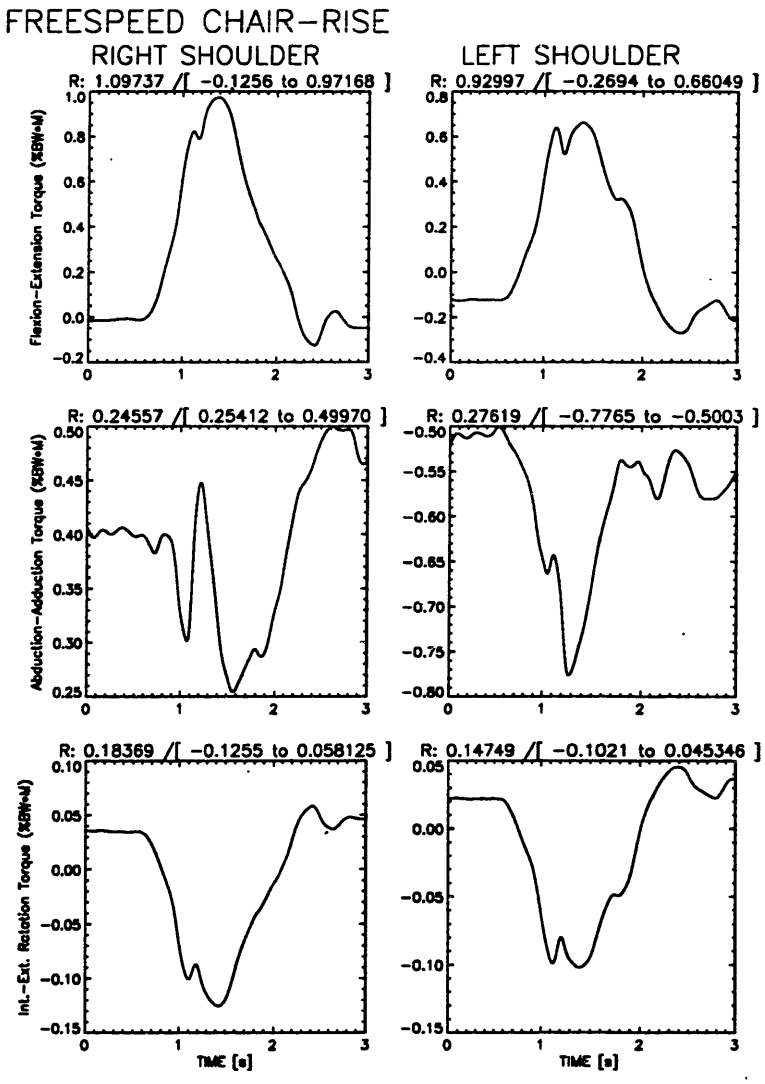

Figure 3.2.d Representative torque curves for the right and left shoulders during chair-rise.

3.1.2 Dynamic Components of Force and Torque

Table 3.5 lists the mean dynamic force and torque as percentages of the total force and torque and the standard deviation for the 20 chair-rise trials. The formulas used to calculate these dynamic percentages are given by equations (2-13) and (2-14) in section 2.3.1. The dynamic percentages were averaged for the left and right lower extremity joints.

TABLE 3.5

DYNAMIC FORCE AND TORQUE AS A PERCENTAGE OF THE TOTAL FORCE AND TORQUE DURING CHAIR-RISE (mean and standard deviation) (n=20)

	Ankle	Knee	Hip	Back	Neck
Dynamic force Percentage	.03% (.01)	.82% (.27)	5.43% (1.57)	17.8% (4.75)	37.7% (9.74)
Dynamic torque Percentage	.04% (.03)	.83% (.49)	7.12% (2.76)	18.1% (3.71)	**

**Dynamic torques on the neck were not reported.

Figures 3.3.a and 3.3.b show the information of Table 3.5 graphically. In Figure 3.3.a, the mean ankle, knee, hip, back, and neck dynamic force components are plotted for the twenty trials. In Figure 3.3.b, the mean ankle, knee, hip, and back dynamic torque components are plotted for the twenty trials.

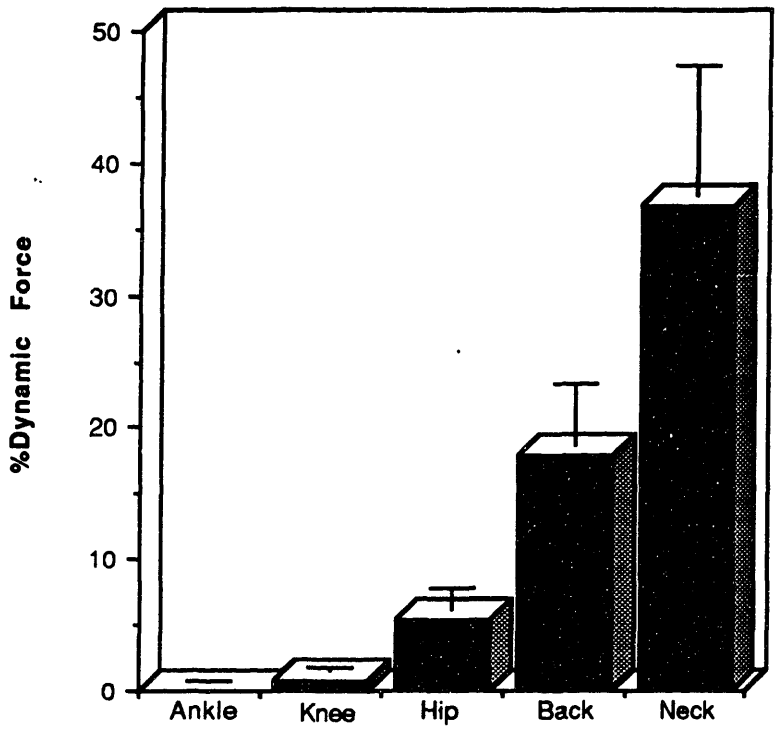

Figure 3.3.a The joint force percentage due to dynamic components. (mean % + std.dev.).

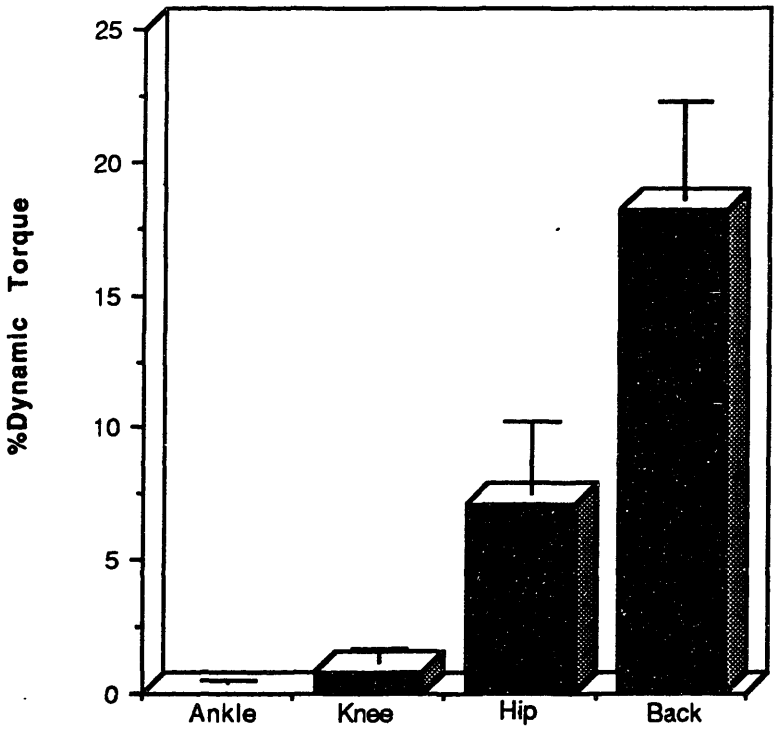

Figure 3.3.b The joint torque percentage due to dynamic components.(mean %+ std.dev.).

3.1.3 Dynamics and the Speed of Ascent

The effect of increasing the speed of ascent on the dynamic components of force and torque is demonstrated in Table 3.6, which lists dynamic force and torque percentages for three slow chair-rise trials (46 beats per minute) and three fast chair-rise trials (92 beats per minute).

TABLE 3.6

DYNAMIC FORCE AND TORQUE AS A PERCENTAGE OF THE TOTAL FORCE AND TORQUE DURING CHAIR-RISE (mean and standard deviation) (n=3)

	Ankle	Knee	Hip	Back	Neck
Dynamic force percentage:					
Slow	.03% (.01)	1.20% (.21)	6.96% (1.28)	23.2% (.26)	45.8% (1.96)
Fast	.05% (.02)	1.96% (.42)	13.3% (1.57)	30.7% (1.76)	61.8% (3.73)
Dynamic torque percentage:					
Slow	.06% (.02)	.92% (.46)	7.59% (1.57)	16.5% (.39)	**
Fast	.09% (.03)	1.33% (.67)	11.6% (1.15)	17.6% (1.81)	**

**Dynamic torques on the neck were not reported.

Figures 3.4.a and 3.4.b show the information of Table 3.6 graphically. In Figure 3.4.a, the mean ankle, knee, hip, back, and neck dynamic force components are plotted for three slow trials and three fast trials. In Figure 3.4.b, the mean ankle, knee, hip, and back dynamic torque components are plotted for three slow trials and three fast trials.

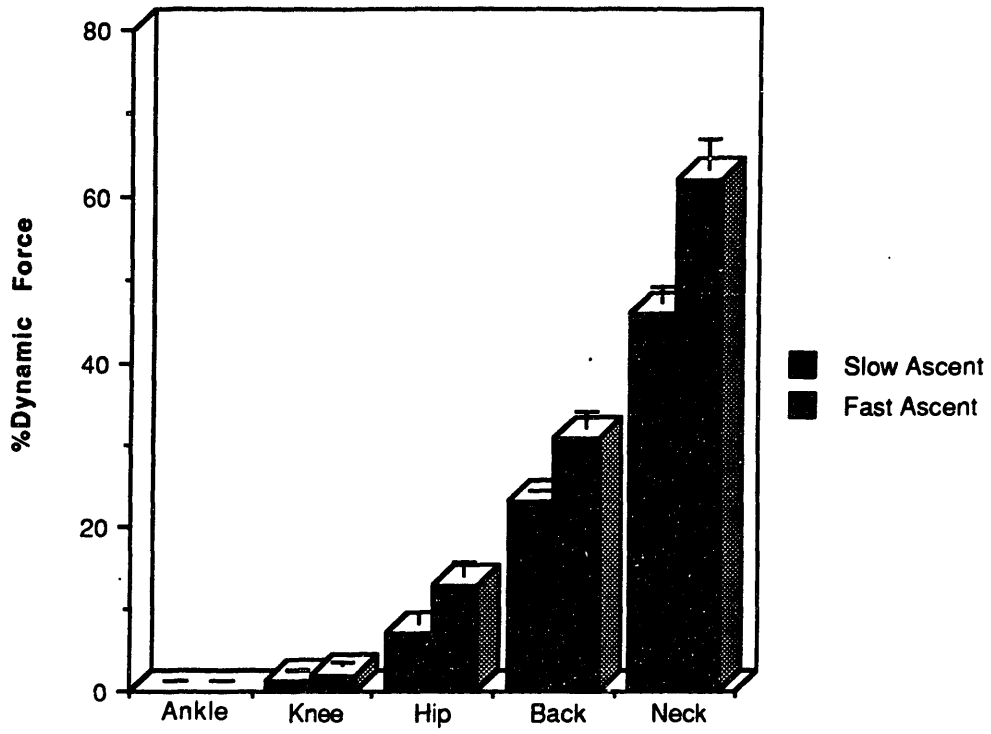

Figure 3.4.a The joint force percentage due to dynamic components. (mean % + std.dev.).

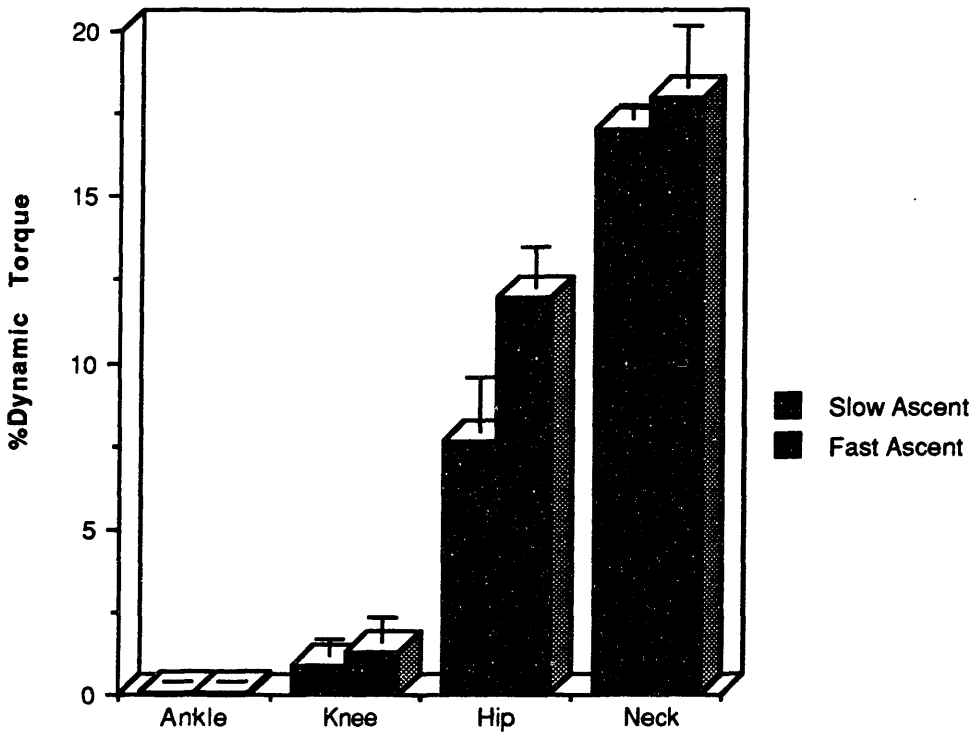

Figure 3.4.b The joint torque percentage due to dynamic components.(mean %+ std.dev.).

3.1.4 Representative Momentum Curves

As was the case with the force and torque curves, the number of plots generated by the dynamic estimator made the inclusion of all the linear and angular momentum curves for all 20 chair-rise trials impractical. The representative chair-rise linear and angular momentum curves given in this section are for a normal male subject (age: 24 years, height: 175 cm, weight: 73 kg).

The representative linear momentum curves are given in Figure 3.5. Figure 3.5.a shows the three components of linear momentum for the right foot, shank and thigh. Figure 3.5.b shows the three components of linear momentum for the left foot, shank and thigh. Figure 3.5.c shows the three components of linear momentum for the head, trunk and pelvis. Figure 3.5.d shows the three components of linear momentum for the left and right arms.

The representative angular momentum curves are given in Figure 3.6. Figure 3.6.a shows the three components of angular momentum for the right foot, shank and thigh. Figure 3.6.b shows the three components of angular momentum for the left foot, shank and thigh. Figure 3.6.c shows the three components of angular momentum for the head, trunk and neck. Figure 3.6.d shows the three components of angular momentum for the left and right arms.

Figure 3.7 shows representative curves for the whole body linear momentum, the whole body angular momentum about the averaged ankle joint and the whole body angular momentum about the body center of mass.

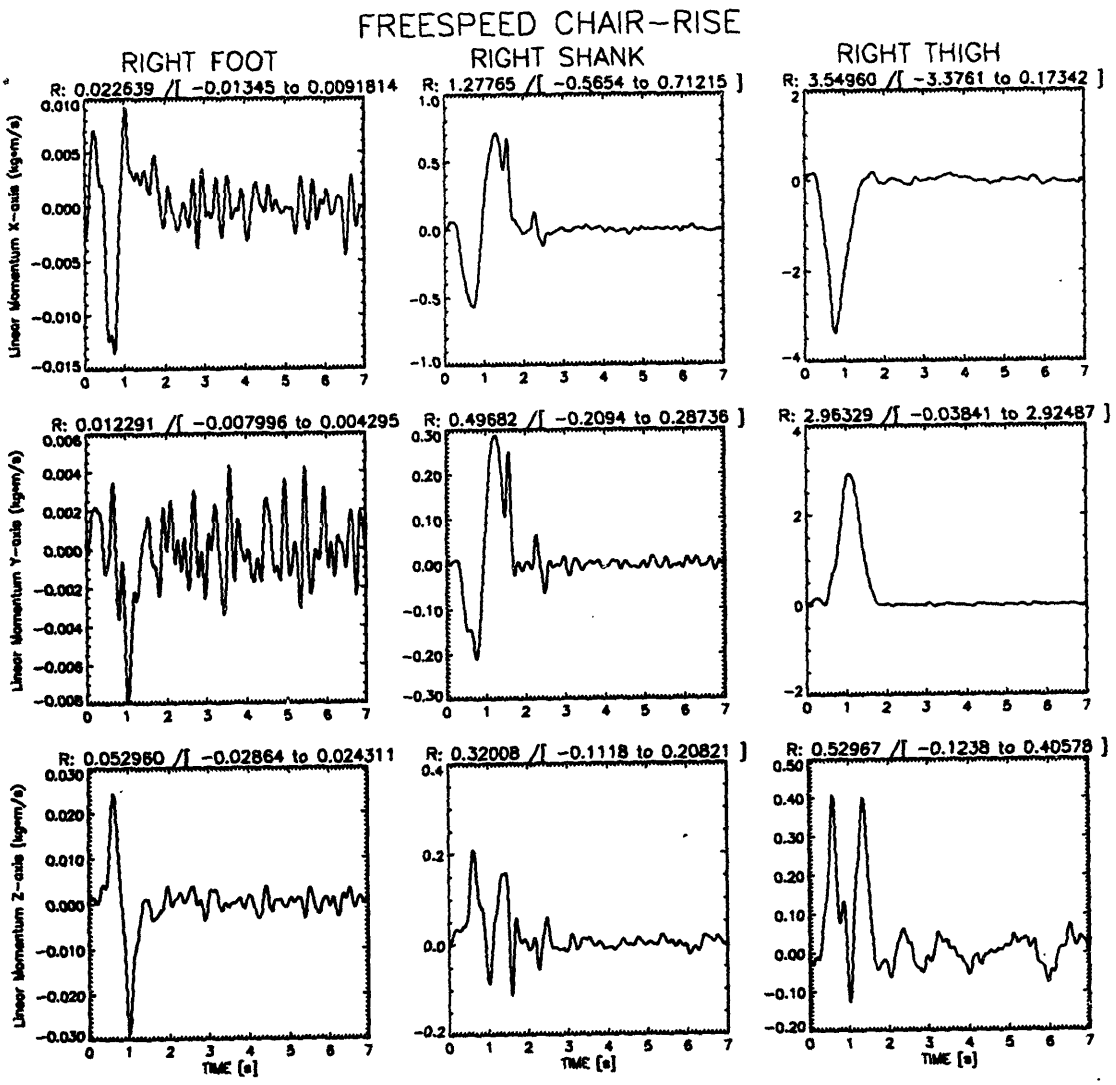

Figure 3.5.a Representative chair-rise linear momentum curves for the right foot, shank and thigh.

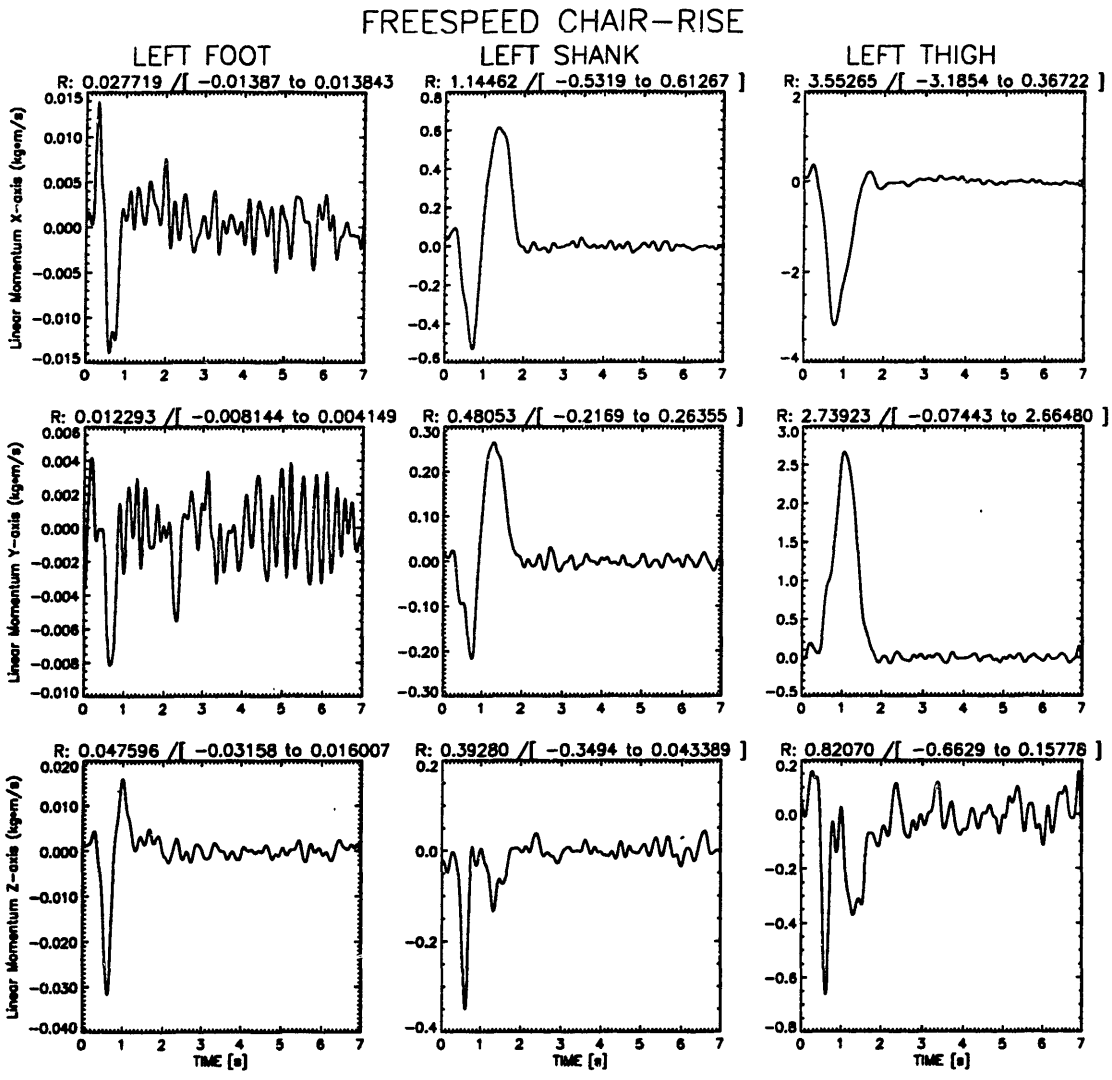

Figure 3.5.b Representative chair-rise linear momentum curves for the left foot, shank and thigh.

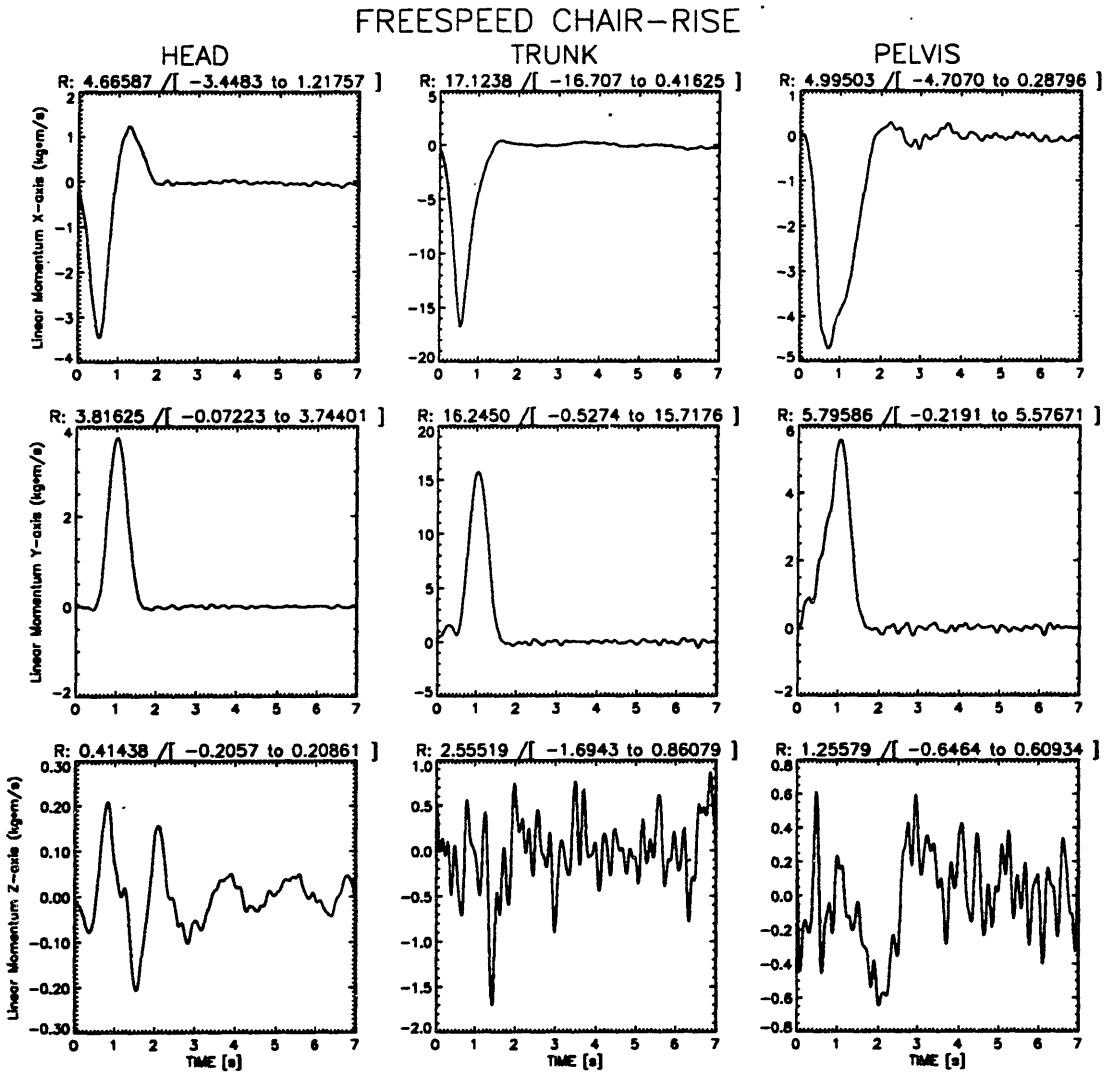

Figure 3.5.c Representative chair-rise linear momentum curves for the head, trunk and pelvis.

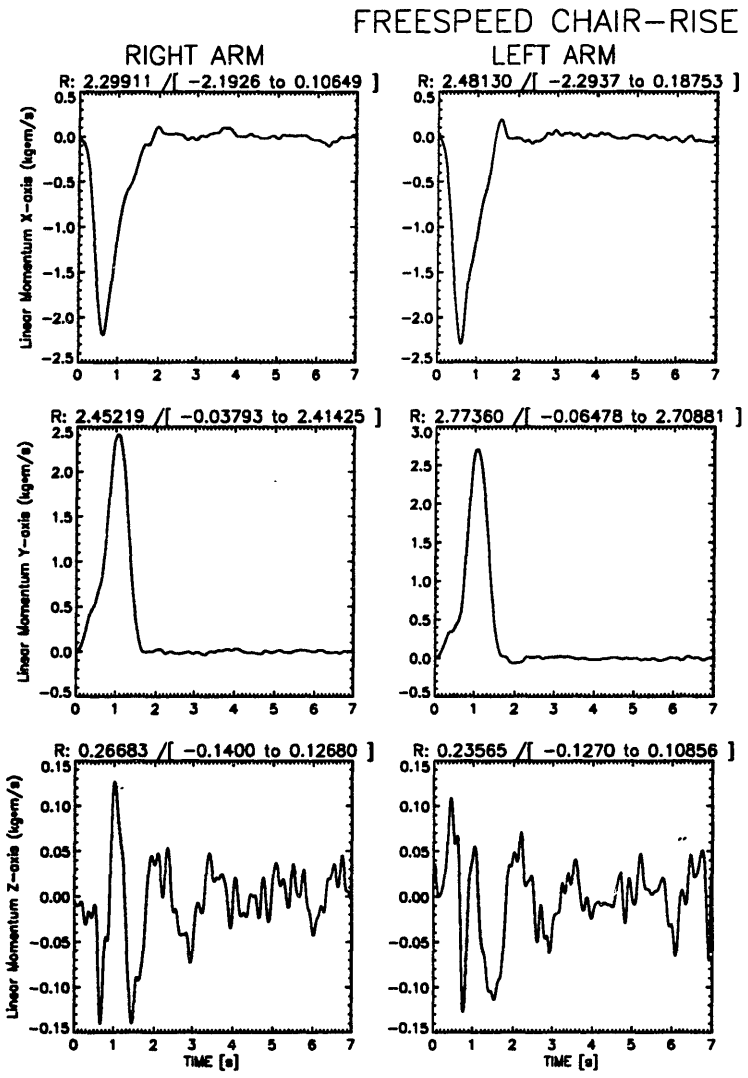

Figure 3.5.d Representative chair-rise linear momentum curves for the right and left arms.

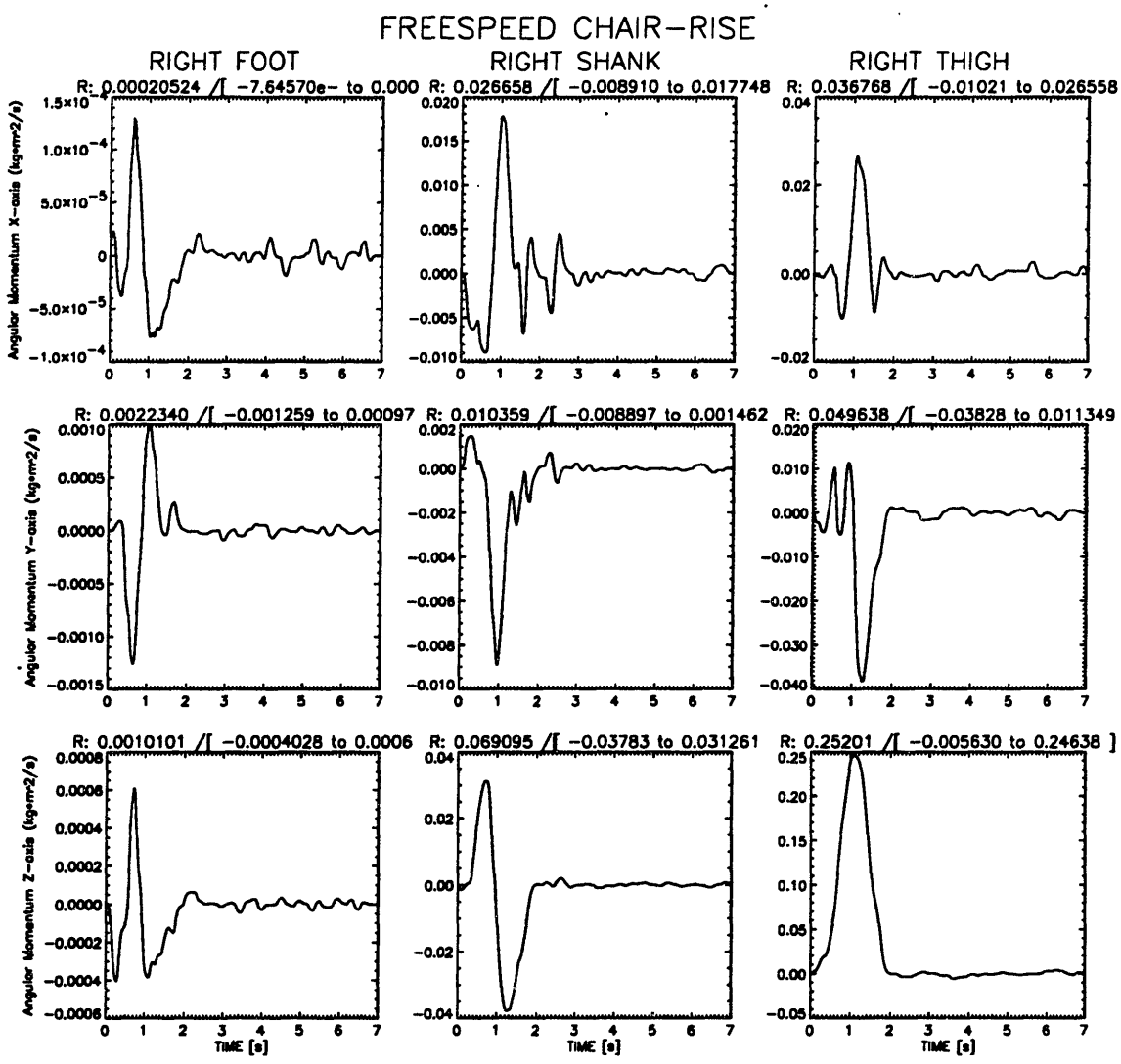

Figure 3.6.a Representative chair-rise angular momentum curves for the right foot, shank and thigh.

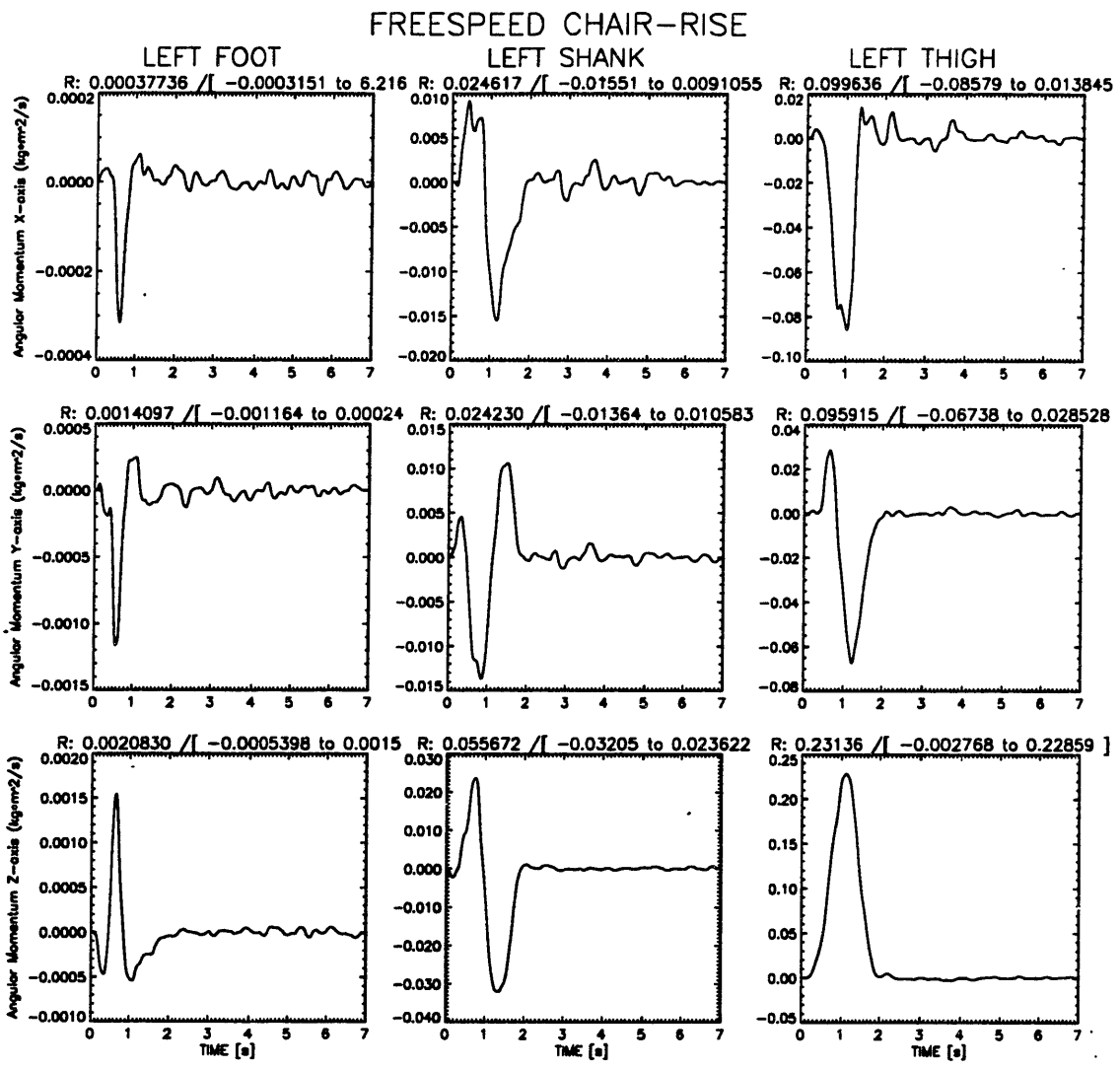

Figure 3.6.b Representative chair-rise angular momentum curves for the left foot, shank and thigh.

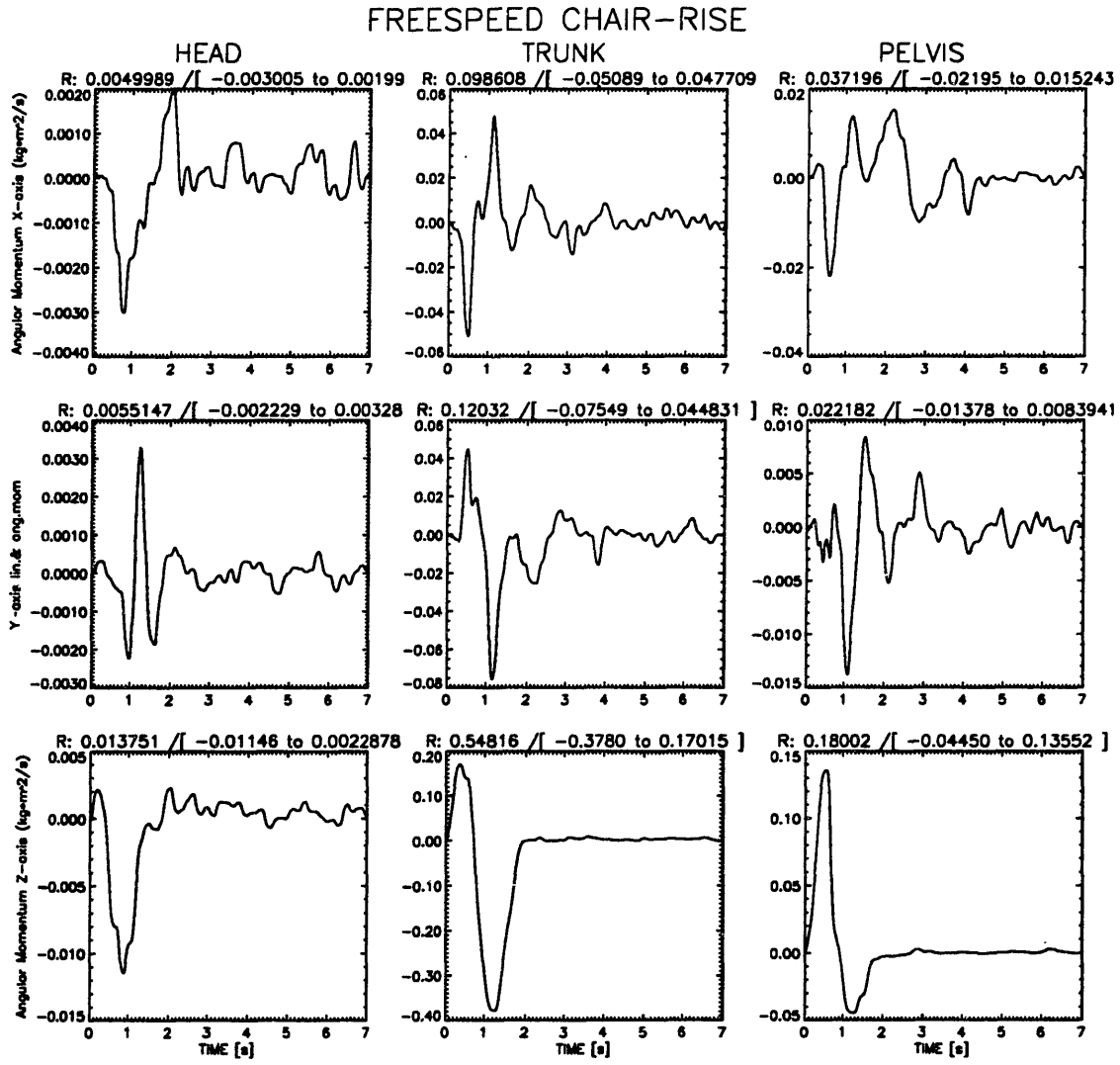

Figure 3.6.c Representative chair-rise angular momentum curves for the head, trunk and pelvis.

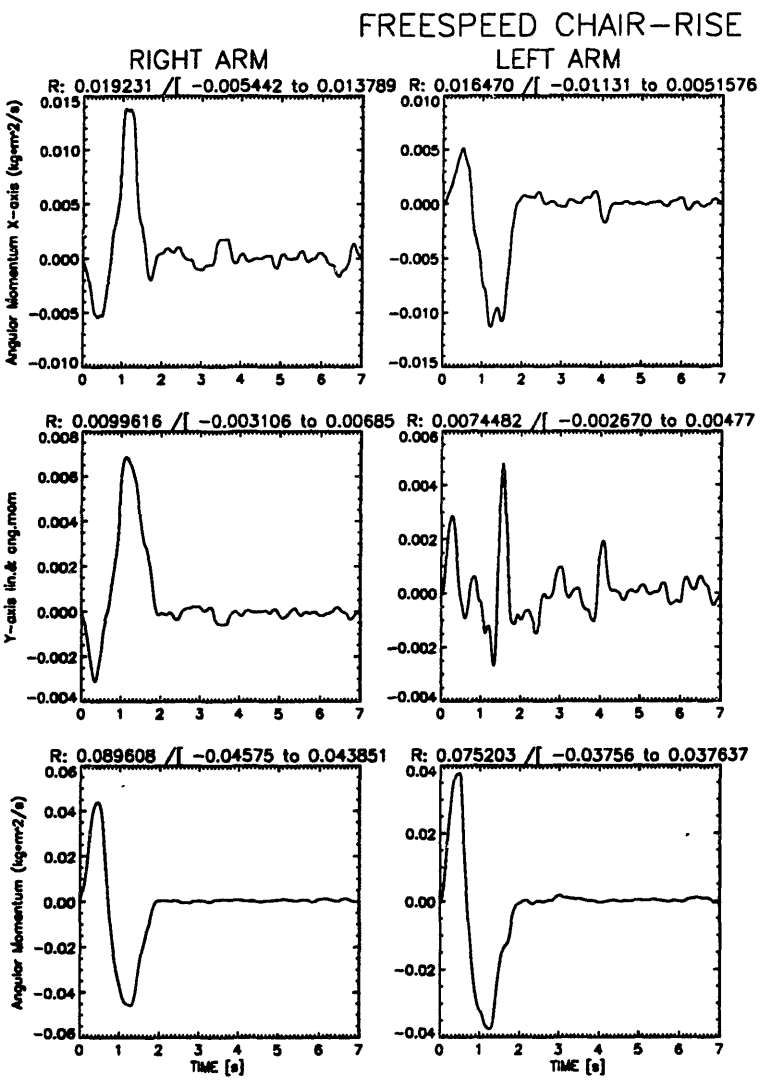

Figure 3.6.d Representative chair-rise angular momentum curves for the right and left arms.

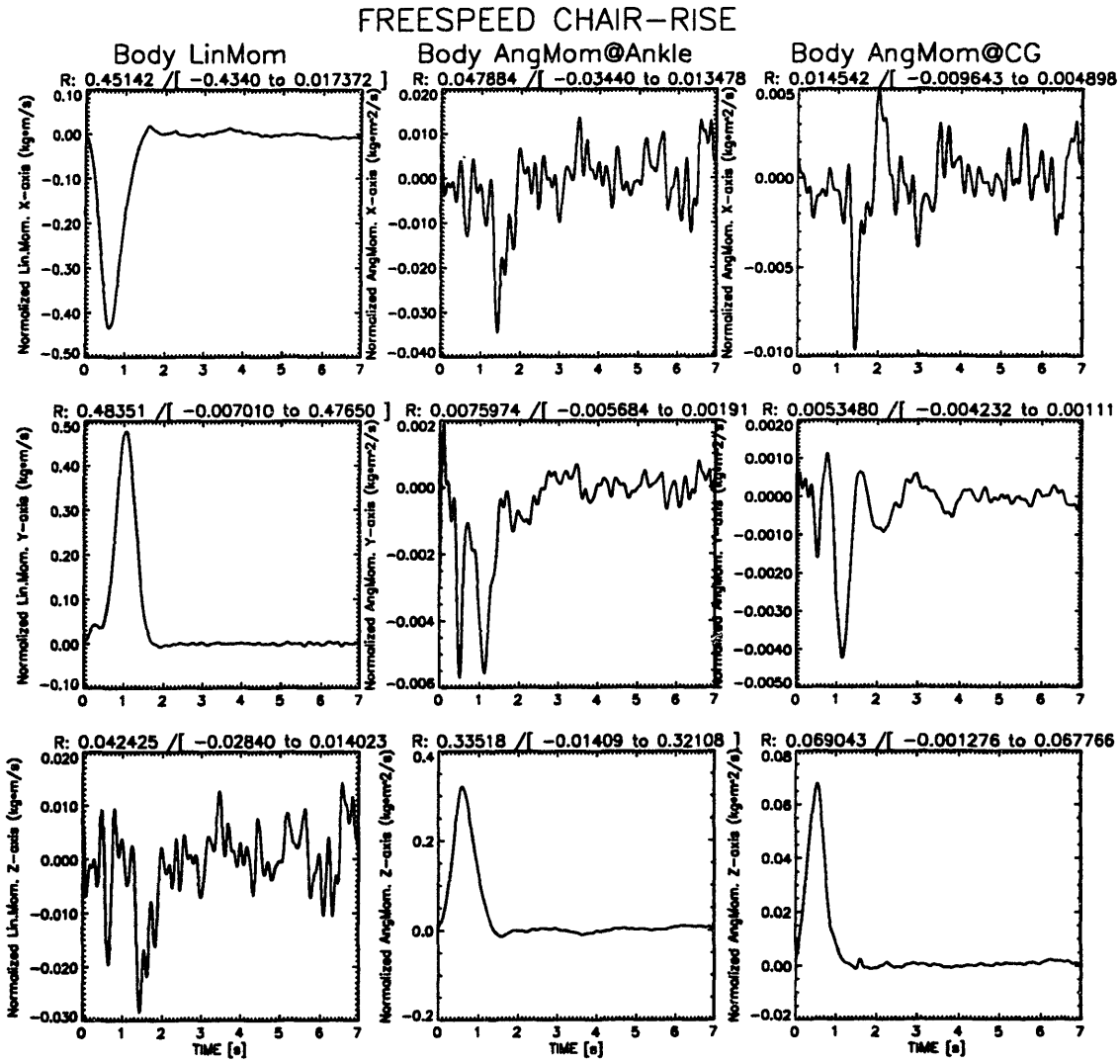

Figure 3.7 Representative chair-rise whole body linear momentum, angular momentum about the average ankle joint and angular momentum about the body center of mass.

3.1.5 Whole Body Momentum

As shown in Figure 3.7, the whole body linear momentum and whole body angular momentum about the average ankle joint and the center of mass were calculated during chair-rise. The peak values for each of these three types of whole body momentum were calculated and averaged for ten freespeed chair-rise trials by five normal subjects. The group consisted of four males and one female and had an average age of 28 (range 24-36) years, height of 176 (range 141-184) cm and weight of 68 (range 55-73) kg. The normalized peak whole body momentum values are listed below in Table 3.7.

TABLE 3.7

NORMALIZED PEAK WHOLE BODY MOMENTUM DURING CHAIR-RISE

(mean and standard deviation, n=10)

	X-axis(AP)	Y-axis(Vertical)	Z-axis(Lateral)
Linear Momentum (m/s)/kg	-.397 (.041)	.524 (.092)	-.079 (.049)
Angular Momentum @ ankle (m^2/s)	-.060 (.036)	-.009 (.014)	.305 (.023)
Angular Momentum @ CG (m^2/s)	-.013 (.012)	-.004 (.004)	.072 (.010)

3.2 The Swing and Stance Phases of Gait

Because the focus of the gait experiments in this thesis was limited to a comparison of swing and stance phase dynamics, only the lower extremity forces and torques will be presented. All three of the directional components of force and torque acting on the ankle, knee and hip joints were calculated during the swing and stance phases of gait. However, only the axial forces and the flexion-extension torques were included, because they were the largest and most consistent between subjects of the three components of force and torque. Hence the axial forces and flexion-extension torques were the most useful in investigating the significance of the dynamic contributions to force and torque. Representative axial force and flexion-extension torque curves are presented for the swing and stance phases of gait. Next, the dynamic components of force and torque are reported as percentages of the total force and torque for the averages of the seven paced gait trials. Following the dynamic percentage results, curves for the three directional components of linear and angular momentum are given for the foot, shank and thigh segments during a representative gait trial. Representative whole body linear momentum and angular momentum about the center of mass curves are also presented, however, the whole body momentum includes both the swing leg and the stance leg, therefore these curves span a cycle that includes the swing phase of one leg and the stance phase of the other leg.

3.2.1 Representative Force and Torque Curves

Fewer force and torque plots were generated for the swing and stance phases of gait than for chair-rise. This is because upper body kinetics were not used for the swing and stance phase comparison, and fewer gait trials were analyzed than chair-rise trials. However, there were still a large number of plots and by presenting force and torque curves from one

representative trial, repetition was avoided. The representative swing and stance phase force curves given in this section are from a normal male subject (age: 29 years, height: 178 cm, weight: 85 kg), and the torque curves are from a normal female subject (age: 30 years, height: 168 cm, weight: 59 kg). As with chair-rise, the net joint forces and torques are presented in this study, which do not include the effects of muscle co-contraction and represent the lower limits of the force and torque reactions which actually occur within the joint.

The representative force curves given in Figure 3.8 show the ankle, knee and hip axial force as calculated with both a quasi-static method and a dynamic method. Figure 3.8.a shows the quasi-static and dynamic axial forces during swing phase, and Figure 3.8.b shows the quasi-static and dynamic axial forces during stance phase.

Likewise, the representative torque curves given in Figure 3.9 show the ankle, knee and hip flexion-extension torque as calculated with both a quasi-static method and a dynamic method. Figure 3.9.a shows the quasi-static and dynamic flexion-extension torques during swing phase, and Figure 3.9.b shows the quasi-static and dynamic flexion-extension torques during stance phase.

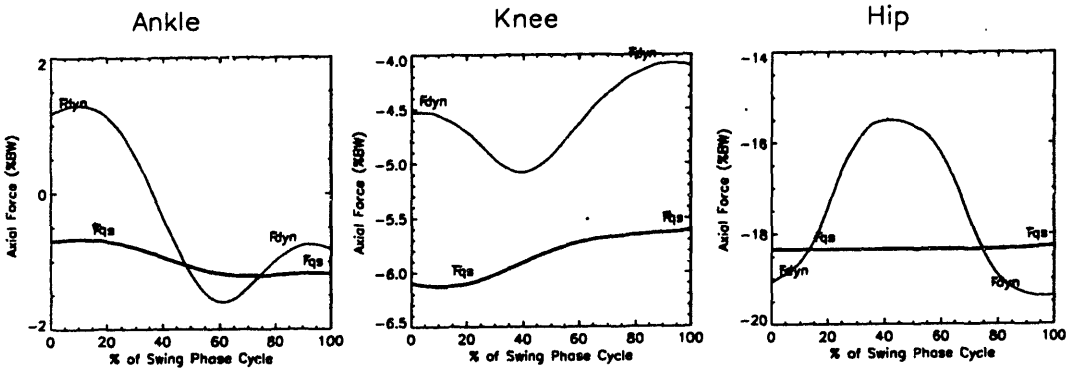

Figure 3.8.a Representative swing phase axial forces calculated with a quasi-static method and a dynamic method.

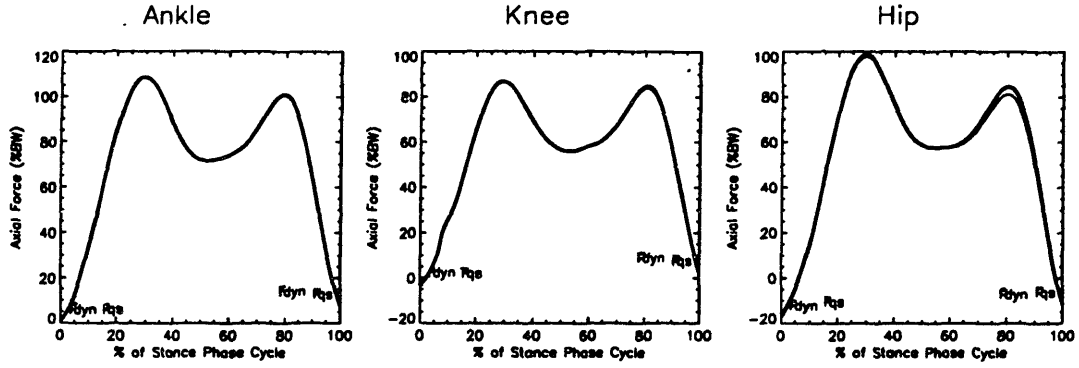

Figure 3.8.b Representative stance phase axial forces calculated with a quasi-static method and a dynamic method.

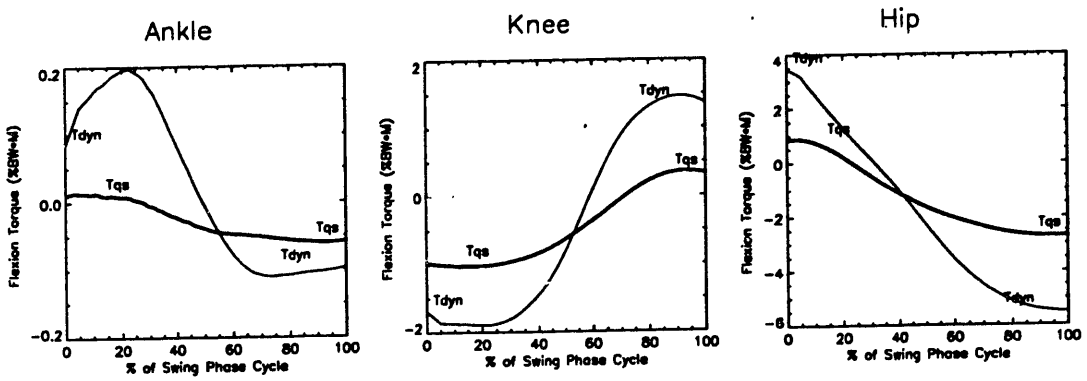

Figure 3.9.a Representative swing phase flexion-extension torques calculated with a quasi-static method and a dynamic method.

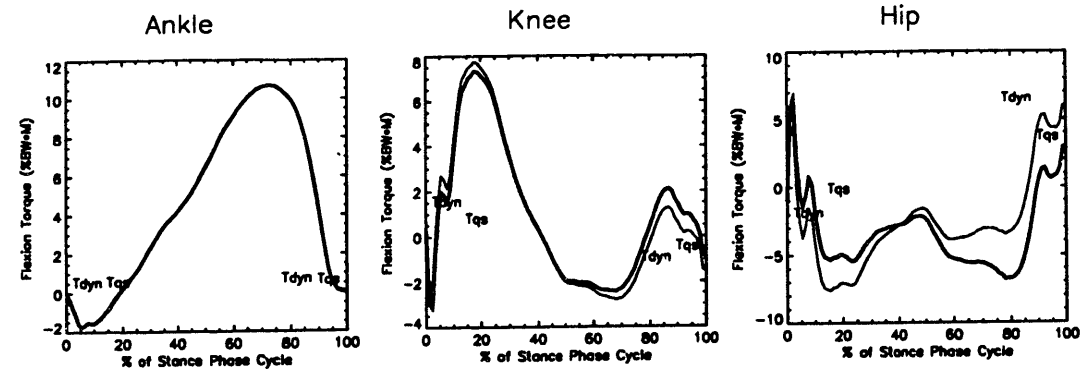

Figure 3.9.b Representative stance phase flexion-extension torques calculated with a quasi-static method and a dynamic method.

3.2.2 Dynamic Contributions to Swing and Stance Phase Joint Forces and Torques.

Table 3.8 lists the peak axial forces calculated with the dynamic estimator and the quasi-static method. These differences at the peaks were averaged for the seven gait trials. The dynamic force equation is given in equation (2-9), and the quasi-static force equation is given in equation (2-17). Normalization was performed by dividing the peak difference by the range of the axial force, as calculated with the dynamic estimator.

TABLE 3.8

PEAK AXIAL FORCE DIFFERENCES RESULTING FROM USING DYNAMIC VS. QUASI-STATIC JOINT FORCE ESTIMATES DURING GAIT (n=7)

	Ankle		Knee		Hip	
	Swing	Stance	Swing	Stance	Swing	Stance
Peak q.s. force (%BW)	-1.25	114.9	-6.27	104.6	-19.7	98.8
Peak dyn. force (%BW)	-1.66	114.7	-5.54	104.7	-21.8	98.5
Peak difference (%BW)	-.414	-.160	.739	.100	-2.12	-.254
Normalized peak difference	-.125	-.002	.462	.001	-.503	-.002

Figures 3.10 and 3.11 show the information of Table 3.8 graphically. Figure 3.10 gives the peak axial forces for the ankle, knee and hip as calculated with both the dynamic and quasi-static methods during swing phase and stance phase. In Figure 3.11, the normalized peak axial force differences are given during swing phase and stance phase.

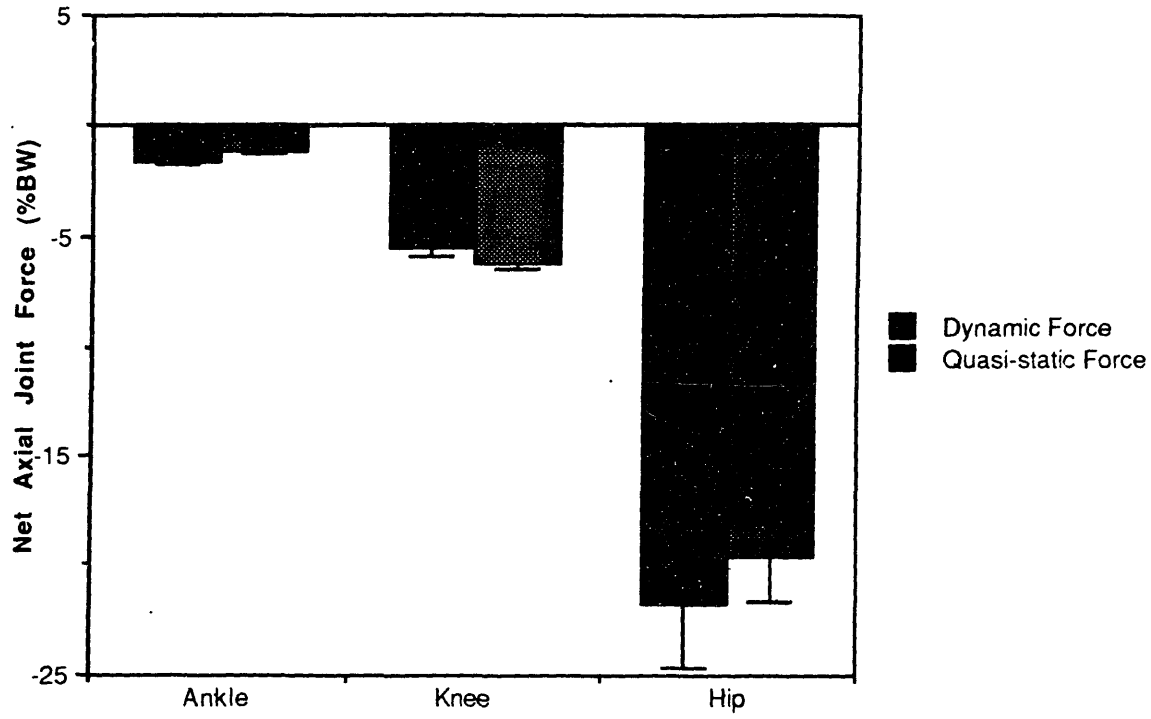

Figure 3.10.a Swing phase quasi-static and dynamic peak axial forces (mean peak + std.dev.).

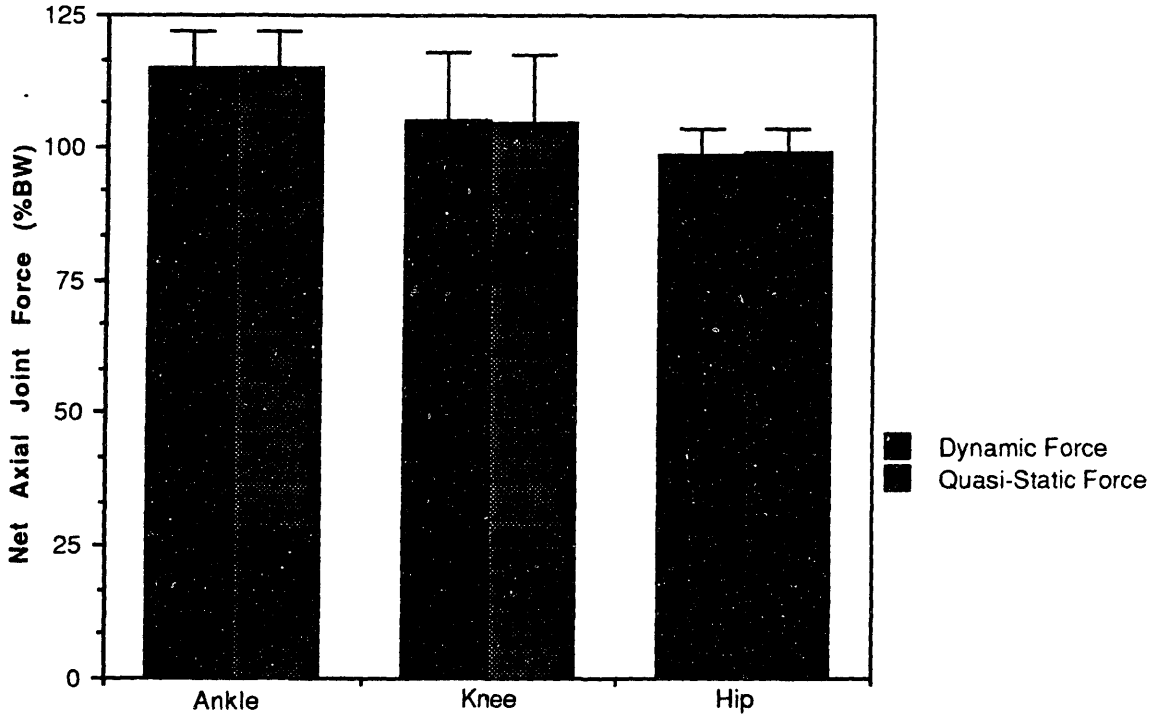

Figure 3.10.b Stance phase quasi-static and dynamic peak axial forces (mean peak + std.dev.).

The peak forces referred to in Table 3.8 and Figures 3.10 and 3.11 occurred simultaneously for the ankle, knee and hip during stance phase and corresponded to the time at which the ground reaction force was a maximum. The peak force during swing phase was more variable, occurring near the middle of the swing phase cycle for the ankle and knee and near the beginning or end for the hip. These peaks, described above, can be seen on the representative force curves of Figure 3.8.

Table 3.9 lists the root mean squared error between dynamic axial force curves and quasi-static axial force curves over complete swing and stance phase cycles. These differences were averaged for the seven gait trials. Normalization was performed by dividing by the range of the dynamic axial force. The root mean square error quantifies the difference between the two curves over the entire cycle, as opposed to the difference at the peaks only.

TABLE 3.9

RMS ERRORS BETWEEN DYNAMIC AND QUASI-STATIC AXIAL FORCE CURVES DURING GAIT (n=7)

	Ankle		Knee		Hip	
	Swing	Stance	Swing	Stance	Swing	Stance
RMS Error (%BW)	1.29	.456	1.35	.801	2.11	1.31
Normalized RMS Error	.390	.005	.840	.007	.501	.011

The normalized root mean square errors of Table 3.9 are plotted in Figure 3.12.

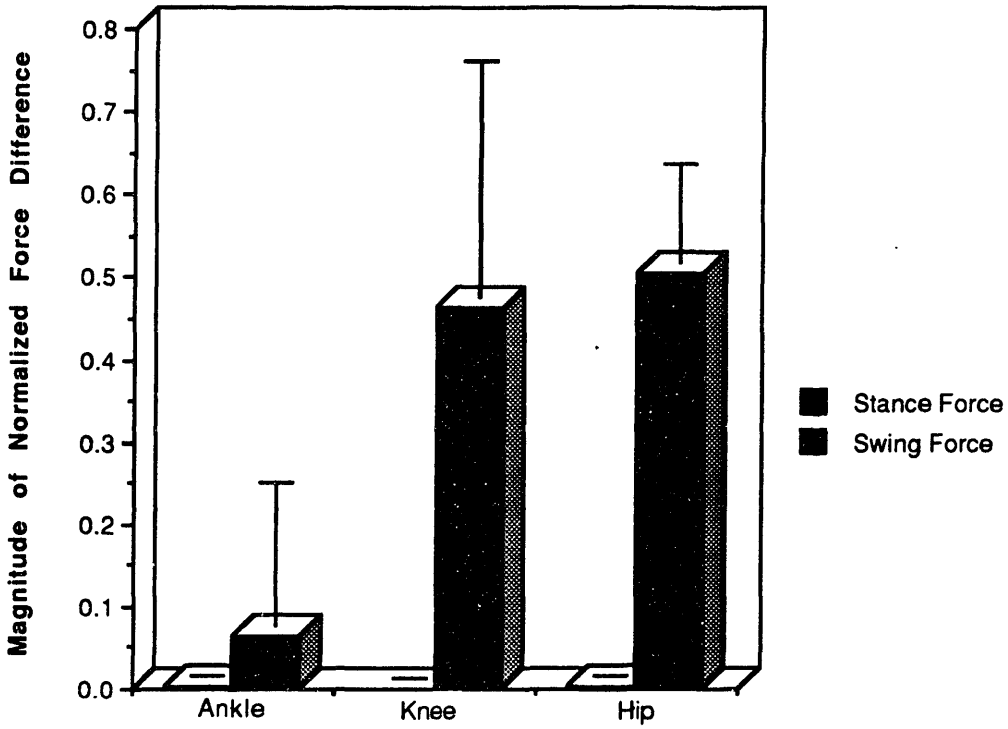

Figure 3.11 Magnitude of the normalized difference between peak dynamic and peak quasi-static axial forces during swing phase and stance phase (mean peak + std.dev.).

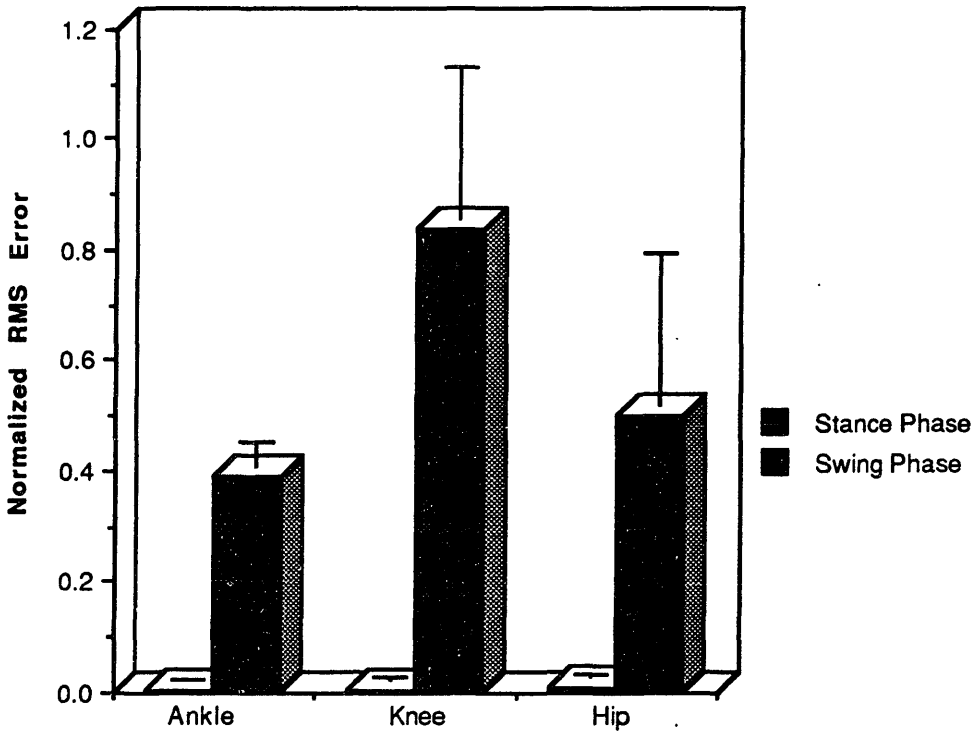

Figure 3.12 Normalized root mean square error between dynamic and quasi-static axial force curves during swing phase and stance phase (mean RMS+ std.dev.).

Table 3.10 lists the peak flexion torques calculated with the dynamic estimator and the quasi-static method. These differences at the peaks were averaged for the seven gait trials. The dynamic torque equation is given in equation (2-10), and the quasi-static torque equation is given in equation (2-18). Normalization was performed by dividing the peak difference by the range of the flexion-extension torque as calculated with the dynamic estimator.

TABLE 3.10

PEAK FLEXION TORQUE DIFFERENCES RESULTING FROM USING DYNAMIC VS. QUASI-STATIC ESTIMATES FOR JOINT TORQUES DURING GAIT (n=7)

	Ankle		Knee		Hip	
	Swing	Stance	Swing	Stance	Swing	Stance
Peak 1 q.s. torque (%BW*M)	.019	15.3	.333	5.43	2.09	7.84
Peak 1 dyn. torque (%BW*M)	.146	15.3	1.54	5.79	4.79	5.22
Peak 1 difference (%BW*M)	.127	.020	1.21	.361	2.70	-2.62
Normalized Peak 1 difference	.476	.001	.376	.003	.270	-.192
Peak 2 q.s. torque (%BW*M)	-.074	-1.32	-1.05	-3.71	-1.80	4.02
Peak 2 dyn.torque (%BW*M)	-.130	-1.35	-1.62	-4.11	-5.22	7.25
Peak 2 difference (%BW*M)	-.056	-.033	-.571	-.403	-3.41	3.22
Normalized Peak 2 difference	-.220	-.002	-.117	-.039	-.341	.236

Figures 3.13 and 3.14 show the information of Table 3.10 graphically. Figure 3.13 gives the peak flexion torques for the ankle, knee and hip as calculated with both the dynamic and quasi-static methods during swing phase and stance phase. In Figure 3.14, the normalized peak flexion torque differences are given during swing phase and stance phase.

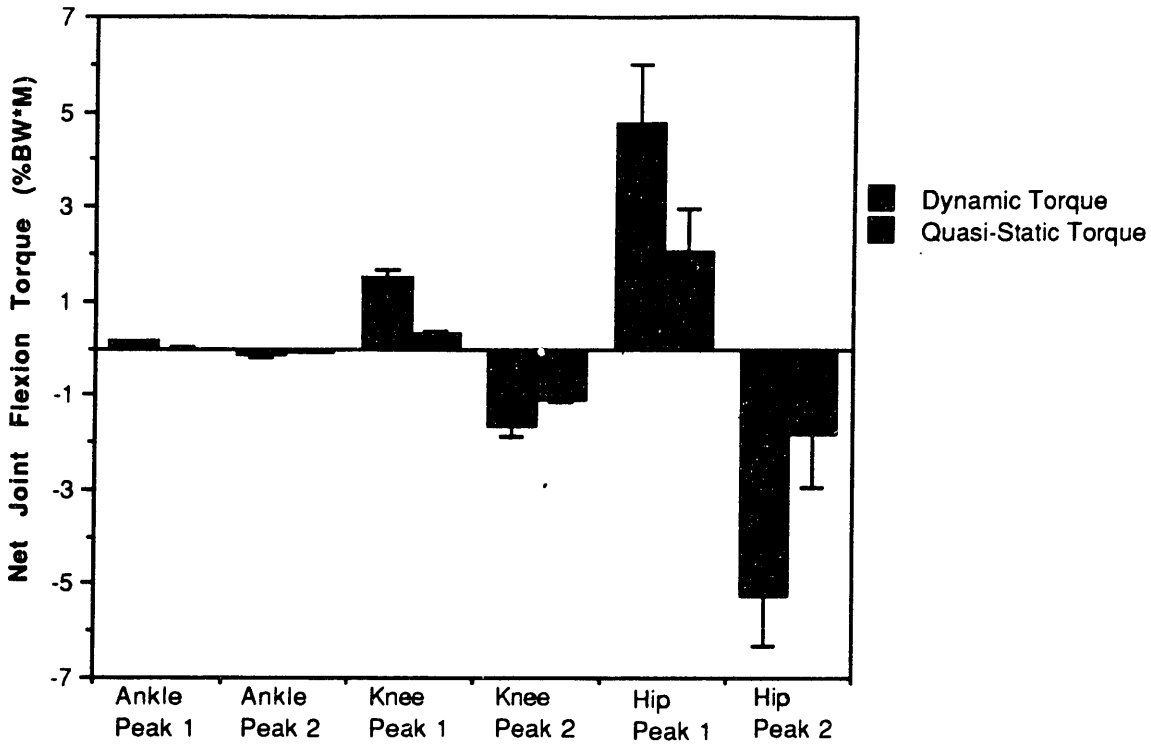

Figure 3.13.a Swing phase quasi-static and dynamic peak flexion-extension torques (mean peak+ std.dev.) (n=7).

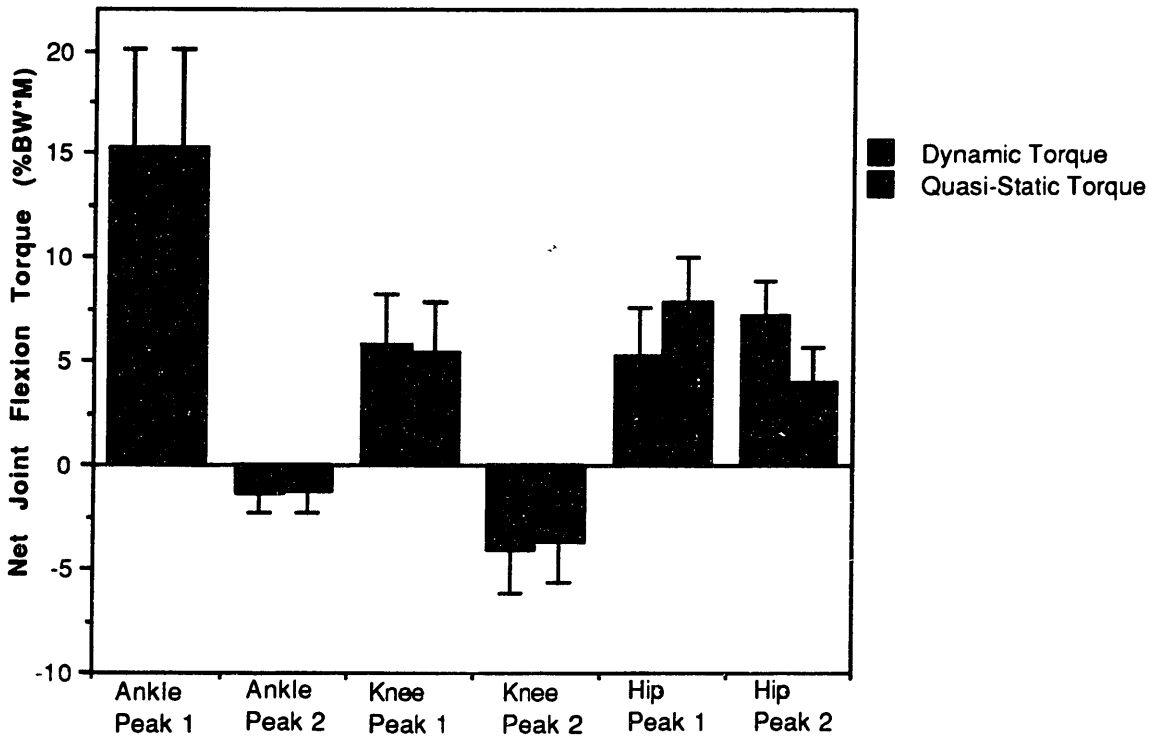

Figure 3.13.a Stance phase quasi-static and dynamic peak flexion-extension torques (mean peak+ std.dev.) (n=7).

Table 3.10 and Figures 3.13 and 3.14 refer to "Peak 1" and "Peak 2" on the torque curves. During swing phase for all three joints, peak 1 is the absolute maximum torque value and peak 2 is the absolute minimum torque value during the cycle. For the ankle and hip, peak 1 occurred at or shortly after toeoff and peak 2 occurred at or shortly before heel strike. The opposite was true for the knee joint. During stance phase the same representation is used for the ankle and knee, where peak 1 refers to the absolute maximum and peak 2 refers to the absolute minimum. For the hip during stance phase, the curves were less consistent. Peak 1 refers to the first local maximum and peak 2 refers to the last local maximum of the hip curve, both of which were always positive. Peak 1 occurred near toeoff for the ankle and closer to heel strike for the knee and hip. Peak 2 occurred near heel strike for the ankle, near the middle of the stance phase cycle for the knee, and towards toeoff for the hip. These peaks, described above, can be seen on the representative torque curves of Figure 3.9.

Table 3.11 lists the root mean squared error between dynamic and quasi-static flexion-extension torque curves over complete swing and stance phase cycles. These differences were averaged for the seven gait trials. Normalization was performed by dividing by the range of the dynamic flexion-extension torque. The root mean square error quantifies the difference between the two curves over the entire cycle, as opposed to the difference at the peaks only. Figure 3.16 shows the normalized RMS errors of Table 3.11.

TABLE 3.11

RMS ERRORS BETWEEN DYNAMIC AND QUASI-STATIC FLEXION TORQUE CURVES DURING GAIT (n=7)

	Ankle		Knee		Hip	
	Swing	Stance	Swing	Stance	Swing	Stance
RMS Error (%BW*M)	.080	.036	.788	.479	2.22	2.57
Normalized RMS Error	.293	.002	.245	.047	.222	.189

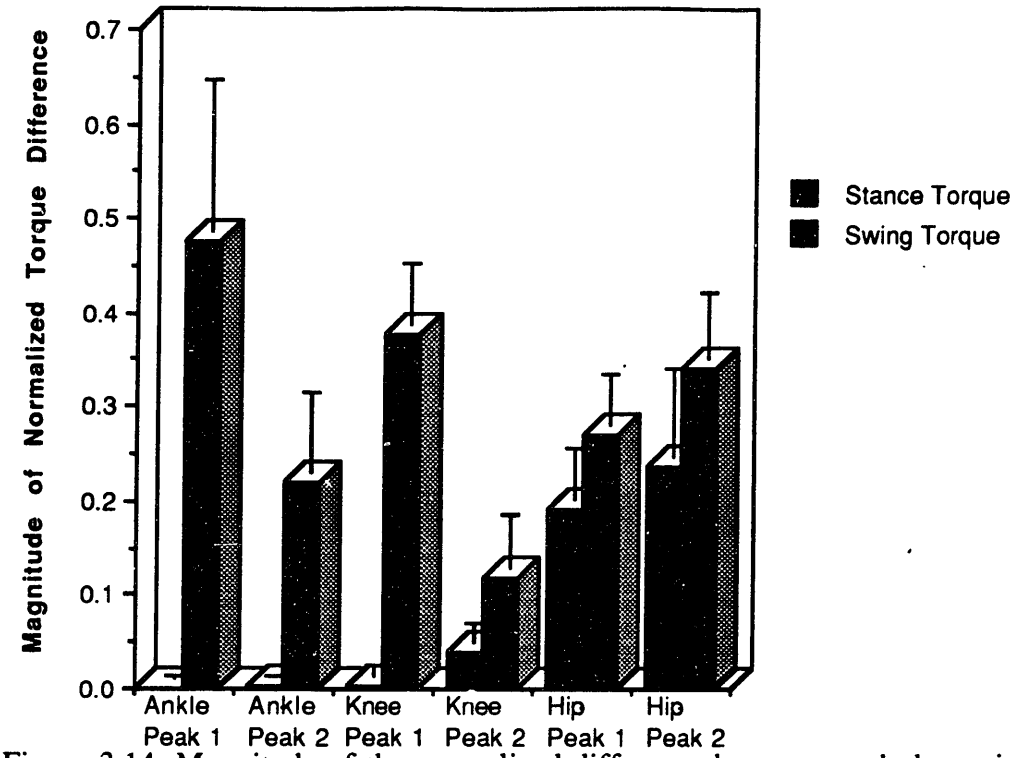

Figure 3.14 Magnitude of the normalized difference between peak dynamic and quasi-static flexion-extension torques during swing and stance phase (mean peak + std.dev.).

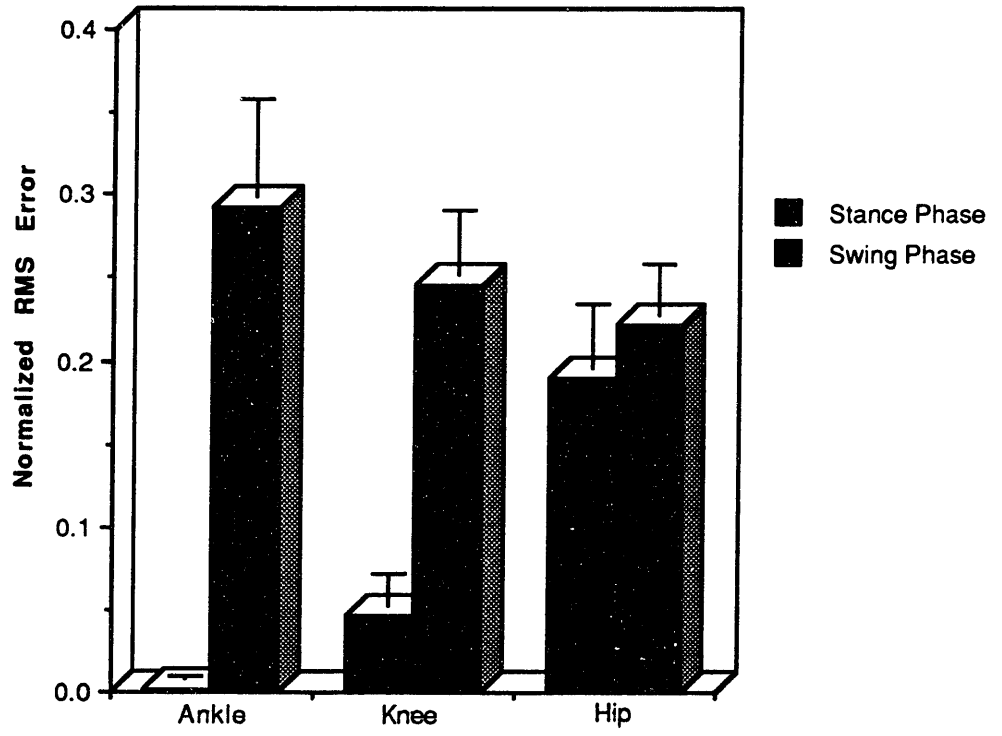

Figure 3.15 Normalized root mean square error between dynamic and quasi-static flexion-torque curves during swing phase and stance phase (mean RMS+ std.dev.) (n=7).

3.2.3 Representative Momentum Curves

As was the case with the force and torque curves, the number of plots generated by the dynamic estimator made the inclusion of all the linear and angular momentum curves for the swing and stance phases of all seven gait trials impractical. The representative swing leg and stance leg linear and angular momentum curves given in this section are for a normal male subject (age: 24 years, height: 175 cm, weight: 73 kg).

The representative linear momentum curves are given in Figure 3.16. Figure 3.16.a shows the three components of linear momentum for the foot, shank and thigh during the swing phase of gait. Figure 3.16.b shows the three components of linear momentum for the foot, shank and thigh during stance phase.

The representative angular momentum curves are given in Figure 3.17. Figure 3.17.a shows the three components of angular momentum for the foot, shank and thigh during swing phase. Figure 3.17.b shows the three components of angular momentum for the foot, shank and thigh during stance phase.

Figure 3.18 shows representative curves for the whole body linear momentum and the whole body angular momentum about the body center of mass during a complete stance phase cycle, which encompasses an entire swing phase cycle of the opposite leg.

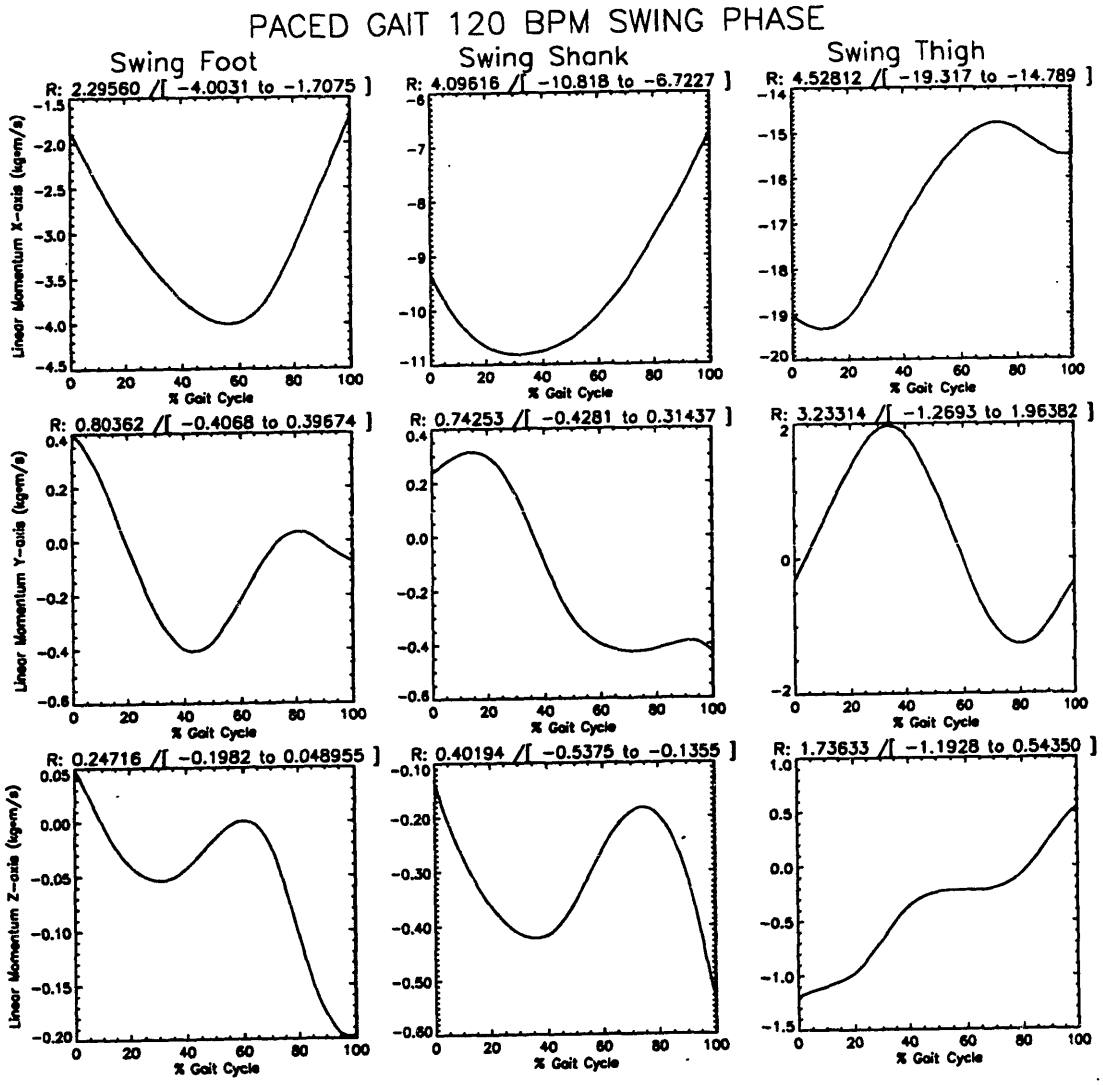

Figure 3.16.a Representative linear momentum curves for the foot, shank and thigh during the swing phase of gait.

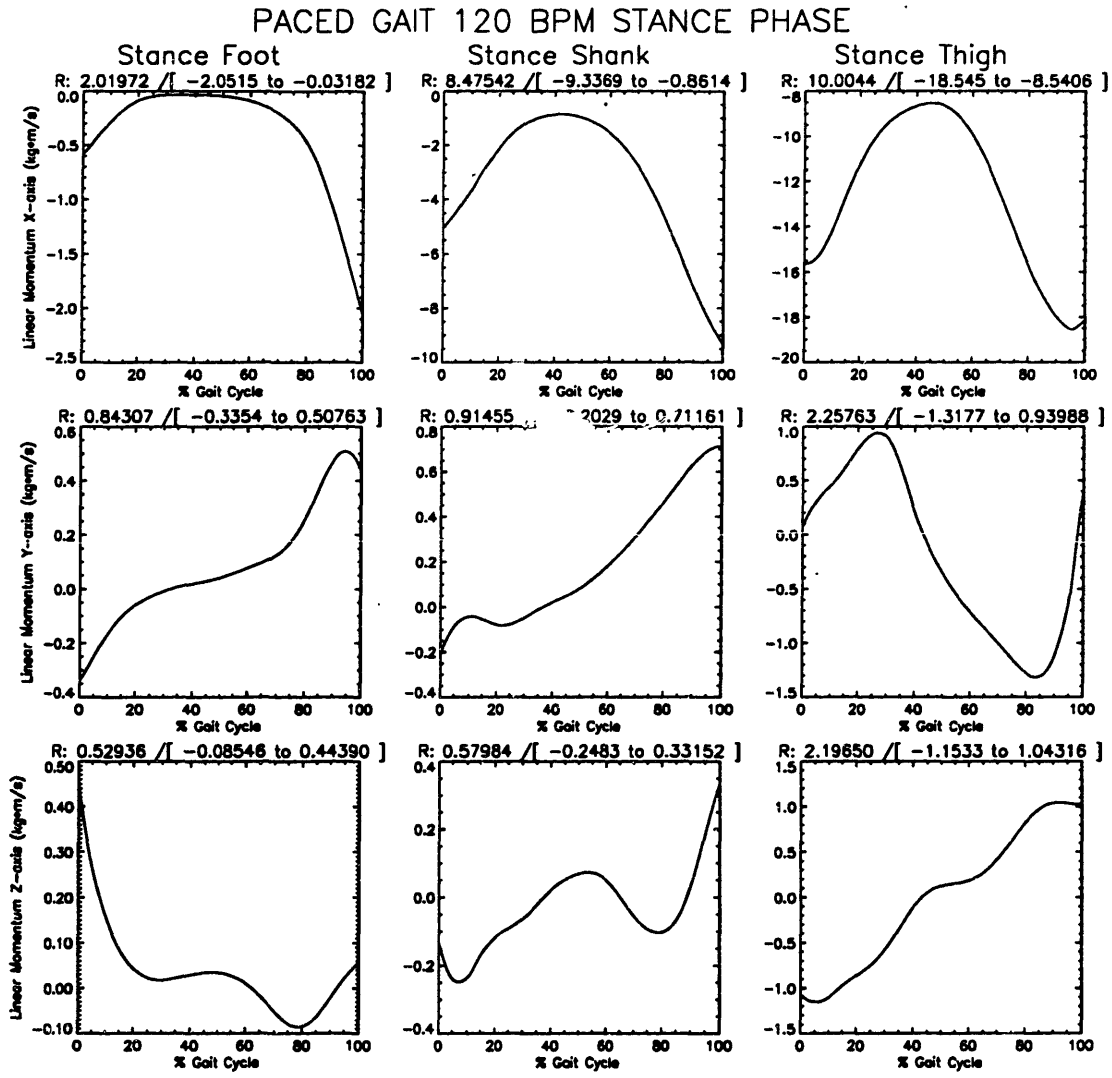

Figure 3.16.b Representative linear momentum curves for the foot, shank and thigh during the stance phase of gait.

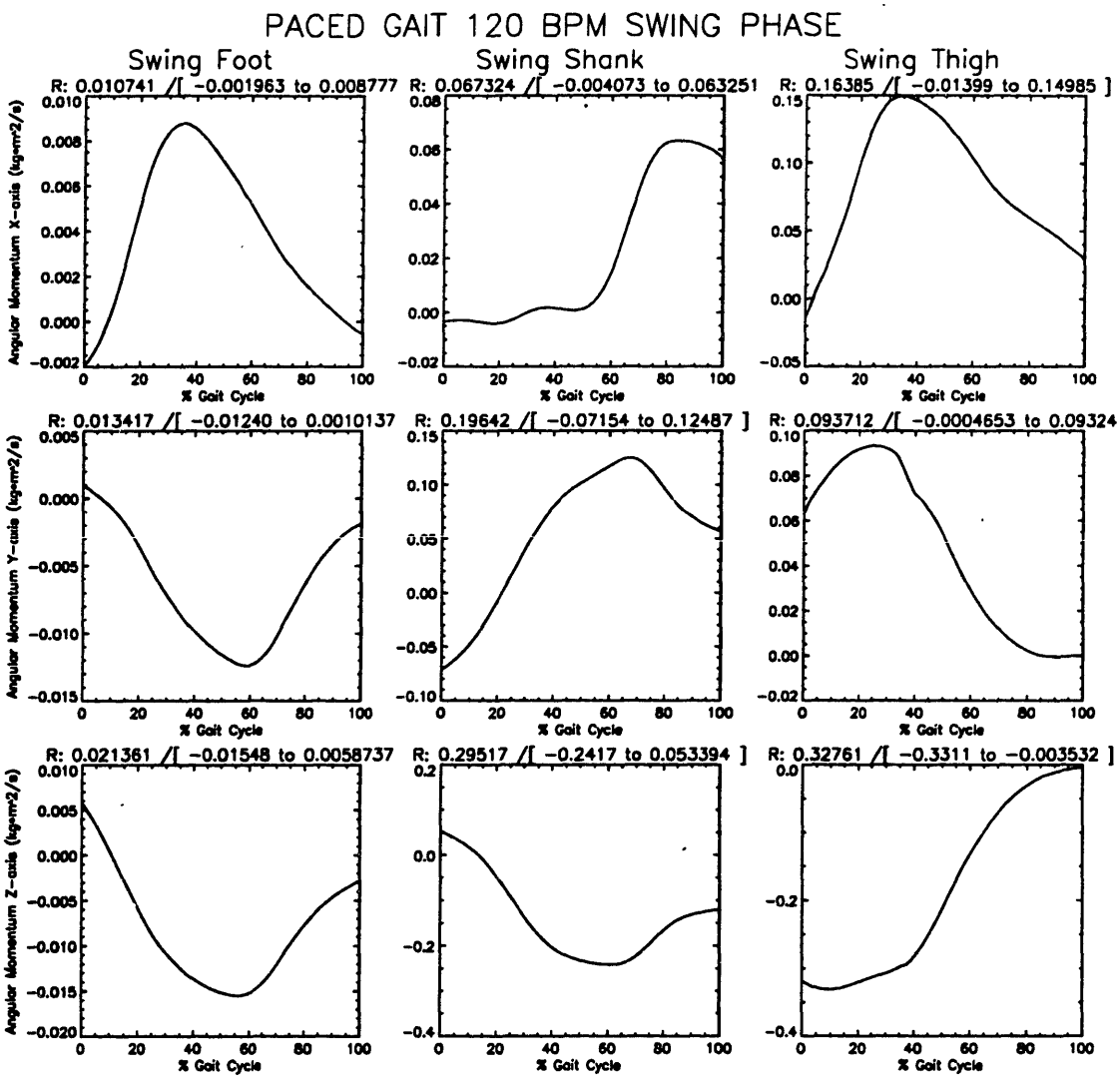

Figure 3.17.a Representative angular momentum curves for the foot, shank and thigh during the swing phase of gait.

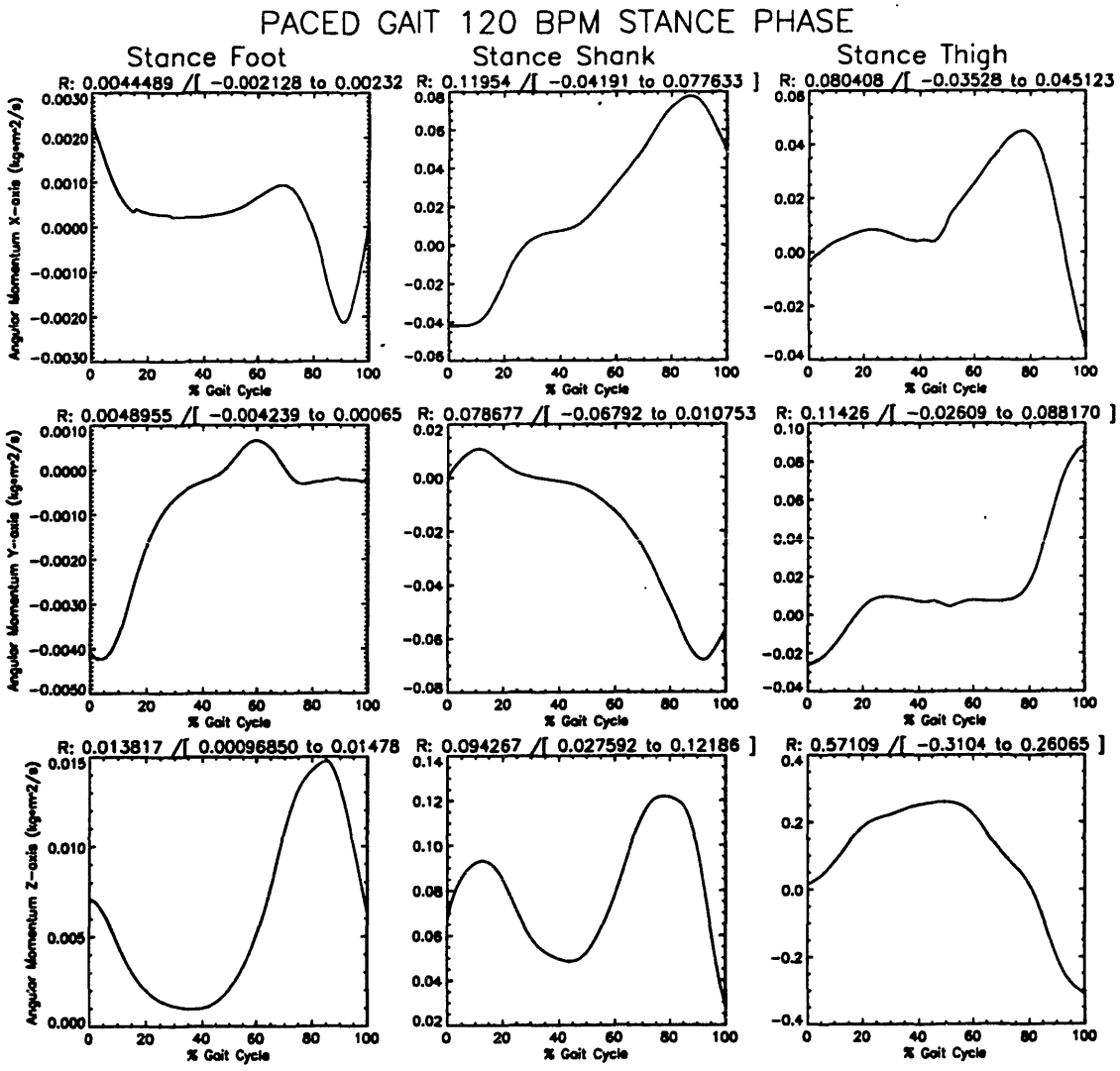

Figure 3.17.b Representative angular momentum curves for the foot, shank and thigh during the stance phase of gait.

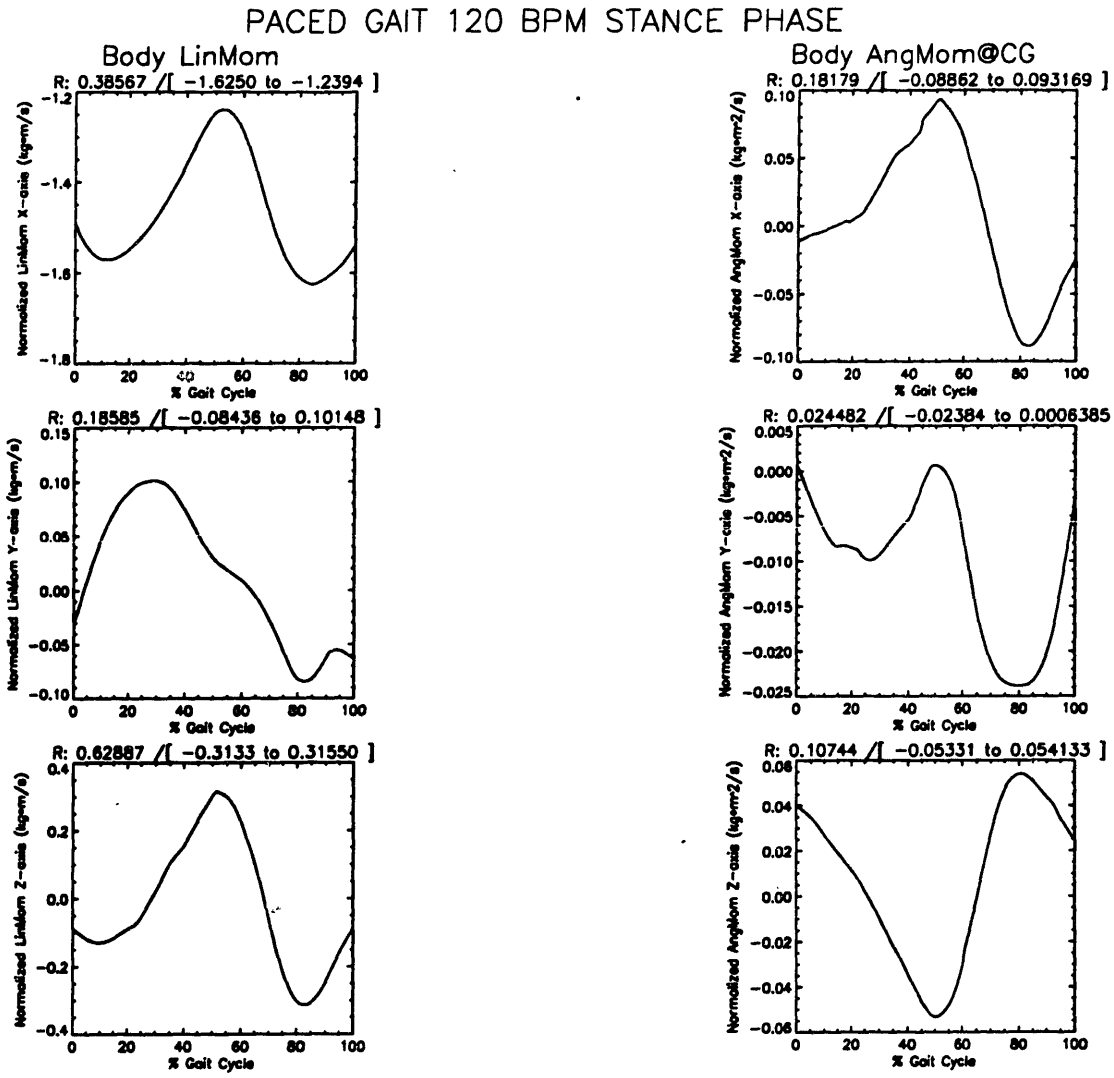

Figure 3.18 Representative whole body linear momentum and angular momentum about the center of mass during a stance phase of gait.

3.3 Lifting

Curves for the three components of force and three components of torque acting on the L2-L3 back joint of the MGH model during lifting are presented for lifting trials representative of those performed by the subject.

3.3.1 Representative Back Force and Torque Curves

The representative curves for the forces and torques on the back joint during lifting trials were very similar in shape and in magnitude for the trials in which the subject lifted with straight legs, and with bent knees. For this reason, only one set of representative back force and torque curves is given. Figure 3.19.a shows representative curves for the forces at the back joint during lifting with straight legs. The torque curves of Figure 3.19.b are for the same lifting trial.

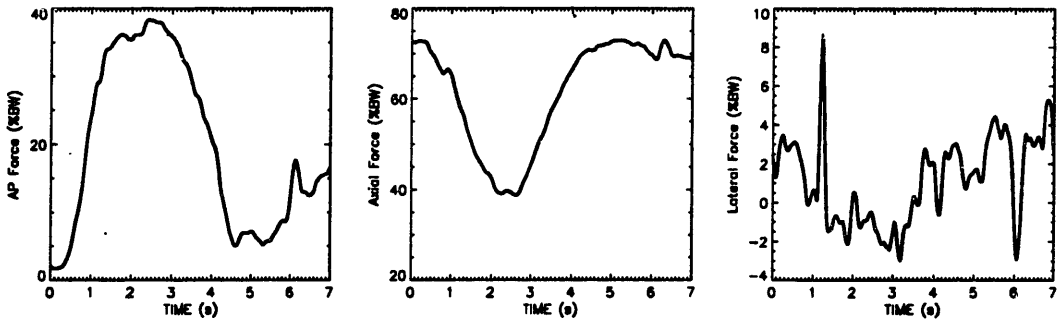

Figure 3.19.a Representative curves for back forces during lifting of a 25 lb. weight.

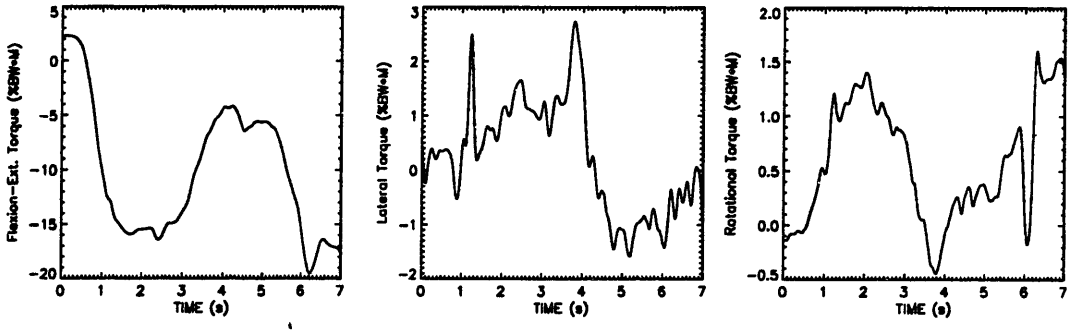

Figure 3.19.b Representative curves for back torques during lifting of a 25 lb. weight.

3.4 Three-Link Model Chair-Rise Simulations

The simulation inputs consisted of ankle, knee and back angular position, velocity and acceleration trajectories from two chair-rise trials performed by the same subject. Table 3.12 gives a comparison of the peak torques calculated with the MGH 11 segment dynamic estimator and those calculated with the three-link simulator. A graphical comparison is provided in Figures 3.20 and 3.21. The first column, labeled 'Actual Torques', shows torques calculated with the MGH 11 segment model. The second column, labeled 'Model Torques', shows the simulated torques. The third column shows the generalized forces for each joint, an intermediate quantity used to calculate the joint torques as described in section 2.4.2. As mentioned previously, the simulation assumed sagittal symmetry, and consequently only the flexion-extension torques were calculated with the three-link model. The torques presented in this section are in newton-meters, in contrast to the rest of the of the torques in this thesis which are in percent body weight-meters.

TABLE 3.12

PEAK FLEXION-EXTENSION TORQUES CALCULATED WITH THE MGH ELEVEN SEGMENT DYNAMIC ESTIMATOR AND SIMULATED WITH THE THREE-LINK MODEL (N*M)

	Joint 1 (Ankle)		Joint 2 (Knee)		Joint 3 (Hip)	
	MGH Model	Simulated	MGH Model	Simulated	MGH Model	Simulated
Trial 1	51	53	-144	-133	-47	-48
Trial 2	45	45	-145	-131	-46	-50

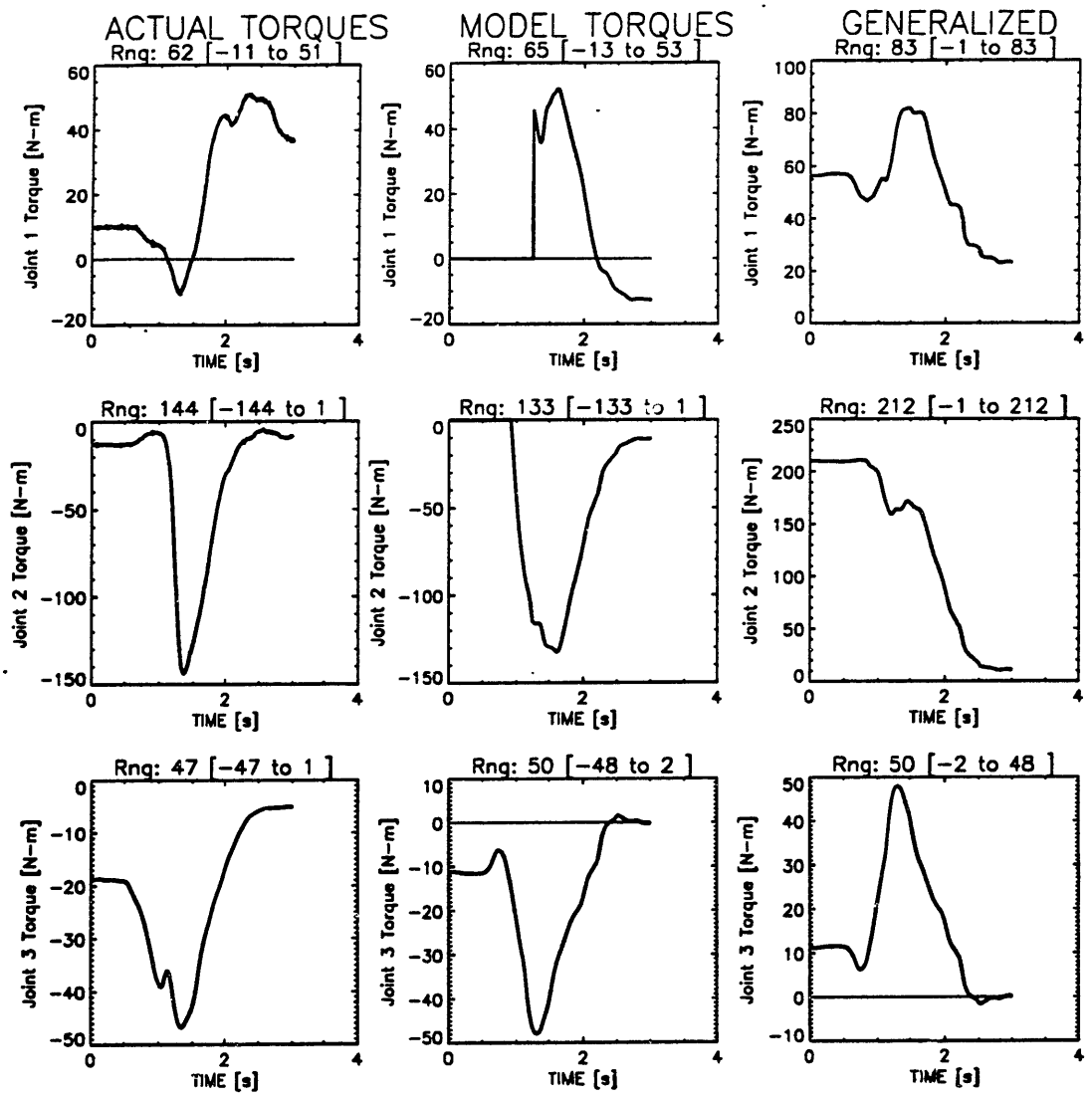

Figure 3.20 Torques calculated with the MGH 11 segment model and the three-link simulation. Also shown are the generalized forces. Chair-rise trial one.

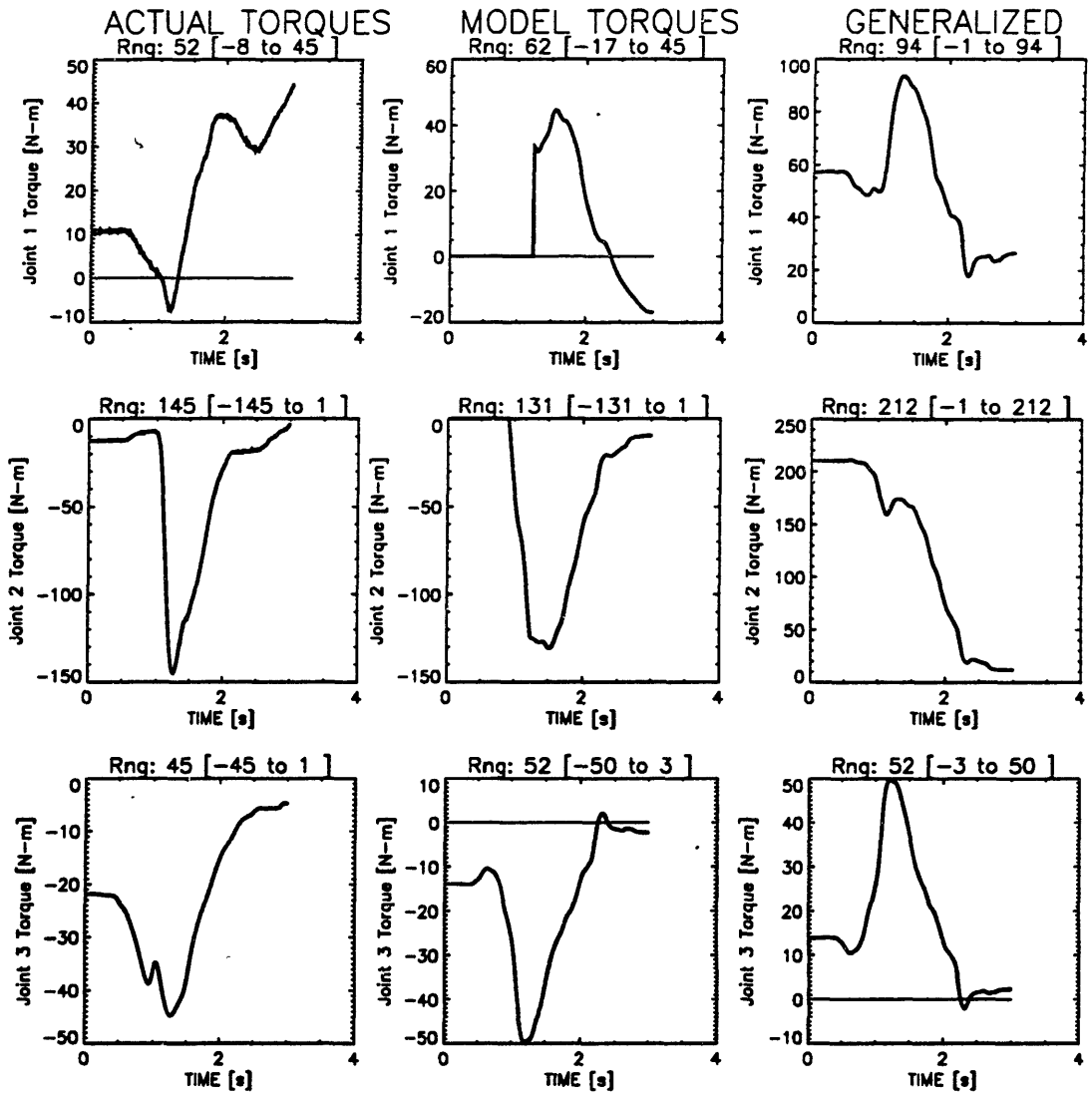

Figure 3.21 Torques calculated with the MGH 11 segment model and the three-link simulation. Also shown are the generalized forces. Chair-rise trial two.

3.5 Validation of the MGH Model

As mentioned in section 2.2.2, the dynamic estimator calculates the upper body kinetics from upper body kinematics, while the lower body kinetics are calculated from lower body kinematics and the ground reaction. The MGH 11 segment model was validated by calculating the forces and torques at the back joint both from the head and arms down to the back joint and from the feet up to the back joint. This method of model validation was suggested by deLooze as a means to validate dynamic linked segment models [100]. A comparison of the back forces and torques calculated from the lower body with the back forces and torques calculated from the upper body is presented for six standing trials and six chair-rise trials. The group had an average age of 28 (range 24-36) years, height of 170 (range 160-185) cm and weight of 66 (range 55-77) kg.

Figure 3.22.a shows the three components of force, and Figure 3.22.b shows the three components of torque acting on the back joint calculated from both the upper body and the lower body during a representative standing trial. Figure 3.23.a shows the three components of force, and Figure 3.23.b shows the three components of torque acting on the back joint calculated from both the upper body and the lower body during a representative chair-rise trial. The representative standing force and torque curves given in Figure 3.22 and the representative chair-rise force and torque curves given in Figure 3.23 are for a normal male subject (age: 25 years, height: 160 cm, weight: 55 kg). Table 3.13 lists the individual and average root mean square (RMS) errors between the back force and torque curves calculated from the upper body and those calculated from the lower body for standing and chair-rise trials. The RMS errors give a measure of the average difference between the upper and lower body back force curves in (%BW) and a measure of the average difference between the upper and lower body back torque curves in (%BW*M).

TABLE 3.13

MODEL VALIDATION: RMS ERRORS FOR BACK JOINT FORCES (%BW) AND TORQUES (%BW*M) CALCULATED FROM THE UPPER AND THE LOWER BODY

	FORCE			TORQUE		
	Axial	Lateral	Ant./Post.	Flexion	Abduction	Rotation
Standing						
Subject 1	5.12	0.75	2.09	1.92	0.32	0.65
Subject 2	3.11	5.23	4.71	3.98	0.76	0.34
Subject 3	0.25	0.88	1.30	2.57	0.06	0.56
Subject 4	3.63	4.97	6.41	0.44	0.14	0.31
Subject 5	3.91	0.44	2.01	1.82	0.64	0.20
Subject 6	6.26	1.00	6.00	1.22	1.43	1.30
Average:	3.68	2.21	3.75	1.99	0.56	0.56
Std.Dev.:	2.12	2.25	2.23	1.21	0.51	0.40
Chair-rise						
Subject 1	4.30	1.23	0.77	3.83	0.24	0.86
Subject 2	8.90	4.40	5.10	2.83	0.27	0.89
Subject 3	9.21	5.02	4.18	4.85	0.60	0.48
Subject 4	3.69	5.87	5.80	2.56	0.67	0.48
Subject 5	3.48	2.09	2.25	3.99	0.58	0.33
Subject 6	5.25	3.79	5.07	3.78	1.12	2.10
Average:	5.81	3.73	3.86	3.64	0.58	0.86
Std.Dev.:	2.59	1.77	1.95	0.83	0.32	0.65

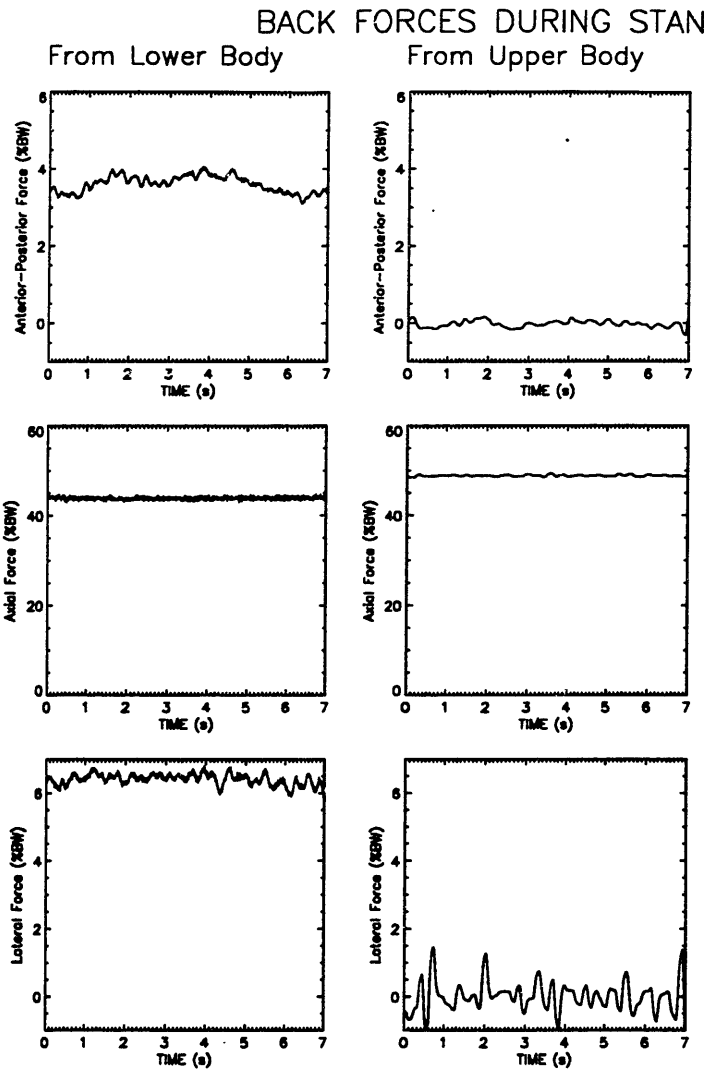

Figure 3.22.a Representative back joint forces calculated from the upper body and lower body during standing.

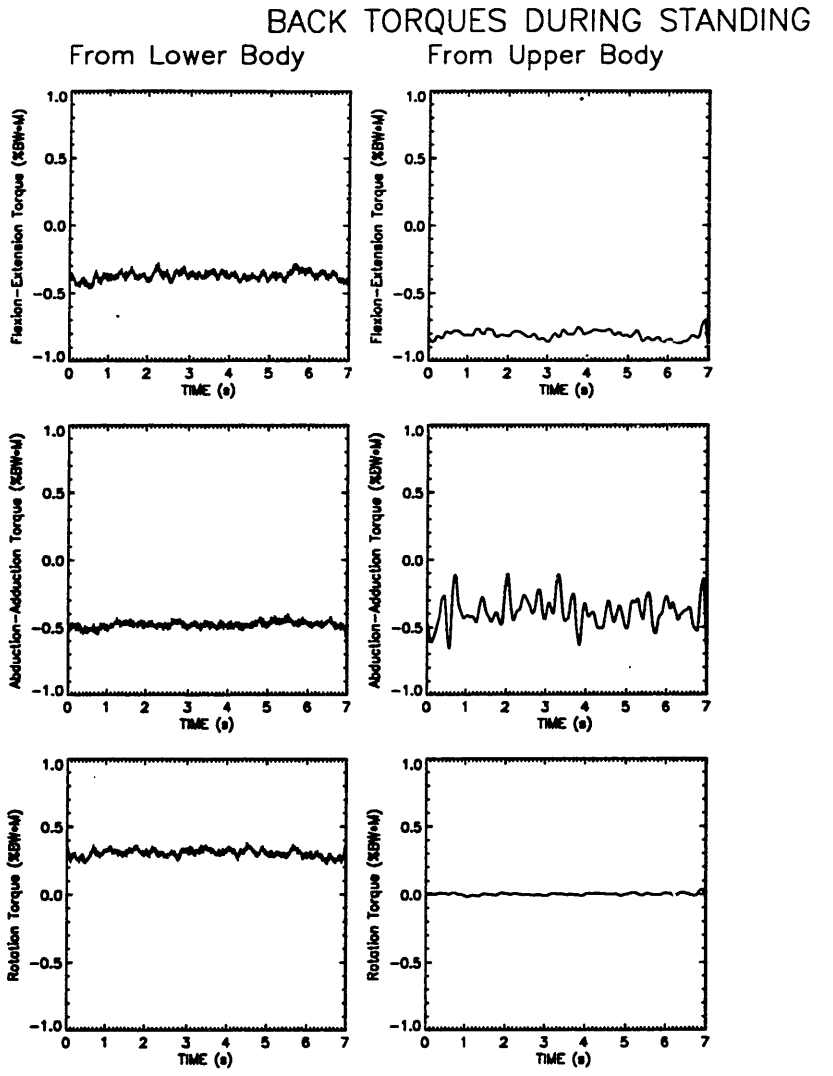

Figure 3.22.b Representative back joint torques calculated from the upper body and lower body during standing.

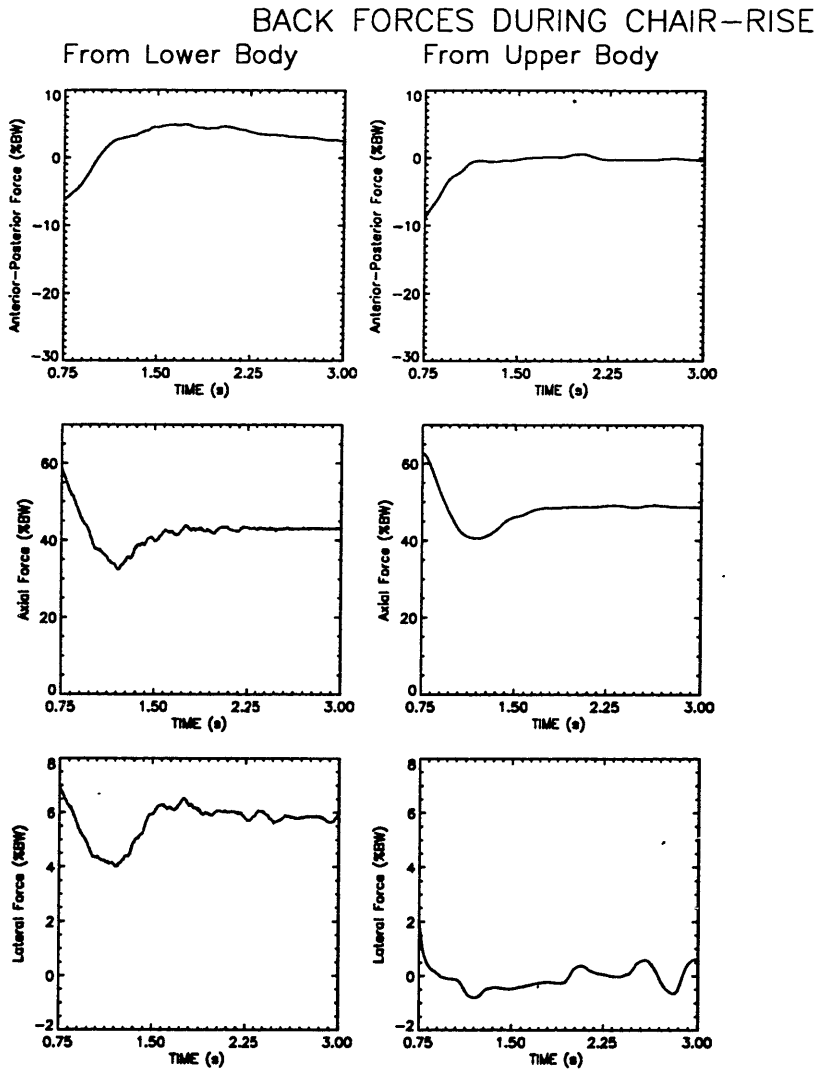

Figure 3.23.a Representative back joint forces calculated from the upper body and lower body during chair-rise.

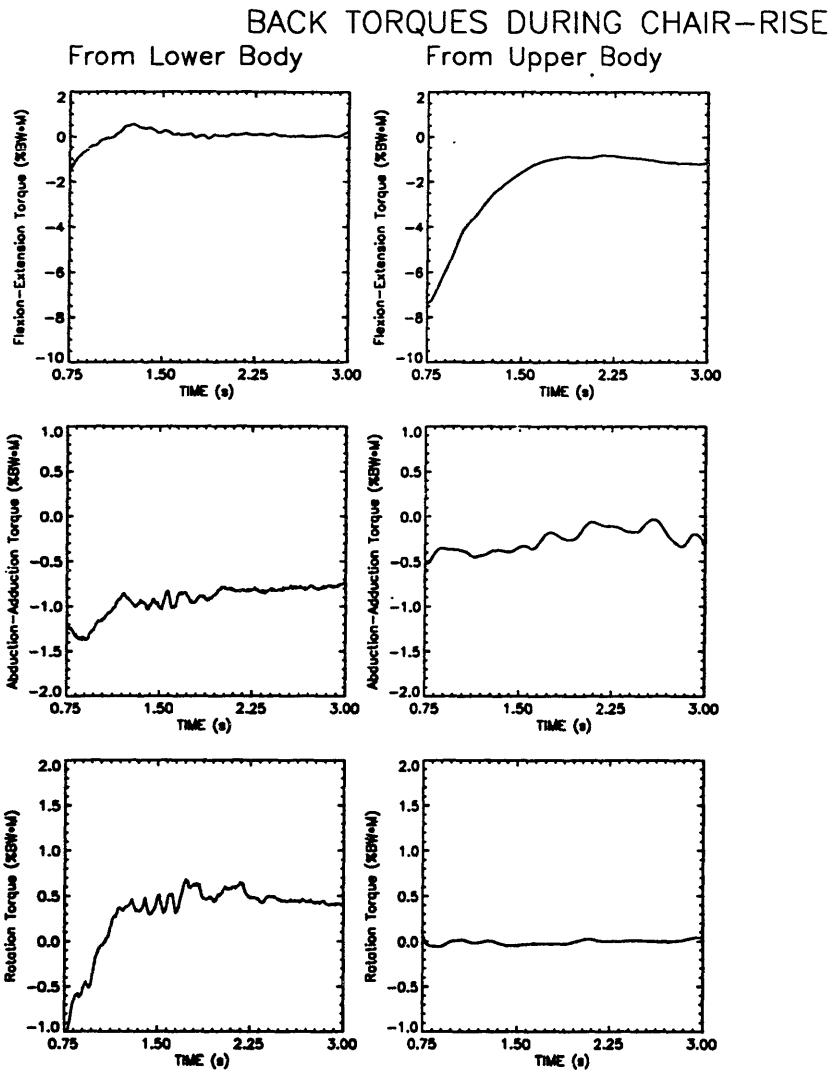

Figure 3.23.b Representative back joint torques calculated from the upper body and lower body during chair-rise.

Chapter 4

Discussion

In this chapter, the results of applying the dynamic estimator to the chair-rise, gait and lifting trials are discussed. In the discussion of these applications of the dynamic estimator, qualitative results consisting of representative force, torque and momentum curves are interpreted, and quantitative results which illustrate the importance of segmental dynamics are discussed. In addition, the three-link chair-rise simulations, and the MGH model validation results are discussed. Whenever possible, the findings presented in this thesis are compared to other published results.

4.1 Chair-Rise

First, the representative ankle, knee, hip, back and neck force and torque curves during chair-rise are briefly discussed. Next, the quantitative contributions of segmental dynamics to joint forces and torques during chair-rise are discussed. The effect that varying the speed of ascent during chair-rise has on the dynamic components of force and torque is also discussed. A brief discussion on linear and angular segmental momentum is included. Finally, whole body linear momentum and whole body angular momentum about the body center of mass and about the combined ankle joint during chair-rise are briefly discussed, including a preliminary discussion on how these quantities may be different for an unsuccessful chair-rise trial.

4.1.1 Representative Force and Torque Curves

The representative ankle, knee and hip flexion-extension torques shown in Figure 3.2 and the flexion-extension torque peak magnitudes listed in Table 3.6 were within the ranges of those reported by other studies which investigated lower limb joint torques during chair rise with quasi-static analysis [10], [36] and dynamic analysis [41]. Abduction-adduction and internal-external rotation torques have not been previously reported by studies on chair-rise. In this study they were found to be relatively small in magnitude and highly variable. In addition, comparisons of the joint forces and the neck and back torques found in this study with those of other studies were not possible, since these were unavailable from the literature. The axial and anterior-posterior force magnitudes were quite consistent across subjects, but the medial-lateral force magnitudes were smaller and more variable.

Most of the discussion of the representative force and torque curves will be limited to the axial component of force and the flexion-extension component of torque. To aid in this discussion, Figure 4.1 shows the axial forces for all the joints of the body and Figure 4.2 shows the flexion-extension torques for all the joints of the body. These force and torque curves were taken from the representative force and torque curves of Figures 3.1 and 3.2. To facilitate interpretation, three vertical line markers have been added signifying key points during the chair-rise trial. The first line marked 'tf' denotes the onset of trunk flexion. The second line marked 'lo' signifies liftoff from the chair or the time at which the subject lost contact with the seat. The third line marked 'ss' denotes the time at which stable stance was achieved. As mentioned in section 2.3.1, the dynamic components of force and torque were averaged from the onset of trunk flexion to stable stance for the upper body joints and from liftoff to stable stance for the lower body joints.

From Figures 4.1 and 4.2, it is apparent that the forces and torques of the right leg joints were similar in magnitude and in shape to those of the left leg. However, closer

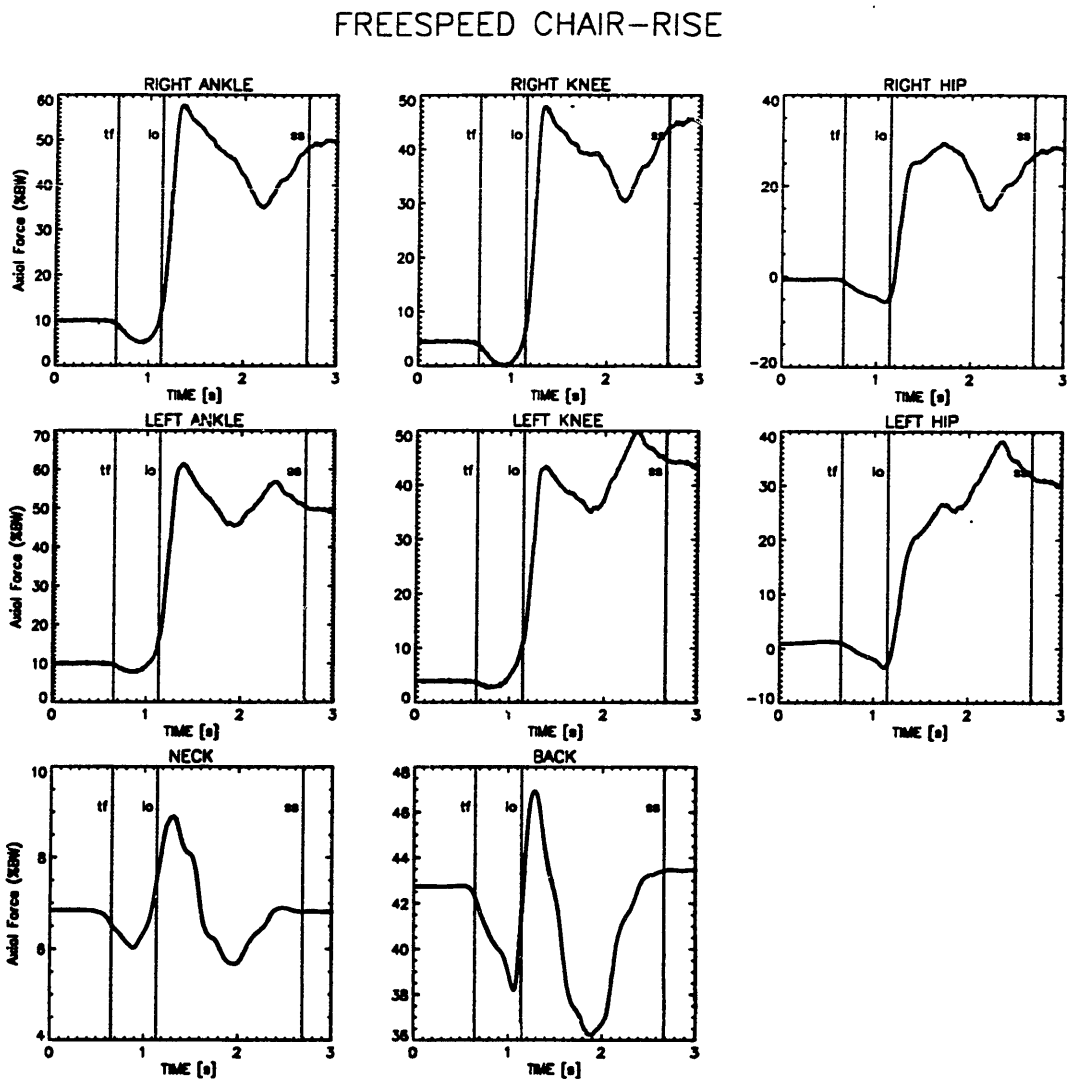

Figure 4.1 Axial force curves for all joints during chair-rise. ('tf': trunk flexion, 'lo': liftoff, 'ss': stable stance).

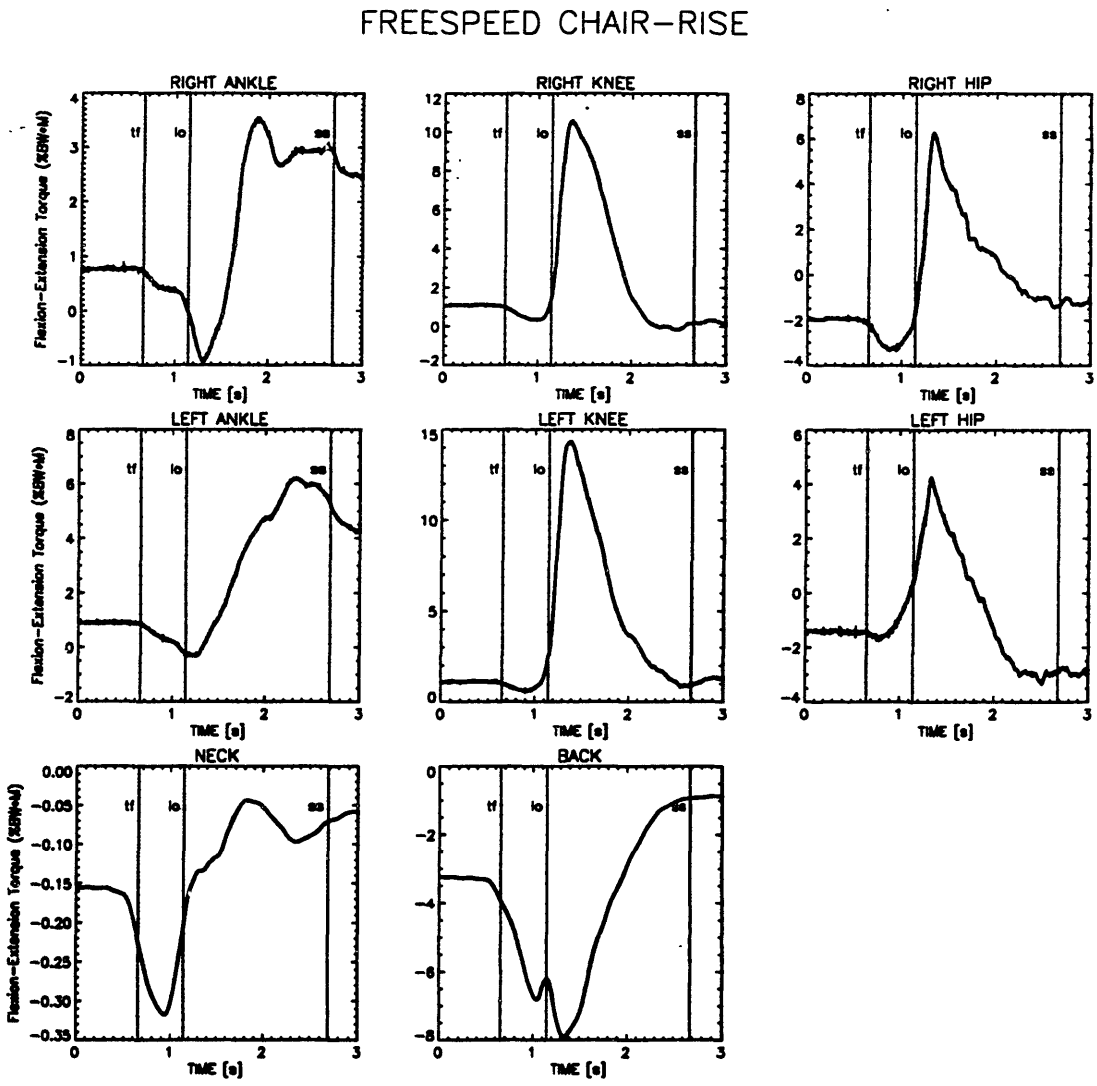

Figure 4.2 Flexion-extension torque curves for all joints during chair-rise. ('tf': trunk flexion, 'lo': liftoff, 'ss': stable stance).

inspection reveals that the magnitudes of the peaks were often different for the corresponding joint on the right and left legs. For example, in Figure 4.1 the left knee peak axial force is approximately 54 %BW, while the right knee peak axial force is only 44 %BW. Similarly the left knee peak extension torque of Figure 4.2 is approximately 13 %BW*M, while the right knee peak extension torque is only 9 %BW*M. The higher axial force and extension torque on the left leg suggest that the subject put more weight on her left leg than her right leg during the chair-rise maneuver. One advantage of the MGH three-dimensional model is that it can detect bilateral asymmetries which would be missed by the more common unilateral, or sagittal plane models often used to study chair-rise.

From the representative axial force curves of Figure 4.1, we can see that the ankle, knee and hip axial force curves are similar in shape. This is because the compressive force on these joints is dominated by the ground reaction force (not shown). The magnitudes of the peak axial force curves also decreases from the ankles to the knees to the hips. This decrease in compressive axial force was expected because as one moves up the body beginning at the feet, there is progressively less load on the joints, as less body mass is above the joints.

From the representative flexion-extension torque curves of Figure 4.2, it was discovered that the maximum dorsiflexion for both ankles, the maximum extension torque for both knees and both hips, and the maximum flexion torque for the back all occurred within 0.06 seconds of each other. The time at which these peak torques occurred was shortly after liftoff as can be seen in Figure 4.2. Perhaps this is the critical time at which sit-back failure occurs in patients who are unable to rise from a chair without the use of their hands. This could be the time at which some elderly people lack the strength to generate enough torque on one or a combination of two or more joints to successfully rise from a chair. For example, it is possible that neither the required knee torque nor the required hip torque alone is enough to prevent a successful chair-rise, but the combination of the two simultaneous peak torques may recruit some of the same upper leg muscles

which cannot meet the torque demands of both joints at the same time. Further investigation of these possibilities would provide more insight into the principles underlying some people's inability to rise from a chair.

Another interesting observation is that the temporary reversal in slope on the back torque curve of Figure 4.2 corresponds to liftoff from the chair. Although the back torque remains in flexion (negative), this temporary reversal suggests that liftoff causes a transient extension torque on the back. While it was not within the scope of this thesis, this would be an interesting phenomena to investigate with musculoskeletal co-contraction models.

4.1.2 Dynamic Components of Force and Torque

The data in Table 3.5 and Figure 3.3 show that the effects of segmental dynamics are negligible at the ankle joint and still very small at the knee joint, but become larger rostrally. The insignificant dynamic components of force and torque on the ankle joint were expected since the feet remain stationary during chair-rise, and the external ground reaction force is applied directly to the feet. For this reason, the ground reaction force dominates the forces and torques on the ankle joint. The increasing dynamic loads that accompany increasing height can be explained because not only do the superior body segments have more motion themselves, but they also have more body segments inferior to them, whose dynamic effects combine to produce a greater total dynamic contribution. In other words, the quasi-static method of calculating joint forces and torques progressively loses accuracy while proceeding in the rostral direction. The results of Table 3.5 indicated that segmental dynamics accounted for less that 1% of the total force and torque on the ankle and knee joints during chair-rise. For this reason, the quasi-static assumption is acceptable for studies on chair-rise that only calculate the ankle and knee forces and torques. However, the larger contributions of segmental dynamics to the hip, back and neck forces and torques suggest that chair-rise studies investigating these joints should employ dynamic analysis

when calculating joint reactions. The data in Table 3.5 suggest that for chair-rise, static loads are always the largest components of joint force and torque, but dynamic components of force and torque become larger rostrally.

4.1.3 Dynamics and the Speed of Ascent

While more data are needed, Table 3.6 provides some preliminary evidence that increases in speed of ascent are accompanied by increasing segmental dynamic contributions to net joint forces and torques. In every joint, the dynamic percentages of force and torque were larger for the fast trials than for the slow trials. While a previous study, using dynamic torque calculations, showed that increases in chair-rise speed are accompanied by increases in joint torques [41], the preliminary results of this study indicate that at least part of the torque increase may be attributed to contributions from segmental dynamics. It is possible that quasi-static torques may also increase when the speed of ascent increases due to a larger ground reaction. Because the dynamic force and torque contributions varied with the speed of ascent, it seems appropriate for studies investigating variations in ascent speed to use a dynamic model to account for the disproportional effects of segmental dynamics at different ascent speeds.

The dynamic torques on the neck varied significantly between subjects and between different trials by the same subject. Due to this inter- and intra-subject inconsistency dynamic percentages for neck torque were not reported. The dynamic axial forces on the neck were consistent between subjects and between different trials by the same subject, indicating that the linear acceleration of the head was consistent. A possible explanation for the inconsistency of the dynamic torques on the neck could be highly variable head angular velocities and accelerations which are only included in the dynamic components of torque.

4.1.4 Representative Momentum Curves

As previously mentioned, the linear and angular segmental momentum curves are in global coordinates. The representative curves of Figures 3.5 and 3.6 indicate that the linear and angular momentum of the feet were very small in relation to the other body segments. This was expected since the feet are approximately stationary during the chair-rise maneuver. Another observation was that the body segments of the upper body tended to have larger linear and angular momentum components than those of the lower body. This was also expected, as the upper body segments tend to be both larger and have more motion than the segments of the lower body.

Because most of the body motions during chair-rise occur in the sagittal plane, the x-axis (anterior-posterior) and y-axis (vertical) components of linear momentum are larger than the z-axis (medial-lateral) linear momentum for most segments. The only exceptions to this were the feet, for which all components were small, and the shanks, which had small vertical and medial-lateral linear momentum components, but larger anterior-posterior components. This can be seen from the representative segmental linear momentum curves of Figure 3.5.

The large motions in the sagittal plane also explain why the z-axis (flexion-extension) angular momentum components were larger than the x-axis (abbduction-adduction) and y-axis (internal-external rotation) components of angular momentum. The z-axis angular momentum component was larger than the other two components for every body segment, as shown in Figure 3.6.

4.1.5 Whole Body Momentum

The representative whole body momentum curves are shown in Figure 3.7 for chair-rise. Again it is evident that the x-axis and y-axis components of linear momentum and the z-axis component of angular momentum are larger than the other components due to the sagittally planar nature of the chair-rise movement. The peak whole body momentums listed in Table 3.7 reaffirm this result. As mentioned previously, the whole body momentum curves were normalized by dividing by the subjects body mass.

In a previous study done in our lab, Schenkman determined the chair-rise maneuver could be divided into four categories with respect to momentum: (1) the flexion momentum phase, (2) the momentum transfer phase, (3) the vertical momentum phase, and (4) the stabilization phase [47]. The representative whole body momentum curves of Figure 3.7 show the flexion momentum phase in the x-axis component of linear momentum and the vertical momentum phase in the y-axis component of linear momentum. The peak vertical (y-axis) momentum occurs approximately 0.5 seconds after the peak horizontal (x-axis) momentum; which indicates the interval over which the momentum transfer phase occurs. The peak z-axis angular momentum occurred at approximately the same time at the peak x-axis linear momentum, probably due to the anterior motion of the head, arms and trunk prior to liftoff. Perhaps, the whole body angular momentum also contributes to vertical whole body linear momentum. An interesting prospect for future study would be to investigate the role of the joint torques during the momentum transfer phase. By studying the timing of peak joint torques in relation to peak whole body momentums, it might be discovered that joint torques are the driving mechanism by which horizontal momentum is transferred to vertical momentum.

Figure 4.3 shows some preliminary results which indicate that joint torques may in fact play a role in transferring horizontal linear momentum to vertical linear momentum

during chair-rise. In the top plot, the magnitudes of normalized whole body horizontal and vertical linear momentum are shown. The middle plot shows normalized whole body angular momentum about the center of mass and about the combined ankle joint. The lower plot shows the combined knee extension torques and the combined hip extension torques. These combined torques are the sums of the torques from the right and left leg joints.

From the plots of Figure 4.3, it was determined that the horizontal momentum peaked at 0.59 seconds and the vertical momentum peaked at 1.05 seconds. The angular momentum peak occurred at approximately the same time as the peak horizontal momentum. Specifically, the angular momentum about the center of mass peaked at 0.54 seconds, and the angular momentum about the combined ankle joint peaked at 0.58 seconds. Both the peak knee torque and the peak hip torque occurred at 0.74 seconds, after the peak horizontal momentum, but before the peak vertical momentum. The timing of these peak values suggests that the knee and hip torques may play a role in transferring the horizontal momentum to the vertical momentum, however further investigation is needed. Another interesting point to observe in Figure 4.3 and in Table 3.7, is that the magnitude of the peak vertical momentum is greater than the magnitude of the peak horizontal momentum. Because the peak whole body angular momentum occurs at approximately the same time as the peak horizontal momentum, it is possible that angular momentum is converted to vertical momentum along with horizontal momentum.

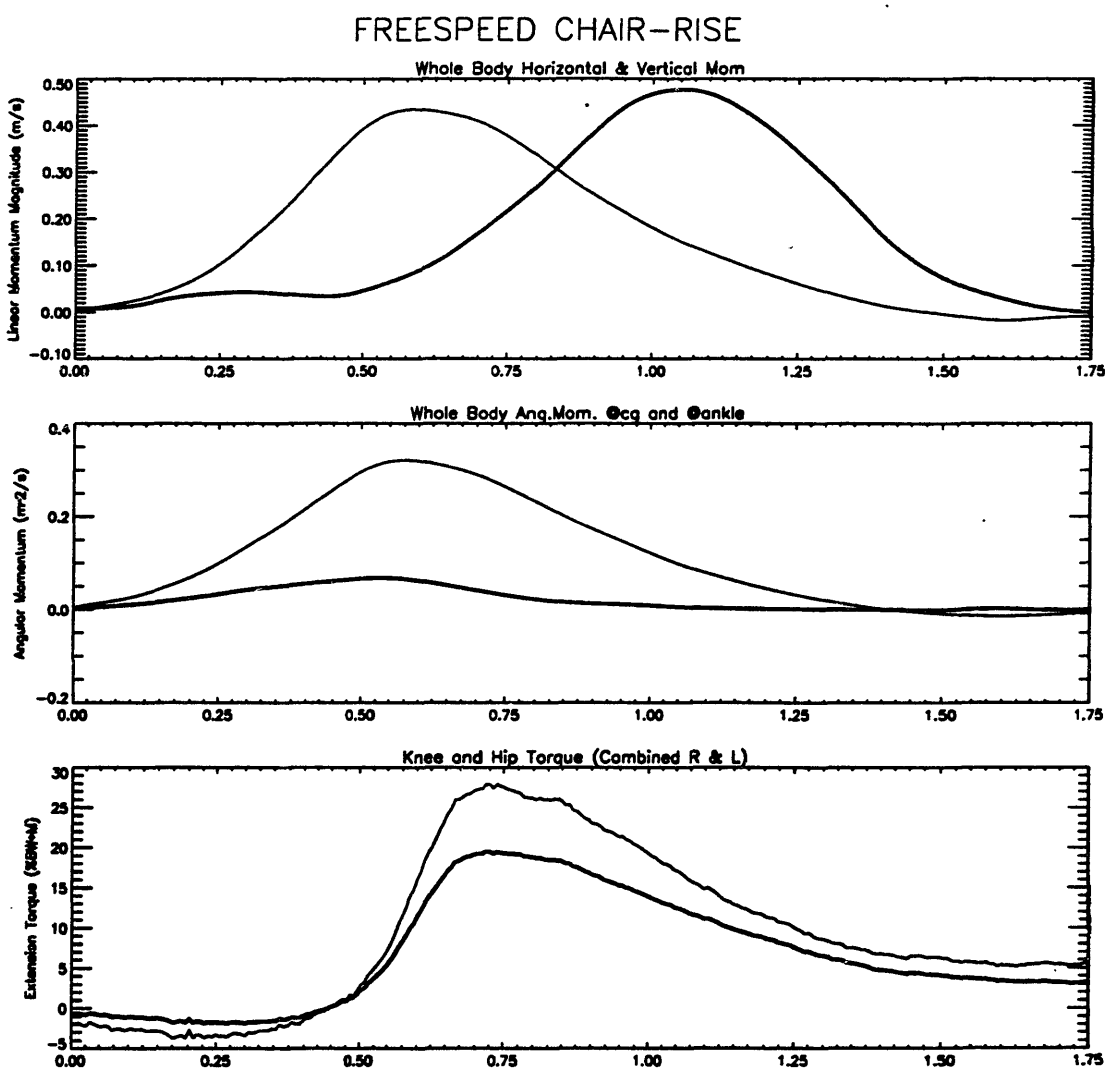

Figure 4.3 Chair-rise. Top plot: Normalized whole body horizontal and vertical linear momentum. Middle Plot: Normalized whole body angular momentum about the center of mass and the combined ankle joint. Lower Plot: Combined knee extension torque and combined hip extension torque.

Another area for future study will be to determine how whole body momentum is linked to successful versus unsuccessful chair-rises. For example, inadequate whole body horizontal momentum or the inability to provide sufficient joint torques to transfer the horizontal momentum to vertical momentum could result in sit-back failure. Another possibility would be that the inability to control whole body momentum would lead to an unstable stance, requiring a step to maintain balance.

In a preliminary investigation, the horizontal (x-axis) and vertical (y-axis) whole body momentum components from a sit-back chair-rise failure were plotted along side those from a successful chair rise. These plots, shown in Figure 4.4 show that in the unsuccessful trial, the subject only generated about half of the horizontal and vertical linear momentum generated by the subject who performed the successful trial. In addition, it appears that during the unsuccessful trial, very little horizontal momentum was transferred to vertical momentum following the peak horizontal momentum. Further study needs to be done to determine the degree to which whole body momentum dictates whether or not a chair-rise is successful or unsuccessful.

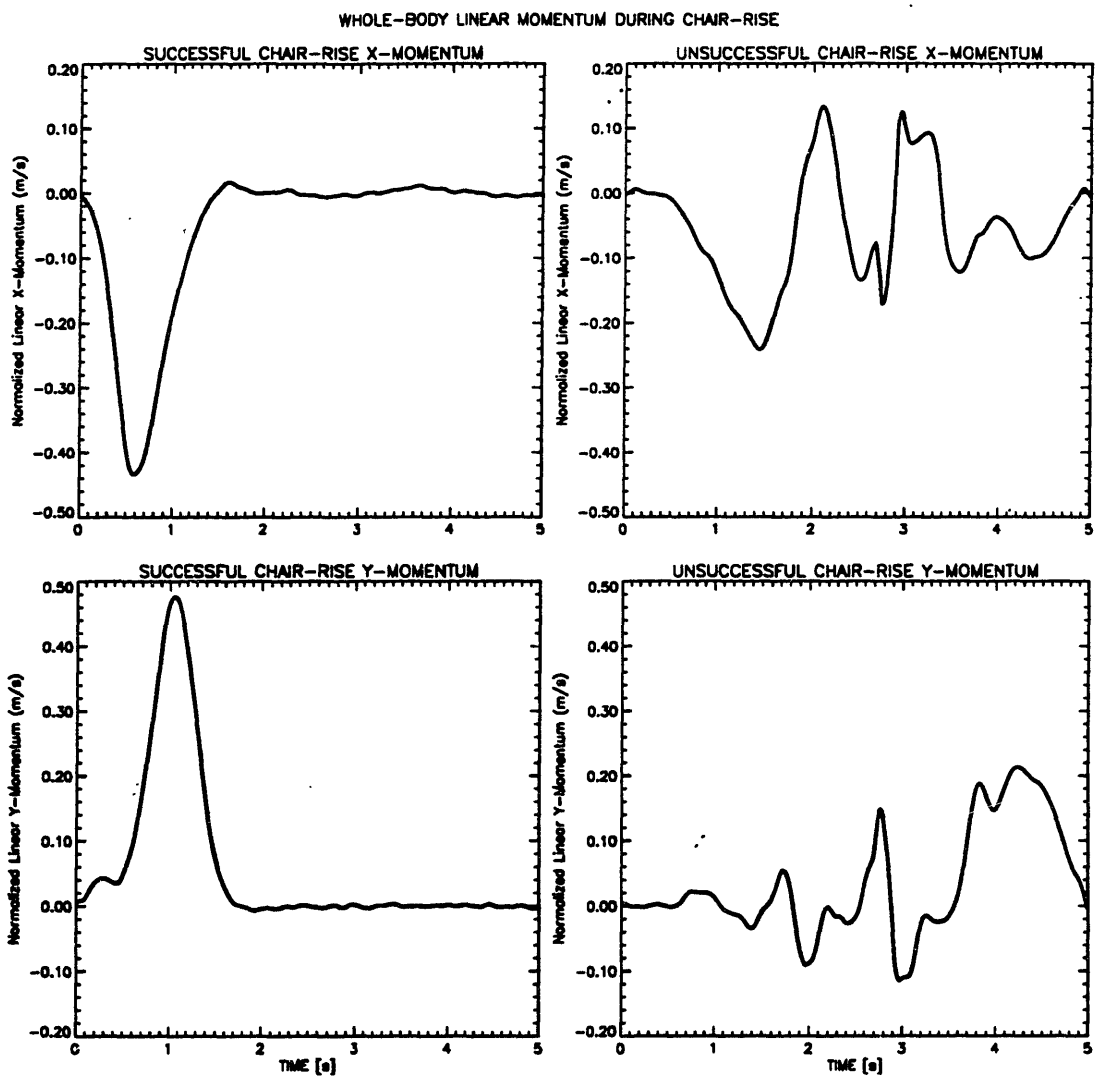

Figure 4.4 Whole body horizontal and vertical linear momentum for a successful and an unsuccessful chair-rise (sit-back) trial.

4.2 The Swing and Stance Phases of Gait

First the representative ankle, knee and hip axial force and flexion-extension torque curves during the swing and stance phases of gait are briefly discussed. Next, the significance of using a dynamic model versus a quasi-static model to calculate joint forces and torques are discussed, as it applies to the swing and stance phases of gait. Also, a brief discussion of the linear and angular segmental and whole body momentum curves is included.

4.2.1 Representative Force and Torque Curves

As mentioned previously, the dynamic estimator calculates the three components of force and the three components of torque for the ten joints of the MGH human body model. However for the swing and stance phases of gait, only the axial forces and flexion-extension torques on the ankle, knee and hip joints will be discussed as these were the only curves presented in the results of section 3.2. The magnitudes given in Table 3.6 and the curve shapes given in Figure 3.8 for the ankle, knee and hip joint axial forces were consistent with previous studies, including those by Bresler [7] and Patriarco [51]. The ankle, knee and hip flexion-extension torque curves given in Figure 3.9 and magnitudes given in Table 3.8 were similar to those reported in other studies, including those by Bresler [7] and Winter [52]. The remainder of the discussion on the representative force and torque curves will be in the following section, which emphasizes the role of segmental dynamics on joint forces and torques.

4.2.2 Dynamic Contributions to Swing and Stance Phase Joint Forces and Torques

As hypothesized, for both the net joint forces and torques, the differences between using a dynamic estimator and a quasi-static estimator were more apparent during the swing phase than the stance phase of gait. For the axial forces given in Table 3.8 and the flexion-extension torques given in Table 3.10, both the absolute and the normalized differences at the peaks were greater during swing phase than stance phase. The normalized differences were larger than the absolute differences, which can be attributed to the larger magnitudes of the stance phase curves relative to the swing phase curves.

From Figures 3.8 and 3.10, it is apparent that segmental dynamics did not greatly affect the axial forces present during stance phase. This can be seen by the virtually identical quasi-static and dynamic force curves of Figure 3.8 and the very close peak force values shown in Figures 3.8 and 3.10, which show that the role of segmental dynamics was more pronounced during swing phase than stance phase. As stated in equation (2-19), the only dynamic effect on axial force results from the mass of the segment multiplied by the linear acceleration of the segment center of mass in the axial direction. The dynamic component of force was slightly larger during swing phase, as shown in Table 3.8 indicating that the acceleration in the axial direction was slightly larger during swing phase.

As illustrated in Figure 3.11, after normalization, the difference between using a quasi-static and a dynamic force estimator was very small during stance phase relative to the magnitude of the total stance phase axial force. This suggests that axial forces are dominated by the ground reaction force during the stance phase of gait. Another observation from the axial force curves of Figure 3.8 was that segmental dynamics usually tend to increase the magnitude of the net axial load on the ankle, but usually tend to decrease the axial loads on the knee and hip during swing phase. Table 3.9 and Figure

3.12 show that the RMS errors were larger and the normalized RMS errors were much larger during swing phase than stance phase, which provides further evidence that segmental dynamics have a larger impact on axial forces during the swing phase than the stance phase of gait.

The flexion-extension torque curves of Figure 3.9 and the quasi-static and dynamic peak torque comparison shown in Figure 3.13 provide an indication that segmental dynamics are more evident during swing phase than stance phase. As stated in equation (2-20), there are three dynamic components of torque, arising from segment angular acceleration, gyroscopic effects and the dynamic component of force. While not a major focus of this thesis, an attempt to further investigate the dynamic components of torque is shown in Figure 4.5, in which the term accounting for the angular acceleration is plotted along with the sum of the three dynamic torque components during the swing and stance phase cycles. The angular acceleration term was usually the largest in magnitude of the three dynamic torque components. The dynamic component of torque, consisting of all three terms, was larger during swing phase, as shown in Table 3.10 and Figure 3.13.

Similar to the axial force analysis, after normalization the difference between using a quasi-static and a dynamic flexion-extension torque estimator was more evident during swing phase than stance phase, as shown in Figure 3.14. This suggests that the ground reaction dominates the flexion-extension torques during the stance phase of gait. From the swing phase curves of Figure 3.9, it was apparent that the segmental dynamics usually acted to increase the magnitude of the net flexion torque at all three lower extremity joints during swing phase. As illustrated in Figure 4.5, this effect may be attributed to segment angular acceleration effects for the knee and to a lesser degree for the ankle and hip.

An additional observation not seen in the axial force analysis, was that during stance phase flexion torques were more affected by segmental dynamics the further the joint was from the floor. This can be seen by comparing the differences between the quasi-static and dynamic torque curves for the ankle, knee and hip joints in Figure 3.9. For the ankle,

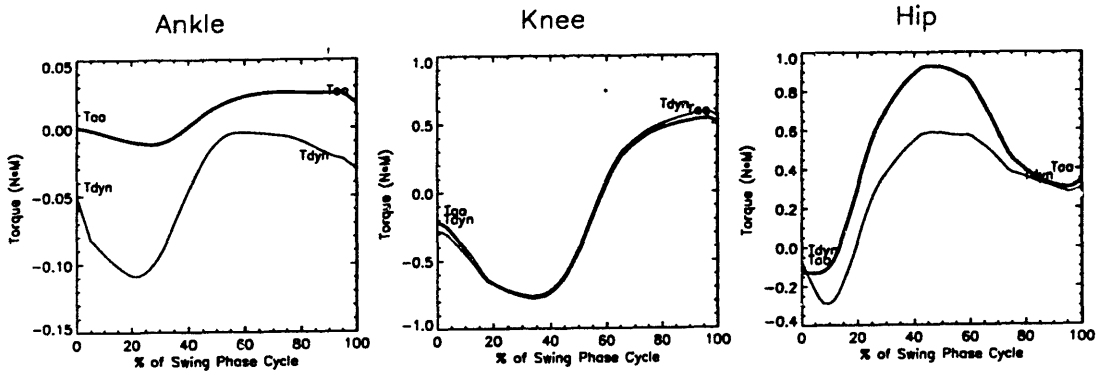

Figure 4.5.a Representative torque curves for the total dynamic component of torque and the component due to body segment angular acceleration during the swing phase of gait.

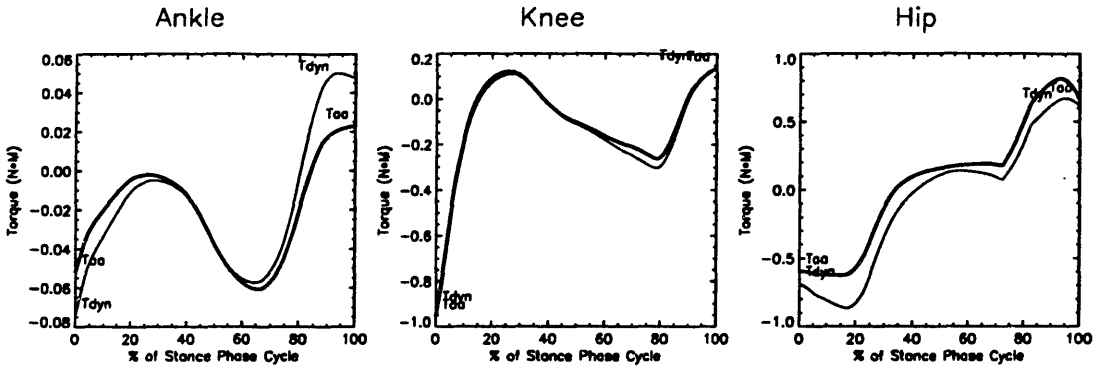

Figure 4.5.b Representative torque curves for the total dynamic component of torque and the component due to body segment angular acceleration during the stance phase of gait.

the quasi-static and dynamic curves were virtually identical, but this was less true for knee and less true still for the hip. As shown in Figure 4.3, this difference which arose from using the dynamic instead of a quasi-static torque estimator was mostly due to the segmental angular acceleration term of the dynamic torque. The increasing dynamic loads that accompany increasing height can be explained because not only do the superior body segments have more motion themselves, but they also have more body segments inferior to them, whose dynamic effects combine to produce a greater total dynamic contribution. This indicates that quasi-static methods of calculating torques become progressively more inaccurate, while proceeding in the rostral direction from the ankle to the hip. Tables 3.9 and 3.10 and Figure 3.15 show that, with the exception of the hip joint, the RMS errors were larger and the normalized RMS errors were much larger during swing phase than stance phase, which provides further evidence that segmental dynamics have a larger impact on flexion torques during the swing phase than the stance phase of gait.

Both the axial force and flexion-extension torque results suggest that the quasi-static assumption is valid for calculating joint loads during the stance phase of gait, with the possible exception of hip torques. However, when calculating joint forces and torques during the swing of gait, a dynamic model provides greater accuracy than a quasi-static model.

4.2.3 Representative Momentum Curves

Figure 3.16 shows representative linear momentum curves for the swing and stance leg segments during gait. It is apparent from these curves that the magnitude of the horizontal (x-direction) momentum is the largest component of linear momentum, which is understandable as the direction of gait was along the x-axis. Another interesting point to note is that although the swing leg was the right leg and the stance leg was the left leg, the end of the stance phase x-direction linear momentum curves fit together fairly well with the

beginning of the swing phase x-direction linear momentum curves, and vice versa. In other words, putting the swing and stance phase curves end to end, would yield a fairly continuous entire gait cycle.

The representative angular momentum curves for the swing and stance leg segments during gait are shown in Figure 3.17. From these curves, it is apparent that since the largest motions during gait occur in the sagittal plane, the largest components of segmental angular momentum are those about the z-axis (flexion-extension). The swing and stance phase angular momentum curves did not fit together as continously end to end as did the linear momentum curves. Because the same numerical differentiation routine was used for linear and angular position data, it is possible that segment orientation angles are not measured as accurately as segment position.

The representative whole body momentum curves of Figure 3.18 indicate that once again the horizontal (x-axis) linear momentum is the largest component of linear momentum. The whole body linear momentum curve could also have been obtained by multiplying the velocity trajectory of the center of mass by the mass of the body. This was used as a method of verifying the whole body linear momentum. Looking at the whole body angular momentum components, it is now evident that the z-axis (flexion-extension) momentum component is no longer the largest. This may be caused by cancelation of the opposing angular momentum effects of the swing and stance legs, for the following reason. The whole body momentum was calculated over a stance phase, however, while one leg is in stance phase the other is in swing phase. Looking at the curves of Figure 3.17, it is apparent that stance leg angular momentums are usually positive about the z-axis, whereas swing leg angular momentums are usually negative. Hence, when calculating the whole body angular momentum, it is possible that the swing and stance leg angular momentums about the z-axis cancel each other out to some degree. An additional topic, worthy of future study is the role of arm swing during gait, and its contributions to whole body momentum.

4.3 Lifting

The analysis of the lifting trials is limited to the forces and torques on the L2-L3 back joint. While not enough lifting trials were obtained to make conclusions with sufficient confidence, a brief interpretation of the limited back force and torque curves obtained in this study is included. Also, the large number of variable parameters involved in the lifting task prevented direct comparisons with results reported in other studies. These variable task parameters include the mass of the weight, handles vs. no handles, straight arms vs. bent arms, straight legs vs. bent legs, the height and distance from which the weight is picked up, the height and distance at which the weight is released (if released), planar vs. twisting motion and subject parameters, which include height, weight, age, and physical condition. Even for the same subject, the way in which he performed the lifting task changed significantly between testing dates. For example, the subject performed straight leg lifts quite well on one test date, but on another test date he noticeably bent his legs for all of his straight leg lifting trials. For this reason, only the lifting trials for one test date were studied.

4.3.1 Back Forces and Torques During Lifting

Several observations were made based on the representative curves for the three components of force and three components of torque acting on the back joint during lifting shown in Figure 3.19. The axial force and the flexion-extension torque were the largest components and are the only components discussed in this thesis.

While not shown, the back forces and torques were also calculated for these lifting trials with the weight artificially removed by the dynamic estimator computer program. By comparing these forces and torques to the forces and torques calculated with the 12.5

pound weight present in each hand, it was found that the back axial forces and flexion-extension torques were effectively increased by roughly %50 due to the 25 pound weight in the hands. As mentioned previously, the estimation of the increased load on the back due to the weight did not include the inertial characteristics of the weight itself, which might contribute significantly the load on the back.

From the representative force and torque curves of Figure 3.19 and the key points during the lifting trial shown in Figure 2.5, several interesting observations were made. First, the time at which the subject initially picked up the weight corresponded to the first of two flexion-extension torque peaks and the minimum axial force. This peak torque might be generated to begin the extension of the back to the upright position. The axial force might be a minimum since the bent over position causes a tensile force on the back, which substantially lowers the normal compressive axial force on the spine. The weight in the hands would seem to amplify the effect of lowering the compressive force on the spine. Another interesting point was the second peak on the torque curve, which occurred when the subject leaned over to place the weight on a shelf. This peak torque may be required to prevent the back from bending over due to the downward force of the weight on the ends of the outstretched arms. A final observation, which seems counterintuitive, was that the back axial force and flexion-extension torque were approximately the same regardless of whether the legs were straight or bent. It was expected that the back forces and torques would be higher for lifting with straight legs than with legs bent, however, it is possible that studying more trials by more subjects might provide different results than those presented here.

In order to obtain more accurate results, several improvements should be made to the model. First of all the dynamics of the weight itself are probably not negligible and could be determined by mounting an irLED array on the weight. By knowing the position of the weight as determined by irLED, the weight linear and angular velocity and acceleration could be determined, which could improve the model accuracy by including the

effects of the dynamic force and torque components. Another improvement would be to model the arm as two segments instead of one. By taking into account motions at the elbows, the effect of the arms bending while lifting could be included. This seems particularly appropriate since lifting of a weight almost always involves bending the arms to some extent.

4.4 Three-Link Chair-Rise Simulations

A brief discussion regarding the validity of the inertial parameters used for the three-link model is presented, followed by a discussion of the differences between torques calculated with the MGH dynamic estimator and simulated torques.

4.4.1 Inertial Parameters

As expected, the summation of the link masses was equivalent to the total weight of the subject minus the weight of the subject's feet. The summation of the link lengths was slightly less than the subject's standing height, but this was accounted for since the distance between the subject's ankle and the ground was not included in the three-link model. Other parameters, such as the moments of inertia and the center of gravity locations, had no direct method of verification. All of the link parameters were comparable to those found by another study which used a similar three-link model [97]. However, they could not be directly compared because this study used anthropometric data from a 58 Kg, 150 cm female subject, and the other study used anthropometric data from a 66 Kg, 143 cm male subject. In addition, this study included the pelvis in link two, while the other study included the pelvis in link three.

4.4.2 Simulated Torques and the MGH Model Torques

As shown in Figures 3.20 and 3.21 and Table 3.12, the MGH model torques and the simulated torques showed fairly good agreement for joint three, the back joint. The overall shapes of the curves were similar and the magnitudes were also reasonably close, although the magnitudes of the simulated torques were slightly larger for both trials.

The agreement between the MGH model knee joint torques and the simulated joint two torques was not quite as good as that between the back joint and joint three. The overall shapes of the curves were similar, however, the magnitudes of the simulated torques were noticeably smaller for both trials. At the beginning of the trial and prior to liftoff, there was a discrepancy due to the artificially imposed condition in the simulation which set the joint two torque equal to zero until liftoff, as mentioned earlier.

The magnitudes of the peak ankle joint torques of the MGH model and the simulated joint one torques were quite close, however, the shapes of the curves were not as closely related. Again, there was an initial discrepancy due to the artificially imposed condition in the simulation which set the joint one torque equal to zero prior to liftoff, as previously mentioned. In both trials, the simulated torques had roughly the same shape and magnitude as the MGH model torques after liftoff and remained fairly close up to the peak magnitude. Once the standing position was achieved, however, the agreement diverged as the simulated joint one torque decreased much more than the ankle joint torque calculated with the MGH model.

The two largest discrepancies between the MGH model and the three-link simulation were the difference in peak torque magnitude at the knee joint, and the difference in ankle joint torques near the end of the trial, once the standing position had been achieved. While no direct explanations for these observed phenomena were determined, it is apparent that contributing factors are associated with fundamental differences between

the two models. These differences include the number and location of body joints and segments. For example, the lack of a foot segment, the combination of the thighs and pelvis, and the combination of the upper body segments most likely contributed to differences in joint torques between the two models.

4.5 Model Validation

The back forces and torques calculated from the lower body kinematics and force plate data were compared to the back forces and torques calculated from the upper body kinematics for five standing and five chair-rise trials, as shown in Table 3.12. In general, the differences as determined by the root mean square error between the upper body and lower body curves were highly variable for different subjects. This suggests that there were not any fundamental problems with the model, but rather the inertial parameters of each subject deviated with those based on the anthropometric studies according to the subject's individual body segment characteristics. The data of Table 3.12 support this idea, in that some subjects had small RMS errors for certain force and torque components, and larger RMS errors for other components of force and torque, whereas the small and large RMS errors for other subjects occurred for different force and torque components. For example, the chair-rise RMS error was largest for the axial force and smaller for the lateral and anterior-posterior forces for subject three, whereas for subject four, the RMS error was smallest for the axial force and larger for the anterior-posterior and lateral forces. Similarly for the standing trials, the flexion torque RMS error was the largest and the abduction RMS error was the smallest for subject three, whereas the opposite was true for subject six.

Looking at the average RMS errors for all five subjects, it is apparent that certain force and torque components usually had larger RMS errors than others. For example, for chair-rise the axial force and flexion torque RMS errors were the largest with one exception. The reason for this could be that these were the largest components force and

torque, as can be seen in Figures 3.1 and 3.2, and consequently the corresponding errors were also of larger magnitude. If percent difference had been used to quantify the difference between upper body and lower body back forces and torques, the axial force and flexion torque curves, being of largest magnitude would probably have had the smallest percent difference. Instead, the RMS error gives a measure of the average error in units of (%BW) for force and (%BW*M) for torque, which is more physiologically meaningful and allows for a normalized comparison between subjects.

The representative upper body and lower body back force and torque curves in Figures 3.23 and 3.24, show graphically the information of subject 4 presented in Table 3.12. For most curves the agreement is fairly good, with the general shapes of the curves being similar. One assessment of the model parameters can be made directly from the axial forces from the upper and lower bodies during standing. The axial force at the back joint is different when determined from the upper and lower bodies, which implies that the body mass is incorrectly distributed among the body segments for this particular subject. While the differences in the other curves do not lend themselves to such direct interpretations, it is possible that when the upper and lower body back forces are in close agreement, but the back torques are not, the model might provide inaccurate estimates for the inertial tensors for the subject in question.

A comparison of the body segment inertial parameters calculated with the MGH model and those calculated by other investigators was performed by Grierson [88]. The variations in the inertial parameters predicted by different studies were large enough that they would affect joint force and torque calculations. Hence, this provides evidence for the widely accepted notion that anthropometric models are rarely able to predict inertial parameters of individual subjects with a high degree of accuracy.

In a similar comparison of back joint forces and torques calculated from both the upper and lower body kinematics, MacKinnon found differences of greater than 20

newton-meters for the L4/L5 joint torques. He concluded that using the inverse solution from the head downward would minimize the transfer of modelling errors [101].

In general the errors, between upper and lower body back forces and torques suggest limitations of the model. As just described, one of these limitations is the ability of the model to accurately predict inertial parameters of individual subjects with the anthropometric regression equations. Another limitation is the rigid body assumption, because in actuality, the soft tissue of a body segment shifts during movement and muscle contraction, thereby changing inertial properties such as the center of mass and the inertial tensor parameters. Modeling the spine as a single joint at the L2-L3 level may also lead to inaccurate back force and torque calculations.

Chapter 5

Conclusions and Recommendations for Further Study

5.1 Conclusions

A dynamic estimator was developed for use in conjunction with the MGH eleven segment, three dimensional, 66 degree of freedom human body model. The applications of the dynamic estimator to the activities of chair-rise and the swing and stance phases of gait indicate that there are portions of these activities which are well suited to biomechanical analysis with a dynamic model rather than a quasi-static model. In particular, the contributions of segmental dynamics to joint forces and torques during both chair-rise and gait become progressively larger while proceeding in the rostral direction from the ankle joint upward. However, the implementation of a dynamic model is only an improvement over a quasi-static model if particular care is taken to obtain meaningful estimates for segment velocities and accelerations, with the effects of noise sufficiently controlled.

During chair-rise static loads are larger than dynamic loads. Quantification of segmental dynamics determined that quasi-static ankle forces and torques are valid, however, hip, back and neck forces and torques should be calculated using a model that includes the effects of segmental dynamics. Some preliminary results also suggest that as the speed of ascent is increased, the dynamic components of force and torque become larger in relation to the total force and torque. While many studies on chair-rise have been performed using quasi-static analysis, the results presented in this thesis suggest that a

dynamic model should be used to obtain more accurate joint force and torque estimates, especially for the hip, back and neck joints and at fast ascent speeds.

During the stance phase of gait, the net axial forces and flexion-extension torques are dominated by the ground reaction force, however segmental dynamics do significantly affect the stance phase hip flexion-extension torque. During swing phase of gait, segmental dynamics are more evident than during stance phase. When calculating the net forces and torques during swing phase a dynamic model should be chosen, with careful attention given to calculating meaningful estimates for segment angular and linear acceleration.

The comparison of the flexion-extension joint torques during chair-rise calculated with the MGH model and those simulated with the three-link model demonstrate that an eleven segment Newton-Euler inverse dynamic model and a three segment Lagrangian inverse dynamic model can both be used to calculate joint torques during chair-rise. However, the torque differences between these two methods and models, may be attributed to effects of oversimplification that occur when modeling the human body with only three segments.

When normalized, the whole body linear momentum, the whole body angular momentum about the average ankle joint, and the whole body angular momentum about the body center of mass were consistent between normal subjects when rising from a chair. The whole body horizontal linear momentum peaks first, and is then followed by the peak vertical whole body linear momentum. The peak whole body angular momentum about both the average ankle joint and the center mass peak close to the time of the peak horizontal linear momentum. Joint torques may play a role in transferring horizontal linear momentum and angular momentum to vertical linear momentum during chair-rise.

5.2 Recommendations for Further Study

Verification of the model revealed that the model was fairly accurate, however, the degree to which it predicted segment inertial parameters of individual subjects was highly variable. As a future project, the anthropometric regression equations should be adapted to allow tailoring to individual subjects. For example, an optimization routine could be used to systematically adjust the segment inertial properties in an iterative process that would minimize the discrepancies between back forces and torques calculated with upper body kinematics and those calculated with lower body kinematics and force plate data. In a further attempt to verify inertial parameters, computerized tomography scanning could be used to investigate the assumptions of constant segment density and segment symmetry.

Other means of model verification also need to be performed. While the Lagrangian differentiation scheme used in this study was verified with another numerical differentiation scheme, further verification of the differentiated position trajectories should be performed with accelerometers as described by Ladin [8]. If a sensitivity analysis were performed which investigated the effects of artificially moving joint centers, as suggested by Ramakrishnan [102], some insight would be gained regarding the magnitudes of joint force and torque errors due to slightly displaced joint centers.

Experimentally, further study of more subjects is needed to verify that the dynamic components of force and torque increase relative to the total force and torque when the speed of ascent during chair-rise is increased. Also, the change in total force and torque should be calculated as well as the change in dynamic force and torque. By doing this calculation, it would be possible to determine the portion of the change in force or torque which is due to segmental dynamics and the portion which is due to changes in the ground reaction force, or other quasi-static components.

In addition further study of whole body linear and angular momentum during chair-rise might relate the inability of some people to rise from a chair to inadequate generation of whole body momentum, unsuccessful transfer of horizontal linear momentum to vertical linear momentum, or the inability to control whole body momentum once it is generated. Specifically, the role of joint torques as mechanisms to transfer horizontal whole body linear momentum to vertical whole body linear momentum during chair-rise should be investigated. The role of whole body angular momentum during chair-rise should also be studied, as it may also contribute to vertical whole body linear momentum.

Bibliography

[1] W. Weber and E. Weber. *Mechanik der menschlichen Gerwerkzeuge* (Mechanics of Human Locomotion). Gottingen, Germany 1836.

[2] E.J. Marey. De la locomotion terrestre chez les bipedes et les quadrupedes (Terrestrial Locomotion of Bipeds and Quadrupeds). *J. de l'Anat. et de la Physiol.*, 9: 42-80, 1873.

[3] E.J. Marey and G. Masson. *La methode graphique dans les sciences experimentales* (The Graphic Method in Experimental Sciences). Paris, France, 1885.

[4] E.J. Marey and G. Demeny. Estudes experimentales de la locomotion humaine (Experimental Studies on Human Locomotion). *Comptes Rendus Acad. des Sciences*, 105: 544-552, 1887.

[5] E.J. Marey. *Movement.* D. Appleton Co., New York, NY, 1895.

[6] W. Braune and O. Fischer. *The Human Gait* (Translated by P. Maquet and R. Furlong). Springer, Berlin, New York, Hiedelberg, 1987.

[7] B. Bresler and J.P. Frankel. The forces and moments in the leg during level walking. *ASME Transactions*, 27-36, 1950.

[8] Z. Ladin, W.C. Flowers and W. Messner. A quantitative comparison of a position measurement system and accelerometry. *J. Biomechanics*, 22: 295-308, 1989.

[9] M.R. Yeadon. The simulation of aerial movement - II. A mathematical inertia model of the human body. *J. Biomechanics*, 23(1): 67-74, 1990.

[10] A.B. Schultz, N.B. Alexander and J.A. Ashton-Miller. Biomechanical analyses of rising from a chair. *J. Biomechanics*, 25(12): 1383-1391, 1992.

[11] E. Harless. The static movements of human limbs. *Treatises of the Math-Physics Class of the Royal Academy of Science of Bavaria*, 69-96, 257-294, 1860.

[12] C. W. Braun and O. Fischer. The center of gravity of the human body as related to the equipment of the german infantryman.*Treatises of the Math-Physics Class of the Royal Academy of Science of Saxony*, 26, 1889.

[13] W.T. Dempster. Space requirements for the seated operator. Tech. Report 55-159, AD No. 87892, Wright-Patterson Air Force Base, 1955.

[14] C.E. Clauser, J.T. McConville, and J.W. Young. Weight, volume and center of mass of segments of the human body. AMRL Technical Report 69-70, Wright-Patterson Air Force Base, 1969.

[15] R.F. Chandler, C.E. Clauser, J.T. McConville, and J.W. Young. Investigation of the inertial properties of the human body. AMRL Technical Report TR-74-137, Wright-Patterson Air Force Base, 1975.

[16] N.A. Bernstein, O.A. Salzgeber, P.P Pavelenko, and N.A. Gurvich. Determination of location of the centers of gravity and mass of the limbs of the human body. *All-Union Institute of Experimental Medicine*, 1936.

[17] R.K. Jensen. Body segment mass, radius and radius of gyration proportions of children. *J. Biomechanics*, 19(5): 359-368, 1986.

[18] J.T. McConville, C.E. Clauser, T.D. Churchill, J. Cuzzi and I. Kaleps. Anthropometric relationships of body and body segments of inertia. Technical Report, Air Force Aerospace Lab, 1980.

[19] J.W. Young, R.F. Chandler, L.L. Snow, K.M. Robinette, G.F. Zehner and M.S. Loftberg. Anthropometric and mass distribution characteristics of the adult female. Technical Report, FAA Civil Aeromedical Institute, 1983.

[20] J. Amar. *Le Moteur Humain*. Paris, 1914.

[21] A. Macovski. *Medical Imaging Systems*. Prentice-Hall, Inc., Englewood Cliffs, NJ, 1983.

[22] L. Sjostrom. A computer-tomography based multicompartment body composition technique and anthropometric predictions of lean body mass, total and subcutaneous adipose tissue. *Int. J. Obes.*, 15: 19-30, 1991.

[23] H.K. Huang and S.C. Wu. The evaluation of mass densities of the human body *in vivo* for CT scans. *Computational biol. Med.*, 6: 337-343, 1976.

[24] H.J. Woltring. *On the use of nuclear magnetic resonance imaging (NMR) for biomechanical parameter acquisition in functional movement studies*. Paper presented at the 1984 Olympics Scientific Congress, Eugene, OR.

[25] M. Mungiole and P.E. Martin. Estimating segment inertial properties: magnetic resonance imaging versus existing methods. In *Proceedings of the North American Congress on Biomechanics*, Montreal, Canada, 1986.

[26] G.A. Brown. *Determination of body segment parameters using computerized tomography and magnetic resonance imaging*. Master's thesis, MIT Press, Cambridge, MA, 1987.

[27] A. Godfredson, J. Gorg, C. Christiansen, and R.B. Mazess. Total body bone mineral content/em *in vivo* by dual photon absorptiometry ii. accuracy. *Clin. Physiology*, 4: 357-362, 1984.

[28] S.B. Heymsfield, J. Wang, S. Heshka, J.J. Keyayias, and R.N. Peirson. Dual-photon absorptiometry: comparison of bone mineral and soft tissue mass measurements *in vivo* with established methods. *Am. J. Clin. Nutr.*, 49: 1283-1289, 1989.

[29] M.R. Yeadon and M. Morlock. The appropriate use of regression equation for the estimation of segment inertia parameters. *J. Biomechanics*, 22(6/7): 683-689, 1989.

[30] R.A. Berger, P.O. Riley, R.W. Mann and W.A. Hodge. Knee kinematics and kinetics during locomotor activities of daily living. *Physical Therapy*, 73: 229-242, 1993.

[31] R.G. Burdett, R. Habasevich, J. Piscotta and S.R. Simon. Biomechanical comparison of rising from two types of chairs. *Physical Therapy* , 65: 1177-1183, 1985.

[32] M.W. Rodosky, A.P. Andriacchi and B.J.A. Gunnar. The influence of chair height on lower limb mechanics during rising. *J. Orthop. Res.*, 7: 266-271, 1989.

[33] J. Wheeler, C. Woodward, R.L. Ucovich, J. Perry and J.M. Walker. Rising from a chair; influence of age and chair design. *Physical Therapy* , 65: 22-26, 1985.

[34] S.J. Fleckenstein, R.L. Kirby and D.A. MacLeod, Effect of limited knee-flexion range on peak hip moments of force while transferring from sitting to standing. *J. Biomechanics*, 21: 915-918, 1988.

[35] N. Alexander, A. Schultz, D. Warwick and J. Ashton-Miller. Rising from a chair; performance biomechanics of healthy elderly and young adult subjects. *Symp. on Biomechanics ASME Annual Meeting*, AMD-89: 333-336, 1989.

[36] U.P. Arborelius, P. Wretenberg and F. Lindberg. The effects of armrests and high seat heights on lower-limb joint load and muscular activity during sitting and rising. *Ergonomics*, 35(11): 1377-1391, 1992.

[37] M. Miller, A. Schultz, N. Alexander, D. Warwick and J. Ashton-Miller. Dynamics of rising from a chair: experimental data collection. *ASME 1989 Biomechanics Symposium* (Edited by P.A.Torzilli and M.H. Friedman), 98: 329-332, 1989.

[38] B.B. Seedhom and K. Terayama. Knee forces during the activity of getting out of a chair with and without the aid of the arms," *Biomedical Engineering*, 8: 278-282, 1976.

[39] Y.C. Pai and M.W. Rogers. Dependence of weight transfer patterns on the speed of ascent during sit-to-stand. *ASB A. Mtg. Proc.*, 79-80, 1988.

[40] Y.C. Pai. Segmental contributions to total body motion in sit-to-stand. *Abstracts XXII Congr. Int. Soc. Biomechanics* (Edited by R.J. Gregor and R.F. Zernicke) *J. Biomechanics*, 22(10): 1105, 1989.

[41] Y.C. Pai and M.W. Rogers. Speed variation and resultant joint torques during sit-to-stand. *Archives of Physical Medicine and Rehabilitation*, 72: 881-885, 1991.

[42] S. Nuzik, R. Lamb, A. VanSant and S. Hirt. Sit-to-stand movement pattern; a kinematic study. *Physical Therapy*, 66: 1708-1713.

[43] A. Kralj, R.J. Jaeger and M. Munih. Analysis of standing up and sitting down in humans: definitions and normative data presentation. *J. Biomechanics*, 23(11): 1123-1138, 1990.

[44] E.B. Hutchinson and P.O. Riley. Dynamic analysis of rising from a chair. *Proceedings of the 15th Annual IEEE/EMBS Conference*, San Diego, CA, 1097-1098, 1993.

[45] E.R. Ikeda, M.L. Schenkman, P.O. Riley, W.A. Hodge. Influence of age on dynamics of rising from a chair. *Physical Therapy*, 71: 473-481, 1991

[46] P.O. Riley, M.L. Schenkman, R.W. Mann and W.A. Hodge. Mechanics of a constrained chair-rise. *J. Biomechanics*, 24(1): 77-85, 1991.

[47] M.L. Schenkman, R.A. Berger, P.O. Riley, R.W. Mann, W.A. Hodge. Whole body movements during rising to standing from sitting. *Physical Therapy*, 70(10): 638-651, 1990.

[48] P.R. Cavanagh and R.J. Gregor. Knee joint torque during the swing phase of normal treadmill walking. *J. Biomechanics*, 8: 337-344, 1975.

[49] D.R.W. May. The development of a comprehensive external prosthetic knee control. *Conference on Human Locomotor Engineering*, 179-198, 1971 Inst. Mech. Engnrs., London.

[50] J.P. Paul. The effect of walking speed on the force actions transmitted at the hip and knee joint. *Proc. R. Soc. Med,* 63: 200, 1970.

[51] A.G. Patriarco, R.W. Mann, S.R. Simon and J.M. Mansour. An Evaluation of the Approaches of Optimization Models in the Prediction of Muscle Forces During Human Gait. *J. Biomechanics*, 14(8): 513-525, 1981.

[52] D.A. Winter. Biomechanics and Motor Control of Human Movement, 2nd Edit. John Wiley and Sons, 35-52, 1990.

[53] D.A. Winter. Overall principle of lower limb support during stance phase of gait. *J. Biomechanics,* 13: 923-927, 1980.

[54] P.R. Cavanagh, M. L. Pollock and J. Landa. A biomechanical comparison of elite and good distance runners. *Ann. N.Y. Acad. Sci.,* 301: 328-345, 1977.

[55] A.E. Chapman and G.E. Caldwell. Factors determining changes in lower limb energy during swing in treadmill running. *J. Biomechanics*, 16(1): 69-77, 1983.

[56] C.J. Dillman. A kinetic analysis of the recovery leg during sprint running. *Selected Topics on Biomechanics*, (Edited by J.M. Cooper), Athletic Institute, Chicago.,137-165, 1971.

[57] S.J. Phillips, and E.M. Roberts. Muscular and non-muscular moments of force in the swing limb of masters runners. *Biomechanics Symposium Proceedings*, (Edited by J.M. Cooper, and B. Haven), Indiana State Board of Health, Bloomington, IN, 256-274, 1980.

[58] F.L. Buczek and P.R. Cavanagh. Stance phase knee and ankle kinematics and kinetics during level and downhill running. *Med. and Sci. in Sports and Exercise*, 22(5): 669-677, 1990.

[59] R.V. Mann. A kinetic analysis of sprinting. *Med. Sci. Sports Exercise* , 13(5): 325-328, 1981.

[60] R. V. Mann and P. Sprague. A kinetic analysis of the ground leg during sprint running. *Res. Q. Exercise Sport.*, 51: 334-348, 1980.

[61] D.G.E. Robertson. Functions of the leg muscles in running. *Biomechanics X-B*, (Edited by B. Jonsson), Human Kinetics Publishers, Champaign, Illinois, 1021-1028, 1987.

[62] S.H. Scott and D.A. Winter. Internal forces at chronic running injury sites. *Med. and Sci. in Sports and Exercise*, 22(3): 357-369, 1990.

[63] D.A. Winter. Moments of force and mechanical power in jogging. *J. Biomechanics.*, 16(1): 91-97, 1983.

[64] Impairments due to injury; United States-1971. Public Health Service, DHEW Publication No. (HRA) 74-1514, Rockville, MD, 1973.

[65] G.B.J. Andersson. Epidemiological aspects on low-back pain in industry. *Spine*, 6:53-60, 1981.

[66] D. B. Chaffin and K.S. Park. A longitudinal study of low back pain as associated with occupational weight lifting factors. *Am.Ind.Hyg.Assoc.J.*, 34: 513, 1973.

[67] A.L. Nachemson. Low back pain: its etiology and treatment. *Clinical Medicine*, 1: 18, 1971.

[68] D. Gagnon and M. Gagnon. The influence of dynamic factors on triaxial net muscular moments at the L5/S1 joint during asymmetrical lifting and lowering. *J. Biomechanics*, 25(8): 891-901, 1992.

[69] C. Bush-Joseph, O. Schipplein, G.B.J. Andersson and T.P. Andriacchi. Influence of dynamic factors on the lumbar spine moment in lifting. *Ergonomics*, 31: 211-216, 1988.

[70] N.D. Kishino, T.G. Mayer, R.J. Gatchel, M. McRate Parrish, C. Anderson, L. Gustin and V. Mooney. Quantification of lumbar function. Part 4: isometric and isokinetic lifting simulation in normal subjects and in low-back dysfunction patients. *Spine*, 10: 921-927, 1985.

[71] S. Kumar, D.B. Chaffin and M.Redfern. Isometric and isokinetic back and arm lifting strengths: device and measurement. *J.Biomechanics*, 21: 35-44, 1988.

[72] J. Sandover. Dynamic loading as possible source of low-back disorders. *Spine*, 8: 652-658, 1983.

[73] J.E. Smeathers and D.N. Joanes. Dynamic compressive properties of human lumbar intervertebral joints: a comparison between fresh and thawed specimens. *J. Biomechanics*, 21: 425-433, 1988.

[74] W. S. Marras and G.A. Mirka. Muscle activities during asymmetric trunk angular accelerations. *J. Orthopaedic Research*, 8(6): 824-832, 1990.

[75] M. Parnianpour, M. Campello and A. Sheikhzadeh. Effect of posture on triaxial trunk strength in different directions.Its biomechanical consideration with respect to incidence of low-back problem in construction industry. *J. Industrial Ergonomics*, 8(3): 279-287, 1991.

[76] O.O. Schipplein, J.H. Trafimow, G.B.J. Andersson and T.P. Andriacchi. Relationship between moments at the L5/S1 level, hip and knee joint when lifting. *J. Biomechanics*, 23(9): 907-912, 1990.

[77] J.M. Stevenson, J.T. Bryant, D.R. Greenhorn, S.L. French and G.M. Andrew. Dynamic analysis of isoinertial lifting technique. *Abstracts XXII Congr. Int. Soc. Biomechanics* (Edited by R.J. Gregor and R.F. Zernicke) *J. Biomechanics*, 22(10): 1086, 1989.

[78] L.C. Masse and M. Lamontagne. Kinematic analysis of wheelchair propulsion for three speeds of propulsion. *Abstracts XXII Congr. Int. Soc. Biomechanics* (Edited by R.J. Gregor and R.F. Zernicke) *J. Biomechanics*, 22(10): 1052, 1989.

[79] L.C. Masse, M. Lamontagne and M.D. O'Rain. Biomechanical analysis of wheelchair propulsion for various seating positions. *J. Rehab. Res. & Dev.*, 29(3): 12-28, 1992.

[80] M. Gagnon and G. Smyth. Biomechanical exploration on dynamic modes of lifting. *Ergonomics*, 35(3): 329-345, 1992.

[81] M.A. Ragheb. Braking horizontal momentum. A method for developing leg power. *Abstracts XXII Congr. Int. Soc. Biomechanics* (Edited by R.J. Gregor and R.F. Zernicke) *J. Biomechanics*, 22(10): 1072, 1989.

[82] R.N. Hinrichs, B.A. Munkasy and S.A. Chinworth. Analysis of angular momentum during the run-up and takeoff in long jumping. *Abstracts XXII Congr. Int. Soc.*

Biomechanics (Edited by R.J. Gregor and R.F. Zernicke) *J. Biomechanics*, 22(10): 1023, 1989.

[83] M. Figgen. Computer simulation of the two-dimensional flight phase of an athlete. *Abstracts XXII Congr. Int. Soc. Biomechanics* (Edited by R.J. Gregor and R.F. Zernicke) *J. Biomechanics*, 22(10): 1010, 1989.

[84] M.R. Yeadon. The simulation of aerial movement - III. The determination of the angular momentum of the human body. *J. Biomechanics*, 23(1): 75-83, 1990.

[85] Y.C. Pai, C.A. Lim and J.L. Lewis. Deviations in dynamic weight transfer of elderly fallers. *1990 Advances in Bioengineering*, New York: The American Society of Mechanical Engineers, 77-79, 1990.

[86] E. Antonsson and R. Mann. Automatic 6-D.o.F. kinematic trajectory acquisition and analysis. *ASME J. Dyn. Syst. Meas. Con.*, 111: 31-39, 1989.

[87] P.O. Riley, R.W. Mann and W.A. Hodge. Modeling of the biomechanics of posture and balance. *J. Biomechanics*, 23(5): 503-506, 1990.

[88] Y.M. Grierson. *Creation of a Three-Dimensional Human Body Model for Dynamic Analysis*. Master's thesis, MIT Press, Cambridge, MA, 1992.

[89] F.B. Hildebrand. *Introduction to Numerical Analysis*. McGraw-Hill, 1956.

[90] M.D. Lesh, J.M. Mansour and S.R. Simon. A gait analysis subsystem for smoothing and differentiation of Human Motion Data. *J. Biomechanical Engineering*, 101: 205-212, 1979.

[91] D.E. Krebs, D. Wong, D. Jevsevar, P.O. Riley and W.A. Hodge. Trunk Kinematics During Locomotor Activities. *Physical Therapy* , 72(7): 505-514, 1992.

[92] C.L. Golliday Jr. and H. Hemami. Postural stability of the two-degree-of-freedom biped by general linear feedback. *Proc. Joint Automat. Contr. Conf.*, Austin, TX. 599-608, 1974.

[93] H. Hemami and V.S. Cvetkovic. Postural Stability of two biped models via Liapunov second method. *IEEE Trans. Automat. Contr.*, AC-22: 66-70, 1977.

[94] J.G.M. Quintern and R.J. Jaeger. Biomechanics of quiet stance. *Proceedings. of the 9th Annual Conf. of the IEEE /EMBS*, Chicago, IL. 1992-1993, 1987.

[95] J.F. Wilson. Dynamics of the human legs subjected to posterior impact. *AMSE J. Dynamic Sys., Meas. and Control*, 113(1): 127-133, 1991.

[96] H. Kallel, H. Hemami and S. Simon. Postural Stability of constrained three dimensional robotic systems. *Proc. of the IEEE Int. Conf. on Rob. and Aut.*, Cincinnati, OH, 2120-2125, 1990.

[97] H. Hemami and B. Wyman. Modeling and control of constrained dynamic systems with application to biped locomotion in the frontal plane. *IEEE Trans. on Automat. Contr.*, 24: 526-535, 1979.

[98] Y. Hurmuzlu. Effect of stance ankle torques on the nonlinear stability of a two degree of freedom, three element bipedal locomotion system. *Modeling and Contr. Issues in Biomech. Sys.*, Winter Annual Meeting of the ASME, Dallas, TX. 73-88, 1988.

[99] H. Hemami. A measurement oriented formulation of the dynamics of natural and robotic systems. *ASME J. of Dynamic Sys., Meas., and Contr.*, 113: 401-408, 1991.

[100] M.P. deLooze, I. Kingma, J.B.J. Bussmann and H.M. Toussaint. Validation of a dynamic linked segment model to calculate joint moments in lifting. *Clinical Biomechanics*, 7(3): 161-169, 1992.

[101] C.D. MacKinnon and D.A. Winter. Control of whole body balance in the frontal plane during human walking. *J.Biomechanics*, 26(6): 633-644, 1993.

[102] H.K. Ramakrishnan and M.P. Kadaba. On the estimation of joint kinematics during gait. *J. Biomechanics*, 24(10): 969-977, 1991.

[103] H. Asada and J.-J.E. Slotine. *Robot Analysis and Control*. John Wiley and Sons, New York, 1986.

Appendix A: C Computer Programs

A.1 Description of Subroutines

Subroutine	Function
TORQUEWHOLEBODY()	Calculates net joint forces and torques
gmprd1([A],b,c)	Multiplies a vector by a rotation matrix [A]*b = c
gmtra(A,B)	[B] = Transpose[A]
crossp(a,b,c)	Takes cross product of two vectors c = a x b
gmprd3([A],[A],[C])	Multiplies two 3x3 matrices [A][B] = [C]
ANGVELACCEL()	Calculates angular velocity, acceleration, momentum
dataread()	Reads in angular position data
angvelaccelsub()	Calculates angular velocity and acceleration
angseg()	Rotates an array from global to segment coordinates
momseg()	Calculates angular momentum - segment coordinates
angglob()	Rotates an array from segment to global coordinates
datawr()	Writes angular output files (.ang, .av, .aa, .am)
free_matrix()	Initializes matrices

Subroutine	Function
LINVELACCEL()	Calculates linear velocity, acceleration, momentum
datard()	Reads linear position data
cgfind()	Locates segment and whole body centers of gravity
cpfind()	Locates the center of pressure
cgvel()	Calculates the center of gravity velocity
delta()	Calculates the change in position w.r.t. 1st data point
cgmom()	Calculates linear segment momentum - global coords
datawr	Writes linear output files (.pos, .lv, .la, .lm)

A.2 Segment Linear and Angular Velocity, Acceleration and Momentum C Code

```
/* Program ANGVELACCEL.C    Erin Hutchinson
This program calculates the segmental angular velocity and acceleration*/

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include  "t4i_global.h"

#define DEBUG
#define BAD -1000

extern float **matrix(int nrl,int nrh,int ncl,int nch);
extern void dataread(int nsegr, int nsegl, int nframe, float ****sgrtrt, float ****sgrtlf, FILE *finr, FILE
*finl, float ***data);
extern void angvelaccelsub(int nframe, int nseg, int freq,  float ***angveldata, float ***angacceldata);
extern void angseg(int nseg,int nsegr,int nsegl, int sf, int lf, float ****sgrtrt, float ****sgrtlf, float
***seg_rotr,float ***seg_rotl, float ***data);
extern void momseg(int nsegr,int nsegl, int sf, int lf, float **moi_r, float **moi_l, float ***angmomdata);
extern void angglob(int nseg, int nsegr, int nsegl, int sf, int lf, float ****sgrtrt, float ****sgrtlf, float
***seg_rotr, float ***seg_rotl, float ***angmomdata);
extern int datawr(char namein_r[20], int nseg,int nframe,int freq, int sf, int lf, char *exten,  float
***data);

        FILE *finr;
        FILE *finl;

angvelaccel(char *namein, float ***angdata,float ***angveldata, float ***angacceldata, float
***angmomdata) {
        char  namein_r[20],  namein_l[20], namein_f[20];
        char *ext[3];
        char *exten;
        int i,j,k,ii,jj,kk;                     /*Counters*/
        int frame,seg,axis;                     /*Counters*/
        int nseg,nsegr,nsegl;                   /*Number of segments*/
        int nframe, freq;
        int sf, lf, sf2, lf2, ifg, lfg;
```

```
        int *good;
        float **moi_r, **moi_l;   /*Moments of inertia*/
        float ***seg_rotr, ***seg_rotl;      /*array to segment rotations*/
        float ****sgrtrt, ****sgrtlf;        /*global to array rotations*/
        float ***posdata;
        float a,arr[36];
        float angveldatatemp[15][3][1200], angacceldatatemp[15][3][1200],
angmomdatatemp[15][3][1200];
        t4header *rfile_header;
        t4header *lfile_header;

/*Memory Allocation for variables*/
        good = calloc(2,sizeof(int));
        sgrtrt = (float ****)malloc((unsigned) 3*sizeof(float ***))-1;
        for(i=1;i<=3;i++) {
                sgrtrt[i] =  (float ***)malloc((unsigned) 3*sizeof(float **))-1;
                for(j=1;j<=3;j++) {
                        sgrtrt[i][j] = matrix(1,7,1,1071);
                }
        }
        sgrtlf = (float ****)malloc((unsigned) 3*sizeof(float ***))-1;
        for(i=1;i<=3;i++) {
                sgrtlf[i] =  (float ***)malloc((unsigned) 3*sizeof(float **))-1;
                for(j=1;j<=3;j++) {
                        sgrtlf[i][j] = matrix(1,7,1,1071);
                }
        }

        rfile_header = (t4header *)calloc(1,sizeof(t4header));
        lfile_header = (t4header *)calloc(1,sizeof(t4header));

/* Assign file names */
        ext[0] = ".dtr";
        ext[1] = ".dtl";            }

        namein[6] = NULL;

        strcpy(namein_r,namein,6);
        namein_r[7] = NULL;
        strcat(namein_r,ext[0]);

        strcpy(namein_l,namein,6);
        namein_l[7] = NULL;
        strcat(namein_l,ext[1]);

/*Open the right side data file and read the header */
        if (( finl = fopen(namein_l, "r")) == NULL) {
                (void) printf("ErrorAL opening input file :%s.\n", namein_l);
                return(0);
        }
        fread(lfile_header, sizeof(t4header),1,finl);
        nsegl = lfile_header ->nb_segment;
        if (( finr = fopen(namein_r, "r")) == NULL) {
                (void) printf("ErrorAR opening input file :%s.\n", namein_r);
```

```
                return(0);
        }
        fread(rfile_header, sizeof(t4header),1,finr);

/*Setting subject parameters*/
        nframe =  rfile_header->nb_frames;
        freq = rfile_header->frequency;
        nsegr = rfile_header ->nb_segment;
        nseg = nsegr + nsegl;

/*Read in data*/
        dataread(nsegr,nsegl,nframe,sgrtrt,sgrtlf,finr,finl,angveldata);

/*Find first frame of good data*/
   for(frame = 1; frame <= nframe; frame++) {
        i=0;
        for(axis = 1; axis<=3;axis++) {
                for(seg=1;seg<=nseg;seg++) {
                        if(angveldata[seg][axis][frame] != BAD) i = i +1;
                }
        }
        if(i == (nseg*3)) {
                ifg = frame;
                frame = nframe;
        }
   }
   if(ifg == nframe) printf("There are no good frames of data");

/*Find last frame of good data*/
   for(frame = ifg; frame<=nframe;frame++) {
        i=0;
        for(axis =1; axis <=3; axis ++) {
                for(seg =1; seg<=nseg; seg++) {
                        if(angveldata[seg][axis][frame] != BAD) i = i + 1;
                }
        }
        if (i == (nseg*3)) lfg = frame;
        else frame = nframe;
   }
   good[0] = ifg;
   good[1] = lfg;

/*Find change in position vector relative to lst frame
        delta(nseg, nframe,good,angveldata);*/
        sf = good[0];
        lf = good[1];

/*Increase size of position data array and extend the ends to prevent spikes in the intial and terminal ends
of the derivative curves*/
        for(kk = 1; kk <= nseg; kk++)
        {
                for (jj = 1; jj <= 3; jj++)
                {
                        for (ii = 1; ii <= nframe; ii++)
```

```
                        {
                                angveldatatemp[kk][jj][ii+50] = angveldata[kk][jj][ii];
                        }
                }
        }

        for(kk = 1; kk <= nseg; kk++)
        {
                for (jj = 1; jj <= 3; jj++)
                {
                        for (ii = 1; ii <= 50; ii++)
                        {
                                angveldatatemp[kk][jj][ii] =
angveldatatemp[kk][jj][50]+(angveldatatemp[kk][jj][100-ii]-angveldatatemp[kk][jj][50]);

                                angveldatatemp[kk][jj][nframe+50+ii] =
angveldatatemp[kk][jj][nframe+50]+(angveldatatemp[kk][jj][nframe+50-ii]-
angveldatatemp[kk][jj][nframe+50]);
                        }
                }
        }

/*temporarily include the frames added to the beginning and end of the data */
        nframe = nframe + 100;
        lf = lf + 100;

/*Smooth angular position data before derivatives are taken*/
/* Median smoothing */
        for(kk = 1; kk <= nseg; kk++)
        {
                for (jj = 1; jj <= 3; jj++)
                {
                        for (ii = 18; ii <= nframe-17; ii++)
                        {
                                for (j=1;j<=35;j++) arr[j] = angveldatatemp[kk][jj][ii-18+j];
                                for (j=2;j<=35;j++)
                                {
                                        a = arr[j];
                                        i = j-1;
                                        while ((i > 0) && (arr[i] > a))
                                        {
                                                arr[i+1] = arr[i];
                                                i--;
                                        }
                                        arr[i+1] = a;
                                }
                                angveldatatemp[kk][jj][ii] = arr[18];
                        }
                }
        }

/* Boxcar average smoothing */
        for(kk = 1; kk <= nseg; kk++)
        {
```

```
                for (jj = 1; jj <= 3; jj++)
                {
                        for (ii = 9; ii <= nframe-8; ii++)
                        {
                                angveldatatemp[kk][jj][ii] = (angveldatatemp[kk][jj][ii-8]+
angveldatatemp[kk][jj][ii-7]+ angveldatatemp[kk][jj][ii-6]+ angveldatatemp[kk][jj][ii-5]+
angveldatatemp[kk][jj][ii-4]+ angveldatatemp[kk][jj][ii-3]+ angveldatatemp[kk][jj][ii-2]+
angveldatatemp[kk][jj][ii-1]+ angveldatatemp[kk][jj][ii]+ angveldatatemp[kk][jj][ii+1]+
angveldatatemp[kk][jj][ii+2]+ angveldatatemp[kk][jj][ii+3]+ angveldatatemp[kk][jj][ii+4]+
angveldatatemp[kk][jj][ii+5]+ angveldatatemp[kk][jj][ii+6]+ angveldatatemp[kk][jj][ii+7]+
angveldatatemp[kk][jj][ii+8])/17.0;/* smooth position data */
                        }
                }
        }
/* Create an ang pos array of original size*/
         nframe=nframe-100;
        lf=lf-100;
        for(kk = 1; kk <= nseg; kk++)
        {
                for (jj = 1; jj <= 3; jj++)
                {
                        for (ii = 1; ii <= nframe; ii++)
                        {
                                angdata[kk][jj][ii] = angveldatatemp[kk][jj][ii+50];
                        }
                }
        }
/*Allocate memory and assign values to subject parameters*/
        moi_r = matrix(1,7,1,3);
        moi_l = matrix(1,7,1,3);
        seg_rotr = (float ***)malloc((unsigned) 7*sizeof(float **))-1;
        for(i=1;i<=7;i++) seg_rotr[i] = matrix(1,3,1,3);
        seg_rotl = (float ***)malloc((unsigned) 7*sizeof(float **))-1;
        for(i=1;i<=7;i++) seg_rotl[i] = matrix(1,3,1,3);

        for(i=1;i<=7; i ++){
                for(j=1;j<=3;j++) {
                        moi_r[i][j] = rfile_header->seginert[i-1][j-1];
                        moi_l[i][j] = lfile_header->seginert[i-1][j-1];
                        for(k=1;k<=3;k++){
                                seg_rotr[i][j][k] = rfile_header->seg_rot[i-1][j-1][k-1];
                                seg_rotl[i][j][k] = lfile_header->seg_rot[i-1][j-1][k-1];
                        }
                }
        }

/*Rotate angle data to global coordinates*/
        angglob(nseg,nsegr, nsegl, sf,lf,sgrtrt, sgrtlf,seg_rotr,seg_rotl,angdata);
/*Write angle data to output file (commented out to save space)*/
        /*exten = "      .ang";
        datawr(namein_r,nseg, nframe,freq, sf, lf,exten, angdata);*/
        lf=lf+100;
        nframe=nframe+100;
```

```
/*Close input files*/
        fclose(finr);
        fclose(finl);

/*Create time standard*/
        angveldata[15][2][1] = 0.0;
        for(i=2;i<=nframe;i++) {
                angveldatatemp[15][2][i] = angveldatatemp[15][2][i-1] + (1.0/freq);
        }

/*Find angular velocities & accelerations in global coordinates */
        angvelaccelsub(nframe,nseg,freq,angveldatatemp,angacceldatatemp);

/* Smooth vel. & accel. again before writing to output files */
/* Velocity Median smoothing */
        for(kk = 1; kk <= nseg; kk++)
        {
                for (jj = 1; jj <= 3; jj++)
                {
                        for (ii = 18; ii <= nframe-17; ii++)
                        {
                                for (j=1;j<=35;j++) arr[j] = angveldatatemp[kk][jj][ii-18+j];
                                for (j=2;j<=35;j++)
                                {
                                        a = arr[j];
                                        i = j-1;
                                        while ((i > 0) && (arr[i] > a))
                                        {
                                                arr[i+1] = arr[i];
                                                i--;
                                        }
                                        arr[i+1] = a;
                                }
                                angveldatatemp[kk][jj][ii] = arr[18];
                        }
                }
        }

/* Acceleration Median smoothing */
        for(kk = 1; kk <= nseg; kk++)
        {
                for (jj = 1; jj <= 3; jj++)
                {
                        for (ii = 18; ii <= nframe-17; ii++)
                        {
                                for (j=1;j<=35;j++) arr[j] = angacceldatatemp[kk][jj][ii-18+j];
                                for (j=2;j<=35;j++)
                                {
                                        a = arr[j];
                                        i = j-1;
                                        while ((i > 0) && (arr[i] > a))
                                        {
                                                arr[i+1] = arr[i];
                                                i--;
```

```
                                        }
                                        arr[i+1] = a;
                                }
                                angacceldatatemp[kk][jj][ii] = arr[18];
                        }
                }
        }

/* Velocity Boxcar average smoothing */
        for(kk = 1; kk <= nseg; kk++)
        {
                for (jj = 1; jj <= 3; jj++)
                {
                        for (ii = 9; ii <= nframe-8; ii++)
                        {
                                angveldatatemp[kk][jj][ii] = (angveldatatemp[kk][jj][ii-8]+
angveldatatemp[kk][jj][ii-7]+ angveldatatemp[kk][jj][ii-6]+ angveldatatemp[kk][jj][ii-5]+
angveldatatemp[kk][jj][ii-4]+ angveldatatemp[kk][jj][ii-3]+ angveldatatemp[kk][jj][ii-2]+
angveldatatemp[kk][jj][ii-1]+ angveldatatemp[kk][jj][ii]+ angveldatatemp[kk][jj][ii+1]+
angveldatatemp[kk][jj][ii+2]+ angveldatatemp[kk][jj][ii+3]+ angveldatatemp[kk][jj][ii+4]+
angveldatatemp[kk][jj][ii+5]+ angveldatatemp[kk][jj][ii+6]+ angveldatatemp[kk][jj][ii+7]+
angveldatatemp[kk][jj][ii+8])/17.0;
                        }
                }
        }

/* Acceleration Boxcar average smoothing */
        for(kk = 1; kk <= nseg; kk++)
        {
                for (jj = 1; jj <= 3; jj++)
                {
                        for (ii = 9; ii <= nframe-8; ii++)
                        {
                                angacceldatatemp[kk][jj][ii] = (angacceldatatemp[kk][jj][ii-8]+
angacceldatatemp[kk][jj][ii-7]+ angacceldatatemp[kk][jj][ii-6]+ angacceldatatemp[kk][jj][ii-5]+
angacceldatatemp[kk][jj][ii-4]+ angacceldatatemp[kk][jj][ii-3]+ angacceldatatemp[kk][jj][ii-2]+
angacceldatatemp[kk][jj][ii-1]+ angacceldatatemp[kk][jj][ii]+ angacceldatatemp[kk][jj][ii+1]+
angacceldatatemp[kk][jj][ii+2]+ angacceldatatemp[kk][jj][ii+3]+ angacceldatatemp[kk][jj][ii+4]+
angacceldatatemp[kk][jj][ii+5]+ angacceldatatemp[kk][jj][ii+6]+ angacceldatatemp[kk][jj][ii+7]+
angacceldatatemp[kk][jj][ii+8])/17.0;
                        }
                }
        }

/* return "frame" to its original size */
nframe = nframe - 100;
lf = lf - 100;

/*Return pos, av,aa,am arrays to original size*/
        for(kk = 1; kk <= nseg; kk++)
        {
                for (jj = 1; jj <= 3; jj++)
                {
                        for (ii = 1; ii <= nframe; ii++)
```

```
                {
                        angveldata[kk][jj][ii] = angveldatatemp[kk][jj][ii+50];
                        angmomdata[kk][jj][ii] = angveldatatemp[kk][jj][ii+50];
                        angacceldata[kk][jj][ii] = angacceldatatemp[kk][jj][ii+50];
                }
            }
        }

/*Rotate angular velocities & acceleration to segmental coordinate system*/
        angseg(nseg,nsegr, nsegl, sf,lf,sgrtrt, sgrtlf,seg_rotr,seg_rotl,angveldata);

/*Compute angular momentum in segmental coordinates*/
        momseg(nsegr, nsegl,sf,lf,moi_r, moi_l,angmomdata);

/*Rotate angular velocity to global coordinates*/
        angglob(nseg,nsegr, nsegl, sf,lf,sgrtrt, sgrtlf,seg_rotr,seg_rotl,angveldata);

/*Rotate angular momentum to global coordinates*/
        angglob(nseg,nsegr, nsegl, sf,lf,sgrtrt, sgrtlf,seg_rotr,seg_rotl,angmomdata);

/*Write data to output file (commented out to save space) */
        /*exten ="     .am";
        datawr(namein_r, nseg,nframe,freq,sf,lf,exten,angmomdata);
        exten ="     .av";
        datawr(namein_r, nseg,nframe,freq,sf,lf,exten,angveldata);
        exten ="     .aa";
        datawr(namein_r, nseg,nframe,freq,sf,lf,exten,angacceldata);  */
}
```

```
/*ANGVELACCELSUB.C
This function calculates the angular velocity and acceleration in global coordinates*/
#define BAD -1000
#include <stdio.h>
#include <stdlib.h>
#include<math.h>

extern float **matrix(int nrl,int nrh,int ncl,int nch);
extern void free_matrix(float **m, int nrl,int nhr, int ncl, int nch);

void angvelaccelsub(int nframe, int nseg, int freq, float angveldata[15][3][1200], float
angacceldata[15][3][1200])
{
   int ifg,lfg;
   int axis, seg, frame;
   int i,fr,ii;
   float h,frequency;
   float **d;
   float **e;
   float **f;

   d = matrix(1,7,1,15);
   e = matrix(1,5,1,15);
   f = matrix(1,4,1,15);
   if( (!d) || (!e) || (!f) ) printf("matrix allocation failure....");

   frequency = freq;
  h = (frequency/12.0);

/*Find first frame of good data*/
   for(frame = 1; frame <= nframe; frame++) {
         i=0;
         for(axis = 1; axis<=3;axis++) {
                  for(seg=1;seg<=nseg;seg++) {
                           if(angveldata[seg][axis][frame] != BAD) i = i +1;
                  }
         }
         if(i == (nseg*3)) {
                  ifg = frame;
                  frame = nframe;
         }
   }
   if(ifg == nframe) printf("There are no good frames of data");

/*Find last frame of good data*/
   for(frame = ifg; frame<=nframe;frame++) {
         i=0;
         for(axis =1; axis <=3; axis ++) {
                  for(seg =1; seg<=nseg; seg++) {
                           if(angveldata[seg][axis][frame] != BAD) i = i + 1;
                  }
         }
         if (i == (nseg*3)) lfg = frame;
         else frame = nframe;
```

```
    }

    fr = ifg + 7;
 /*Set up initial sequence*/

    for(axis = 1;axis <= 3; axis++) {
          i = 1;
          for(frame = ifg; frame < fr; frame ++) {
                for(seg = 1; seg<=nseg;seg++) {
                      d[i][seg] = angveldata[seg][axis][frame];
                }
                i = i + 1;
          }

 /*Startup Sequence*/

          for(seg=1;seg<=nseg;seg++) {
                e[1][seg] = h * (-25 * d[1][seg] + 48 * d[2][seg]
                      - 36 * d[3][seg] + 16*d[4][seg] - 3*d[5][seg]);
                e[2][seg] = h * (-3 *d[1][seg]-10 * d[2][seg] +
                      18 * d[3][seg] - 6*d[4][seg]  + d[5][seg]);
                e[3][seg] = h * (d[1][seg] - d[5][seg] +
                      8 * (d[4][seg] - d[2][seg]));
                e[4][seg] = h * (d[2][seg] - d[6][seg] +
                      8  * (d[5][seg] - d[3][seg]));
                e[5][seg] = h * (d[3][seg] - d[7][seg] +
                      8 * (d[6][seg] - d[4][seg]));

                f[1][seg] = h * (-25 * e[1][seg] + 48*e[2][seg]
                      -36*e[3][seg] + 16 * e[4][seg] - 3*e[5][seg]);
                f[2][seg] =  h * (-3 *e[1][seg]-10 * e[2][seg] +
                      18 * e[3][seg] - 6*e[4][seg]  + e[5][seg]);
          }

          for(seg = 1; seg <= nseg; seg ++) {
                angveldata[seg][axis][1] = e[1][seg];
                angveldata[seg][axis][2] = e[2][seg];
                angacceldata[seg][axis][1] = f[1][seg];
                angacceldata[seg][axis][2] = f[2][seg];
          }

 /* Main differentiation loop*/
          for(frame =(fr-1); frame<=lfg; frame++) {
                for(seg = 1; seg <= nseg; seg ++) {
                      if(angveldata[seg][axis][frame] == BAD)
                       angveldata[seg][axis][frame] = angveldata[seg][axis][frame-1];
                      if(angacceldata[seg][axis][frame] == BAD)
                       angacceldata[seg][axis][frame] = angacceldata[seg][axis][frame-1];
                      d[7][seg] = angveldata[seg][axis][frame];
                      e[5][seg] = h * (d[3][seg]-d[7][seg] +
                            8 *(d[6][seg]-d[4][seg]));
                      f[2][seg]=  h *(e[1][seg] - e[5][seg] +
                            8 * (e[4][seg] - e[2][seg]));
                      for(i=1;i<=6;i++) d[i][seg] = d[i+1][seg];
```

```
                    for(i=1;i<=4;i++) e[i][seg] = e[i+1][seg];
                    f[1][seg] = f[2][seg];
                    angveldata[seg][axis][frame-4] = e[2][seg];
                    angacceldata[seg][axis][frame-4] = f[2][seg];
                }
        }

/*End Sequence*/
        for(seg = 1; seg <= nseg ;seg++) {
                e[5][seg] = h * (-d[2][seg] + 6 * d[3][seg] -
                        18 * d[4][seg] + 10 * d[5][seg] + 3 * d[6][seg]);
                f[1][seg] = h * (e[1][seg] - e[5][seg] +
                        8 * (e[4][seg] - e[2][seg]));
                for( i =1; i <=5; i++) d[i][seg] = d[i+1][seg];
                for(i=1;i<=4;i++) e[i][seg] = e[i+1][seg];
                e[5][seg] = h * (3 * d[1][seg] - 16 * d[2][seg] +
                        36 * d[3][seg] - 48 * d[4][seg] + 25 * d[5][seg]);
                f[2][seg] = h *(e[1][seg] - e[5][seg] +
                        8 * (e[4][seg] - e[2][seg]));
                f[3][seg]= h * (-e[1][seg] + 6 * e[2][seg] - 18 * e[3][seg]
                         + 10 * e[4][seg] + 3 * e[5][seg]);
                f[4][seg]=h*(3 * e[1][seg] - 16 * e[2][seg] + 36 * e[3][seg]
                        - 48 * e[4][seg] + 25 * e[5][seg]);
        }

        for(seg =1; seg <=nseg; seg++) {
                for(i=1;i<=4;i++)
                        angveldata[seg][axis][frame + i -5] = e[i+1][seg];
                for(ii=1;ii<=3;ii++)
                        angacceldata[seg][axis][frame + ii -5] = f[ii+1][seg];

        }
  }
/*BAD all data before ifg and after lfg*/
  for(frame=1;frame < ifg;frame++) {
        for(seg = 1; seg<=nseg;seg++)
                for(axis = 1; axis <=3;axis ++)
                        angveldata[seg][axis][frame] = BAD;
                        angacceldata[seg][axis][frame] = BAD;
   }
  if(lfg != nframe) {
        for(frame=lfg;frame <= nframe;frame++) {
                for(seg = 1; seg<=nseg;seg++)
                        for(axis = 1; axis <=3;axis ++)
                                angveldata[seg][axis][frame] = BAD;
                                angacceldata[seg][axis][frame] = BAD;
        }
  }
  free_matrix(d,1,7,1,15);
  free_matrix(e,1,5,1,15);
  free_matrix(f,1,4,1,15);
}
```

```
/*LINVELACCEL.C - Erin Hutchinson November 1993
This program calculates the segmental linear velocity & acceleration.*/

#include <stdio.h>
#include<stdlib.h>
#include <math.h>
#include  "t4i_global.h"

#define DEBUG
#define BAD -1000

/*Global Variables*/
char  namein_r[20],  namein_l[20], namein_f[20];
char *ext[3];
int frame;
int nb_segr;
int nb_segl, nb_seg;
float freq, totmass;
float fpxscl,fpyscl,fpzscl;
float *segmass;
float **fppos;
float **fy, **ax, **az;
float **rpr, **rpl;
float **rdr, **rdl;
float **rcgr,  **rcgl;
float ***fp_rotmx;
float ***dr, ***dl;
float linveldatatemp[15][3][1200], linacceldatatemp[15][3][1200], linmomdatatemp[15][3][1200];
/*float ***linveldata;
float ***linacceldata;*/
float ****sgrtrt, ****sgrtlf;
t4header *rfile_header, *lfile_header;
FILE *finr, *finl, *fpd;

/* External subroutines*/
/* matrix is a numerical recipes subroutine to initialize matrices*/
extern float **matrix(int nrl,int nrh,int ncl,int nch);

/*DATARD reads in the .dtr, .dtl, and .fpd data*/
extern void datard(int segr, int segl, int nframe, float ***dr,  float ****sgrtrt, float ***dl, float ****sgrtlf,
float **fy, float **ax, float **az,FILE *finr,FILE *finl, FILE *fpd);

/*CGFIND locates the segment and whole body centers of gravity*/
extern void cgfind(int frame, int rseg, int lseg, float totmass,   float ****sgrtrt, float ****sgrtlf, float
***dr, float ***dl, float *segmass, float ***linveldata, float **rcgr, float **rcgl);

/*CPFIND locates the center of pressure*/
extern void cpfind(int nseg,int nframe, float **fy, float **ax, float **az, float ***fp_rotmx,float
**fppos,float ***linveldata);

/*CGVEL calculates the velocity of the center of gravity*/
extern void  linvelaccelsub(int nseg, int nframe, int freq, int sf, float ***linveldata, float
***linacceldata);
```

```
/*DELTA calculates the change in position of the cg with respect to the initial frame of good data*/
/*extern void delta(int nseg, int nframe, int *good, float ***data );*/

/*CGMOM calculates the linear momentum*/
extern void cgmom(int nframe, int nseg, float totmass, float segmass[15], float ***linmomdata );

/*DATAWR writes the data to the root.lm file*/
extern int datawr( char namin_r[20], int nseg, int nframe, int freq, int sf, int lf, char *exten, float ***data,
t4header *rfile_header);


linvelaccel(char *namein, float ***linveldata, float ***linacceldata, float ***linmomdata) {


        int i,j,k,l,sf,lf,ii,jj,kk;
        int *good;
        float a,arr[36];
        float volumecenter[4];
        char *exten;
        rcgr = matrix(1,7,1,3);
        rcgl = matrix(1,7,1,3);
        fppos = matrix(1,2,1,3);
        fy = matrix(1,2,1,1071);
        ax = matrix(1,2,1,1071);
        az = matrix(1,2,1,1071);

        good = calloc(2,sizeof(int));
        segmass = (float *)malloc((unsigned) 15*sizeof(float))-1;

        dr = (float ***)malloc((unsigned) 3*sizeof(float **))-1;
        for(i=1;i<=3;i++) dr[i] = matrix(1,7,1,1071);

        dl = (float ***)malloc((unsigned) 3*sizeof(float **))-1;
        for(i=1;i<=3;i++) dl[i] = matrix(1,7,1,1071);

        fp_rotmx = (float ***)malloc((unsigned) 2*sizeof(float **))-1;
        for(i=1;i<=2;i++) fp_rotmx[i] = matrix(1,3,1,3);

        sgrtrt = (float ****)malloc((unsigned) 3*sizeof(float ***))-1;
        for(i=1;i<=3;i++) {
                sgrtrt[i] =  (float ***)malloc((unsigned) 3*sizeof(float **))-1;
                for(j=1;j<=3;j++) {
                        sgrtrt[i][j] = matrix(1,7,1,1071);
                }
        }
        sgrtlf = (float ****)malloc((unsigned) 3*sizeof(float ***))-1;
        for(i=1;i<=3;i++) {
                sgrtlf[i] =  (float ***)malloc((unsigned) 3*sizeof(float **))-1;
                for(j=1;j<=3;j++) {
                        sgrtlf[i][j] = matrix(1,7,1,1071);
                }
        }

        rfile_header = (t4header *)calloc(1,sizeof(t4header));
```

```
        lfile_header = (t4header *)calloc(1,sizeof(t4header));

/* Assign file names */
        ext[0] = ".dtr";
        ext[1] = ".dtl";
        ext[2] = ".fpd";

        strncpy(namein_r,namein,6);
        namein_l[7] = NULL;
        strcat(namein_r,ext[0]);

        strncpy(namein_l,namein,6);
        namein_l[7] = NULL;
        strcat(namein_l,ext[1]);

        strncpy(namein_f,namein,6);
        strcat(namein_f,ext[2]);

/*Open the right side data file and read the header */
        if (( finr= fopen(namein_r, "r+")) == NULL) {
                (void) printf("ErrorLR opening input file :%s.\n", namein_r);
                return(0);
        }
        fread(rfile_header, sizeof(t4header),1,finr);

/*Open the left side data file and read the header*/
        if (( finl= fopen(namein_l, "r+")) == NULL) {
                (void) printf("ErrorLL opening input file :%s.\n", namein_l);
                return(0);
        }
        fread(lfile_header, sizeof(t4header),1,finl);

/*Open the force plate data file*/
        if (( fpd= fopen(namein_f, "r+")) == NULL) {
                (void) printf("Error opening input file :%s.\n", namein_f);
                return(0);
        }

/*Setting subject parameters*/
        frame =  rfile_header->nb_frames;
        freq = rfile_header->frequency;
        nb_segr = rfile_header ->nb_segment;
        nb_segl = lfile_header ->nb_segment;
        fpxscl = rfile_header->fpxscl;
        fpyscl = rfile_header->fpyscl;
        fpzscl = rfile_header->fpzscl;
        for(i=1;i<=2; i++) {
                for(j = 1; j <=3; j++) {
                        fppos[i][j] = rfile_header->fppos[i-1][j-1];
                        for(k=1;k<=3;k++) {
                                fp_rotmx[i][j][k] = rfile_header->fp_rotmtx[i-1][j-1][k-1];
                        }
                }
        }
```

```
        for(i=1;i<=7; i ++){
                for(j=1;j<=3;j++) {
                        rcgr[i][j] = rfile_header->rcg[i-1][j-1];
                        rcgl[i][j] = lfile_header->rcg[i-1][j-1];
                }
        }

        for (i=0;i<=6;i++) segmass[i] =  rfile_header->segmas[i];
        for (i=7;i<15;i++) segmass[i] =  lfile_header->segmas[i-7];

        nb_seg = nb_segr + nb_segl;

/*Calculate the center of the viewing volume*/
                for(j = 1; j <=3; j++) {
                                        volumecenter[j] = (fppos[1][j]+fppos[2][j])/2.0;
                                        }

/*Call function to read data*/
        datard(nb_segr, nb_segl,frame ,dr,sgrtrt,dl,sgrtlf,fy,ax,az,finr, finl,fpd);

/*Close files*/
        fclose(finr);
        fclose(finl);

/*Find the center of gravity*/
        cgfind( frame, nb_segr, nb_segl,totmass, sgrtrt, sgrtlf, dr,dl,segmass,linveldata,rcgr,rcgl);

/*Find the center of pressure*/
        if ((fpxscl != 0.0) && (fpyscl != 0.0) && (fpzscl != 0.0)) cpfind(nb_seg,frame,fy,ax,az,
fp_rotmx,fppos,linveldata);


/*Throw out position data out of the viewing volume*/
for (i=1;i<=nb_seg;i++){  for (k=1;k<=frame;k++){
        if ( ((linveldata[i][1][k] - volumccenter[1]) > 1.0) || ((linveldata[i][1][k] - volumecenter[1]) < -
1.0) ) linveldata[i][1][k] = linveldata[i][1][k-1];
        if ( ((linveldata[i][2][k] - volumecenter[2]) > 2.0) || ((linveldata[i][2][k] - volumecenter[2]) < -
2.0) ) linveldata[i][2][k] = linveldata[i][2][k-1];
        if ( ((linveldata[i][3][k] - volumecenter[3]) > 1.0) || ((linveldata[i][3][k]  - volumecenter[3]) < -
1.0) ) linveldata[i][3][k] = linveldata[i][3][k-1];
}}

/*Find change in position vector relative to lst frame*/
        delta(nb_seg, frame,good,linveldata);
        sf = good[0];
        lf = good[1];

/*Increase size of position data array and extend the ends to prevent spikes in the intial and terminal ends
of the derivative curves*/
        for(kk = 1; kk <= nb_seg; kk++)
        {
                for (jj = 1; jj <= 3; jj++)
                {
                        for (ii = 1; ii <= frame; ii++)
```

```
                {
                        linveldatatemp[kk][jj][ii+50] = linveldata[kk][jj][ii];
                }
        }
    }

    for(kk = 1; kk <= nb_seg; kk++)
    {
        for (jj = 1; jj <= 3; jj++)
        {
            for (ii = 1; ii <= 50; ii++)
            {
                linveldatatemp[kk][jj][ii] =
linveldatatemp[kk][jj][50]+(linveldatatemp[kk][jj][100-ii]-linveldatatemp[kk][jj][50]);
                linveldatatemp[kk][jj][frame+50+ii] =
linveldatatemp[kk][jj][frame+50]+(linveldatatemp[kk][jj][frame+50-ii]-
linveldatatemp[kk][jj][frame+50]);
            }
        }
    }

/*temporarily include the frames added to the beginning and end of the data */
    frame = frame + 100;
    lf = lf + 100;
/*Smooth Position Data Before Derivatives are taken*/
/* Median smoothing */
    for(kk = 1; kk <= nb_seg; kk++)
    {
        for (jj = 1; jj <= 3; jj++)
        {
            for (ii = 7; ii <= frame-6; ii++)
            {
                for (j=1;j<=13;j++) arr[j] = linveldatatemp[kk][jj][ii-7+j];
                for (j=2;j<=13;j++)
                {
                    a = arr[j];
                    i = j-1;
                    while ((i > 0) && (arr[i] > a))
                    {
                        arr[i+1] = arr[i];
                        i--;
                    }
                    arr[i+1] = a;
                }
                linveldatatemp[kk][jj][ii] = arr[7];
            }
        }
    }

/* Boxcar average smoothing */
    for(kk = 1; kk <= nb_seg; kk++)
    {
        for (jj = 1; jj <= 3; jj++)
        {
```

```
                        for (ii = 9; ii <= frame-8; ii++)
                        {
                                linveldatatemp[kk][jj][ii] = (linveldatatemp[kk][jj][ii-8]+
linveldatatemp[kk][jj][ii-7]+ linveldatatemp[kk][jj][ii-6]+ linveldatatemp[kk][jj][ii-5]+
linveldatatemp[kk][jj][ii-4]+ linveldatatemp[kk][jj][ii-3]+ linveldatatemp[kk][jj][ii-2]+
linveldatatemp[kk][jj][ii-1]+ linveldatatemp[kk][jj][ii]+ linveldatatemp[kk][jj][ii+1]+
linveldatatemp[kk][jj][ii+2]+ linveldatatemp[kk][jj][ii+3]+ linveldatatemp[kk][jj][ii+4]+
linveldatatemp[kk][jj][ii+5]+ linveldatatemp[kk][jj][ii+6]+ linveldatatemp[kk][jj][ii+7]+
linveldatatemp[kk][jj][ii+8])/17.0;
                        }
                }
        }

/* Create an ang pos array of original size*/
        frame=frame-100;
        lf=lf-100;
        for(kk = 1; kk <= nb_seg; kk++)
        {
                for (jj = 1; jj <= 3; jj++)
                {
                        for (ii = 1; ii <= frame; ii++)
                        {
                                linveldata[kk][jj][ii] = linveldatatemp[kk][jj][ii+50];
                        }
                }
        }

/*Create time standard */
        linveldata[15][2][1] = 0.0;
        for(i = 2; i<= frame; i++)
                linveldata[15][2][i] = linveldata[15][2][i-1] + 1/freq;
        exten = "     .pos";
        datawr(namein_r,nb_seg, frame,freq, sf, lf,exten, linveldata,rfile_header);
        lf=lf+100;
        frame=frame+100;


/*Calculate the segment cg velocities and accelerations*/
        linvelaccelsub(nb_seg,frame,freq,sf,linveldatatemp,linacceldatatemp);


/* Smooth Vel. & Accel. Data Before Writing to output files */
/* Velocity Median smoothing */
        for(kk = 1; kk <= nb_seg; kk++)
        {
                for (jj = 1; jj <= 3; jj++)
                {
                        for (ii = 7; ii <= frame-6; ii++)
                        {
                                for (j=1;j<=13;j++) arr[j] = linveldatatemp[kk][jj][ii-7+j];
                                for (j=2;j<=13;j++)
                                {
                                        a = arr[j];
                                        i = j-1;
```

```
                                        while ((i > 0) && (arr[i] > a))
                                        {
                                                arr[i+1] = arr[i];
                                                i--;
                                        }
                                        arr[i+1] = a;
                                }
                                linveldatatemp[kk][jj][ii] = arr[7];
                        }
                }
        }

/* Acceleration Median smoothing */
        for(kk = 1; kk <= nb_seg; kk++)
        {
                for (jj = 1; jj <= 3; jj++)
                {
                        for (ii = 18; ii <= frame-17; ii++)
                        {
                                for (j=1;j<=35;j++) arr[j] = linacceldatatemp[kk][jj][ii-18+j];
                                for (j=2;j<=35;j++)
                                {
                                        a = arr[j];
                                        i = j-1;
                                        while ((i > 0) && (arr[i] > a))
                                        {
                                                arr[i+1] = arr[i];
                                                i--;
                                        }
                                        arr[i+1] = a;
                                }
                                linacceldatatemp[kk][jj][ii] = arr[18];
                        }
                }
        }


/* Velocity Boxcar average smoothing */
        for(kk = 1; kk <= nb_seg; kk++)
        {
                for (jj = 1; jj <= 3; jj++)
                {
                        for (ii = 9; ii <= frame-8; ii++)
                        {
                                linveldatatemp[kk][jj][ii] = (linveldatatemp[kk][jj][ii-8]+
linveldatatemp[kk][jj][ii-7]+ linveldatatemp[kk][jj][ii-6]+ linveldatatemp[kk][jj][ii-5]+
linveldatatemp[kk][jj][ii-4]+ linveldatatemp[kk][jj][ii-3]+ linveldatatemp[kk][jj][ii-2]+
linveldatatemp[kk][jj][ii-1]+ linveldatatemp[kk][jj][ii]+ linveldatatemp[kk][jj][ii+1]+
linveldatatemp[kk][jj][ii+2]+ linveldatatemp[kk][jj][ii+3]+ linveldatatemp[kk][jj][ii+4]+
linveldatatemp[kk][jj][ii+5]+ linveldatatemp[kk][jj][ii+6]+ linveldatatemp[kk][jj][ii+7]+
linveldatatemp[kk][jj][ii+8])/17.0;
                        }
                }
        }
```

```
/* Acceleration Boxcar average smoothing */
        for(kk = 1; kk <= nb_seg; kk++)
        {
                for (jj = 1; jj <= 3; jj++)
                {
                        for (ii = 9; ii <= frame-8; ii++)
                        {
                                linacceldatatemp[kk][jj][ii] = (linacceldatatemp[kk][jj][ii-
8]+linacceldatatemp[kk][jj][ii-7]+linacceldatatemp[kk][jj][ii-6]+ linacceldatatemp[kk][jj][ii-5]+
linacceldatatemp[kk][jj][ii-4]+ linacceldatatemp[kk][jj][ii-3]+ linacceldatatemp[kk][jj][ii-2]+
linacceldatatemp[kk][jj][ii-1]+ linacceldatatemp[kk][jj][ii]+ linacceldatatemp[kk][jj][ii+1]+
linacceldatatemp[kk][jj][ii+2]+ linacceldatatemp[kk][jj][ii+3]+ linacceldatatemp[kk][jj][ii+4]+
linacceldatatemp[kk][jj][ii+5]+ linacceldatatemp[kk][jj][ii+6]+ linacceldatatemp[kk][jj][ii+7]+
linacceldatatemp[kk][jj][ii+8])/17.0;
                        }
                }
        }

for (i=1;i<=nb_seg;i++){ for (j=1;j<=3;j++) { for (k=1;k<=frame;k++){
        linmomdatatemp[i][j][k] = linveldatatemp[i][j][k];
}}}

/*Calculate the linear momentums */
        cgmom(frame, nb_seg, totmass, segmass, linmomdatatemp);

/* return "frame" to its original size */
frame = frame - 100;
lf = lf - 100;

/*Return pos, lv, la, lm arrays to original size*/
        for(kk = 1; kk <= nb_seg; kk++)
        {
                for (jj = 1; jj <= 3; jj++)
                {
                        for (ii = 1; ii <= frame; ii++)
                        {
                                linveldata[kk][jj][ii] = linveldatatemp[kk][jj][ii+50];
                                linmomdata[kk][jj][ii] = linmomdatatemp[kk][jj][ii+50];
                                linacceldata[kk][jj][ii] = linacceldatatemp[kk][jj][ii+50];
                        }
                }
        }


/*Output data (commented out to save space)*/
        /* exten ="     .lv";
        datawr(namein_r, nb_seg,frame,freq,sf,lf,exten, linveldata,rfile_header);
        exten ="     .la";
        datawr(namein_r, nb_seg,frame,freq,sf,lf,exten, linacceldata,rfile_header);
        exten ="     .lm";
        datawr(namein_r, nb_seg,frame,freq,sf,lf,exten, linmomdata,rfile_header);*/
}
```

```
/*LINVELACCELSUB.C  This subroutine
calculates the cg velocity and acceleration from the position data.
Current method is the Lagrangian 5 pt differentiation.*/
#define BAD -1000
#include<stdio.h>
#include<stdlib.h>
#include<math.h>

extern float **matrix(int nrl,int nrh,int ncl,int nch);

void linvelaccelsub(int nseg, int nframe, int freq, int sf, float veldata[15][3][1200], float
acceldata[15][3][1200])
{
   int axis, seg, frame;
   int i,ii,jj,kk, fr;
   float h,frequency;
   float **d;
   float **e;
   float **f;
   /*float temp[12][3][1071];*/

   d = matrix(1,8,1,15);
   e = matrix(1,6,1,15);
   f = matrix(1,4,1,15);

   frequency = freq;
   h = frequency/12.0;
   fr = sf + 7;

/* Set up initial sequence */

   for(axis = 1;axis <= 3; axis++) {
         i = 1;
         for(frame = sf; frame < fr+1; frame ++) {
                  for(seg = 1; seg<=nseg+1;seg++) {
                           d[i][seg] = veldata[seg][axis][frame];
                  }
         i = i + 1;
         }


/*Startup Sequence*/

         for(seg=1;seg<=nseg+1;seg++) {
                  e[1][seg] = h * (-25 * d[1][seg] + 48 * d[2][seg] - 36 * d[3][seg] + 16*d[4][seg] -
3*d[5][seg]);
                  e[2][seg] = h * (-3 *d[1][seg]-10 * d[2][seg] +18 * d[3][seg] - 6*d[4][seg]  + d[5][seg]);
                  e[3][seg] = h * (d[1][seg] - d[5][seg] + 8 * (d[4][seg] - d[2][seg]));
                  e[4][seg] = h * (d[2][seg] - d[6][seg] + 8  * (d[5][seg] - d[3][seg]));
                  e[5][seg] = h * (d[3][seg] - d[7][seg] + 8 * (d[6][seg] - d[4][seg]));
                  e[6][seg] = h * (d[4][seg] - d[8][seg] + 8 * (d[7][seg] - d[5][seg]));

                  f[1][seg] = h * (-25 * e[1][seg] + 48*e[2][seg] -36*e[3][seg] + 16 * e[4][seg] -
3*e[5][seg]);
```

```
                f[2][seg] =  h * (-3 *e[1][seg]-10 * e[2][seg] +18 * e[3][seg] - 6*e[4][seg]  + e[5][seg]);
                f[3][seg] = h * (e[1][seg] - e[5][seg] + 8 * (e[4][seg] - e[2][seg]));
                f[4][seg] = h * (e[2][seg] - e[6][seg] + 8  * (e[5][seg] - e[3][seg]));


        for(seg = 1; seg <= nseg + 1; seg ++) {
                veldata[seg][axis][1] = e[1][seg];
                veldata[seg][axis][2] = e[2][seg];
                acceldata[seg][axis][1] = f[1][seg];
                acceldata[seg][axis][2] = f[2][seg];
                acceldata[seg][axis][3] = f[3][seg];
                acceldata[seg][axis][4] = f[4][seg];
        }

/* Main differentiation loop */
        for(frame = fr; frame<=nframe;frame++) {
                for(seg = 1; seg <= nseg +1;seg ++) {
                        if(veldata[seg][axis][frame] == BAD) veldata[seg][axis][frame] =
veldata[seg][axis][frame-1];
                        if(acceldata[seg][axis][frame] == BAD) acceldata[seg][axis][frame] =
acceldata[seg][axis][frame-1];
                        d[7][seg] = veldata[seg][axis][frame];
                        e[5][seg] = h * (d[3][seg]-d[7][seg] + 8 *(d[6][seg]-d[4][seg]));
                        f[2][seg]=  h *(e[1][seg] - e[5][seg] +8 * (e[4][seg] - e[2][seg]));
                        for(i=1;i<=7;i++) d[i][seg] = d[i+1][seg];
                        for(i=1;i<=5;i++) e[i][seg] = e[i+1][seg];
                        f[1][seg] = f[2][seg];
                        veldata[seg][axis][frame-4] = e[2][seg];
                        acceldata[seg][axis][frame-4] = f[2][seg];
                }
        }

/*End Sequence*/

        for(seg = 1; seg <= nseg +1;seg++) {
                e[5][seg] = h * (-d[2][seg] + 6 * d[3][seg] -18 * d[4][seg] + 10 * d[5][seg] + 3 *
d[6][seg]);
                f[1][seg] = h * (e[1][seg] - e[5][seg] + 8 * (e[4][seg] - e[2][seg]));
                for( i =1; i <=5; i++) d[i][seg] = d[i+1][seg];
                for(i=1;i<=4;i++) e[i][seg] = e[i+1][seg];
                e[5][seg] = h * (3 * d[1][seg] - 16 * d[2][seg] + 36 * d[3][seg] - 48 * d[4][seg] + 25 *
d[5][seg]);

                f[2][seg] = h *(e[1][seg] - e[5][seg] + 8 * (e[4][seg] - e[2][seg]));
                f[3][seg]= h * (-e[1][seg] + 6 * e[2][seg] - 18 * e[3][seg] + 10 * e[4][seg] + 3 *
e[5][seg]);
                f[4][seg]=h*(3 * e[1][seg] - 16 * e[2][seg] + 36 * e[3][seg]- 48 * e[4][seg] + 25 *
e[5][seg]);
        }

        for(seg =1; seg <=nseg +1; seg++) {
                for(i=1;i<=4;i++)
                        veldata[seg][axis][frame + i -5] = e[i+1][seg];
                for(ii;ii<=4;ii++)
```

```
                        acceldata[seg][axis][frame + i -5] = f[i+1][seg];

                }
        }
        for (ii = 1; ii <= 9; ii++)
        {
                for (jj = 1; jj <= 3; jj++)
                {
                        for(kk = 1; kk <= nseg; kk++)
                        {
                                acceldata[kk][jj][ii] = 0;
                        }
                }
        }

}
}
```

A.3 Force, Torque and Whole Body Momentum C Code

```
/*****************************************************************************
      TORQWHOLEBODY.C
* subroutine to calculate the entire body joint net torques
* Using dynamic estimator
* by Erin Hutchinson              November 1993
* MGH Biomotion Laboratory

Sign Conventions for Joint Forces and Torques, listed as they act on the
body segment distal to the joint.

                                        Joint Forces
                                  Positive          Negative

Anterior-Posterior      Ankle     Posterior         Anterior
                        Knee      Posterior         Anterior
                        Hip       Posterior         Anterior
                        Back      Posterior         Anterior
                        Neck      Posterior         Anterior
                        Shlder    Posterior         Anterior

Axial                   Ankle     Compression       Tension
                        Knee      Compression       Tension
                        Hip       Compression       Tension
                        Back      Compression       Tension
                        Neck      Compression       Tension
                        Shlder    Compression       Tension

Lateral Medial          Ankle     Lateral           Medial
                        Knee      Lateral           Medial
                        Hip       Lateral           Medial
                        Back      Left              Right
                        Neck      Left              Right
                        Shlder    Lateral           Medial

                                        Joint Torques
                                  Positive          Negative

Sagittal Plane          Ankle     Plantarflexor     Dorsiflexor
```

	Knee	Extensor	Flexor
	Hip	Extensor	Flexor
	Back	Extensor	Flexor
	Neck	Extensor	Flexor
	Shlder	Extensor	Flexor
Coronal Plane	Ankle	Valgus	Varus
	Knee	Valgus	Varus
	Hip	Abductor	Adductor
	Back	Left Bend	Right Bend
	Neck	Left Bend	Right Bend
	Shlder	Abductor	Adductor
Transverse Plane	Ankle	Internal	External
	Knee	Internal	External
	Hip	Internal	External
	Back	Left Rotation	Right Rotation
	Neck	Left Rotation	Right Rotation
	Shlder	Internal	External

Sign Conventions for Segment Linear and Angular Momentum (global coordinates):

Momentum	Sagittal Plane	Frontal Plane	Transverse Plane
Linear	Left (+z)	Posterior (+x)	Upward (+y)
Angular	ccw @ +z axis	ccw @ +x axis	ccw @ +y axis

For angular momentum, the global coordinates are located as follows:

Segment	Location of Global Coordinate Origin
Foot	Ankle
Shank	Ankle
Thigh	Knee
Pelvis	Hip(s) (momentums averaged)
Trunk	Back Joint (L2-L3) (momentums averaged)
Neck	Neck Joint
Arm	Shoulder

```
**************************************************************************/
#include <t4i_global.h>
#define bad -1000.00

void torqwholebody(float seg_vec_r[], float seg_vec_l[], float seg_rot_r[],
        float seg_rot_l[], float seg_rp_r[], float seg_rp_l[], float seg_rd_r[],
        float seg_rd_l[], float seg_rcg_r[], float seg_rcg_l[], float seg_mass_r[],
        float seg_mass_l[], float seg_inert_r[], float seg_inert_l[], float fppos[],
        loat fprot[], float *rdata, float *ldata, float *fpdata, float *outarrr,
        float *outarrl, float *outarrs, float *outarru, float *outarrmr, float *outarrml,
        float *outarrms, float *outarrmu, float *outarrmw, int nframe, int nseg_r,
        int nseg_l, int ifpr, int ifpl, float wgt[], float lv[13][1071][3],
        float la[13][1071][3],float av[13][1071][3],float aa[13][1071][3],
        float lm[13][1071][3],float am[13][1071][3])
```

```
{

int i,j,k,ijk,ik,ia;

float arrrot[13][3][3], segrtr[3][3], segrtl[3][3], segrcg[13][3], tempr[3], templ[3],
        rd1r[3], rd1l[3], fp[13][3],fd[13][3], tp[13][3], td[13][3], fpg[13][3], fdg[13][3],
        tpg[13][3], tdg[13][3],rdcfdr[3],rdcfdl[3], rp1r[3], rp1l[3], rpcfpr[3],rpcfpl[3],
        cpressr[3], cpressl[3], rotseg[13][3][3], g[3], fppr[3], fppl[3], segmas[13],
        vecseg[13][3], segrp[13][3], segrd[13][3], segint[13][3], arrpos[13][3], data[1],
        fpxyzr[3], fpxyzl[3], fpaxzr[2], fpaxzl[2], fpr[3][3], fpl[3][3], ymr, yml, fmagr,
        fmagl, faxr, faxl, fayr, fayl, fazr, fazl, xr, xl, yr, yl, zr, zl, wgtnt;

float fp15[3], fd15[3], tp15[3], td15[3], fdg15[3], tdg15[3], fpg15[3], tpg15[3], rd1tr[3],
        rd1tl[3], rdcfdt[3], rpcfpt[3], rp1t[3], rcgh[3], temp33t[3][3], inertia15[3][3],
        segrtt[3][3], lv15[3], la15[3], av15[3], aa15[3],iaa15[3], iav15[3], fdg155[3],
        fdg1512[3],fdg156[3], tdg155[3] ,tdg1512[3], tdg156[3], avciav15[3], tdyn15[3],
        dtdyn15[3], tempt[3], arrrot15[3][3], rotseg15[3][3], segmas15, rdcfd155[3],
        rdcfd1512[3], rdcfd156[3],fd155[3],fd1512[3],fd156[3], td155[3], td1512[3],
        td156[3]; /* arrays used to calculate the combined trunk torque */

float fp13[3], fd13[3], tp13[3], td13[3], fdg13[3], tdg13[3], fpg13[3], tpg13[3], rd1p[3],
        rp1p[3], rdcfdp[3], rpcfpp[3], temp33p[3][3], inertia13[3][3], segrtp[3][3], lv13[3],
        la13[3], av13[3], aa13[3],iaa13[3], iav13[3], avciav13[3], tdyn13[3], dtdyn13[3],
        tempp[3], arrrot13[3][3], rotseg13[3][3], segmas16;
        /* arrays used to calculate the combined pelvis torque */

float arshoulder[3], ptrshoulder[3], dtrshoulder[3], alshoulder[3], ptlshoulder[3]
        dtlshoulder[3], trunkcg[3];/* arrays used to calculate the shoulder position */

float rag[3], lag[3], r[13][3], P[3], A[3], rcp[13][3], CG[3], gsegrcg[13][3], segmastotal,
        rcg[13][3],rcgcp[13][3],ACG[3]; /* arrays used to calculate whole-body momentum */

        float nsegm = nseg_r;
        float inertia[13][3][3]; /*inertia matrices(Ixx Ixy Ixz; Ixy Iyy Iyz; Ixz Iyz Izz)*/
        float iaa[13][3]; /*arrays for segment (inertia)*(angular acceleration)*/
        float iav[13][3]; /*arrays for segment (inertia)*(angular velocity)*/
        float avciav[13][3];/*arrays for segment(ang.velocity)X(inertia)*(angular velocity)*/
        float temp33r[3][3],temp33l[3][3];
        float pinertia[13][3][3];/*principle moment 3X3 matrices*/
        float sum[3],suml[3],sumr,pweight, avehipforce[3],avetrunkforce[3];
        g[0] = 0.0, g[1] = -9.8, g[2] = 0.0;/* Gravity vector in global coordinates */

/* Form the array to segment vectors and rotation matrices */
        xr = 0.0, yr = 0.0, zr = 0.0, ymr = 0.0;
        xl = 0.0, yl = 0.0, zl = 0.0, yml = 0.0;
        wgtnt = wgt[0] * 9.8 / 2.2;

        for (i = 0; i < 3; i++)
        {
                fppr[i] = fppos[3*(ifpr-1)+i];
                fppl[i] = fppos[3*(ifpl-1)+i];
                for (j = 0; j < 3; j++)
                {
                        fpr[j][i] = fprot[9*(ifpr-1)+3*(i)+j];
```

```
                        fpl[j][i] = fprot[9*(ifpl-1)+3*(i)+j];
                }
        }

        for (i = 0; i < nseg_r ; i++)
        {
                segmas[i] = seg_mass_r[i];
                for (j = 0; j < 3; j++)
                {
                vecseg[i][j] = seg_vec_r[3*(i)+j];          /* array to seg origin vector */
                segrp[i][j] = seg_rp_r[3*(i)+j];   /* array to proximal jc vector */
                segrd[i][j] = seg_rd_r[3*(i)+j];   /* array to distal jc vector */
                segrcg[i][j] = seg_rcg_r[3*(i)+j];          /* array to seg cg vector */
                segint[i][j] = seg_inert_r[3*(i)+j];/*seg principal moment of inertia*/
                for (k = 0; k < 3; k++)
                        {
                                rotseg[i][k][j] = seg_rot_r[9*(i)+3*(j)+k];

                        }
                }
        }
        for (i = nseg_r; i < nseg_r+nseg_l ; i++)
        {
                segmas[i] = seg_mass_l[i-7];
                for (j = 0; j < 3; j++)
                {
                vecseg[i][j] = seg_vec_l[3*(i-7)+j];  /* array to seg origin vector */
                segrp[i][j] = seg_rp_l[3*(i-7)+j];          /* array to proximal jc vector */
                segrd[i][j] = seg_rd_l[3*(i-7)+j];          /* array to distal jc vector */
                segrcg[i][j] = seg_rcg_l[3*(i-7)+j];  /* array to seg cg vector */
                segint[i][j] = seg_inert_l[3*(i-7)+j];/*seg principal moment of inertia*/

                        for (k = 0; k < 3; k++)
                        {
                                rotseg[i][k][j] = seg_rot_l[9*(i-7)+3*(j)+k];
                        }
                }
        }

/* Initialize principle moment 3X3 matrix */
        for(k=0;k<nseg_r+nseg_l;k++)
        {
                for(i=0;i<3;i++)
                {
                        for(j=0;j<3;j++)
                        pinertia[k][i][j] = 0.0;
                }
                pinertia[k][0][0]= segint[k][0];
                pinertia[k][1][1]= segint[k][1];
                pinertia[k][2][2]= segint[k][2];
        }

/* Estimate joint net force and torque  (dynamic estimator) */
for (i = 0; i < nframe ; i++)
```

```
{
        for (j = 0; j < nseg_r; j++)
        {
                for (k = 0; k < 3; k++)
                {

                        arrpos[j][k] = rdata[16*nseg_r*(i)+16*(j)+k];
                        for (ijk =0; ijk < 3; ijk++)
                        {
                        arrrot[j][k][ijk]= rdata[16*nseg_r*(i)+16*(j)+7+3*(k)+ijk];

                        }
                }
        }

        for (j = nseg_r; j < nseg_r+nseg_l; j++)
        {
                for (k = 0; k < 3; k++)
                {

                        arrpos[j][k] = ldata[16*nseg_l*(i)+16*(j-7)+k];
                        for (ijk =0; ijk < 3; ijk++)
                        {
                        arrrot[j][k][ijk]= ldata[16*nseg_l*(i)+16*(j-7)+7+3*(k)+ijk];

                        }
                }
        }

/* Find Trunk and Pelvis Average rotation matrices */
                for (j=0;j<3;j++)
                {
                        for(k=0;k<3;k++)
                        {
                        arrrot13[j][k]= (arrrot[3][j][k] + arrrot[10][j][k])/2.0;
                        rotseg13[j][k]= (rotseg[3][j][k] + rotseg[10][j][k])/2.0;
                        arrrot15[j][k]= (arrrot[4][j][k] + arrrot[11][j][k])/2.0;
                        rotseg15[j][k]= (rotseg[4][j][k] + arrrot[11][j][k])/2.0;
                        }
                }

/* Calculate Shoulder Joint Positions */

/* right shoulder position in global coordinates as calculated from right arm */
        gmprd1(arrrot[5],segrp[5],tempr);
        gmprd1(arrrot[5],segrcg[5],templ);
        for (ia=0;ia<3;ia++)
        {
                arshoulder[ia]=tempr[ia]+arrpos[5][ia]+templ[ia];
        }
        gmprd1(arrrot[4],segrp[4],tempr);
        gmprd1(arrrot[4],segrcg[4],templ);
        for (ia=0;ia<3;ia++)
        {
```

```
                ptrshoulder[ia]=tempr[ia]+arrpos[4][ia]+templ[ia];
        }
        gmprd1(arrrot[4],segrd[4],tempr);
        gmprd1(arrrot[4],segrcg[4],templ);
        for (ia=0;ia<3;ia++)
        {
                dtrshoulder[ia]=tempr[ia]+arrpos[4][ia]+templ[ia];
        }

/* left shoulder position in global coordinates as calculated from left arm */
        gmprd1(arrrot[12],segrp[12],tempr);
        gmprd1(arrrot[12],segrcg[12],templ);
        for (ia=0;ia<3;ia++)
        {
                alshoulder[ia]=tempr[ia]+arrpos[12][ia]+templ[ia];
        }
        gmprd1(arrrot[11],segrp[11],tempr);
        gmprd1(arrrot[11],segrcg[11],templ);
        for (ia=0;ia<3;ia++)
        {
                ptlshoulder[ia]=tempr[ia]+arrpos[11][ia]+templ[ia];
        }
        gmprd1(arrrot[11],segrd[11],tempr);
        gmprd1(arrrot[11],segrcg[11],templ);
        for (ia=0;ia<3;ia++)
        {
                dtlshoulder[ia]=tempr[ia]+arrpos[11][ia]+templ[ia];
        }

/* Calculation of the moment arm from the trunk cg to the shoulder joints */
        gmprd1(arrrot[4],segrcg[4],tempr);
        gmprd1(arrrot[11],segrcg[11],templ);
        for(ia=0;ia<3;ia++)
                {
                trunkcg[ia]= (arrpos[4][ia]+tempr[ia]+arrpos[11][ia]+templ[ia])/2.0;
                }

/* distance from trunk cg to right shoulder joint in global coordinates */
        for (ia=0;ia<3;ia++)
        {
                tempr[ia]= arshoulder[ia]-trunkcg[ia];

        }
        gmtra(arrrot15,segrtr);
/*trunk cg to right shoulder in bcs (moment arm for shoulder reaction force,fd155)*/
        gmprd1(segrtr,tempr,rd1tr);

/* distance from trunk cg to left shoulder joint in global coordinates */
        for (ia=0;ia<3;ia++)
        {
                templ[ia]= alshoulder[ia]-trunkcg[ia];

        }
        gmtra(arrrot15,segrtr);
```

```
/*trunk cg to left shoulder in bcs (moment arm for shoulder reaction force, fd1512)*/
        gmprd1(segrtr,templ,rd1tl);

/*Calculate the whole body linear and angular momentum
(about a point which is the average of the two ankle joints and about the CG) */

for (j=0;j<13;j++)
{
gmprd1(arrrot[j],segrcg[j],gsegrcg[j]);/*array to segment cg in global coordinates*/
}

/* about the combined ankle joint */
/*right ankle joint location in global coordinates*/
gmprd1(arrrot[0],segrp[0],templ);
for (ia=0;ia<3;ia++) rag[ia]= arrpos[0][ia] + gsegrcg[0][ia] + templ[ia];

/*left ankle joint location in global coordinates*/
gmprd1(arrrot[7],segrp[7],templ);
for (ia=0;ia<3;ia++) lag[ia]= arrpos[7][ia] + gsegrcg[7][ia] + templ[ia];

/*vectors from the average ankle to segment cg's in global coordinates*/
for (j=0;j<13;j++)
{
for (ia=0;ia<3;ia++) r[j][ia]= arrpos[j][ia]+gsegrcg[j][ia]-(rag[ia]+lag[ia])/2;
}

/* AM about the  whole body CG */

/* Whole Body CG location */
segmastotal = segmas[0] + segmas[1] + segmas[2] + (segmas[3] + segmas[10])/2
+ (segmas[4] + segmas[11])/2 + segmas[5] + segmas[6] + segmas[7] + segmas[8]
+ segmas[9] + segmas[12];

for (ia=0; ia<3; ia++)
{
CG[ia] = (segmas[0]*(arrpos[0][ia] + gsegrcg[0][ia]) + segmas[1]*(arrpos[1][ia]
+ gsegrcg[1][ia]) + segmas[2]*(arrpos[2][ia] + gsegrcg[2][ia]) + (segmas[3]*
(arrpos[3][ia] + gsegrcg[3][ia])  + segmas[10]*(arrpos[10][ia] +
gsegrcg[10][ia]))/2 + (segmas[4]*(arrpos[4][ia] + gsegrcg[4][ia])+ segmas[11]
*(arrpos[11][ia] + gsegrcg[11][ia]))/2 + segmas[5]*(arrpos[5][ia] +
gsegrcg[5][ia]) + segmas[6]*(arrpos[6][ia] + gsegrcg[6][ia]) + segmas[7]*
(arrpos[7][ia] + gsegrcg[7][ia])+ segmas[8]*(arrpos[8][ia] + gsegrcg[8][ia])
+ segmas[9]*(arrpos[9][ia] + gsegrcg[9][ia]) + segmas[12]*(arrpos[12][ia] +
gsegrcg[12][ia]))/segmastotal;
}

/*vectors from the average ankle to segment cg's in global coordinates*/
for (j=0;j<13;j++)
{
for (ia=0;ia<3;ia++) rcg[j][ia]= arrpos[j][ia] + gsegrcg[j][ia] - CG[ia];
}

if(i==440) printf("CG = %f %f %f\n",CG[0],CG[1],CG[2]);
```

```
 /*whole body linear momentum*/
for (ia=0;ia<3;ia++) P[ia] = 0;
for (ia=0;ia<3;ia++) P[ia] = lm[0][i][ia] + lm[1][i][ia] + lm[2][i][ia] +
(lm[3][i][ia] + lm[10][i][ia])/2 + (lm[4][i][ia] + lm[11][i][ia])/2 +
lm[5][i][ia] + lm[6][i][ia] + lm[7][i][ia] + lm[8][i][ia] + lm[9][i][ia]
+ lm[12][i][ia]; /* whole body linear momentum */

/*whole body angular momentum*/
for (ia=0;ia<3;ia++) A[ia] = 0;
for (ia=0;ia<3;ia++) ACG[ia] = 0;
for (j=0;j<13;j++)
{
crossp(r[j],lm[j][i],rcp[j]); /* about combined ankle joint */
crossp(rcg[j],lm[j][i],rcgcp[j]); /* about combined ankle joint */
}
 /* whole body angular momentum about combined ankle joint*/
for (ia=0;ia<3;ia++) A[ia] =  am[1][i][ia] + rcp[1][ia] + am[2][i][ia] +
rcp[2][ia] + (am[3][i][ia] + rcp[3][ia] + am[10][i][ia] + rcp[10][ia])/2
+ (am[4][i][ia] + rcp[4][ia] + am[11][i][ia] + rcp[11][ia])/2 + am[5][i][ia]
+ rcp[5][ia] + am[6][i][ia] + rcp[6][ia] + am[8][i][ia] + rcp[8][ia] +
am[9][i][ia] + rcp[9][ia] + am[12][i][ia] + rcp[12][ia];

/* whole body angular momentum about CG*/
for (ia=0;ia<3;ia++) ACG[ia] = am[0][i][ia] + rcgcp[0][ia] +am[1][i][ia]
+ rcgcp[1][ia] + am[2][i][ia] + rcgcp[2][ia] + (am[3][i][ia] + rcgcp[3][ia]
+ am[10][i][ia] + rcgcp[10][ia])/2 + (am[4][i][ia] + rcgcp[4][ia] +
am[11][i][ia] + rcgcp[11][ia])/2 + am[5][i][ia] + rcgcp[5][ia] + am[6][i][ia]
+ rcgcp[6][ia] + am[7][i][ia] + rcgcp[7][ia] + am[8][i][ia] + rcgcp[8][ia] +
am[9][i][ia] + rcgcp[9][ia] + am[12][i][ia] + rcgcp[12][ia];

/* Read in the force plate data */
        for (k =0; k < 3 ; k++)
        {
        fpxyzr[k] = fpdata[12*(i)+6*(ifpr-1)+k];/*three force components*/
        fpxyzl[k] = fpdata[12*(i)+6*(ifpl-1)+k];/*three force components*/
        }
        fpaxzr[0] = fpdata[12*(i)+6*(ifpr-1)+3];/*ap cop displacement*/
        ymr = fpdata[12*(i)+6*(ifpr-1)+4];          /*free moment*/
        fpaxzr[1] = fpdata[12*(i)+6*(ifpr-1)+5];/*lateral cop displacement*/
        fpaxzl[0] = fpdata[12*(i)+6*(ifpl-1)+3];/*ap cop displacement*/
        yml = fpdata[12*(i)+6*(ifpl-1)+4];          /*free moment*/
        fpaxzl[1] = fpdata[12*(i)+6*(ifpl-1)+5];/*lateral cop displacement*/

/* Do segment ([0] & [7]) calculations (feet)*/
        if ((arrpos[0][0] == bad)||(arrpos[0][1] == bad)||arrpos[0][2] == bad)
        {
                for (ik = 0; ik < 3; ik++)
                {
                        for (ia=0;ia<4;ia++)
                        {
                        fd[ia][ik] = 0.0;
                        td[ia][ik] = 0.0;
                        fp[ia][ik] = 0.0;
                        tp[ia][ik] = 0.0;
```

```
                fdg[ia][ik]= 0.0;
                tdg[ia][ik]= 0.0;
                fpg[ia][ik]= 0.0;
                tpg[ia][ik]= 0.0;
                arrpos[ia][0] == bad;
                arrpos[ia][1] == bad;
                arrpos[ia][2] == bad;
                }
        }
}
if ((arrpos[7][0] == bad)||(arrpos[7][1] == bad)||arrpos[7][2] == bad)
{
        for (ik = 0; ik < 3; ik++)
        {
                for (ia=7;ia<11;ia++)
                {
                fd[ia][ik] = 0.0;
                td[ia][ik] = 0.0;
                fp[ia][ik] = 0.0;
                tp[ia][ik] = 0.0;
                fdg[ia][ik]= 0.0;
                tdg[ia][ik]= 0.0;
                fpg[ia][ik]= 0.0;
                tpg[ia][ik]= 0.0;
                arrpos[ia][0] == bad;
                arrpos[ia][1] == bad;
                arrpos[ia][2] == bad;
                }
        }
}
        gmtra(arrrot[0],segrtr);/* transpose segment rotation matrix*/
        gmtra(arrrot[7],segrtl);/* transpose segment rotation matrix*/
        gmprd1(arrrot[0],segrcg[0],tempr);/*cg vector global coords*/
        gmprd1(arrrot[7],segrcg[7],templ);/*cg vector global coords*/
        xr = arrpos[0][0]+tempr[0];/*segment cg position global*/
        yr = arrpos[0][1]+tempr[1];
        zr = arrpos[0][2]+tempr[2];
        xl = arrpos[7][0]+templ[0];/*segment cg position global*/
        yl = arrpos[7][1]+templ[1];
        zl = arrpos[7][2]+templ[2];

        for (ia = 0; ia < 3; ia++)
        {
                fdg[0][ia]=fpxyzr[ia];
                fdg[7][ia]=fpxyzl[ia];
        }

        cpressr[0] = -fpaxzr[0];
        cpressr[1] = 0.0;
        cpressr[2] = -fpaxzr[1];
        cpressl[0] = -fpaxzl[0];
        cpressl[1] = 0.0;
        cpressl[2] = -fpaxzl[1];
        tdg[0][1] = ymr;
```

```
tdg[0][0] = 0.0;
tdg[0][2] = 0.0;
tdg[7][1] = yml;
tdg[7][0] = 0.0;
tdg[7][2] = 0.0;
fmagr = sqrt(fdg[0][0]*fdg[0][0]
        + fdg[0][1]*fdg[0][1] + fdg[0][2]*fdg[0][2]);
fmagl = sqrt(fdg[7][0]*fdg[7][0]
        + fdg[7][1]*fdg[7][1] + fdg[7][2]*fdg[7][2]);

if (fmagr < 20)
{
        fdg[0][0] = 0.0;
        fdg[0][1] = 0.0;
        fdg[0][2] = 0.0;
}
if (fmagl < 20)
{
        fdg[7][0] = 0.0;
        fdg[7][1] = 0.0;
        fdg[7][2] = 0.0;
}

rforce(fdg[0],tdg[0],cpressr,fpr);/*rotate fp data into global coords*/
rforce(fdg[7],tdg[7],cpressl,fpl);/*rotate fp data into global coords*/

faxr = cpressr[0];
fayr = cpressr[1];
fazr = cpressr[2];
fdg[0][1] = (-1)*fdg[0][1];
tdg[0][1] = (-1)*tdg[0][1];
faxl = cpressl[0];
fayl = cpressl[1];
fazl = cpressl[2];
fdg[7][1] = (-1)*fdg[7][1];
tdg[7][1] = (-1)*tdg[7][1];

/* Calculate the vector to the forceplate center of force in global coordinates */
faxr = faxr + fppr[0]; /* cop in global coordinates */
fayr = fayr + fppr[1]; /* cop in global coordinates */
fazr = fazr + fppr[2]; /* cop in global coordinates */
tempr[0] = faxr - xr;
tempr[1] = fayr - yr;
tempr[2] = fazr - zr;
faxl = faxl + fppl[0]; /* cop in global coordinates */
fayl = fayl + fppl[1]; /* cop in global coordinates */
fazl = fazl + fppl[2]; /* cop in global coordinates */
templ[0] = faxl - xl;
templ[1] = fayl - yl;
templ[2] = fazl - zl;
gmprd1(segrtr,tempr,rd1r);
gmprd1(segrtl,templ,rd1l);

for (ia = 0; ia < 3; ia++)
```

```
                {
                        fpg[0][ia] = (-1)*segmas[0]*g[ia]-fdg[0][ia]+segmas[0]*la[0][i][ia];
                        fpg[7][ia] = (-1)*segmas[7]*g[ia]-fdg[7][ia]+segmas[7]*la[7][i][ia];
                        rp1r[ia] = segrp[0][ia];
                        rp1l[ia] = segrp[7][ia];
                }

/* Change forces and torques into bcs coordinates */
                gmprd1(segrtr,fpg[0],fp[0]);
                gmprd1(segrtl,fpg[7],fp[7]);
                gmprd1(segrtr,fdg[0],fd[0]);
                gmprd1(segrtl,fdg[7],fd[7]);
                gmprd1(segrtr,tdg[0],td[0]);
                gmprd1(segrtl,tdg[7],td[7]);
                gmprd1(segrtr,av[0][i],av[0][i]);
                gmprd1(segrtr,aa[0][i],aa[0][i]);
                gmprd1(segrtl,av[7][i],av[7][i]);
                gmprd1(segrtl,aa[7][i],aa[7][i]);

                crossp(rd1r,fd[0],rdcfdr);
                crossp(rd1l,fd[7],rdcfdl);
                crossp(rp1r,fp[0],rpcfpr);
                crossp(rp1l,fp[7],rpcfpl);

                gmprd3(arrrot[0],pinertia[0],temp33r);
                gmprd3(arrrot[7],pinertia[7],temp33l);
                gmprd3(temp33r,segrtr,inertia[0]);
                gmprd3(temp33l,segrtl,inertia[7]);
                gmprd1(inertia[0],aa[0][i],iaa[0]);
                gmprd1(inertia[7],aa[7][i],iaa[7]);
                gmprd1(inertia[0],av[0][i],iav[0]);
                gmprd1(inertia[7],av[7][i],iav[7]);
                crossp(av[0][i],iav[0], avciav[0]);
                crossp(av[7][i],iav[7], avciav[7]);

                for (ia = 0; ia < 3; ia++)
                {
                        tdyn[0][ia] = avciav[0][ia] + iaa[0][ia];
                        tdyn[7][ia] = avciav[7][ia] + iaa[7][ia];
                }

                for (ia = 0; ia < 3; ia++)
                {
                        tp[0][ia] = (-1)*rdcfdr[ia] - td[0][ia] - rpcfpr[ia] +tdyn[0][ia];
                        tp[7][ia] = (-1)*rdcfdl[ia] - td[7][ia] - rpcfpl[ia] +tdyn[7][ia];
                }
                gmprd1(arrrot[0],tp[0],tpg[0]);
                gmprd1(arrrot[7],tp[7],tpg[7]);
                gmprd1(arrrot[0],fp[0],fpg[0]);
                gmprd1(arrrot[7],fp[7],fpg[7]);

        if ((arrpos[0][0] == bad) || (arrpos[0][1] == bad) || arrpos[0][2] == bad)
        {
                for (ik = 0; ik < 3; ik++)
```

```
                {
                        for (ia=0;ia<4;ia++)
                        {
                        fd[ia][ik] = 0.0;
                        td[ia][ik] = 0.0;
                        fp[ia][ik] = 0.0;
                        tp[ia][ik] = 0.0;
                        fdg[ia][ik]= 0.0;
                        tdg[ia][ik]= 0.0;
                        fpg[ia][ik]= 0.0;
                        tpg[ia][ik]= 0.0;
                        arrpos[ia][0] == bad;
                        arrpos[ia][1] == bad;
                        arrpos[ia][2] == bad;
                        }
                }
        }

        if ((arrpos[7][0] == bad) || (arrpos[7][1] == bad) || arrpos[7][2] == bad)
        {
                for (ik = 0; ik < 3; ik++)
                {
                        for (ia=7;ia<11;ia++)
                        {
                        fd[ia][ik] = 0.0;
                        td[ia][ik] = 0.0;
                        fp[ia][ik] = 0.0;
                        tp[ia][ik] = 0.0;
                        fdg[ia][ik]= 0.0;
                        tdg[ia][ik]= 0.0;
                        fpg[ia][ik]= 0.0;
                        tpg[ia][ik]= 0.0;
                        arrpos[ia][0] == bad;
                        arrpos[ia][1] == bad;
                        arrpos[ia][2] == bad;
                        }
                }
        }

/* Do segment ([1]&[8])calculations (shanks)*/
                if ((arrpos[1][0] == bad) || (arrpos[1][1] == bad) || (arrpos[1][2] == bad))
                {
                        for (ik = 0; ik < 3; ik++)
                        {
                                for (ia=1;ia<4;ia++)
                                {
                                fd[ia][ik]=0.0;
                                td[ia][ik]=0.0;
                                fp[ia][ik]=0.0;
                                tp[ia][ik]=0.0;
                                fdg[ia][ik]=0.0;
                                tdg[ia][ik]=0.0;
                                fpg[ia][ik]=0.0;
                                tpg[ia][ik]=0.0;
```

```
                    arrpos[ia][0] == bad;
                    arrpos[ia][1] == bad;
                    arrpos[ia][2] == bad;
                    }
                }
        }
        if ((arrpos[8][0] == bad) || (arrpos[8][1] == bad) || (arrpos[8][2] == bad))
        {
                for (ik = 0; ik < 3; ik++)
                {
                        for (ia=8;ia<11;ia++)
                        {
                        fd[ia][ik]=0.0;
                        td[ia][ik]=0.0;
                        fp[ia][ik]=0.0;
                        tp[ia][ik]=0.0;
                        fdg[ia][ik]=0.0;
                        tdg[ia][ik]=0.0;
                        fpg[ia][ik]=0.0;
                        tpg[ia][ik]=0.0;
                        arrpos[ia][0] == bad;
                        arrpos[ia][1] == bad;
                        arrpos[ia][2] == bad;
                        }
                }
        }
                gmtra(arrrot[1],segrtr); /* Transpose segment rotation matrix */
                gmtra(arrrot[8],segrtl); /* Transpose segment rotation matrix */

                for (ia = 0; ia < 3; ia++)
                {
                        fdg[1][ia]= (-1)*fpg[0][ia];
                        tdg[1][ia]= (-1)*tpg[0][ia];
                        rd1r[ia]= segrd[1][ia];
                        rp1r[ia]= segrp[1][ia];
                        fdg[8][ia]= (-1)*fpg[7][ia];
                        tdg[8][ia]= (-1)*tpg[7][ia];
                        rd1l[ia]= segrd[8][ia];
                        rp1l[ia]= segrp[8][ia];
                }
                for (ia = 0; ia < 3; ia++)
                {
                        fpg[1][ia] = (-1)*segmas[1]*g[ia]
                                        -fdg[1][ia]+segmas[1]*la[1][i][ia];
                        fpg[8][ia] = (-1)*segmas[8]*g[ia]
                                        -fdg[8][ia]+segmas[8]*la[8][i][ia];
                }

/* Change forces and torques into bcs coordinates */
                gmprd1(segrtr,fpg[1],fp[1]);
                gmprd1(segrtl,fpg[8],fp[8]);
                gmprd1(segrtr,fdg[1],fd[1]);
                gmprd1(segrtl,fdg[8],fd[8]);
                gmprd1(segrtr,tdg[1],td[1]);
```

```
gmprd1(segrtl,tdg[8],td[8]);
gmprd1(segrtr,av[1][i],av[1][i]);
gmprd1(segrtr,aa[1][i],aa[1][i]);
gmprd1(segrtl,av[8][i],av[8][i]);
gmprd1(segrtl,aa[8][i],aa[8][i]);

crossp(rd1r,fd[1],rdcfdr);
crossp(rd1l,fd[8],rdcfdl);
crossp(rp1r,fp[1],rpcfpr);
crossp(rp1l,fp[8],rpcfpl);

gmprd3(arrrot[1],pinertia[1],temp33r);
gmprd3(arrrot[8],pinertia[8],temp33l);
gmprd3(temp33r,segrtr,inertia[1]);
gmprd3(temp33l,segrtl,inertia[8]);
gmprd1(inertia[1],aa[1][i],iaa[1]);
gmprd1(inertia[8],aa[8][i],iaa[8]);
gmprd1(inertia[1],av[1][i],iav[1]);
gmprd1(inertia[8],av[8][i],iav[8]);
crossp(av[1][i],iav[1], avciav[1]);
crossp(av[8][i],iav[8], avciav[8]);

for (ia = 0; ia < 3; ia++)
{
        tdyn[1][ia] = avciav[1][ia] + iaa[1][ia];
        tdyn[8][ia] = avciav[8][ia] + iaa[8][ia];
}

for (ia = 0; ia < 3; ia++)
{
        tp[1][ia]= (-1)*(rdcfdr[ia]+td[1][ia]
                        +rpcfpr[ia]-tdyn[1][ia]);
        tp[8][ia]= (-1)*(rdcfdl[ia]+td[8][ia]+
                        rpcfpl[ia]-tdyn[8][ia]);
}

gmprd1(arrrot[1], tp[1], tpg[1]);
gmprd1(arrrot[8], tp[8], tpg[8]);
gmprd1(arrrot[1], fp[1], fpg[1]);
gmprd1(arrrot[8], fp[8], fpg[8]);

if ((arrpos[1][0] == bad) || (arrpos[1][1] == bad) || (arrpos[1][2] == bad))
{
        for (ik = 0; ik < 3; ik++)
        {
                for (ia=1;ia<4;ia++)
                {
                fd[ia][ik]=0.0;
                td[ia][ik]=0.0;
                fp[ia][ik]=0.0;
                tp[ia][ik]=0.0;
                fdg[ia][ik]=0.0;
                tdg[ia][ik]=0.0;
                fpg[ia][ik]=0.0;
```

```
                        tpg[ia][ik]=0.0;
                        arrpos[ia][0] == bad;
                        arrpos[ia][1] == bad;
                        arrpos[ia][2] == bad;
                        }
                }
        }
        if ((arrpos[8][0] == bad) || (arrpos[8][1] == bad) || (arrpos[8][2] == bad))
        {
                for (ik = 0; ik < 3; ik++)
                {
                        for (ia=8;ia<11;ia++)
                        {
                        fd[ia][ik]=0.0;
                        td[ia][ik]=0.0;
                        fp[ia][ik]=0.0;
                        tp[ia][ik]=0.0;
                        fdg[ia][ik]=0.0;
                        tdg[ia][ik]=0.0;
                        fpg[ia][ik]=0.0;
                        tpg[ia][ik]=0.0;
                        arrpos[ia][0] == bad;
                        arrpos[ia][1] == bad;
                        arrpos[ia][2] == bad;
                        }
                }
        }

/* Do segment ([2]&[9]) calculations (thighs)*/
                if ((arrpos[2][0] == bad) || (arrpos[2][1] == bad) || (arrpos[2][2] == bad))
                        {
                                for (ik = 0; ik < 3; ik++)
                                {
                                        for (ia=2;ia<4;ia++)
                                        {
                                        fd[ia][ik]=0.0;
                                        td[ia][ik]=0.0;
                                        fp[ia][ik]=0.0;
                                        tp[ia][ik]=0.0;
                                        fdg[ia][ik]=0.0;
                                        tdg[ia][ik]=0.0;
                                        fpg[ia][ik]=0.0;
                                        tpg[ia][ik]=0.0;
                                        arrpos[3][0] == bad;
                                        arrpos[3][1] == bad;
                                        arrpos[3][2] == bad;
                                        }
                                }
                        }
                if ((arrpos[9][0] == bad) || (arrpos[9][1] == bad) || (arrpos[9][2] == bad))
                        {
                                for (ik = 0; ik < 3; ik++)
                                {
                                        for (ia=9;ia<11;ia++)
```

```
                                {
                                fd[ia][ik]=0.0;
                                td[ia][ik]=0.0;
                                fp[ia][ik]=0.0;
                                tp[ia][ik]=0.0;
                                fdg[ia][ik]=0.0;
                                tdg[ia][ik]=0.0;
                                fpg[ia][ik]=0.0;
                                tpg[ia][ik]=0.0;
                                arrpos[10][0] == bad;
                                arrpos[10][1] == bad;
                                arrpos[10][2] == bad;
                                }
                        }
                }
                gmtra(arrrot[2],segrtr); /*Transpose segment rotation matrix*/
                gmtra(arrrot[9],segrtl); /*Transpose segment rotation matrix*/

                for (ia = 0; ia < 3; ia++)
                {
                                        fdg[2][ia]= (-1)*fpg[1][ia];
                                        tdg[2][ia]= (-1)*tpg[1][ia];
                                        rd1r[ia]= segrd[2][ia];
                                        rp1r[ia]= segrp[2][ia];
                                        fdg[9][ia]= (-1)*fpg[8][ia];
                                        tdg[9][ia]= (-1)*tpg[8][ia];
                                        rd1l[ia]= segrd[9][ia];
                                        rp1l[ia]= segrp[9][ia];
                }

                for (ia = 0; ia < 3; ia++)
                {
                fpg[2][ia] = -segmas[2]*g[ia]-fdg[2][ia]+segmas[2]*la[2][i][ia];
                fpg[9][ia] = -segmas[9]*g[ia]-fdg[9][ia]+segmas[9]*la[9][i][ia];
                }

/* Change forces and torques into bcs coordinates */
                                        gmprd1(segrtr,fpg[2],fp[2]);
                                        gmprd1(segrtl,fpg[9],fp[9]);
                                        gmprd1(segrtr,fdg[2],fd[2]);
                                        gmprd1(segrtl,fdg[9],fd[9]);
                                        gmprd1(segrtr,tdg[2],td[2]);
                                        gmprd1(segrtl,tdg[9],td[9]);
                                        gmprd1(segrtr,av[2][i],av[2][i]);
                                        gmprd1(segrtr,aa[2][i],aa[2][i]);
                                        gmprd1(segrtl,av[9][i],av[9][i]);
                                        gmprd1(segrtl,aa[9][i],aa[9][i]);

                                        crossp(rd1r,fd[2],rdcfdr);
                                        crossp(rd1l,fd[9],rdcfdl);
                                        crossp(rp1r,fp[2],rpcfpr);
                                        crossp(rp1l,fp[9],rpcfpl);

                                        gmprd3(arrrot[2],pinertia[2],temp33r);
```

```
                    gmprd3(arrrot[9],pinertia[9],temp33l);
                    gmprd3(temp33r,segrtr,inertia[2]);
                    gmprd3(temp33l,segrtl,inertia[9]);
                    gmprd1(inertia[2],aa[2][i],iaa[2]);
                    gmprd1(inertia[9],aa[9][i],iaa[9]);
                    gmprd1(inertia[2],av[2][i],iav[2]);
                    gmprd1(inertia[9],av[9][i],iav[9]);
                    crossp(av[2][i],iav[2], avciav[2]);
                    crossp(av[9][i],iav[9], avciav[9]);

                    for (ia = 0; ia < 3; ia++)
                    {
                            tdyn[2][ia] = avciav[2][ia] + iaa[2][ia];
                            tdyn[9][ia] = avciav[9][ia] + iaa[9][ia];
                    }


                    for (ia =0; ia < 3; ia++)
                    {
                            tp[2][ia]= -(rdcfdr[ia]+td[2][ia]
                                            +rpcfpr[ia]-tdyn[2][ia]);
                            tp[9][ia]= -(rdcfdl[ia]+td[9][ia]
                                            +rpcfpl[ia]-tdyn[9][ia]);
                    }

                    gmprd1(arrrot[2],tp[2],tpg[2]);
                    gmprd1(arrrot[9],tp[9],tpg[9]);
                    gmprd1(arrrot[2],fp[2],fpg[2]);
                    gmprd1(arrrot[9],fp[9],fpg[9]);

if ((arrpos[2][0] == bad) || (arrpos[2][1] == bad) || (arrpos[2][2] == bad))
        {
                for (ik = 0; ik < 3; ik++)
                {
                        for (ia=2;ia<4;ia++)
                        {
                        fd[ia][ik]=0.0;
                        td[ia][ik]=0.0;
                        fp[ia][ik]=0.0;
                        tp[ia][ik]=0.0;
                        fdg[ia][ik]=0.0;
                        tdg[ia][ik]=0.0;
                        fpg[ia][ik]=0.0;
                        tpg[ia][ik]=0.0;
                        arrpos[3][0] == bad;
                        arrpos[3][1] == bad;
                        arrpos[3][2] == bad;
                        }
                }
        }
if ((arrpos[9][0] == bad) || (arrpos[9][1] == bad) || (arrpos[9][2] == bad))
        {
                for (ik = 0; ik < 3; ik++)
                {
```

```
                                        for (ia=9;ia<11;ia++)
                                        {
                                        fd[ia][ik]=0.0;
                                        td[ia][ik]=0.0;
                                        fp[ia][ik]=0.0;
                                        tp[ia][ik]=0.0;
                                        fdg[ia][ik]=0.0;
                                        tdg[ia][ik]=0.0;
                                        fpg[ia][ik]=0.0;
                                        tpg[ia][ik]=0.0;
                                        arrpos[10][0] == bad;
                                        arrpos[10][1] == bad;
                                        arrpos[10][2] == bad;
                                        }
                                }
                        }

/* Do segment ([5] & [12]) (arms) calculations */
                if ((arrpos[5][0] == bad) || (arrpos[5][1] == bad) || (arrpos[5][2] == bad))
                                                {
                                                        for (ik = 0; ik < 3; ik++)
                                                        {
                                                        fd[5][ik] = 0.0;

                                                        td[5][ik]=0.0;
                                                        fp[5][ik]=0.0;
                                                        tp[5][ik]=0.0;
                                                        fdg[5][ik]=0.0;
                                                        tdg[5][ik]=0.0;
                                                        fpg[5][ik]=0.0;
                                                        tpg[5][ik]=0.0;
                                                        fd15[ik] = 0.0;

                                                        td15[ik]=0.0;
                                                        td155[ik]=0.0;
                                                        fp15[ik]=0.0;
                                                        tp15[ik]=0.0;
                                                        fdg15[ik]=0.0;
                                                        tdg15[ik]=0.0;
                                                        fpg15[ik]=0.0;
                                                        tpg15[ik]=0.0;
                                                        }
                                                }
        if ((arrpos[12][0] == bad) || (arrpos[12][1] == bad) || (arrpos[12][2] == bad))
                                                {
                                                        for (ik = 0; ik < 3; ik++)
                                                        {
                                                        fd[12][ik] = 0.0;

                                                        td[12][ik]=0.0;
                                                        fp[12][ik]=0.0;
                                                        tp[12][ik]=0.0;
                                                        fdg[12][ik]=0.0;
                                                        tdg[12][ik]=0.0;
```

```
                                        fpg[12][ik]=0.0;
                                        tpg[12][ik]=0.0;
                                        td1512[ik]=0.0;
                                        fd15[ik] = 0.0;

                                        td15[ik]=0.0;
                                        fp15[ik]=0.0;
                                        tp15[ik]=0.0;
                                        fdg15[ik]=0.0;
                                        tdg15[ik]=0.0;
                                        fpg15[ik]=0.0;
                                        tpg15[ik]=0.0;
                                        }
                                }

gmtra(arrot[5],segrtr); /* Transpose segment rotation matrix */
gmtra(arrot[12],segrtl); /* Transpose segment rotation matrix */

for (ia = 0; ia < 3; ia++)
        {
                                                fdg[5][ia]= 0;
                                                tdg[5][ia]= 0;
                                                rd1r[ia]= segrd[5][ia];
                                                rp1r[ia]= segrp[5][ia];
                                                fdg[12][ia]= 0;
                                                tdg[12][ia]= 0;
                                                rd1l[ia]= segrd[12][ia];
                                                rp1l[ia]= segrp[12][ia];
        }

for (ia = 0; ia < 3; ia++)
        {
        fpg[5][ia] = (-1)*segmas[5]*g[ia]-fdg[5][ia]
                        +segmas[5]*la[5][i][ia];
        fpg[12][ia] =(-1)*segmas[12]*g[ia]-fdg[12][ia]
                        +segmas[12]*la[12][i][ia];
                                }

/* Change forces and torques into bcs coordinates */

                                gmprd1(segrtr,fpg[5],fp[5]);
                                gmprd1(segrtl,fpg[12],fp[12]);
                                gmprd1(segrtr,fdg[5],fd[5]);
                                gmprd1(segrtl,fdg[12],fd[12]);
                                gmprd1(segrtr,tdg[5],td[5]);
                                gmprd1(segrtl,tdg[12],td[12]);
                                gmprd1(segrtr,av[5][i],av[5][i]);
                                gmprd1(segrtr,aa[5][i],aa[5][i]);
                                gmprd1(segrtl,av[12][i],av[12][i])
                                gmprd1(segrtl,aa[12][i],aa[12][i]);

                                crossp(rd1r,fd[5],rdcfdr);
                                crossp(rd1l,fd[12],rdcfdl);
                                crossp(rp1r,fp[5],rpcfpr);
```

```
crossp(rp1l,fp[12],rpcfpl);

gmprd3(arrrot[5],pinertia[5],temp33r);
gmprd3(arrrot[12],pinertia[12],temp33l);
gmprd3(temp33r,segrtr,inertia[5]);
gmprd3(temp33l,segrtl,inertia[12]);
gmprd1(inertia[5],aa[5][i],iaa[5]);
gmprd1(inertia[12],aa[12][i],iaa[12]);
gmprd1(inertia[5],av[5][i],iav[5]);
gmprd1(inertia[12],av[12][i],iav[12]);
crossp(av[5][i],iav[5], avciav[5]);
crossp(av[12][i],iav[12], avciav[12]);

for (ia = 0; ia < 3; ia++)
{
tdyn[5][ia] = avciav[5][ia] + iaa[5][ia];
tdyn[12][ia] = avciav[12][ia] + iaa[12][ia];
}

for (ia = 0; ia < 3; ia++)
{
tp[5][ia]= (-1)*(rdcfdr[ia]+td[5][ia]
+rpcfpr[ia]-tdyn[5][ia]);
tp[12][ia]= (-1)*(rdcfdl[ia]+td[12][ia]
+rpcfpl[ia]-tdyn[12][ia]);
}

gmprd1(arrrot[5], tp[5], tpg[5]);
gmprd1(arrrot[12], tp[12], tpg[12]);
gmprd1(arrrot[5], fp[5], fpg[5]);
gmprd1(arrrot[12], fp[12], fpg[12]);

if ((arrpos[5][0] == bad) || (arrpos[5][1] == bad) || (arrpos[5][2] == bad))
{
        for (ik = 0; ik < 3; ik++)
        {
        fd[5][ik] = 0.0;

        td[5][ik]=0.0;
        fp[5][ik]=0.0;
        tp[5][ik]=0.0;
        fdg[5][ik]=0.0;
        tdg[5][ik]=0.0;
        fpg[5][ik]=0.0;
        tpg[5][ik]=0.0;
        fd15[ik] = 0.0;

        td15[ik]=0.0;
        td155[ik]=0.0;
        fp15[ik]=0.0;
        tp15[ik]=0.0;
        fdg15[ik]=0.0;
        tdg15[ik]=0.0;
        fpg15[ik]=0.0;
```

```
                                        tpg15[ik]=0.0;
                                        }
                                }
        if ((arrpos[12][0] == bad) || (arrpos[12][1] == bad) || (arrpos[12][2] == bad))
                                {
                                        for (ik = 0; ik < 3; ik++)
                                        {
                                        fd[12][ik] = 0.0;

                                        td[12][ik]=0.0;
                                        fp[12][ik]=0.0;
                                        tp[12][ik]=0.0;
                                        fdg[12][ik]=0.0;
                                        tdg[12][ik]=0.0;
                                        fpg[12][ik]=0.0;
                                        tpg[12][ik]=0.0;
                                        td1512[ik]=0.0;
                                        fd15[ik] = 0.0;

                                        td15[ik]=0.0;
                                        fp15[ik]=0.0;
                                        tp15[ik]=0.0;
                                        fdg15[ik]=0.0;
                                        tdg15[ik]=0.0;
                                        fpg15[ik]=0.0;
                                        tpg15[ik]=0.0;
                                        }
                                }

/* Do segment [6] (head) calculations */
                if ((arrpos[6][0] == bad) || (arrpos[6][1] == bad) || (arrpos[6][2] == bad))
                                        {
                                                for (ik = 0; ik < 3; ik++)
                                                {
                                                fd[6][ik] = 0.0;

                                                td[6][ik]=0.0;
                                                fp[6][ik]=0.0;
                                                tp[6][ik]=0.0;
                                                fdg[6][ik]=0.0;
                                                tdg[6][ik]=0.0;
                                                fpg[6][ik]=0.0;
                                                tpg[6][ik]=0.0;
                                                td15[ik]=0.0;
                                                td156[ik]=0.0;
                                                fp15[ik]=0.0;
                                                tp15[ik]=0.0;
                                                fdg15[ik]=0.0;
                                                tdg15[ik]=0.0;
                                                fpg15[ik]=0.0;
                                                tpg15[ik]=0.0;
                                                }
                                        }
```

```
                                        /*Transpose segment rotation matrix*/
                                        gmtra(arrrot[6],segrtr);
                                          for (ia = 0; ia < 3; ia++)
                                          {
                                                  fdg[6][ia]= 0;
                                                  tdg[6][ia]= 0;
                                                  rd1r[ia]= segrd[6][ia];
                                                  rp1r[ia]= segrp[6][ia];
                                          }

                                          for (ia = 0; ia < 3; ia++)
                                          {
                                          fpg[6][ia] = (-1)*segmas[6]*g[ia]-fdg[6][ia]
                                                          +segmas[6]*la[6][i][ia];
                                          }


/* Change forces and torques into bcs coordinates */
                                          gmprd1(segrtr,fpg[6],fp[6]);
                                          gmprd1(segrtr,fdg[6],fd[6]);
                                          gmprd1(segrtr,tdg[6],td[6]);
                                          gmprd1(segrtr,av[6][i],av[6][i]);
                                          gmprd1(segrtr,aa[6][i],aa[6][i]);

                                          crossp(rd1r,fd[6],rdcfdr);
                                          crossp(rp1r,fp[6],rpcfpr);

                                          gmprd3(arrrot[6],pinertia[6],temp33r);
                                  gmprd3(temp33r,segrtr,inertia[6]);
                                          gmprd1(inertia[6],aa[6][i],iaa[6]);
                                          gmprd1(inertia[6],av[6][i],iav[6]);

                                          crossp(av[6][i],iav[6], avciav[6]);

                                          for (ia = 0; ia < 3; ia++)
                                          {
                                          tdyn[6][ia] = avciav[6][ia] + iaa[6][ia];
                                          }


                                          for (ia = 0; ia < 3; ia++)
                                          {
                                          tp[6][ia]= (-1)*(rdcfdr[ia]+td[6][ia]
                                                          +rpcfpr[ia]-tdyn[6][ia]);
                                          }

                                          gmprd1(arrrot[6], tp[6], tpg[6]);
                                          gmprd1(arrrot[6], fp[6], fpg[6]);

        if ((arrpos[6][0] == bad) || (arrpos[6][1] == bad) || (arrpos[6][2] == bad))
                                                  {
                                                          for (ik = 0; ik < 3; ik++)
                                                          {
```

```
                                                fd[6][ik] = 0.0;

                                                td[6][ik]=0.0;
                                                fp[6][ik]=0.0;
                                                tp[6][ik]=0.0;
                                                fdg[6][ik]=0.0;
                                                tdg[6][ik]=0.0;
                                                fpg[6][ik]=0.0;
                                                tpg[6][ik]=0.0;
                                                td15[ik]=0.0;
                                                td156[ik]=0.0;
                                                fp15[ik]=0.0;
                                                tp15[ik]=0.0;
                                                fdg15[ik]=0.0;
                                                tdg15[ik]=0.0;
                                                fpg15[ik]=0.0;
                                                tpg15[ik]=0.0;
                                                }
                                        }

/* Do segment ([4] & [11]): right & left trunk, 15:combined trunk */
        if ((arrpos[4][0] == bad) || (arrpos[4][1] == bad) || (arrpos[4][2] == bad))
                                                {
                                                for (ik = 0; ik < 3; ik++)
                                                        {
                                                        fd[5][ik] = 0.0;

                                                        td[5][ik]=0.0;
                                                        fp[5][ik]=0.0;
                                                        tp[5][ik]=0.0;
                                                        fdg[5][ik]=0.0;
                                                        tdg[5][ik]=0.0;
                                                        fpg[5][ik]=0.0;
                                                        tpg[5][ik]=0.0;
                                                        td15[ik]=0.0;
                                                        td155[ik]=0.0;
                                                        td156[ik]=0.0;
                                                        td1512[ik]=0.0;
                                                        fp15[ik]=0.0;
                                                        tp15[ik]=0.0;
                                                        fdg15[ik]=0.0;
                                                        tdg15[ik]=0.0;
                                                        fpg15[ik]=0.0;
                                                        tpg15[ik]=0.0;
                                                        }
                                                }
        if ((arrpos[11][0] == bad) || (arrpos[11][1] == bad) || (arrpos[11][2] == bad))
                                                {
                                                for (ik = 0; ik < 3; ik++)
                                                        {
                                                        fd[11][ik] = 0.0;

                                                        td[11][ik]=0.0;
                                                        fp[11][ik]=0.0;
```

```
                                                    tp[11][ik]=0.0;
                                                    fdg[11][ik]=0.0;
                                                    tdg[11][ik]=0.0;
                                                    fpg[11][ik]=0.0;
                                                    tpg[11][ik]=0.0;
                                                    td15[ik]=0.0;
                                                    td155[ik]=0.0;
                                                    td156[ik]=0.0;
                                                    td1512[ik]=0.0;
                                                    fp15[ik]=0.0;
                                                    tp15[ik]=0.0;
                                                    fdg15[ik]=0.0;
                                                    tdg15[ik]=0.0;
                                                    fpg15[ik]=0.0;
                                                    tpg15[ik]=0.0;
                                                    }
                                                }

for (ia=0;ia<3;ia++)
{
        aa15[ia] = (aa[4][i][ia]+aa[11][i][ia])/2.0;
        av15[ia] = (av[4][i][ia]+av[11][i][ia])/2.0;
        la15[ia] = (la[4][i][ia]+la[11][i][ia])/2.0;
        lv15[ia] = (lv[4][i][ia]+lv[11][i][ia])/2.0;
}
segmas15 = (segmas[4]+segmas[11])/2.0;

gmtra(arrrot[4],segrtr);/*Transpose segment rotation matrix*/
gmtra(arrrot[11],segrtl);/*Transpose segment rotation matrix*/
gmtra(arrrot15,segrtt);/*Transpose segment rotation matrix*/

for (ia = 0; ia < 3; ia++)
{
fdg[4][ia]= (-1)*(fpg[5][ia]);/*right trunk segment reaction forces w/right arm*/
tdg[4][ia]= (-1)*(tpg[5][ia]);/*right trunk segment reaction torques w/right arm*/
rd1r[ia]= segrd[4][ia];
rp1r[ia]= segrp[4][ia];
fdg[11][ia]= (-1)*(fpg[12][ia]);
tdg[11][ia]= (-1)*(tpg[12][ia]);
rd1l[ia]= segrd[11][ia];
rp1l[ia]= segrp[11][ia];
fdg155[ia]= (-1)*(fpg[5][ia]);
fdg1512[ia]= (-1)*(fpg[12][ia]);
fdg156[ia]= (-1)*(fpg[6][ia]);
tdg155[ia]= (-1)*(tpg[5][ia]);
tdg1512[ia]= (-1)*(tpg[12][ia]);
tdg156[ia]= (-1)*(tpg[6][ia]);
rp1t[ia]= (segrd[11][ia]+segrd[4][ia])/2.0;/*distance from trunk cg to back joint*/
rcgh[ia]= (segrp[11][ia]+segrp[4][ia])/2.0;/*distance from trunk cg to neck*/
}

gmprd1(segrtt,rp1t,tempr);
gmprd1(segrtt,rcgh,templ);
```

```
        for (ia = 0; ia < 3; ia++)
        {
        fpg[4][ia] = (-1)*segmas[4]*g[ia]-fdg[4][ia]+segmas[4]*la[4][i][ia];
        fpg[11][ia] = (-1)*segmas[11]*g[ia]-fdg[11][ia]+segmas[11]*la[11][i][ia];
        fpg15[ia] = (-1)*segmas15*g[ia]-fdg155[ia]-fdg1512[ia]-fdg156[ia]+segmas15*la15[ia];
        }

/* Change forces and torques into bcs coordinates */
        gmprd1(segrtr,fpg[4],fp[4]);
        gmprd1(segrtl,fpg[11],fp[11]);
        gmprd1(segrtt,fpg15,fp15);
        gmprd1(segrtr,fdg[4],fd[4]);
        gmprd1(segrtl,fdg[11],fd[11]);
        gmprd1(segrtt,fdg155,fd155)
        gmprd1(segrtt,fdg1512,fd1512);
        gmprd1(segrtt,fdg156,fd156);
        gmprd1(segrtr,tdg[4],td[4]);
        gmprd1(segrtl,tdg[11],td[11]);\
        gmprd1(segrtt,tdg155,td155);
        gmprd1(segrtt,tdg1512,td1512);
        gmprd1(segrtt,tdg156,td156);
        gmprd1(segrtr,av[4][i],av[4][i]);
        gmprd1(segrtr,aa[4][i],aa[4][i]);
        gmprd1(segrtl,av[11][i],av[11][i]);
        gmprd1(segrtl,aa[11][i],aa[11][i]);
        gmprd1(segrtt,av15,av15);
        gmprd1(segrtt,aa15,aa15);

        crossp(rp1r,fp[4],rpcfpr);
        crossp(rp1l,fp[11],rpcfpl);
        crossp(rd1r,fd[4],rdcfdr);
        crossp(rd1l,fd[11],rdcfdl);
        crossp(rd1tr,fd155,rdcfd155);
        crossp(rd1tl,fd1512,rdcfd1512);
        crossp(rcgh,fd156,rdcfd156);
        crossp(rp1t,fp15,rpcfpt);

        gmprd3(arrrot[4],pinertia[4],temp33r);
        gmprd3(arrrot[11],pinertia[11],temp33l);
        gmprd3(arrrot15,pinertia[11],temp33t); /* note: pinertia[11]=pinertia[4]*/
        gmprd3(temp33r,segrtr,inertia[4]);
        gmprd3(temp33l,segrtl,inertia[11]);
        gmprd3(temp33t,segrtt,inertia15);
        gmprd1(inertia[4],aa[4][i],iaa[4]);
        gmprd1(inertia[11],aa[11][i],iaa[11]);
        gmprd1(inertia15,aa15,iaa15);
        gmprd1(inertia[4],av[4][i],iav[4]);
        gmprd1(inertia[11],av[11][i],iav[11]);
        gmprd1(inertia15,av15,iav15);

        crossp(av[4][i],iav[4], avciav[4]);
        crossp(av[11][i],iav[11], avciav[11]);
        crossp(av15,iav15, avciav15);
```

```
for (ia = 0; ia < 3; ia++)
        {
        tdyn[4][ia] = avciav[4][ia] + iaa[4][ia];
        tdyn[11][ia] = avciav[11][ia] + iaa[11][ia];
        tdyn15[ia] = avciav15[ia] + iaa15[ia];
        }

for (ia = 0; ia < 3; ia++)
        {
        tp[4][ia]= (-1)*(rdcfdr[ia]+td[4][ia]+rpcfpr[ia]-tdyn[4][ia]);
        tp[11][ia]= (-1)*(rdcfdl[ia]+td[11][ia]+rpcfpl[ia]-tdyn[11][ia]);
        tp15[ia]= (-1)*(rdcfd155[ia]+rdcfd1512[ia]+rdcfd156[ia]+
        td155[ia]+td1512[ia]+td156[ia]+rpcfpt[ia]-tdyn15[ia]);
        }

gmprd1(arrrot[4], tp[4], tpg[4]);
gmprd1(arrrot[11], tp[11], tpg[11]);
gmprd1(arrrot15, tp15, tpg15);
gmprd1(arrrot[4], fp[4], fpg[4]);
gmprd1(arrrot[11], fp[11], fpg[11]);
gmprd1(arrrot15, fp15, fpg15);

        if ((arrpos[4][0] == bad) || (arrpos[4][1] == bad) || (arrpos[4][2] == bad))
                                                                {
                                                                for (ik = 0; ik < 3; ik++)
                                                                        {
                                                                        fd[5][ik] = 0.0;

                                                                        td[5][ik]=0.0;
                                                                        fp[5][ik]=0.0;
                                                                        tp[5][ik]=0.0;
                                                                        fdg[5][ik]=0.0;
                                                                        tdg[5][ik]=0.0;
                                                                        fpg[5][ik]=0.0;
                                                                        tpg[5][ik]=0.0;
                                                                        td15[ik]=0.0;
                                                                        td155[ik]=0.0;
                                                                        td156[ik]=0.0;
                                                                        td1512[ik]=0.0;
                                                                        fp15[ik]=0.0;
                                                                        tp15[ik]=0.0;
                                                                        fdg15[ik]=0.0;
                                                                        tdg15[ik]=0.0;
                                                                        fpg15[ik]=0.0;
                                                                        tpg15[ik]=0.0;
                                                                        }
                                                                }
        if ((arrpos[11][0] == bad) || (arrpos[11][1] == bad) || (arrpos[11][2] == bad))
                                                                {
                                                                for (ik = 0; ik < 3; ik++)
                                                                        {
                                                                        fd[11][ik] = 0.0;

                                                                        td[11][ik]=0.0;
```

```
                                                    fp[11][ik]=0.0;
                                                    tp[11][ik]=0.0;
                                                    fdg[11][ik]=0.0;
                                                    tdg[11][ik]=0.0;
                                                    fpg[11][ik]=0.0;
                                                    tpg[11][ik]=0.0;
                                                    td15[ik]=0.0;
                                                    td155[ik]=0.0;
                                                    td156[ik]=0.0;
                                                    fd1512[ik] = 0.0;

                                                    td1512[ik]=0.0;
                                                    fp15[ik]=0.0;
                                                    tp15[ik]=0.0;
                                                    fdg15[ik]=0.0;
                                                    tdg15[ik]=0.0;
                                                    fpg15[ik]=0.0;
                                                    tpg15[ik]=0.0;
                                                    }
                                                }
/* Do segment (3&10=13) calculations (pelvis from the upper body forces and torques)*/
        if ((arrpos[3][0] == bad) || (arrpos[3][1] == bad) || (arrpos[3][2] == bad))
                                            {
                                                    for (ik = 0; ik < 3; ik++)
                                                    {
                                                    fd[3][ik] = 0.0;

                                                    td[3][ik]=0.0;
                                                    fp[3][ik]=0.0;
                                                    tp[3][ik]=0.0;
                                                    fdg[3][ik]=0.0;
                                                    tdg[3][ik]=0.0;
                                                    fpg[3][ik]=0.0;
                                                    tpg[3][ik]=0.0;
                                                    }
                                            }
    if ((arrpos[10][0] == bad) || (arrpos[10][1] == bad) || (arrpos[10][2] == bad))
                                            {
                                                    for (ik = 0; ik < 3; ik++)
                                                    {
                                                    fd[10][ik] = 0.0;

                                                    td[10][ik]=0.0;
                                                    fp[10][ik]=0.0;
                                                    tp[10][ik]=0.0;
                                                    fdg[10][ik]=0.0;
                                                    tdg[10][ik]=0.0;
                                                    fpg[10][ik]=0.0;
                                                    tpg[10][ik]=0.0;
                                                    }
                                            }

                                                    gmtra(arrrot13,segrtt);
                                                    gmtra(arrrot[3],segrtr);
```

```
                                        gmtra(arrrot[10],segrtl);
                                        for (ia = 0; ia < 3; ia++)
                                        {
                                                fdg[3][ia] = -fpg[2][ia];
                                                tdg[3][ia] = -tpg[2][ia];
                                                fdg[10][ia] = -fpg[9][ia];
                                                tdg[10][ia] = -tpg[9][ia];
                                                fdg13[ia] = -fpg15[ia];
                                                tdg13[ia] = -tpg15[ia];
                                        }

/* Change forces and torques into bcs coordinates */
                                        gmprd1(segrtr,fdg[3],fd[3]);
                                        gmprd1(segrtl,fdg[10],fd[10]);
                                        gmprd1(segrtr,tdg[3],td[3]);
                                        gmprd1(segrtl,tdg[10],td[10]);
                                        gmprd1(segrtt,fdg13,fd13);
                                        gmprd1(segrtt,tdg13,td13);
                                        gmprd1(segrtr,av[3][i],av[3][i]);
                                        gmprd1(segrtr,aa[3][i],aa[3][i]);
                                        gmprd1(segrtl,av[10][i],av[10][i]);
                                        gmprd1(segrtl,aa[10][i],aa[10][i]);
                                        gmprd1(segrtt,av13,av13);
                                        gmprd1(segrtt,aa13,aa13);

                                        for (ia = 0; ia < 3; ia++)
                                        {
                                                fpg[3][ia]=0;
                                                tpg[3][ia]=0;
                                                fp[3][ia]=0;
                                                tp[3][ia]=0;
                                                fpg[10][ia]=0;
                                                tpg[10][ia]=0;
                                                fp[10][ia]=0;
                                                tp[10][ia]=0;
                                                fpg13[ia]=0;
                                                tpg13[ia]=0;
                                                fp13[ia]=0;
                                                tp13[ia]=0;
                                        }
if ((arrpos[3][0] == bad) || (arrpos[3][1] == bad) || (arrpos[3][2] == bad))
                                                {
                                        for (ik = 0; ik < 3; ik++)
                                        {
                                        fd[3][ik] = 0.0;

                                        td[3][ik]=0.0;
                                        fp[3][ik]=0.0;
                                        tp[3][ik]=0.0;
                                        fdg[3][ik]=0.0;
                                        tdg[3][ik]=0.0;
                                        fpg[3][ik]=0.0;
                                        tpg[3][ik]=0.0;
                                        }
```

```
                                        }
if ((arrpos[10][0] == bad) || (arrpos[10][1] == bad) || (arrpos[10][2] == bad))
                                        {
                                                for (ik = 0; ik < 3; ik++)
                                                {
                                                fd[10][ik] = 0.0;

                                                td[10][ik]=0.0;
                                                fp[10][ik]=0.0;
                                                tp[10][ik]=0.0;
                                                fdg[10][ik]=0.0;
                                                tdg[10][ik]=0.0;
                                                fpg[10][ik]=0.0;
                                                tpg[10][ik]=0.0;
                                                }
                                        }

if ((arrpos[5][0] == bad) || (arrpos[5][1] == bad) || (arrpos[5][2] == bad))
                                                {
                                                        for (ik = 0; ik < 3; ik++)
                                                        {
                                                        fd[5][ik] = 0.0;

                                                        td[5][ik]=0.0;
                                                        fp[5][ik]=0.0;
                                                        tp[5][ik]=0.0;
                                                        fdg[5][ik]=0.0;
                                                        tdg[5][ik]=0.0;
                                                        fpg[5][ik]=0.0;
                                                        tpg[5][ik]=0.0;
                                                        fd15[ik] = 0.0;

                                                        td15[ik]=0.0;
                                                        td155[ik]=0.0;
                                                        fp15[ik]=0.0;
                                                        tp15[ik]=0.0;
                                                        fdg15[ik]=0.0;
                                                        tdg15[ik]=0.0;
                                                        fpg15[ik]=0.0;
                                                        tpg15[ik]=0.0;
                                                        }
                                                }
if ((arrpos[12][0] == bad) || (arrpos[12][1] == bad) || (arrpos[12][2] == bad))
                                                {
                                                        for (ik = 0; ik < 3; ik++)
                                                        {
                                                        fd[12][ik] = 0.0;

                                                        td[12][ik]=0.0;
                                                        fp[12][ik]=0.0;
                                                        tp[12][ik]=0.0;
                                                        fdg[12][ik]=0.0;
                                                        tdg[12][ik]=0.0;
                                                        fpg[12][ik]=0.0;
```

```
                                        tpg[12][ik]=0.0;
                                        fd1512[ik] = 0.0;

                                        td1512[ik]=0.0;
                                        fd15[ik] = 0.0;

                                        td15[ik]=0.0;
                                        fp15[ik]=0.0;
                                        tp15[ik]=0.0;
                                        fdg15[ik]=0.0;
                                        tdg15[ik]=0.0;
                                        fpg15[ik]=0.0;
                                        tpg15[ik]=0.0;
                                        }
                                }

        if ((arrpos[6][0] == bad) || (arrpos[6][1] == bad) || (arrpos[6][2] == bad))
                                                {
                                                        for (ik = 0; ik < 3; ik++)
                                                        {
                                                        fd[6][ik] = 0.0;

                                                        td[6][ik]=0.0;
                                                        fp[6][ik]=0.0;
                                                        tp[6][ik]=0.0;
                                                        fdg[6][ik]=0.0;
                                                        tdg[6][ik]=0.0;
                                                        fpg[6][ik]=0.0;
                                                        tpg[6][ik]=0.0;
                                                        td15[ik]=0.0;
                                                        td156[ik]=0.0;
                                                        fp15[ik]=0.0;
                                                        tp15[ik]=0.0;
                                                        fdg15[ik]=0.0;
                                                        tdg15[ik]=0.0;
                                                        fpg15[ik]=0.0;
                                                        tpg15[ik]=0.0;
                                                        }
                                                }

/* Rotate the bcs coordinate forces into the coordinates of the limb sement */
/* Foot */
        for (ia = 0; ia < 3; ia++)
        {
                tempr[ia] = fp[0][ia];
                templ[ia] = fp[7][ia];
        }
        gmprd1(rotseg[0],tempr,fp[0]);
        gmprd1(rotseg[7],templ,fp[7]);
        for (ia = 0; ia < 3; ia++)
        {
                tempr[ia]=fd[0][ia];
                templ[ia]=fd[7][ia];
```

```
        }
        gmprd1(rotseg[0],tempr,fd[0]);
        gmprd1(rotseg[7],templ,fd[7]);
        for (ia = 0; ia < 3; ia++)
        {
                tempr[ia]=tp[0][ia];
                templ[ia]=tp[7][ia];
        }
        gmprd1(rotseg[0],tempr,tp[0]);
        gmprd1(rotseg[7],templ,tp[7]);
        for (ia = 0; ia < 3; ia++)
        {
                tempr[ia]=td[0][ia];
                templ[ia]=td[7][ia];

        }
        gmprd1(rotseg[0],tempr,td[0]);
        gmprd1(rotseg[7],templ,td[7]);

/* shank */
        for (ia =0; ia < 3; ia++)
        {
                tempr[ia] = fp[1][ia];
                templ[ia] = fp[8][ia];
        }
        gmprd1(rotseg[1],tempr,fp[1]);
        gmprd1(rotseg[8],templ,fp[8]);
        for (ia = 0; ia < 3; ia++)
        {
                tempr[ia]=fd[1][ia];
                templ[ia]=fd[8][ia];
        }
        gmprd1(rotseg[1],tempr,fd[1]);
        gmprd1(rotseg[8],templ,fd[8]);
        for (ia = 0; ia < 3; ia++)
        {
                tempr[ia]=tp[1][ia];
                templ[ia]=tp[8][ia];
        }
        gmprd1(rotseg[1],tempr,tp[1]);
        gmprd1(rotseg[8],templ,tp[8]);
        for (ia = 0; ia < 3; ia++)
        {
                tempr[ia]=td[1][ia];
                templ[ia]=td[8][ia];
        }
        gmprd1(rotseg[1],tempr,td[1]);
        gmprd1(rotseg[8],templ,td[8]);

/* thigh */
        for (ia = 0; ia < 3; ia++)
        {
                tempr[ia] = fp[2][ia];
                templ[ia] = fp[9][ia];
```

```
}
gmprd1(rotseg[2],tempr,fp[2]);
gmprd1(rotseg[9],templ,fp[9]);
for (ia = 0; ia < 3; ia++)
{
        tempr[ia]=fd[2][ia];
        templ[ia]=fd[9][ia];
}
gmprd1(rotseg[2],tempr,fd[2]);
gmprd1(rotseg[9],templ,fd[9]);
for (ia = 0; ia < 3; ia++)
{
        tempr[ia]=tp[2][ia];
        templ[ia]=tp[9][ia];
}
gmprd1(rotseg[2],tempr,tp[2]);
gmprd1(rotseg[9],templ,tp[9]);
for (ia = 0; ia < 3; ia++)
{
        tempr[ia]=td[2][ia];
        templ[ia]=td[9][ia];
}
gmprd1(rotseg[2],tempr,td[2]);
gmprd1(rotseg[9],templ,td[9]);
for (ia = 0; ia < 3; ia++)

/* pelvis (from lower body forces and torques) */

        for (ia = 0; ia < 3; ia++)
        {
                tempr[ia] = fp[3][ia];
                templ[ia] = fp[10][ia];
        }
        gmprd1(rotseg[3],tempr,fp[3]);
        gmprd1(rotseg[10],templ,fp[10]);
        for (ia = 0; ia < 3; ia++)
        {
                tempr[ia]=fd[3][ia];
                templ[ia]=fd[10][ia];
        }
        gmprd1(rotseg[3],tempr,fd[3]);
        gmprd1(rotseg[10],templ,fd[10]);
        for (ia = 0; ia < 3; ia++)
        {
                tempr[ia]=tp[3][ia];
                templ[ia]=tp[10][ia];
        }
        gmprd1(rotseg[3],tempr,tp[3]);
        gmprd1(rotseg[10],templ,tp[10]);
        for (ia = 0; ia < 3; ia++)
        {
                tempr[ia]=td[3][ia];
                templ[ia]=td[10][ia];
        }
```

```
        gmprd1(rotseg[3],tempr,td[3]);
        gmprd1(rotseg[10],templ,td[10]);

/* pelvis (from upper body forces & torques) */

        for (ia = 0; ia < 3; ia++)
        {
                tempr[ia] = fp13[ia];
        }
        gmprd1(rotseg13,tempr,fp13);
        for (ia = 0; ia < 3; ia++)
        {
                tempr[ia]=fd13[ia];
        }
        gmprd1(rotseg13,tempr,fd13);
        for (ia = 0; ia < 3; ia++)
        {
                tempr[ia]=tp13[ia];
        }
        gmprd1(rotseg13,tempr,tp13);
        for (ia = 0; ia < 3; ia++)
        {
                tempr[ia]=td13[ia];
        }
        gmprd1(rotseg13,tempr,td13);

/* arms */

        for (ia = 0; ia < 3; ia++)
        {
                tempr[ia] = fp[5][ia];
                templ[ia] = fp[12][ia];
        }
        gmprd1(rotseg[5],tempr,fp[5]);
        gmprd1(rotseg[12],templ,fp[12]);
        for (ia = 0; ia < 3; ia++)
        {
                tempr[ia]=fd[5][ia];
                templ[ia]=fd[12][ia];
        }
        gmprd1(rotseg[5],tempr,fd[5]);
        gmprd1(rotseg[12],templ,fd[12]);
        for (ia = 0; ia < 3; ia++)
        {
                tempr[ia]=tp[5][ia];
                templ[ia]=tp[12][ia];
        }
        gmprd1(rotseg[5],tempr,tp[5]);
        gmprd1(rotseg[12],templ,tp[12]);
        for (ia = 0; ia < 3; ia++)
        {
                tempr[ia]=td[5][ia];
                templ[ia]=td[12][ia];
        }
```

```
        gmprd1(rotseg[5],tempr,td[5]);
        gmprd1(rotseg[12],templ,td[12]);

/* head */

        for (ia = 0; ia < 3; ia++)
        {
                tempr[ia] = fp[6][ia];
        }
        gmprd1(rotseg[6],tempr,fp[6]);
        for (ia = 0; ia < 3; ia++)
        {
                tempr[ia]=fd[6][ia];
        }
        gmprd1(rotseg[6],tempr,fd[6]);
        for (ia = 0; ia < 3; ia++)
        {
                tempr[ia]=tp[6][ia];
        }
        gmprd1(rotseg[6],tempr,tp[6]);
        for (ia = 0; ia < 3; ia++)
        {
                tempr[ia]=td[6][ia];
        }
        gmprd1(rotseg[6],tempr,td[6]);

/* trunk */

        for (ia = 0; ia < 3; ia++)
        {
                tempr[ia] = fp[4][ia];
                templ[ia] = fp[11][ia];
                tempt[ia] = fp15[ia];
        }
        gmprd1(rotseg[4],tempr,fp[4]);
        gmprd1(rotseg[11],templ,fp[11]);
        gmprd1(rotseg15,tempt,fp15);
        for (ia = 0; ia < 3; ia++)
        {
                tempr[ia]=fd[4][ia];
                templ[ia]=fd[11][ia];
                tempt[ia]=fd155[ia];
        }
        gmprd1(rotseg[4],tempr,fd[4]);
        gmprd1(rotseg[11],templ,fd[11]);
        gmprd1(rotseg15,tempt,fd155);
        for (ia = 0; ia < 3; ia++)
        {
                tempt[ia]=fd1512[ia];
        }
        gmprd1(rotseg15,tempt,fd1512);
        for (ia = 0; ia < 3; ia++)
        {
                tempt[ia]=fd156[ia];
```

```
        }
        gmprd1(rotseg15,tempt,fd156);
        for (ia = 0; ia < 3; ia++)
        {
                tempr[ia]=tp[4][ia];
                templ[ia]=tp[11][ia];
                tempt[ia]=tp15[ia];
        }
        gmprd1(rotseg[4],tempr,tp[4]);
        gmprd1(rotseg[11],templ,tp[11]);
        gmprd1(rotseg15,tempt,tp15);

        for (ia = 0; ia < 3; ia++)
        {
                tempr[ia]=td[4][ia];
                templ[ia]=td[11][ia];
                tempt[ia]=td155[ia];
        }
        gmprd1(rotseg[4],tempr,td[4]);
        gmprd1(rotseg[11],templ,td[11]);
        gmprd1(rotseg15,tempt,td155);
        for (ia = 0; ia < 3; ia++)
        {
                tempt[ia]=td156[ia];
        }
        gmprd1(rotseg15,tempt,td156);
        for (ia = 0; ia < 3; ia++)
        {
                tempt[ia]=td1512[ia];
        }
        gmprd1(rotseg15,tempt,td1512);

/* Write calculated torques to output array */

/* Right Ankle, Right Knee, Right Hip */
        /* Right Ankle */
outarrr[18*i] = fp[0][0]*100.0/wgtnt; /* x component of force (ant-post direction)*/
        outarrr[18*(i)+1] = tp[0][2]*100.0/wgtnt;/* flexion torques */
outarrr[18*(i)+6] = -fp[0][1]*100.0/wgtnt; /* y component of force (vertical direction) */
        outarrr[18*(i)+7] = -tp[0][0]*100.0/wgtnt; /* abduction torques */
outarrr[18*(i)+12] = -fp[0][2]*100.0/wgtnt; /* z component of force (lateral direction) */
        outarrr[18*(i)+13] = tp[0][1]*100.0/wgtnt; /* rotation torques */

        /* Right Knee */
outarrr[18*(i)+2] = fp[1][0]*100.0/wgtnt; /* x component of force (ant-post direction)*/
        outarrr[18*(i)+3] = -tp[1][2]*100.0/wgtnt; /* flexion torques */
outarrr[18*(i)+8] = -fp[1][1]*100.0/wgtnt;/* y component of force (vertical direction) */
        outarrr[18*(i)+9] = tp[1][0]*100.0/wgtnt;/* abduction torques */
outarrr[18*(i)+14] = fp[1][2]*100.0/wgtnt; /* z component of force (lateral direction) */
        outarrr[18*(i)+15] =  tp[1][1]*100.0/wgtnt; /* rotation torques */

        /* Right Hip */
outarrr[18*(i)+4] =      fp[2][0]*100.0/wgtnt; /* x component of force (ant-post direction)*/
        outarrr[18*(i)+5] =      tp[2][2]*100.0/wgtnt; /* flexion torques */
```

```
outarrr[18*(i)+10] = -fp[2][1]*100.0/wgtnt;/* y component of force (vertical direction) */
        outarrr[18*(i)+11] = tp[2][0]*100.0/wgtnt;/* abduction torques */
outarrr[18*(i)+16] = -fp[2][2]*100.0/wgtnt; /* z component of force (lateral direction) */
        outarrr[18*(i)+17] = tp[2][1]*100.0/wgtnt; /* rotation torques */

/* Left Ankle, Left Knee, Left Hip */
        /* Left Ankle */
outarrl[18*i] =  -fp[7][0]*100.0/wgtnt; /* x component of force (ant-post direction)*/
        outarrl[18*(i)+1] = -tp[7][2]*100.0/wgtnt; /* flexion torques */
outarrl[18*(i)+6] = -fp[7][1]*100.0/wgtnt;/* y component of force (vertical direction) */
        outarrl[18*(i)+7] = tp[7][0]*100.0/wgtnt;/* abduction torques */
outarrl[18*(i)+12] = -fp[7][2]*100.0/wgtnt; /* z component of force (lateral direction) */
        outarrl[18*(i)+13] = tp[7][1]*100.0/wgtnt; /* rotation torques */


        /* Left Knee */
outarrl[18*(i)+2] =       -fp[8][0]*100.0/wgtnt; /* x component of force (ant-post direction)*/
        outarrl[18*(i)+3] = tp[8][2]*100.0/wgtnt; /* flexion torques */
outarrl[18*(i)+8] = -fp[8][1]*100.0/wgtnt ;/* y component of force (vertical direction) */
        outarrl[18*(i)+9] = tp[8][0]*100.0/wgtnt;/* abduction torques */
outarrl[18*(i)+14] =  fp[8][2]*100.0/wgtnt; /* z component of force (lateral direction) */
        outarrl[18*(i)+15] = tp[8][1]*100.0/wgtnt; /* rotation torques */


        /* Left Hip */
outarrl[18*(i)+4] = -fp[9][0]*100.0/wgtnt; /* x component of force (ant-post direction)*/
        outarrl[18*(i)+5] = -tp[9][2]*100.0/wgtnt; /* flexion torques */
outarrl[18*(i)+10] =  -fp[9][1]*100.0/wgtnt;/* y component of force (vertical direction) */
        outarrl[18*(i)+11] = tp[9][0]*100.0/wgtnt;/* abduction torques */
outarrl[18*(i)+16] =  -fp[9][2]*100.0/wgtnt; /* z component of force (lateral direction) */
        outarrl[18*(i)+17] = -tp[9][1]*100.0/wgtnt; /* rotation torques */

/* Shoulders */
outarrs[18*i] = 0.0; /* x component of force (ant-post direction)*/
        outarru[18*(i)+1] = 0.0; /* flexion torques */
outarrs[18*(i)+6] = 0.0;/* y component of force (vertical direction) */
        outarrs[18*(i)+7] = 0.0; /* abduction torques */
outarrs[18*(i)+12] = 0.0; /* z component of force (lateral direction) */
        outarrs[18*(i)+13] = 0.0; /* rotation torques */


        /* Right Shoulder */
outarrs[18*(i)+2] = -fp[5][0]*100.0/wgtnt; /* x component of force (ant-post direction)*/
        outarrs[18*(i)+3] = tp[5][2]*100.0/wgtnt; /* flexion torques*/
outarrs[18*(i)+8] = fp[5][1]*100.0/wgtnt;/* y component of force (vertical direction)*/
        outarrs[18*(i)+9] = tp[5][0]*100.0/wgtnt;/* abduction torques*/
outarrs[18*(i)+14] = -fp[5][2]*100.0/wgtnt; /* z component of force (lateral direction) */
        outarrs[18*(i)+15] = tp[5][1]*100.0/wgtnt;          /* rotation torques */


        /* Left Shoulder */
outarrs[18*(i)+4] = fp[12][0]*100.0/wgtnt; /* x component of force (ant-post direction)*/
        outarrs[18*(i)+5] = fp[12][2]*100.0/wgtnt; /* flexion torques */
outarrs[18*(i)+10] = fp[12][1]*100.0/wgtnt;/* y component of force (vertical direction)*/
```

```
        outarrs[18*(i)+11] = tp[12][0]*100.0/wgtnt;/* abduction torques */
outarrs[18*(i)+16] = -fp[12][2]*100.0/wgtnt; /* z component of force (lateral direction) */
        outarrs[18*(i)+17] = -tp[12][1]*100.0/wgtnt; /* rotation torques */

/* Neck and Back Forces and Torques */
        /* Neck (in neck coordinates) */
outarru[18*i] = -fp[6][0]*100.0/wgtnt; /* x component of force (ant-post direction)*/
        outarru[18*(i)+1] = tp[6][2]*100.0/wgtnt; /* flexion torques */
outarru[18*(i)+6] = fp[6][1]*100.0/wgtnt; /* y component of force (vertical direction) */
        outarru[18*(i)+7] = -tp[6][0]*100.0/wgtnt; /* abduction torques*/
outarru[18*(i)+12] = -fp[6][2]*100.0/wgtnt; /* z component of force (lateral direction) */
        outarru[18*(i)+13] = tp[6][1]*100.0/wgtnt; /* rotation torques */


        /* Back (in back coordinates) */
outarru[18*(i)+2] = -fp15[0]*100.0/wgtnt;  /* x component of force (ant-post direction) */
        outarru[18*(i)+3] = tp15[2]*100.0/wgtnt; /* flexion torques*/
outarru[18*(i)+8] = fp15[1]*100.0/wgtnt;  /* y component of force (vertical direction)*/
        outarru[18*(i)+9] = -tp15[0]*100.0/wgtnt;/* abduction torques*/
outarru[18*(i)+14] = -fp15[2]*100.0/wgtnt;        /* z component of force (lateral direction) */
        outarru[18*(i)+15] = tp15[1]*100.0/wgtnt; /* rotation torques */


outarru[18*(i)+4] = 0.0; /* x component of force (ant-post direction)*/
        outarru[18*(i)+5] = 0.0; /* flexion torques */
outarru[18*(i)+10] = 0.0; /* y component of force (vertical direction) */
        outarru[18*(i)+11] = 0.0;             /* abduction torques */
outarru[18*(i)+16] = 0.0; /* z component of force (lateral direction) */
        outarru[18*(i)+17] = 0.0; /* rotation torques */

/* Linear and Angular Momentum - Right Foot, Shank & Thigh */
        /* Right Foot */
                outarrmr[18*i] = lm[0][i][0]; /* x axis linear momentum */
        outarrmr[18*(i)+1] = am[0][i][0]; /* x axis angular momentum */
                outarrmr[18*(i)+6] = lm[0][i][1];/* y axis linear momentum */
        outarrmr[18*(i)+7] = am[0][i][1];  /* y axis angular momentum */
                outarrmr[18*(i)+12] = lm[0][i][2];/*  z axis linear momentum */
        outarrmr[18*(i)+13] = am[0][i][2]; /*  z axis angular momentum*/

        /* Right Shank */
                outarrmr[18*(i)+2] = lm[1][i][0];/*        x axis linear momentum */
        outarrmr[18*(i)+3] = am[1][i][0]; /* x axis angular momentum */
                outarrmr[18*(i)+8] = lm[1][i][1];/*        y axis linear momentum */
        outarrmr[18*(i)+9] = am[1][i][1];  /* y axis angular momentum */
                outarrmr[18*(i)+14] = lm[1][i][2];/*  z axis linear momentum */
        outarrmr[18*(i)+15] = am[1][i][2]; /*  z axis angular momentum*/

        /* Right Thigh */
                outarrmr[18*(i)+4] = lm[2][i][0]; /* x axis linear momentum */
        outarrmr[18*(i)+5] =  am[2][i][0]; /* x axis angular momentum */
                outarrmr[18*(i)+10] = lm[2][i][1];/* y axis linear momentum */
        outarrmr[18*(i)+11] =  am[2][i][1]; /*  y axis angular momentum */
                outarrmr[18*(i)+16] = lm[2][i][2]; /* z axis linear momentum */
        outarrmr[18*(i)+17] = am[2][i][2];        /* z axis angular momentum*/
```

```
/* Linear and Angular Momentum - Left Foot, Shank & Thigh */
        /* Left Foot */
                outarrml[18*i] = lm[7][i][0]; /* x axis linear momentum */
        outarrml[18*(i)+1] = am[7][i][0]; /* x axis angular momentum */
                outarrml[18*(i)+6] = lm[7][i][1]; /* y axis linear momentum */
        outarrml[18*(i)+7] = am[7][i][1];  /* y axis angular momentum */
                outarrml[18*(i)+12] = lm[7][i][2]; /* z axis linear momentum */
        outarrml[18*(i)+13] = am[7][i][2]; /*  z axis angular momentum*/

        /* Left Shank */
                outarrml[18*(i)+2] = lm[8][i][0]; /* x axis linear momentum */
        outarrml[18*(i)+3] = am[8][i][0]; /* x axis angular momentum */
                outarrml[18*(i)+8] = lm[8][i][1]; /* y axis linear momentum */
        outarrml[18*(i)+9] = am[8][i][1];/*          y axis angular momentum */
                outarrml[18*(i)+14] = lm[8][i][2]; /* z axis linear momentum */
        outarrml[18*(i)+15] = am[8][i][2];/*  z axis angular momentum*/

        /* Left Thigh */
                outarrml[18*(i)+4] = lm[9][i][0]; /* x axis linear momentum */
        outarrml[18*(i)+5] =  am[9][i][0]; /* x axis angular momentum */
                outarrml[18*(i)+10] = lm[9][i][1]; /* y axis linear momentum */
        outarrml[18*(i)+11] =  am[9][i][1];         /* y axis angular momentum */
                outarrml[18*(i)+16] = lm[9][i][2]; /* z axis linear momentum */
        outarrml[18*(i)+17] = am[9][i][2];         /* z axis angular momentum*/


/* Linear and Angular Momentum - Left Arm & Right Arm */
        /* Right Arm */
                outarrms[18*i] = lm[5][i][0]; /* x axis linear momentum */
        outarrms[18*(i)+1] = am[5][i][0]; /* x axis angular momentum */
                outarrms[18*(i)+6] = lm[5][i][1];/* y axis linear momentum */
        outarrms[18*(i)+7] = am[5][i][1];  /* y axis angular momentum */
                outarrms[18*(i)+12] = lm[5][i][2]; /* z axis linear momentum */
        outarrms[18*(i)+13] = am[5][i][2]; /*  z axis angular momentum*/

        /* Left Arm     */
                outarrms[18*(i)+2] = lm[12][i][0]; /* x axis linear momentum */
        outarrms[18*(i)+3] = am[12][i][0];         /*x axis angular momentum*/
                outarrms[18*(i)+8] = lm[12][i][1];/* y axis linear momentum */
        outarrms[18*(i)+9] = am[12][i][1];         /* y axis angular momentum*/
                outarrms[18*(i)+14] = lm[12][i][2]; /* z axis linear momentum */
        outarrms[18*(i)+15] = am[12][i][2];        /* z axis angular momentum*/

        /* */
                outarrms[18*(i)+4] = 0.0; /* x axis */
        outarrms[18*(i)+5] = 0.0; /* */
                outarrms[18*(i)+10] = 0.0;/* y axis */
        outarrms[18*(i)+11] =  0.0; /* */
                outarrms[18*(i)+16] = 0.0; /* z axis */
        outarrms[18*(i)+17] = 0.0;        /* */

/* Linear and Angular Momentum - Head and Trunk */
        /* Head*/
```

```
                outarrmu[18*i] = lm[6][i][0]; /* x axis linear momentum */
            outarrmu[18*(i)+1] = am[6][i][0]; /* x axis angular momentum */
                outarrmu[18*(i)+6] = lm[6][i][1];/* y axis linear momentum */
            outarrmu[18*(i)+7] = am[6][i][1];  /* y axis angular momentum */
                outarrmu[18*(i)+12] = lm[6][i][2]; /* z axis linear momentum */
            outarrmu[18*(i)+13] = am[6][i][2]; /*  z axis angular momentum*/

            /* Trunk */
  outarrmu[18*(i)+2] = (lm[11][i][0]+lm[4][i][0])/2.0; /* x axis linear momentum */
            outarrmu[18*(i)+3] = (am[11][i][0]+am[4][i][0])/2.0; /* x axis angular momentum */
  outarrmu[18*(i)+8] = (lm[11][i][1]+lm[4][i][1])/2.0;/* y axis linear momentum */
            outarrmu[18*(i)+9] = (am[11][i][1]+am[4][i][1])/2.0; /*  y axis angular momentum */
outarrmu[18*(i)+14] = (lm[11][i][2]+lm[4][i][2])/2.0; /* z axis linear momentum */
            outarrmu[18*(i)+15] = (am[11][i][2]+am[4][i][2])/2.0; /*  z axis angular momentum*/

            /* Pelvis */
  outarrmu[18*(i)+4] = (lm[10][i][0]+lm[3][i][0])/2.0;  /*x axis linear momentum */
            outarrmu[18*(i)+5] =  (am[10][i][0]+am[3][i][0])/2.0; /* x axis angular momentum */
  outarrmu[18*(i)+10] = (lm[10][i][1]+lm[3][i][1])/2.0; /*y axis linear momentum */
            outarrmu[18*(i)+11] =  (am[10][i][1]+am[3][i][1])/2.0; /*  y axis angular momentum */
  outarrmu[18*(i)+16] = (lm[10][i][2]+lm[3][i][2])/2.0; /* z axis linear momentum */
            outarrmu[18*(i)+17] = (am[10][i][2]+am[3][i][2])/2.0;  /* z axis angular momentum*/

  /* Linear and Angular Momentum - Whole Body */

            /* Whole Body Linear Momentum  */
                outarrmw[18*i] = P[0]/segmastotal; /* x axis linear momentum */
            outarrmw[18*(i)+1] = 0.0; /* */
                outarrmw[18*(i)+6] = P[1]/segmastotal;/* y axis linear momentum */
            outarrmw[18*(i)+7] = 0.0;  /*  */
                outarrmw[18*(i)+12] =P[2]/segmastotal; /* z axis linear momentum */
            outarrmw[18*(i)+13] = 0.0; /*  */

            /* Whole Body Angular Momentum about the combined ankle joint */
                outarrmw[18*(i)+2] =A[0]/segmastotal; /* x axis angular momentum */
            outarrmw[18*(i)+3] = 0.0;  /*  */
                outarrmw[18*(i)+8] = A[1]/segmastotal; /*y axis angular momentum */
            outarrmw[18*(i)+9] = 0.0;  /* */
                outarrmw[18*(i)+14] = A[2]/segmastotal;  /*z axis angular momentum*/
            outarrmw[18*(i)+15] = 0.0;       /* */

            /* Whole Body Angular Momentum about the CG */
                outarrmw[18*(i)+4] = ACG[0]/segmastotal; /* x axis angular momentum */
            outarrmw[18*(i)+5] = 0.0; /* */
                outarrmw[18*(i)+10] = ACG[1]/segmastotal;/* y axis angular momentum */
            outarrmw[18*(i)+11] =  0.0; /* */
                outarrmw[18*(i)+16] = ACG[2]/segmastotal; /* z axis angular momentum */
            outarrmw[18*(i)+17] = 0.0;       /* */

            }
  }
```

A.4 Three-Link Simulation C Algorithm

```
/* Parameters for three link simulation */
if (i==0)
        {

/* calculate combined link masses */
                m1= segmas[1]+segmas[8];
                m2= segmas[2]+segmas[9]+(segmas[3]+segmas[10])/2.0;
                m3= segmas[6]+(segmas[4]+segmas[11])/2.0+segmas[5]+segmas[12];

/* rotate vectors & matrices into global coordinates */
        for (ia=0;ia<13;ia++) gmprd1(arrrot[ia],segrcg[ia],gsegrcg[ia]);
        for (ia=0;ia<13;ia++) gmprd1(arrrot[ia],segrp[ia],gsegrp[ia]);
        for (ia=0;ia<13;ia++) gmprd1(arrrot[ia],segrd[ia],gsegrd[ia]);
        for (ia=0;ia<13;ia++){ gmprd3(arrrot[ia],pinertia[ia],temp33r);
                               gmtra(arrrot[ia],segrtr);
                               gmprd3(temp33r,segrtr,temp33l);
                               for (k=0;k<3;k++) gsegint[ia][k] = temp33l[k][k];}

/* calculate link lengths */
                for (ia=0;ia<3;ia++)
                {
                l1[ia]= (gsegrp[1][ia]+gsegrp[8][ia]-gsegrd[1][ia]-gsegrd[8][ia])/2.0;
                l2[ia]= (gsegrp[2][ia]+gsegrp[9][ia]-gsegrd[2][ia]-gsegrd[9][ia]
                        +gsegrp[3][ia]+gsegrp[10][ia]-gsegrd[3][ia]
                        -gsegrd[10][ia])/2.0;
                l3[ia]= ((gsegrp[4][ia]+gsegrp[11][ia]-gsegrd[4][ia]-gsegrd[11][ia])/2.0
                        - gsegrp[6][ia] + gsegrd[6][ia]);
                }
                l1mag = sqrt(l1[0]*l1[0]+l1[1]*l1[1]+l1[2]*l1[2]);
                l2mag = sqrt(l2[0]*l2[0]+l2[1]*l2[1]+l2[2]*l2[2]);
                l3mag = sqrt(l3[0]*l3[0]+l3[1]*l3[1]+l3[2]*l3[2]);

/* calculate length to link cg's */
                for (ia=0;ia<3;ia++)
                {
                lc1[ia]= (-gsegrd[1][ia]-gsegrd[8][ia])/2.0;
                lc2[ia]= (1/m2)*((segmas[2]*(-gsegrd[2][ia])+segmas[9]*(-gsegrd[9][ia]))/2.0 +
                        (segmas[3]*(l2[ia]-gsegrp[3][ia])+segmas[10]*(l2[ia]-gsegrp[10][ia]))/2.0);
                        lc15[ia]= (-gsegrd[4][ia]-gsegrd[11][ia])/2.0;
                        lc6[ia]= l3[ia]-gsegrd[6][ia];
                        lc5[ia]= (arrpos[5][ia]+gsegrcg[5][ia]-(arrpos[4][ia]+gsegrcg[4][ia])+lc15[ia]);
                        lc12[ia]= (arrpos[12][ia]+gsegrcg[12][ia]-
```

```
                        (arrpos[11][ia]+gsegrcg[11][ia])+lc15[ia]);
        lc3[ia]= ((segmas[4]+segmas[11])*(.5)*lc15[ia] +
                segmas[6]*lc6[ia] + segmas[5]*lc5[ia] + segmas[12]*lc12[ia])/m3;
        }


        lc1mag = sqrt(lc1[0]*lc1[0]+lc1[1]*lc1[1]+lc1[2]*lc1[2]);
        lc2mag = sqrt(lc2[0]*lc2[0]+lc2[1]*lc2[1]+lc2[2]*lc2[2]);
        lc3mag = sqrt(lc3[0]*lc3[0]+lc3[1]*lc3[1]+lc3[2]*lc3[2]);

/* calculate combined link inertias */
        for (ia=0;ia<3;ia++){
        i1[ia] = gsegint[1][ia] + gsegint[8][ia];
        i2[ia] = gsegint[2][ia] + gsegint[9][ia] + (gsegint[3][ia] + gsegint[10][ia])*(.5) +
                (segmas[2]+segmas[9])*(lc2[ia]-(-gsegrd[2][ia]-gsegrd[9][ia])*(.5))*
                (lc2[ia]-(-gsegrd[2][ia]-gsegrd[9][ia])*(.5))
                +(segmas[3]+segmas[10])*(.5)*(l2[ia]-
                (gsegrp[3][ia]+gsegrp[10][ia])*(.5)-lc2[ia])*
                (l2[ia]-(gsegrp[3][ia]+gsegrp[10][ia])*(.5)-lc2[ia]);
        i3[ia] = (gsegint[4][ia]+gsegint[11][ia])*(.5) + gsegint[5][ia] + gsegint[12][ia] +
                gsegint[6][ia] +segmas15*(lc3[ia]-lc15[ia])*(lc3[ia]-lc15[ia]) +
                segmas[6]*(lc3[ia]-lc6[ia])*(lc3[ia]-lc6[ia]) +segmas[5]*
                (lc3[ia]-lc5[ia])*(lc3[ia]-lc5[ia]) +
                segmas[12]*(lc3[ia]-lc12[ia])*(lc3[ia]-lc12[ia]);}

/* Calculation of the torques for a 3-link model */
/* calculate angles, calculate and smooth angular velocities & angular accelerations */
        theta1 =  - (ang[1][i][2] + ang[8][i][2])/2.0;
        theta2 =  ang[9][i][2];
        theta3 = 1.5707963268 - (ang[4][i][2] + ang[11][i][2])/2.0;

        dtheta1 = (av[1][i][2] - av[8][i][2])/2.0;
        dtheta2 = -av[9][i][2];
        dtheta3 = (av[4][i][2] + av[11][i][2])/2.0;


        ddtheta1 = (aa[1][i][2] - aa[8][i][2])/2.0;
        ddtheta2 = -aa[9][i][2];
        ddtheta3 = (aa[4][i][2] + aa[11][i][2])/2.0;

/* calculate inertia matrix */
        H11 = m1*lc1mag*lc1mag + i1[2] + m2*l1mag*l1mag + m3*l1mag*l1mag;
        H12 = (m2*l1mag*lc2mag+m3*l1mag*l2mag)*cos(theta1+theta2);
        H13 = m3*l1mag*lc3mag*cos(theta1-theta3);
        H21 = H12;
        H22 = i2[2] + m2*lc2mag*lc2mag + m3*l2mag*l2mag;
        H23 = m3*l2mag*lc3mag*cos(theta2+theta3);
        H31 = H13;
        H32 = H23;
        H33 = i3[2] + m3*lc3mag*lc3mag;

/* calculate non-linear coefficients */
        h122 = -(m2*l1mag*lc2mag+m3*l1mag*l2mag)*sin(theta1+theta2);
        h133 = -m3*l1mag*lc3mag*sin(theta1-theta3);
```

```
                h211 = h122;
                h233 = -m3*l2mag*lc3mag*sin(theta2+theta3);
                h311 = h133;
                h322 = h233;

/* calculate gravity terms */
                G1 = -(m1*g[1]*lc1mag*sin(theta1)+ m2*g[1]*l1mag*sin(theta1) +
                        m3*g[1]*l1mag*sin(theta1));
                G2 = -(m2*g[1]*lc2mag*sin(theta2) + m3*g[1]*l2mag*sin(theta2));
                G3 = -(m3*g[1]*lc3mag*sin(theta3));

/* calculate generalized forces */
                Q1 = H11*ddtheta1 + H12*ddtheta2 + H13*ddtheta3 + h122*dtheta2*dtheta2 +
                        h133*dtheta3*dtheta3 + G1;
                Q2 = H21*ddtheta1 + H22*ddtheta2 + H23*ddtheta3 + h211*dtheta1*dtheta1 +
                        h233*dtheta3*dtheta3 + G2;
                Q3 = H31*ddtheta1 + H32*ddtheta2 + H33*ddtheta3 + h322*dtheta2*dtheta2 +
                        h311*dtheta1*dtheta1 + G3;

                tau1 = Q2 - Q3- Q1;
                tau2 = Q3 - Q2;
                tau3 = -Q3;
```

Appendix B:

Formulation of the Lagrangian Dynamic Equations of Motion for the Three-Link Model

The general equations used in this derivation are in the form described by Asada [103].

General form of equations of motion:

$$\Sigma_j H_{ij}\Theta''_j + \Sigma_j\Sigma_k h_{ijk}\Theta'_j\Theta'_k + G_i = Q_i \quad i = 1,2,3$$

Calculation of the inertia tensor, [H]:

$\omega_1 = \Theta_1'$
$\omega_2 = \Theta_2'$
$\omega_3 = \Theta_3'$

$$x_{c1} = \begin{bmatrix} l_{c1}\sin\Theta_1 \\ l_{c1}\cos\Theta_1 \end{bmatrix} \qquad v_{c1} = \begin{bmatrix} l_{c1}\cos\Theta_1 \\ -l_{c1}\sin\Theta_1 \end{bmatrix}\Theta_1'$$

$$x_{c2} = \begin{bmatrix} l_1\sin\Theta_1 - l_{c2}\sin\Theta_2 \\ l_1\cos\Theta_1 + l_{c2}\cos\Theta_2 \end{bmatrix} \qquad v_{c2} = \begin{bmatrix} l_1\cos\Theta_1 & -l_{c2}\cos\Theta_2 \\ -l_1\sin\Theta_1 & -l_{c2}\sin\Theta_2 \end{bmatrix}\begin{bmatrix}\Theta_1' \\ \Theta_2'\end{bmatrix}$$

$$x_{c3} = \begin{bmatrix} l_1\sin\Theta_1 - l_2\sin\Theta_2 + l_{c3}\sin\Theta_3 \\ l_1\cos\Theta_1 + l_2\cos\Theta_2 + l_{c3}\cos\Theta_3 \end{bmatrix} \qquad v_{c2} = \begin{bmatrix} l_1\cos\Theta_1 & -l_2\cos\Theta_2 & l_{c3}\cos\Theta_3 \\ -l_1\sin\Theta_1 & -l_2\sin\Theta_2 & -l_{c3}\sin\Theta_3 \end{bmatrix}\begin{bmatrix}\Theta_1' \\ \Theta_2' \\ \Theta_3'\end{bmatrix}$$

$$[H] = \Sigma_i \left(m_i J_L^{(i)T} J_L^{(i)} + J_A^{(i)T} I_i J_A^{(i)}\right) \quad i=1,\ldots,n$$

$$[H] = \begin{bmatrix} m_1 l_{c1}^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{bmatrix} \begin{bmatrix} l_1\cos\Theta_1 & -l_1\sin\Theta_1 & 0 \\ l_1\cos\Theta_1 & -l_{c2}\cos\Theta_2 & 0 \end{bmatrix}\begin{bmatrix} l_1\cos\Theta_1 & -l_{c2}\cos\Theta_2 & 0 \\ -l_1\sin\Theta_1 & -l_{c2}\sin\Theta_2 & 0 \end{bmatrix}$$
$$+ m_2 \begin{bmatrix} -l_{c2}\cos\Theta_2 & -l_{c2}\sin\Theta_2 & 0 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} -l_1\sin\Theta_1 & -l_{c2}\sin\Theta_2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$
$$+ m_3 \begin{bmatrix} l_1\cos\Theta_1 & -l_1\sin\Theta_1 & 0 \\ -l_2\cos\Theta_2 & -l_2\sin\Theta_2 & 0 \\ l_{c3}\cos\Theta_3 & -l_{c3}\sin\Theta_3 & 0 \end{bmatrix}\begin{bmatrix} l_1\cos\Theta_1 & -l_2\cos\Theta_2 & l_{c3}\cos\Theta_3 \\ -l_1\sin\Theta_1 & -l_2\sin\Theta_2 & -l_{c3}\sin\Theta_3 \\ 0 & 0 & 0 \end{bmatrix}$$

$$[H] = \begin{bmatrix} I_1+m_1l_{c1}^2+m_2l_1^2+m_3l_1^2 & (m_2l_1l_{c2}+m_3l_1l_2)\cos(\Theta_1+\Theta_2) & m_3l_1l_{c3}\cos(\Theta_1-\Theta_3) \\ (m_2l_1l_{c2}+m_3l_1l_2)\cos(\Theta_1+\Theta_2) & I_2+m_2l_{c2}^2+m_3l_2^2 & m_3l_2l_{c3}\cos(\Theta_2+\Theta_3) \\ m_3l_1l_{c3}\cos(\Theta_1-\Theta_3) & m_3l_2l_{c3}\cos(\Theta_2+\Theta_3) & I_3+m_3l_{c3}^2 \end{bmatrix}$$

Calculation of nonlinear coefficients, h_{ijk}:

$$h_{ijk} = \delta H_{ij}/\delta\Theta_k - (1/2)*(\delta H_{jk}/\delta\Theta_i) \quad \text{for i,j,k} = 1,2,3$$

$h_{111} = h_{123} = h_{132} = h_{213} = h_{222} = h_{231} = h_{312} = h_{321} = h_{333} = 0$

$h_{122} = h_{211} = A$
$h_{133} = h_{331} = B$
$h_{233} = h_{332} = C$
$h_{121} = h_{212} = A/2$
$h_{131} = h_{313} = B/2$
$h_{232} = h_{323} = C/2$
$h_{112} = h_{221} = -A/2$
$h_{113} = h_{331} = -B/2$
$h_{223} = h_{332} = -C/2$

where: $A = -(m_2l_1l_{c2}+m_3l_1l_2)\sin(\Theta_1+\Theta_2)$
$B = -m_3l_1l_{c3}\sin(\Theta_1-\Theta_3)$
$C = -m_3l_2l_{c3}\sin(\Theta_2+\Theta_3)$

Calculation of gravity terms, G_i:

$$G_i = \Sigma_j m_j g^T J_{Li}^{(j)} \qquad \text{where } g^T = [\,0\ \ -g\ \ 0\,] \text{ and } g = 9.8 m/s^2$$

$$G_1 = m_1*[0\ -g\ 0]\begin{bmatrix} l_{c1}\cos\Theta_1 \\ -l_{c1}\sin\Theta_1 \\ 0 \end{bmatrix} + m_2*[0\ -g\ 0]\begin{bmatrix} l_1\cos\Theta_1 \\ -l_1\sin\Theta_1 \\ 0 \end{bmatrix} + m_3*[0\ -g\ 0]\begin{bmatrix} l_1\cos\Theta_1 \\ -l_1\sin\Theta_1 \\ 0 \end{bmatrix}$$

$G_1 = m_1gl_{c1}\sin\Theta_1 + m_2gl_1\sin\Theta_1 + m_3gl_1\sin\Theta_1$
$G_2 = m_2gl_{c2}\sin\Theta_2 + m_3gl_2\sin\Theta_2$
$G_3 = m_3gl_{c3}\sin\Theta_3$

Calculation of generalized forces, Q_i:

$\delta Work = -(\tau_1 + \tau_2)\delta Q_1 - (\tau_2+\tau_3)\delta Q_2 - \tau_3\delta Q_3$

$Q_1 = -(\tau_1 + \tau_2)$ or $\tau_1 = -Q_3$
$Q_2 = -(\tau_2 + \tau_3)$ $\tau_2 = Q_3 - Q_2$
$Q_3 = -\tau_3$ $\tau_3 = Q_2 - Q_3 - Q_1$

Substituting terms in general equation to form the equations of motion:

$$\Sigma_j H_{ij}\Theta''_j + \Sigma_j\Sigma_k h_{ijk}\Theta'_j\Theta'_k + G_i = Q_i \quad i = 1,2,3$$

$$H_{11}\Theta''_1 + H_{12}\Theta''_2 + H_{13}\Theta''_3 + h_{111}\Theta'^2_1 + h_{112}\Theta'_1\Theta'_2 + h_{113}\Theta'_1\Theta'_3 + h_{121}\Theta'_1\Theta'_2 + h_{122}\Theta'^2_2 + h_{123}\Theta'_2\Theta'_3 + h_{131}\Theta'_1\Theta'_3 + h_{132}\Theta'_3\Theta'_2 + h_{133}\Theta'^2_3 + G_1 = Q_1 \quad (i=1)$$

$$H_{22}\Theta''_2 + H_{21}\Theta''_1 + H_{23}\Theta''_3 + h_{211}\Theta'^2_1 + h_{212}\Theta'_1\Theta'_2 + h_{213}\Theta'_1\Theta'_3 + h_{221}\Theta'_1\Theta'_2 + h_{222}\Theta'^2_2 + h_{223}\Theta'_2\Theta'_3 + h_{231}\Theta'_1\Theta'_3 + h_{232}\Theta'_3\Theta'_2 + h_{233}\Theta'^2_3 + G_2 = Q_2 \quad (i=2)$$

$$H_{33}\Theta''_3 + H_{31}\Theta''_1 + H_{32}\Theta''_2 + h_{311}\Theta'^2_1 + h_{312}\Theta'_1\Theta'_2 + h_{313}\Theta'_1\Theta'_3 + h_{321}\Theta'_1\Theta'_2 + h_{322}\Theta'^2_2 + h_{323}\Theta'_2\Theta'_3 + h_{331}\Theta'_1\Theta'_3 + h_{332}\Theta'_3\Theta'_2 + h_{333}\Theta'^2_3 + G_3 = Q_3 \quad (i=3)$$

After cancelation and reduction:

$$H_{11}\Theta''_1 + H_{12}\Theta''_2 + H_{13}\Theta''_3 + h_{122}\Theta'^2_2 + h_{133}\Theta'^2_3 + G_1 = Q_1$$
$$H_{22}\Theta''_2 + H_{21}\Theta''_1 + H_{23}\Theta''_3 + h_{211}\Theta'^2_1 + h_{233}\Theta'^2_3 + G_2 = Q_2$$
$$H_{33}\Theta''_3 + H_{31}\Theta''_1 + H_{32}\Theta''_2 + h_{311}\Theta'^2_1 + h_{322}\Theta'^2_2 + G_3 = Q_3$$

where

$H_{11} = I_1 + M_1L_{c1}^2 + M_2L_1^2 + M_3L_1^2$

$H_{12} = H_{21} = (M_2L_1L_{c2} + M_3L_1L_2)\cos(\Theta_1+\Theta_2)$

$H_{13} = H_{31} = M_3L_1L_3\cos(\Theta_1-\Theta_3)$

$H_{22} = I_2 + M_2L_{c2}^2 + M_3L_2^2$

$H_{23} = H_{32} = M_3L_2L_{c3}\cos(\Theta_2+\Theta_3)$

$H_{33} = I_3 + M_3L_{c3}^2$

$h_{122} = h_{211} = -(M_2L_1L_{c2} + M_3L_1L_2)\sin(\Theta_1+\Theta_2)$

$h_{133} = h_{311} = -M_3L_1L_{c3}\sin(\Theta_1-\Theta_3)$

$h_{233} = h_{322} = -M_3L_2L_{c3}\sin(\Theta_2+\Theta_3)$

$G_1 = m_1gL_{c1}\sin\Theta_1 + m_2gL_1\sin\Theta_1 + m_3gL_1\sin\Theta_1$

$G_2 = m_2gL_{c2}\sin\Theta_2 + m_3gL_2\sin\Theta_2$

$G_3 = m_3gL_{c3}\sin\Theta_3$

$Q_1 = -(\tau_1 + \tau_2)$

$Q_2 = -(\tau_2 + \tau_3)$

$Q_3 = -\tau_3$