



Swansea University
Prifysgol Abertawe



Swansea University E-Theses

Development of a parameterised human musculoskeletal model.

Christodoulakis, Georgios

How to cite:

Christodoulakis, Georgios (2007) *Development of a parameterised human musculoskeletal model.* thesis, Swansea University.

<http://cronfa.swan.ac.uk/Record/cronfa43181>

Use policy:

This item is brought to you by Swansea University. Any person downloading material is agreeing to abide by the terms of the repository licence: copies of full text items may be used or reproduced in any format or medium, without prior permission for personal research or study, educational or non-commercial purposes only. The copyright for any work remains with the original author unless otherwise specified. The full-text must not be sold in any format or medium without the formal permission of the copyright holder. Permission for multiple reproductions should be obtained from the original author.

Authors are personally responsible for adhering to copyright and publisher restrictions when uploading content to the repository.

Please link to the metadata record in the Swansea University repository, Cronfa (link given in the citation reference above.)

<http://www.swansea.ac.uk/library/researchsupport/ris-support/>



DEVELOPMENT OF A PARAMETERISED HUMAN MUSCULOSKELETAL MODEL

Georgios Christodoulakis

Submitted to the University of Wales in fulfilment of the
requirements for the Degree of Master of Philosophy of
Engineering

Swansea University

2007

ProQuest Number: 10821573

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



ProQuest 10821573

Published by ProQuest LLC (2018). Copyright of the Dissertation is held by the Author.

All rights reserved.

This work is protected against unauthorized copying under Title 17, United States Code
Microform Edition © ProQuest LLC.

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346



CONTENTS

Abstract	i
Acknowledgements	ii
Declaration	iii
List of Figures	iv
List of Tables	xi
Chapter 1 Introduction	1
1.1 Motivation.....	1
1.2 Objectives.....	1
1.3 Layout of Thesis.....	2
Chapter 2 Musculoskeletal System	3
2.1 Introduction.....	3
2.2 The Skeleton.....	5
2.2.1 Function of the Skeleton.....	5
2.2.2 Terminology.....	5
2.2.3 The Axial Skeleton.....	7
2.2.3.1 The Skull.....	7
2.2.3.2 The Vertebral Column.....	8
2.2.3.3 The Rib Cage.....	8
2.2.4 The Appendicular Skeleton.....	9
2.2.4.1 The Upper Limb.....	9
2.2.4.2 The Lower Limb.....	11
2.3 The Articular System.....	12
2.3.1 Introduction.....	12
2.3.2 Terminology.....	12
2.3.3 Classification of Joints.....	14
2.4 The Skeletal Muscles.....	17

2.4.1	Function of Muscles.....	17
2.4.2	Muscles of the Head & Neck.....	17
2.4.3	Muscles of the Trunk.....	18
2.4.4	Muscles of the Upper Extremity.....	20
2.4.5	Muscles of the Lower Extremity.....	20
2.5	Conclusions.....	22

Chapter 3 Computer Modelling & Anthropometry 23

3.1	Introduction.....	23
3.2	Why Simulate Human Movement?.....	23
3.3	Limitations of Computer Models.....	24
3.4	Research Method.....	25
3.5	Anthropometry.....	26
3.5.1	Introduction.....	26
3.5.2	Cadaver Studies.....	27
3.5.3	Mathematical Modelling.....	34
3.5.4	Scanning & Imaging Techniques.....	36
3.5.5	Kinematic Techniques.....	37
3.6	Summary.....	37

Chapter 4 Musculoskeletal Software Systems & Application Areas 39

4.1	Introduction.....	39
4.2	Musculoskeletal Software Systems.....	39
4.2.1	Anybody Technology.....	39
4.2.2	SIMM Software Suite.....	40
4.2.2.1	SIMM.....	40
4.2.2.2	Dynamics Pipeline.....	41
4.2.2.3	Generic Knee Model.....	42
4.2.2.4	Full Body Model.....	42
4.2.2.5	Real Time Motion Module.....	43

4.2.2.6	C3D Module.....	44
4.2.2.7	FIT Module.....	44
4.2.3	LifeMOD.....	44
4.2.4	SimMechanics.....	47
4.2.5	MADYMO.....	48
4.3	Application Areas.....	49
4.3.1	Automotive.....	49
4.3.2	Medical/Rehabilitation.....	52
4.3.3	Aerospace.....	54
4.3.4	Occupational Health.....	57
4.3.5	Sports.....	58
4.4	Conclusions.....	60

Chapter 5 The Anybody Modelling System 61

5.1	Introduction.....	61
5.2	The AnyScript Language.....	61
5.2.1	Definition.....	61
5.2.2	Basic Concepts.....	62
5.3	Mechanical Elements.....	64
5.3.1	Segments.....	64
5.3.2	Joints.....	64
5.3.3	Drivers.....	65
5.3.4	Kinematic Measures.....	65
5.3.5	Forces.....	65
5.3.6	Muscles.....	66
5.4	Inverse Dynamics.....	66
5.4.1	Introduction.....	66
5.4.2	Mathematical Formulation.....	67
5.5	Muscle Recruitment Solver.....	69
5.5.1	Redundancy.....	69

5.5.2	Optimisation.....	70
5.6	Summary.....	74
Chapter 6	Development of a Triple Inverted Pendulum	75
6.1	Introduction.....	75
6.2	Structure.....	75
6.3	Single Inverted Pendulum.....	78
6.3.1	Defining Global Reference Frame.....	78
6.3.2	Defining Segments.....	80
6.3.3	Connecting Segments By Joints.....	82
6.3.4	Definition of Movement.....	83
6.3.5	Definition of Muscles.....	84
6.3.6	Adding Real Geometries.....	87
6.4	Double Inverted Pendulum.....	90
6.5	Triple Inverted Pendulum.....	94
6.6	Advanced Study.....	98
6.6.1	Initial Conditions.....	98
6.6.2	Kinematic Measures & Motion.....	101
6.7	Conclusions.....	111
Chapter 7	Development of a Simplified Human Model	112
7.1	Introduction.....	112
7.2	Structure.....	112
7.3	Lower Extremity.....	116
7.3.1	Segments.....	116
7.3.2	Initial Conditions.....	119
7.3.3	Joints.....	120
7.3.4	Muscles.....	122
7.3.5	Kinematic Measures & Drivers.....	124
7.4	Trunk.....	132

7.4.1	Segments.....	132
7.4.2	Joints.....	133
7.4.3	Muscles.....	134
7.4.4	Kinematic Measures & Drivers.....	136
7.5	Upper Extremity & Box.....	141
7.5.1	Segments.....	141
7.5.2	Initial Conditions.....	144
7.5.3	Joints.....	145
7.5.4	Muscles.....	146
7.5.5	Kinematic Measures & Drivers.....	148
7.6	Conclusions.....	153

Chapter 8 Development of a Parameterised Human Model 155

8.1	Introduction.....	155
8.2	Lower Extremity.....	157
8.2.1	Segments.....	157
8.2.2	Joints.....	163
8.2.3	Validation.....	164
8.3	Lumbar, Thorax & Head.....	165
8.3.1	Segments.....	165
8.3.2	Joints.....	171
8.3.3	Validation.....	172
8.4	Upper Limb.....	173
8.4.1	Segments.....	173
8.4.2	Joints.....	180
8.4.3	Validation.....	182
8.5	Application.....	187
8.6	Conclusions.....	190

Chapter 9 Concluding Remarks 191

9.1	Summary.....	191
-----	--------------	-----

9.2	Conclusions.....	191
9.3	Further Work.....	192

Appendix A		193
-------------------	--	------------

References & Bibliography		198
--------------------------------------	--	------------

ABSTRACT

One of the most important and challenging areas of the rapid developing science of biomechanics nowadays, is the area of musculoskeletal modelling. Hence, simulation packages have been proved to be a very useful tool that has enabled scientists to quickly develop and analyze musculoskeletal models.

This project aims to develop a parameterized musculoskeletal model of the human body using one of these simulation tools called AnyBody. The whole investigation will start with a brief analysis of the musculoskeletal system, followed by a review in anthropometry and the existing geometrical models of the human body, and a review in the musculoskeletal software packages and their application areas. An introduction to the Anybody modelling system is then presented, describing its major functions. The investigation will continue with the development of two simple models which intend to be the basic platform for more complicated models. Finally, a scalable human model will be developed, aiming to be a useful tool for present and future users of Anybody Technology.

ACKNOWLEDGEMENTS

First of all I am very grateful to my parents and grandparents for their support all the years of my studies and for their great patience and heroism putting my needs above their own. I owe everything to them.

I would also like to express my gratitude to my supervisor Dr Johann Sienz who trusted me to do research and gave me the opportunity to act like a “free spirit”.

I am also grateful to the Anybody Group for the Repository, which was essential for the implementation of this thesis.

Many thanks to the team of the Virtual Human Project at the University of Brussels for the bone geometries, and the Computational Mechanics Group of the Laboratory of Medical Technology of Italy for the Data Manager software.

I would also like to thank the software development company “Floating Point Solutions Private Limited” for the discount price in purchasing the AutoCAD Plug-Ins “STL Import & Export”.

DECLARATION

This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed

Date...05/03/2007.....

STATEMENT 1

This thesis is the result of my own investigations, except where otherwise stated. Where correction services have been used, the extent and nature of the correction is clearly marked in a footnote(s).

Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signed

Date...05/03/2007.....

STATEMENT 2

I hereby give consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed

Date...05/03/2007.....

LIST OF FIGURES

Figure 2.1 – Skeletal System.....	3
Figure 2.2 – Muscular System.....	4
Figure 2.3 – Axial/Appendicular Skeleton.....	5
Figure 2.4 – Reference Planes.....	6
Figure 2.5 – The Skull.....	7
Figure 2.6 – The Vertebral Column.....	8
Figure 2.7 – The Rib Cage.....	9
Figure 2.8 – Upper Limbs.....	9
Figure 2.9 – The Shoulder Girdle.....	10
Figure 2.10 – Lower Limbs.....	11
Figure 2.11 – Reference Axes.....	12
Figure 2.12 – Abduction/Adduction of Shoulder, Hip & Wrist joints.....	13
Figure 2.13 – Abduction/Adduction of Hand & Foot Joints.....	13
Figure 2.14 – Flexion/Extension of Shoulder, Hip & Wrist Joints.....	13
Figure 2.15 – Flexion/Extension of Elbow & Knee Joints.....	14
Figure 2.16 – Flexion/Extension of Trunk Joints.....	14
Figure 2.17 – Synarthrosis Joint.....	14
Figure 2.18 – Amphiarthrosis Joint.....	15
Figure 2.19 – Ball & Socket Joint.....	15
Figure 2.20 – Hinge Joint.....	15
Figure 2.21 – Pivot Joint.....	16
Figure 2.22 – Gliding Joint.....	16
Figure 2.23 – Saddle Joint.....	16
Figure 2.24 – Head & Neck Muscles.....	18
Figure 2.25 – Muscles of Trunk (Anterior View).....	19

Figure 2.26 – Muscles of Trunk (Posterior View)	19
Figure 2.27 – Muscles of Forearm (Flexors/Extensors)	20
Figure 2.28 – Muscles of Thigh (Posterior/Anterior)	21
Figure 2.29 – Muscles of Leg (Posterior/Anterior)	21
Figure 3.1 – Reuleaux’s Method	28
Figure 3.2 – Body Segment Lengths Expressed as a Fraction of Body Height H	33
Figure 3.3 – Hanavan’s Model	34
Figure 3.4 – Inertial Properties of Hanavan’s Model Segments	35
Figure 3.5 – Hatze’s & Yeadon’s Models	36
Figure 4.1 – The Anybody Modelling System	39
Figure 4.2 – The SIMM Modelling System	40
Figure 4.3 – The LifeMOD Modelling System	45
Figure 4.4 – The SimMechanics Modelling System	47
Figure 4.5 – The MADYMO Modelling System	48
Figure 4.6 – Egress Model	50
Figure 4.7 – Pedal Design Model	50
Figure 4.8 – Rider Model	51
Figure 4.9 – Crash Test Model	51
Figure 4.10 – Gait Analysis Model	52
Figure 4.11 – Wheelchair Model	53
Figure 4.12 – Hip Replacement Model	53
Figure 4.13 – Bet Settling Model	54
Figure 4.14 – Cockpit Design Model	54
Figure 4.15 – Passengers Spaces Model	55
Figure 4.16 – Galileo Model	55
Figure 4.17 – Mars Rover Model	56
Figure 4.18 – Workplace Design Model	57
Figure 4.19 – Pushing Model	57
Figure 4.20 – Equipment Design Model	58
Figure 4.21 – Bicycling Model	58

Figure 4.22 – Fitness Equipment Model	59
Figure 4.23 – Virtual Golfer Model	59
Figure 5.1 – AnyScript Model	62
Figure 5.2 – AnyScript Model’s Overall Structure	62
Figure 5.3 – AnyScript Model’s Actual Body	63
Figure 5.4 – 3D Free-Body Diagram for the Inverse Dynamics Equations	67
Figure 5.5 – Flow Chart of Min/Max Algorithm	72
Figure 5.6 – The Process of Dynamic Analysis Within AnyBody	73
Figure 6.1 – Structure of Triple Inverted Pendulum Windows Folder	76
Figure 6.2 – Structure of Main.any File	76
Figure 6.3 – Mechanical Drawing of Single Inverted Pendulum	78
Figure 6.4 – Environment’s Model View	79
Figure 6.5 – Geometrical Properties of Pendulum	80
Figure 6.6 – Segment1’s Model View	81
Figure 6.7 – Motion of Single Inverted Pendulum	84
Figure 6.8 – Motion of Muscle Driven Single Inverted Pendulum	85
Figure 6.9 – Motion of Muscle Driven Single Inverted Pendulum ($F_0 = 50N$)	86
Figure 6.10 – F_m of <i>Seg1Front</i> Muscle	86
Figure 6.11 – <i>Activity</i> of <i>Seg1Front</i> Muscle	87
Figure 6.12 – Seg1.any File with STL/AnySurf Files	88
Figure 6.13 – AutoCAD/STL Model	88
Figure 6.14 – AutoCAD/STL Model	89
Figure 6.15 – STL/AnySurf Format	89
Figure 6.16 – Mechanical Drawing of Double Inverted Pendulum	90
Figure 6.17 – Motion of Muscle Driven Double Inverted Pendulum	93
Figure 6.18 – Mechanical Drawing of Triple Inverted Pendulum	94
Figure 6.19 – Motion of Muscle Driven Triple Inverted Pendulum	97
Figure 6.20 – Xyz Cardan Sequence of Rotations	98
Figure 6.21 – 3-D Initial Position of Triple Inverted Pendulum	103
Figure 6.22 – Updated KinematicMeasures.any File	104

Figure 6.23 – Updated Driver2.any File.....	105
Figure 6.24 – Updated KinematicMeasures.any File.....	106
Figure 6.25 – InitialConditions.any File.....	107
Figure 6.26 – Model with Initial Conditions at Time Step 0.....	107
Figure 6.27 – Updated Drivers2.any File.....	108
Figure 6.28 – Motion of Triple Inverted Pendulum with only One Driver.....	109
Figure 6.29 – Linear Displacement & Velocity of Node12.....	110
Figure 6.30 – Angular Displacement & Velocity of Node12.....	110
Figure 6.31 – Key Steps of the Triple Inverted Pendulum Motion Implementation.....	110
Figure 7.1 – Structure of the Simplified Human Model Windows Folder.....	112
Figure 7.2 – Main.any File of Simplified Human Model.....	113
Figure 7.3 – Segments of Lower Extremity.....	116
Figure 7.4 – Geometrical Properties of Segment 3.....	117
Figure 7.5 – Joints of Lower Extremity.....	120
Figure 7.6 – Joints.any File of the Lower Extremity.....	121
Figure 7.7 – Muscles of the Lower Extremity.....	122
Figure 7.8 – Muscles.any File of the Lower Extremity.....	122
Figure 7.9 – <i>Position</i> Anyfolder of the Main.any file.....	126
Figure 7.10 – Motion of Lower Extremity.....	128
Figure 7.11 – Inputs of CoM driver.....	130
Figure 7.12 – Motion of Lower Extremity with CoM Drive.....	130
Figure 7.13 – Linear Displacement of <i>CoM</i>.....	131
Figure 7.14 – <i>Fm</i> and <i>Activity</i> of <i>Seg2Front</i> Muscle.....	131
Figure 7.15 – <i>C</i> Joint Reaction Forces.....	131
Figure 7.16 – Segments of Trunk.....	132
Figure 7.17 – Geometrical Properties of Trunk Segments.....	132
Figure 7.18 – Joints of the Trunk.....	134
Figure 7.19 – Joints.any File of the Trunk.....	134
Figure 7.20 – Muscles of the Trunk.....	135
Figure 7.21 – Muscles.any File of the Trunk.....	135

Figure 7.22 – Motion of Trunk.....	137
Figure 7.23 – Angular Displacement of Trunk Joints.....	137
Figure 7.24 – Angular Velocity of Trunk Joints.....	138
Figure 7.25 – Kinematic Measure of Centre of Mass with the Trunk Segments.....	140
Figure 7.26 – Motion of Lower Extremity and Trunk with CoM Driver.....	140
Figure 7.27 – Segments of UpperExtremity & Box.....	141
Figure 7.28 – Geometrical Properties of Box.....	141
Figure 7.29 – Joints of Upper Extremity & Box.....	145
Figure 7.30 – Muscles of the Upper Extremity.....	146
Figure 7.31 – Muscles.any File of the Upper Extremity.....	146
Figure 7.32 – Kinematic Measures of Upper Extremities.....	148
Figure 7.33 – Drivers2.any File of the Simplified Human Model.....	149
Figure 7.34 – Box Driven Motion of the Simplified Human Model.....	152
Figure 8.1 – Body Segment Lengths Expressed as a Fraction of Body Height H.....	155
Figure 8.2 – Segments of the Lower Extremity.....	157
Figure 8.3 – Geometrical Properties of Feet.....	158
Figure 8.4 –RightFoot.any File of the Parameterized Model.....	158
Figure 8.5 – Geometrical Properties of Lower Legs.....	159
Figure 8.6 – RightLowerLeg.any File of the Parameterized Model.....	159
Figure 8.7 – Geometrical Properties of Upper Legs.....	160
Figure 8.8 – RightUpperLeg.any File of the Parameterised Model.....	160
Figure 8.9 – Geometrical Properties of the Pelvis.....	161
Figure 8.10 – Key Dimensions of the Pelvis of the Repository2 Model.....	161
Figure 8.11 – Pelvis.any File of the Parameterised model.....	162
Figure 8.12 – Joints of the Lower Extremity.....	163
Figure 8.13 – Joints.any File of the Lower Extremity.....	163
Figure 8.14 – Motion of the Lower Extremity in Three Versions.....	164
Figure 8.15 – L5L4, L3L2, Thorax & Head Segments.....	165
Figure 8.16 – Geometrical Properties of L5L4.....	166
Figure 8.17 – L5L4Seg.any File of the Parameterized Model.....	167

Figure 8.18 – Geometrical Properties of L3L2.....	167
Figure 8.19 – L3L2Seg.any File of the Parameterized Model.....	168
Figure 8.20 – Geometrical Properties of Head.....	168
Figure 8.21 – Geometrical Properties of Thorax.....	169
Figure 8.22 – CoM & MoI of HAT.....	169
Figure 8.23 – HATSeg.any File of the Parameterised Model.....	170
Figure 8.24 – Joints of Sacrum, L5L4, L4L3 & HAT Segments.....	171
Figure 8.25 – Joints.any File of Sacrum, L5L4, L3L2 & HAT Segments.....	171
Figure 8.26 – Motion of the Lower Extremity with L5L4, L3L2 & HAT Segments.....	172
Figure 8.27 – Segments of the Upper Limb.....	173
Figure 8.28 – Geometrical Properties of Clavicle.....	174
Figure 8.29 – RightClavicle.any File of the Parameterised Model.....	174
Figure 8.30 – Geometrical Properties of Scapula.....	175
Figure 8.31 – RightScapula.any File of the Parameterised Model.....	175
Figure 8.32 – Geometrical Properties of Humerus.....	176
Figure 8.33 – RightHumerus.any File of the Parameterised Model.....	176
Figure 8.34 – Geometrical Properties of Ulna.....	177
Figure 8.35 – RightUlna.any File of the Parameterised Model.....	177
Figure 8.36 – Geometrical Properties of Radius.....	178
Figure 8.37 – RightRadius.any File of the Parameterised Model.....	178
Figure 8.38 – Geometrical Properties of Hand.....	179
Figure 8.39 – RightHand.any File of the Parameterised Model.....	179
Figure 8.40 – Joints of the Upper Limb.....	180
Figure 8.41 – Joints.any File of the Right Upper Limb.....	180
Figure 8.42 – Motion of the Human Parameterized Models.....	182
Figure 8.43 – Anterior View of the Motion of the 1.7m & 1.5m Human Models.....	183
Figure 8.44 – Posterior View of the Motion of the 1.7m & 1.5m Human Models.....	184
Figure 8.45 – Anterior View of the Motion of the 1.9m & 1.5m Human Models.....	185
Figure 8.46 – Posterior View of the Motion of the 1.9m & 1.5m Human Models.....	186

Figure 8.47 – Anterior View of Case 1 Application of the Human Model.....187
Figure 8.48 – Anterior View of Case 2 Application of the Human Model.....187
Figure 8.49 – Right Lateral View of Case 1.....188
Figure 8.50 – Right Lateral View of Case 2.....188
Figure 8.51 – Cases 1 & 2 Reaction Forces of Sacrum/Lumbar L5 Joint.....189
Figure 8.52 – Cases 1 & 2 Muscle Force of Right Rectus Femoris Muscle.....189

LIST OF TABLES

Table 3.1 – Dempster’s Segment Endpoints Definitions.....	28
Table 3.2 – Dempster’s Body Segment Parameters.....	30
Table 3.3 – Clauser et al.’s Segment Endpoints Definitions.....	31
Table 3.4 – Clauser, McConville, and Young’s Body Segment Parameters.....	32
Table 3.5 – Vaughan, Davis, and O’Connor’s Body Segment Parameters.....	33
Table 6.1 – Main.any/Enironment.any Files.....	79
Table 6.2 – Main.any/Seg1.any Files.....	81
Table 6.3 – Main.any/Joints.any Files.....	82
Table 6.4 – Main.any/Drivers1.any Files.....	83
Table 6.5 - Main.any/Muscles.any Files.....	84
Table 6.6 – Double Inverted Pendulum Anyfiles.....	91
Table 6.7 – Triple Inverted Pendulum Anyfiles.....	95
Table 6.8 – Main.any/Initial Conditions.any Files.....	99
Table 6.9 – Main.any/Joints.any/KinematicMeasures.any/Drivers2.any Files.....	101
Table 7.1 – Anyfiles of Lower Extremity Segments.....	117
Table 7.2 – Initial Conditions Anyfiles.....	119
Table 7.3 – Kinematic Measures/Driver of Right Heel (<i>Seg0 - NodeHeel</i>).....	124
Table 7.4 – Kinematic Measures/Driver of Left Heel (<i>Seg0L - NodeHeelL</i>).....	125
Table 7.5 – Kinematic Measures/Driver of Pelvis (<i>Seg3 - NodeSeg3_0</i>).....	125
Table 7.6 – Kinematic Measure/Driver of Centre of Mass of the Lower Extremity.....	129
Table 7.7 – Anyfiles of the Trunk Segments.....	133
Table 7.8 – Kinematic Measure/Driver of Trunk.....	136
Table 7.9 – Anyfiles of Upper Extremity Segments & Box.....	142
Table 7.10 – Initial Conditions Anyfiles of Upper Extremities.....	144
Table 7.11 – Joints.any Files of the Upper Extremity & Box.....	145

Table 7.12 – CoM/Box Kinematic Measures & Box Driver.....151
Table 8.1 – Dempster’s Body Segment Parameters.....156
Table 8.2 – Repository2 Model’s Length of L5L4/L3L2/HAT Segments.....165
Table 8.3 – Rough Estimation of L5L4/L3L2/HAT Segments in Contini Model.....166

CHAPTER 1

INTRODUCTION

1.1 Motivation

All living organisms are made up of one or more cells. The human body, like most organisms, is made up of billions of cells organized into complex functional groups. One of these groups is the musculoskeletal system which is responsible for bringing about human movement under the control of the nervous system. By coordinating activity among the various muscle groups, forces generated by the muscles are transmitted by the bones and joints to enable an individual to maintain an upright or partially-upright body posture and to voluntarily move. The musculoskeletal system, also referred to as the locomotor system, is essentially a machine - a powered mechanism for generating and transmitting forces to counteract the effects of gravity and to bring about desirable movements of the body.

Scientists, being fascinated by human movement, have performed an extensive range of experiments to characterize muscle mechanics, describe musculoskeletal geometry and quantify movement dynamics. However, they early realised the need of computer simulations to complement experimental studies because important elements of movement, including neural signals and muscle forces, are extremely difficult to measure experimentally. With the PC becoming more and more powerful, researchers around the world have developed software systems that let users create computer simulations of human movement. Once researchers develop and test a computer model of the musculoskeletal system, they can use it to explore movement control, analyze athletic performance, and simulate treatments for musculoskeletal disorders.

1.2 Objectives

In performing the computer simulation of a person performing a task, the human body is considered to be a system of mechanical links, with each link of known physical size and form. Science of Anthropometry is the one that defines these size and form properties. Without anthropometrical data, biomechanical models to predict human reach, force, and space requirements cannot be developed.

The overall goal of this project is the development of a parameterised musculoskeletal model of the human body combining the powerful modelling capabilities of a musculoskeletal software with the valuable information that anthropometrical studies provide. The final result will be a human model which can be scaled to global parameters such as weight and height.

1.3 Layout of Thesis

The research that is undertaken in this thesis is described as follows. Chapter 2 is dealing with a brief analysis of the human musculoskeletal system. It describes the bones which comprise the skeleton, the types of joints of the articular system and the various muscle groups of the muscular system. Chapter 3 refers on the advantages and limitations of computer models, and outlines the research method that is widely used for the investigation of human movement. It also contains a review in anthropometry concerning the categories in which the attempts for quantifying body segment parameters fall, and describes how this science is related to the current project. In the chapter that follows, Chapter 4, the most popular musculoskeletal software packages are outlined, followed with some of their real application examples. Chapter 5 then, comes to give an introduction to the software system that will be used in this research, the Anybody modelling system. It starts with the description of the programming language and the basic concepts of this system, and continues with the mechanical elements, the types of simulation and the muscle recruitment solvers that this system provides. Further chapters are dealing with the development of models using Anybody technology. In Chapter 6, the development of a single, double and triple inverted pendulum is being investigated, giving to a novel user, a smooth introduction of how a multibody system can be created, and perform a wide range of motion in space, with or without muscles. Being confident with the simple model of Chapter 6, thesis is continued with the development of a simple human-like model which intends to be the basic platform on which a more realistic model of the human body will be developed. This is discussed in Chapter 7. The final target of this project, the development of a parameterized musculoskeletal model is analyzed in Chapter 8. Conclusions of this research and some further work that has to be done for future development of this project are then discussed in Chapter 9.

CHAPTER 2 MUSCULOSKELETAL SYSTEM

2.1 Introduction

The musculoskeletal system consists of the skeletal system (Figure 2.1) and the muscular system (Figure 2.2). The skeletal system consists of the skeleton, the bones and the various structures forming the joints between the bones. It is similar to the metal framework of a building in that it gives the body its basic shape and provides a framework for all the other systems. Bone is an ideal support material since it is not only strong, but also fairly lightweight. The adult skeletal system normally has 206 bones, more than 200 joints, and accounts for between 12% and 15% of the total body weight¹.

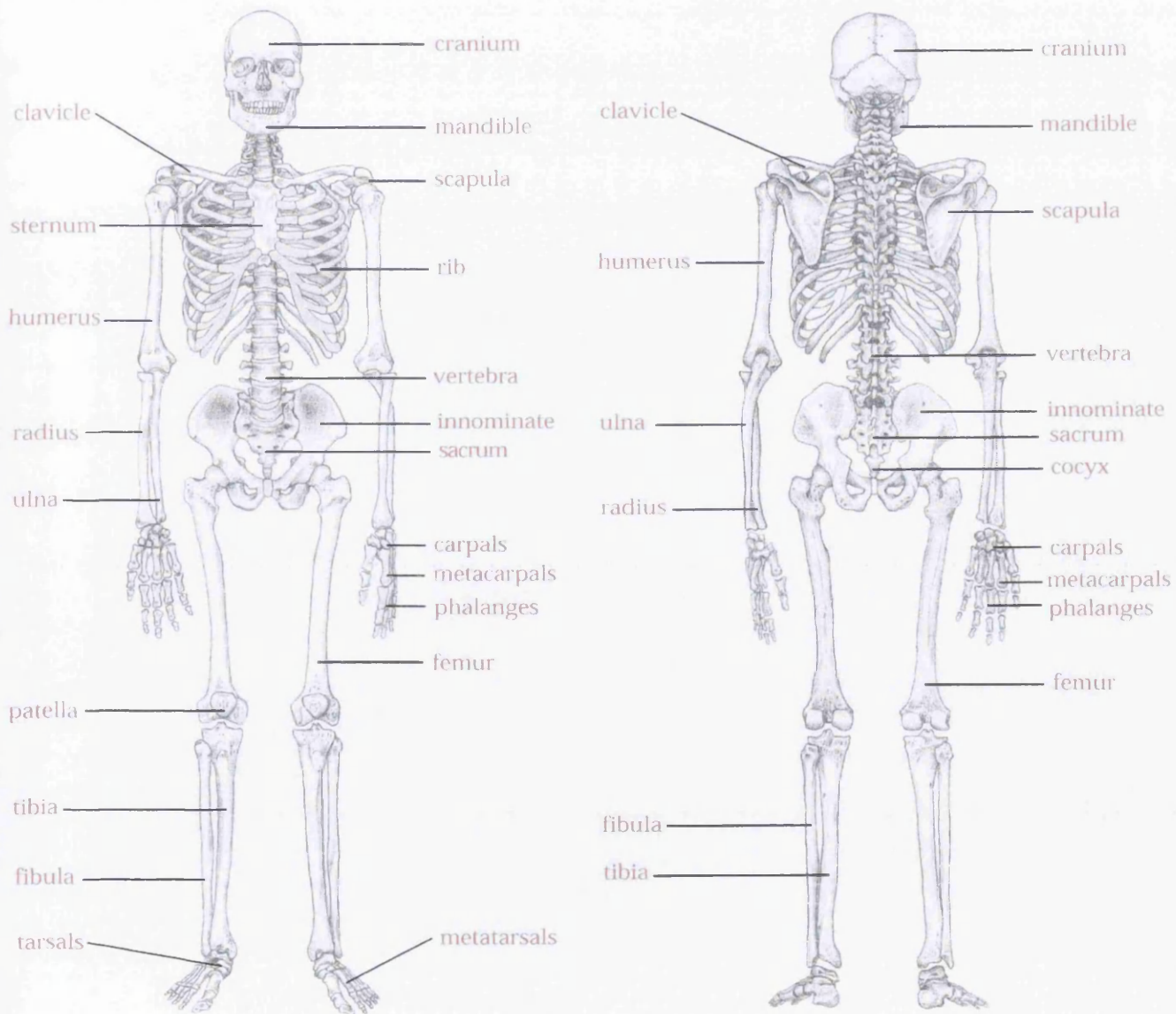


Figure 2.1 – Skeletal System [2]

The muscular system, refers only to the skeletal muscles. Visceral and cardiac muscle tissue belong to other systems. For example, cardiac muscle tissue, found only in the heart, is part of the cardiovascular system. The skeletal muscles are the powerhouse of the body; they are the organs that enable us to make voluntary, controlled movements of the body. With regard to the muscular system, there are approximately 640 skeletal muscles, and they account for about 36% and 45% of total body weight in untrained females and males, respectively¹.

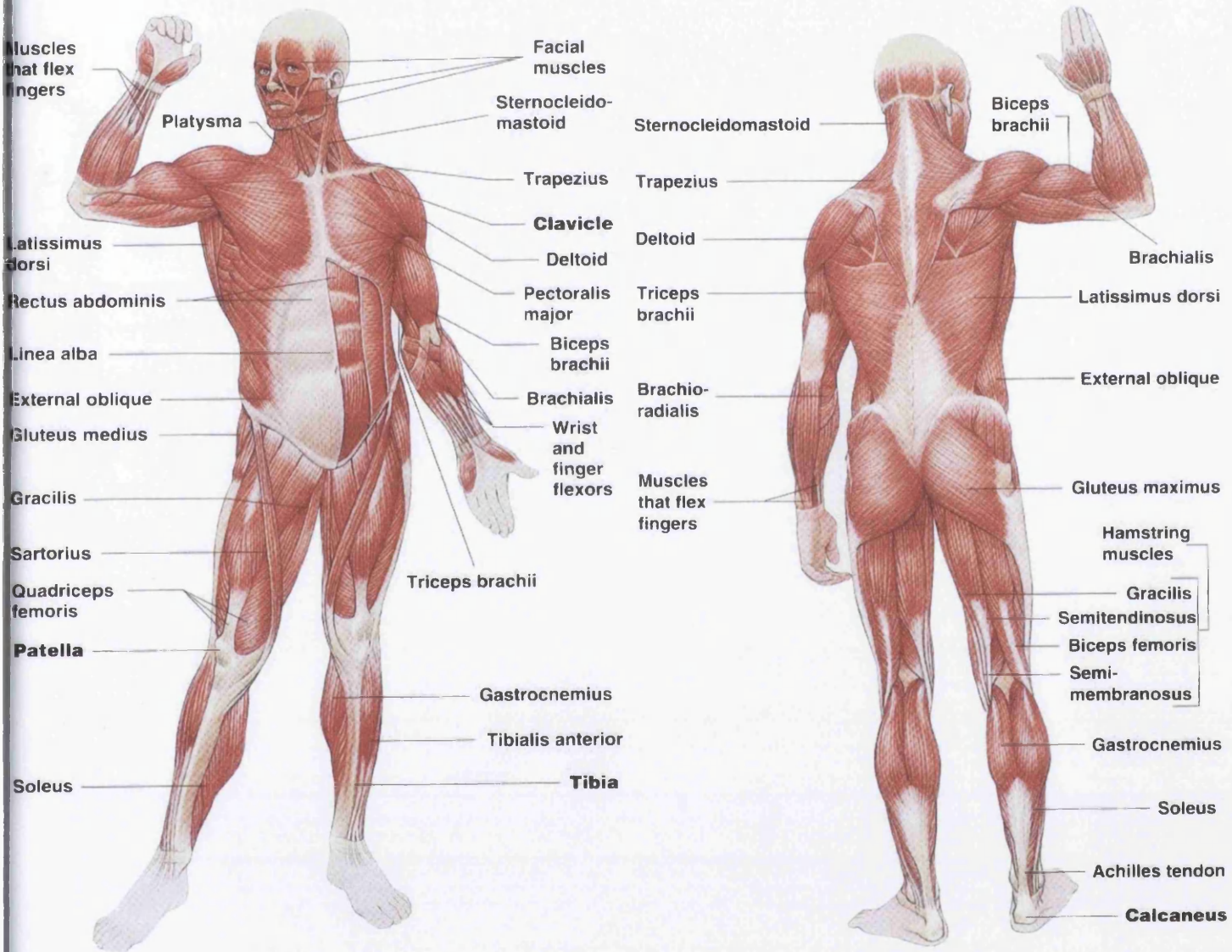


Figure 2.2 – Muscular System [3]

2.2 The Skeleton

2.2.1 Function of the Skeleton

The skeleton performs three main mechanical functions:

- It acts as a supporting framework for the rest of the body.
- It acts as a system of levers on which the muscles can pull in order to stabilize and move the body.
- It protects certain organs. For example the spinal cord and the rib cage help to protect the heart and the lungs.

2.2.2 Terminology

For descriptive purposes the bones are usually divided into two main groups: the **axial skeleton** and the **appendicular skeleton** (Figure 2.3). The adult axial skeleton consists of 80 bones comprising the skull, vertebral column, and ribs. The adult appendicular skeleton consists of 126 bones that make up the upper limbs (arms and hands) and the lower limbs (legs and feet). These two groups will be discussed later in this chapter.

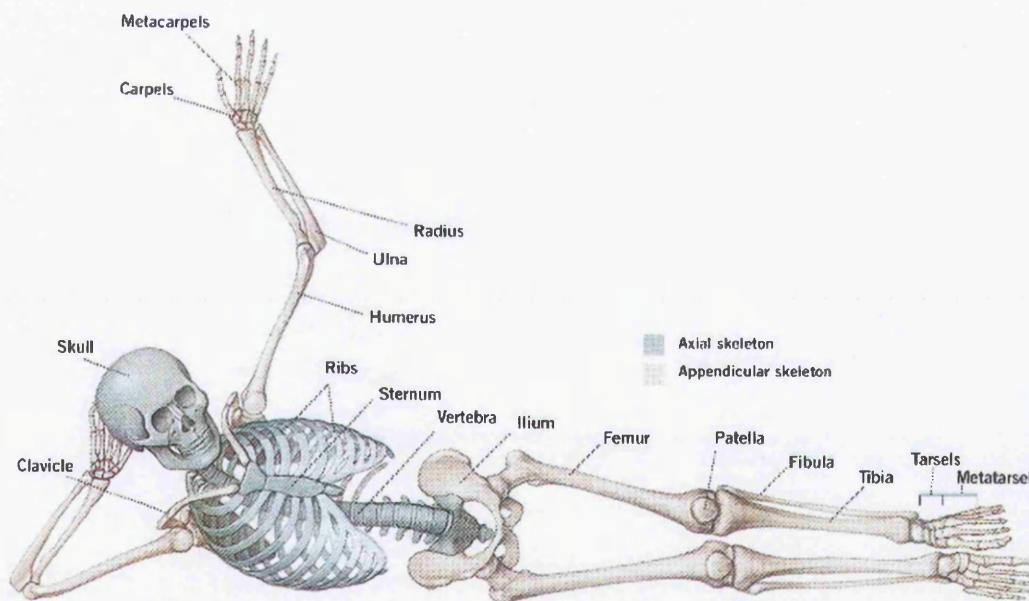


Figure 2.3 – Axial/Appendicular Skeleton [3]

In anatomy, the term **aspect** (Figure 2.4) refers to the appearance of a particular bone (or any other part of the body) from a particular viewpoint. For example, the anterior aspect of the skeleton refers to the features of the skeleton seen from an anterior (frontal) viewpoint. Similarly, lateral aspect (view from the

side), posterior aspect (view from the back), superior aspect (view from above), and inferior aspect (view from below) describe other viewpoints.

In order to describe the spatial orientation of the position of one bone (or a body part) in relation to another, it is necessary to use standard terminology with reference to a standard body posture. In the standard posture, also called the anatomical position (Figure 2.4), the body is upright with the arms by the sides and palms of the hands facing forward. There are three main **reference planes**: median, coronal, and transverse.

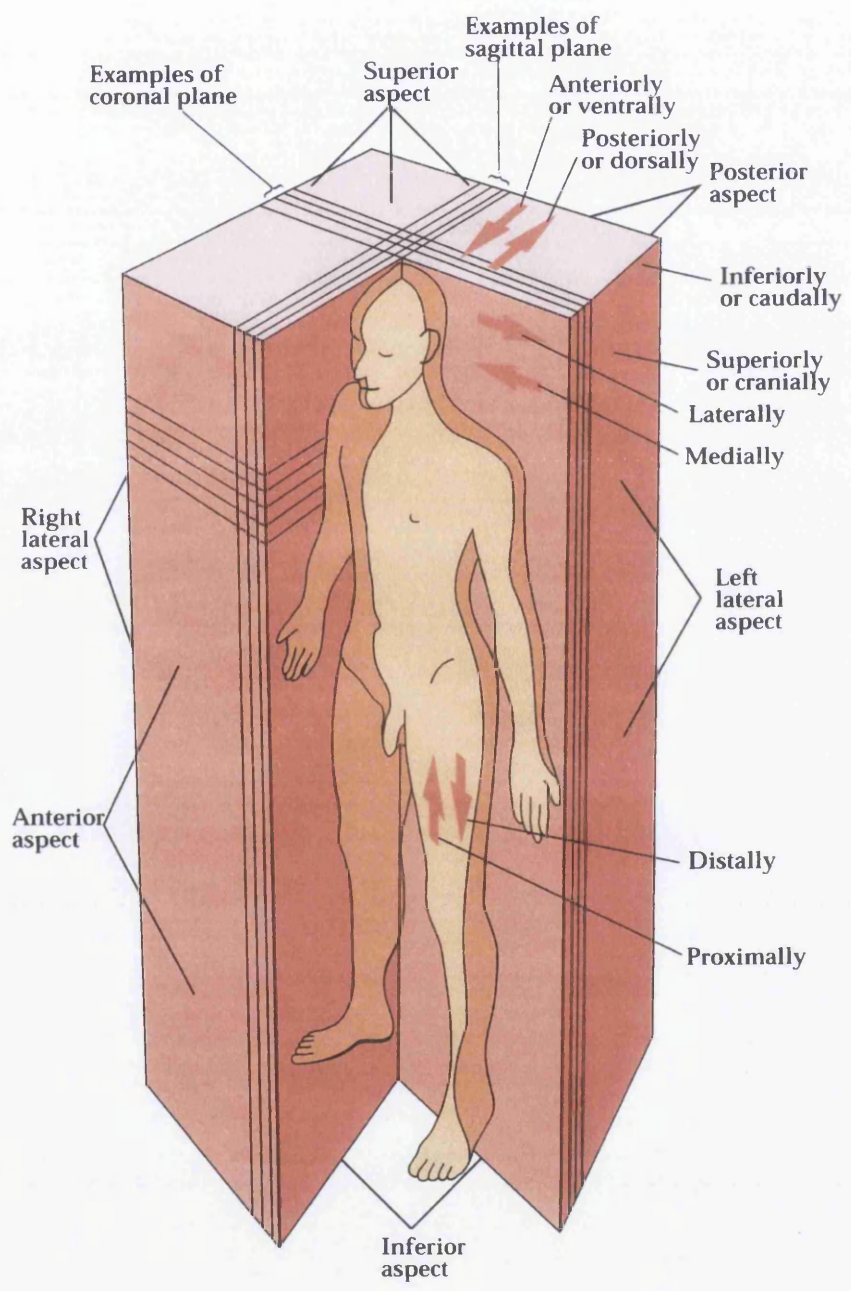


Figure 2.4 – Reference Planes [2]

The median plane is a vertical plane that divides the body down the middle into more-or-less symmetrical left and right portions (Figure 2.4). The median plane is also frequently referred to as the sagittal plane.

The terms lateral and medial are used to describe the relationship of a body part to the median plane. Lateral means farther away from the median plane and medial means closer to the median plane. Any vertical plane perpendicular to the median plane is called a coronal plane (Figure 2.4). The terms anterior (in front) and posterior (behind) describe the position of structures with respect to coronal planes. Any plane perpendicular to both the median and coronal planes is called a transverse plane. All transverse planes are horizontal (Figure 2.4). The terms superior (above or upward) and inferior (below or downward) describe the position of structures with respect to transverse planes. However, there are some spatial terms that apply to some bones, but not to others. For example, the terms proximal and distal (Figure 2.4) are normally only used in reference to the long bones of the limbs. Superior features of these bones (with respect to the anatomical position) are referred to as proximal, whereas inferior features of these bones are referred to as distal. For example, in each arm the proximal end of the humerus (upper arm bone) articulates with its corresponding scapula to form the shoulder joint. The distal end of the humerus articulates with the proximal ends of the radius and ulna (forearm bones) to form the elbow joint. The distal ends of the radius and ulna articulate with the carpals (wrist bones) to form the wrist joint.

2.2.3 The Axial Skeleton

2.2.3.1 The Skull

The skull (Figure 2.5) consists of 29 fairly flat or irregular bones that encase the brain, provide bases for the major sense organs, and form the upper and lower jaws. The 29 bones of the skull comprise

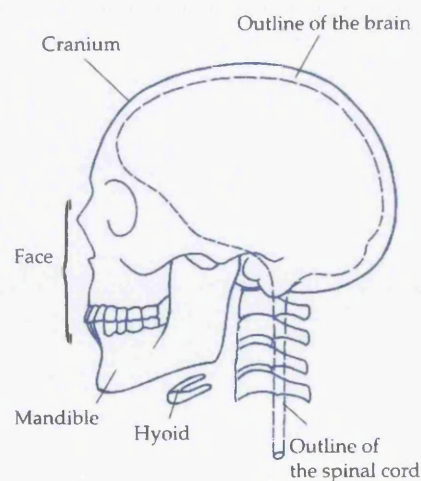


Figure 2.5 – The Skull [4]

8 cranial bones (cranium), 13 facial bones (face), 6 ear ossicles (3 in each middle ear), the mandible (lower jaw), and the hyoid bone (part of the larynx). The bones of the cranium and face form a single unit

that makes up most of the skull. The cranium encloses the brain and is made up of 8 relatively flat irregular bones.

2.2.3.2 The Vertebral Column

The vertebral column-also referred to as the backbone or spine- comprises 26 irregular bones (called vertebrae) stacked on top of each other. The vertebral column supports the weight of the head, arms, and trunk, and provides protection for the spinal cord. The vertebrae are divided into five fairly distinct groups: cervical, thoracic, lumbar, sacral, and coccygeal (Figure 2.6). The neck consists of 7 cervical vertebrae. The thoracic or chest region consists of 12 thoracic vertebrae that provide articulation for the 12 pairs of ribs. The lower back consists of 5 lumbar vertebrae. The 5 sacral vertebrae form the posterior part of the pelvis; at maturity the sacral vertebrae fuse together to form the sacrum. The 4 or 5 coccygeal vertebrae are small and represent a vestigial tail. The coccygeal vertebrae normally fuse together at maturity to form the coccyx or tailbone.

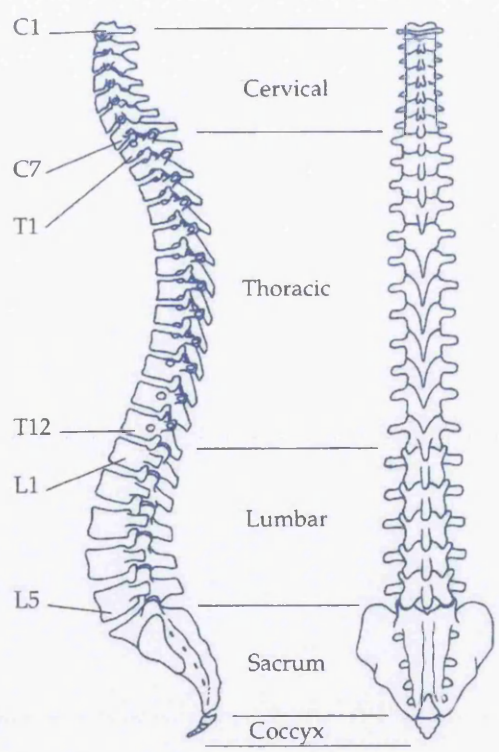


Figure 2.6 – The Vertebral Column [4]

2.2.3.3 The Rib Cage

The rib cage is roughly the shape of an upright cone partially flattened from front to back (Figure 2.7). The rib cage consists of 12 pairs of ribs and the sternum. The rib cage as a whole is a fairly flexible

structure providing protection for the heart and lungs, and is also very important in the ventilation of the lungs during breathing.

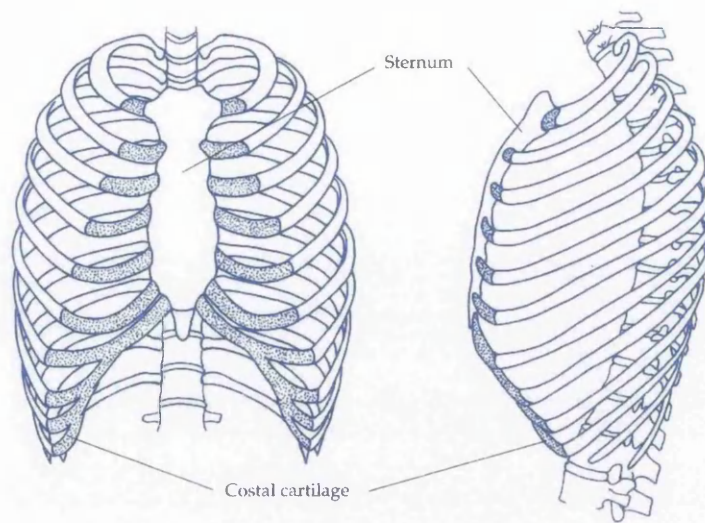


Figure 2.7 – The Rib Cage [4]

2.2.4 The Appendicular Skeleton

2.2.4.1 The Upper Limb

The upper limbs (Figure 2.8) consist of 64 bones, 32 bones in each limb, and can be divided into five regions: shoulder, upper arm, lower arm, wrist and hand.

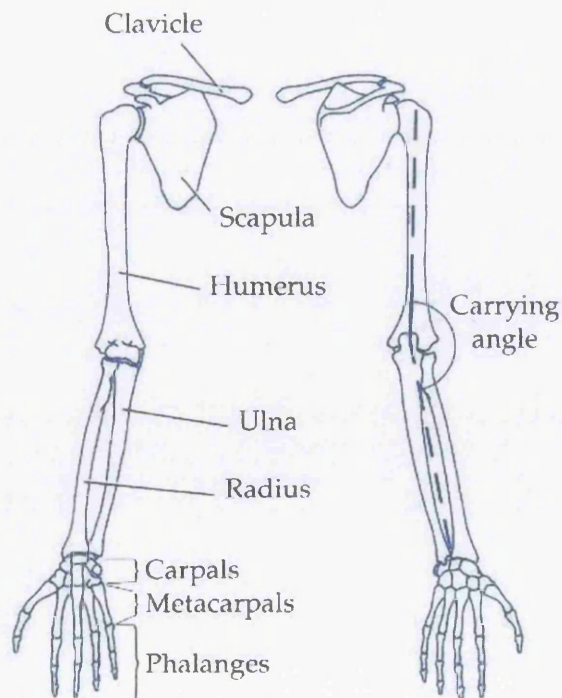


Figure 2.8 – Upper Limbs [4]

The shoulder region consists of the scapula (shoulder blade) and clavicle (collarbone). The medial end of the clavicle articulates with the manubrium to form the sternoclavicular joint linking the upper limb and the axial skeleton. The lateral end of the clavicle articulates with the scapula to form the acromioclavicular joint. Together with the manubrium, the scapulae and clavicles of both upper limbs form an incomplete ring called shoulder girdle (Figure 2.9).

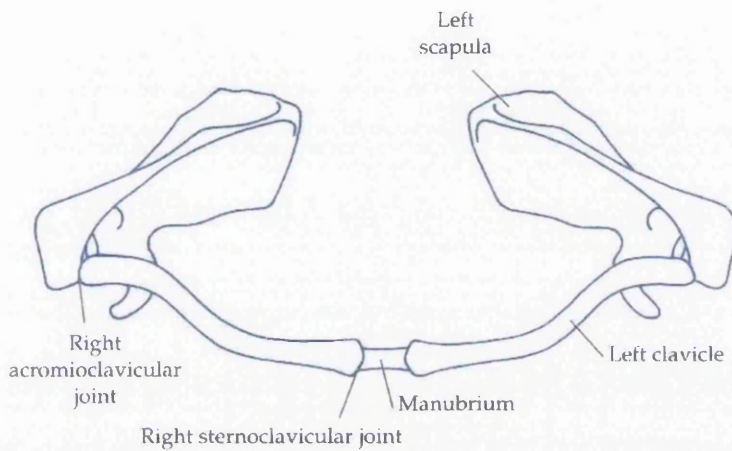


Figure 2.9 – The Shoulder Girdle [4]

The humerus is the only bone of the upper arm. The proximal end articulates with the scapula to form the shoulder joint (glenohumeral joint), and the distal end articulates with the radius and ulna to form the elbow joint (Figure 2.8).

The radius and the ulna are the two long bones in the lower arm. The proximal ends of the radius and ulna articulate with the distal end of the humerus to form the elbow joint, and the distal ends articulate with the proximal row of carpals to form the wrist joint (Figure 2.8). The radius and ulna articulate with each other at their proximal and distal ends such that the radius is able to move relative to ulna. Movement of the radius toward the anatomical position is called supination and movement of the radius away from the anatomical position is called pronation.

The wrist consists of eight small irregular bones called carpals that articulate with each other to form the carpus. The carpals are arranged in a proximal and distal row. The proximal row articulates with the distal ends of the radius and ulna to form the wrist joint (Figure 2.8). The series of joints between the proximal and distal rows is called the midcarpal joint.

The hand consists of five metacarpals and fourteen phalanges that are all miniature long bones. Each finger consists of three phalanges, whereas the thumb has only two phalanges. The joints between the phalanges, two joints in each finger and one joint in the thumb, are called interphalangeal joints.

2.2.4.2 The Lower Limb

The lower limbs consist of 62 bones, 31 bones in each limb, and can be divided into four regions: pelvis, upper leg, lower leg and foot (Figure 2.10).

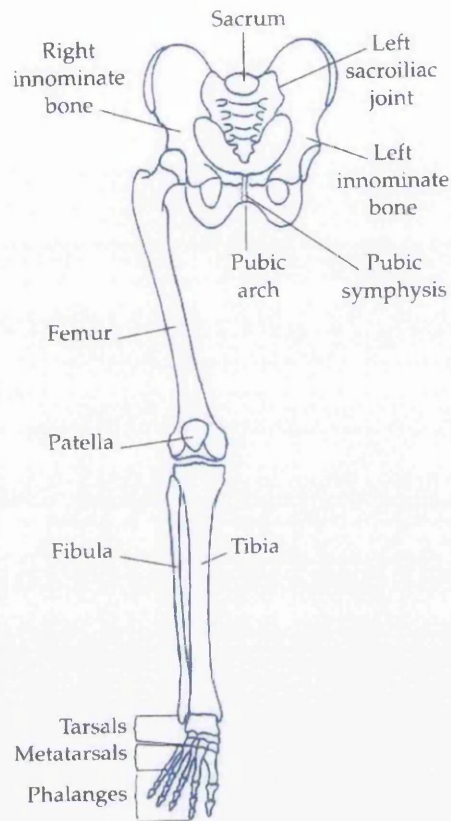


Figure 2.10 – Lower Limbs [4]

Together with the sacrum, the right and the left innominate bones form the pelvic girdle or pelvis (Figure 2.10). The innominate bones link the lower limbs to the axial skeleton. Each innominate bone develops from three bones called ilium, ischium, and pubis, which fuse together at maturity.

The femur and patella make up the upper leg (Figure 2.10). The femur is the longest and strongest bone in the skeleton. The proximal end of the femur articulates with the innominate bone to form the hip joint. The distal end articulates with the tibia to form the tibiofemoral joint and with the patella to form the patellofemoral joint.

The tibia and the fibula are the two long bones in the lower leg (Figure 2.10). The tibia is much thicker than the fibula and is almost completely responsible for transmitting loads between the upper leg and foot. The fibula helps form the ankle joint and provides additional area of attachment for muscles of the lower leg. The tibia and fibula articulate with each other at their proximal and distal ends but have little or no movement relative to each other.

The foot consists of seven tarsals, five metatarsals, and fourteen phalanges. The tarsals are irregular bones articulating with each other to form the tarsus, which forms the posterior half of the foot. The lower leg articulates with the tarsus at the ankle joint.

2.3 The Articular System

2.3.1 Introduction

A joint, also referred to as an articulation or an arthrosis, is defined as a region where two or more bones are connected. The adult skeleton normally has 206 bones linked by approximately 320 joints. The articular system refers to all of the joints of the body.

The human body is capable of a broad range of movements facilitated by the combined effects of the open-chain arrangement of the bones, the number of joints linking the bones, the different types of joints, and the range of movement in the joints. Most joints allow a certain amount of movement and all joints transmit forces. The function of joints is largely mechanical-to facilitate relative motion between bones and transmission of force from one bone to another.

2.3.2 Terminology

In describing a particular joint's movement, it is useful to refer to three mutually perpendicular reference axes. The three reference axes denote anteroposterior (front to back), transverse (side to side), and vertical directions with respect to the anatomical position. Figure 2.11 shows the position of the reference axes in relation to the shoulder joint. With respect to the reference axes, there are six possible directions, called degrees of freedom, in which the shoulder joint, or any other joint, might be able to move, depending upon its structure. The six directions consist of three linear directions (along the axes) and three angular directions (around the axes). A joint with six degrees of freedom could move in any direction by a combination of linear and angular movements.

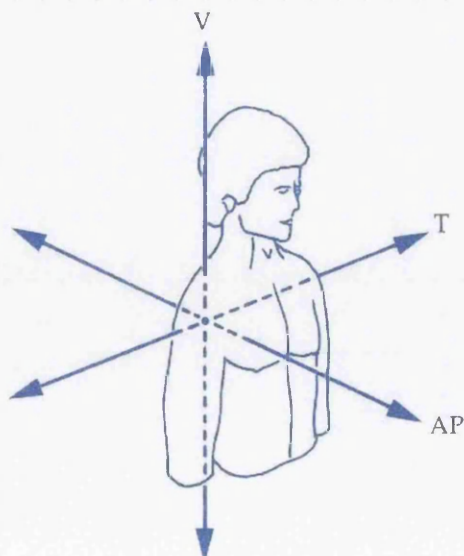


Figure 2.11 – Reference Axes [4]

In most joints the terms abduction and adduction refer to rotations around the anteroposterior axis. In the shoulder, wrist, and hip joints, abduction and adduction refer to movement of the arm, hand, and leg away from and toward the median plane, respectively (Figure 2.12).

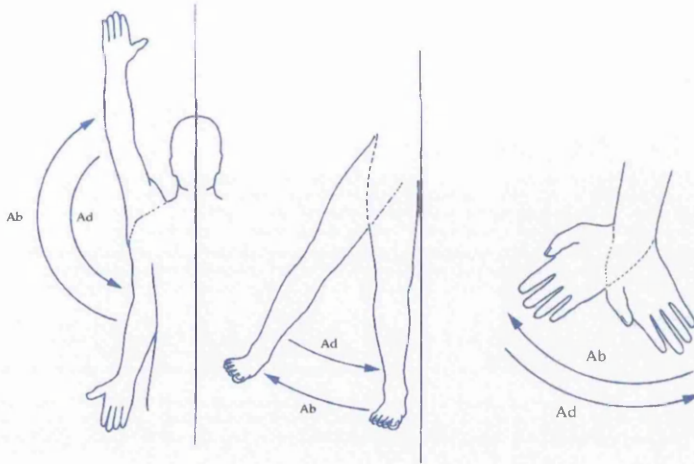


Figure 2.12 – Abduction/Adduction of Shoulder, Hip & Wrist joints [4]

In the hand and foot, abduction occurs when the fingers and toes are spread, and adduction occurs when the fingers and toes are returned to the reference position (Figure 2.13).



Figure 2.13 – Abduction/Adduction of Hand & Foot Joints [4]

In most joints the terms flexion and extension refer to rotation around the transverse axis. In the shoulder, wrist, and hip joints, flexion refers to movement of the arm, hand, and leg forward, and extension refers to movement of the arm, hand, and leg backward (Figure 2.14).

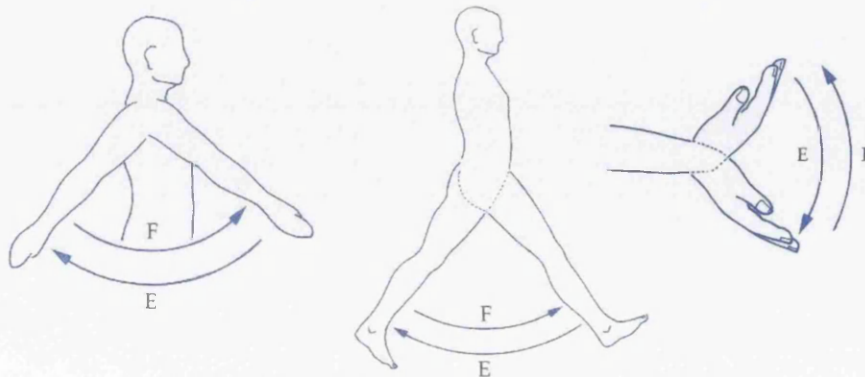


Figure 2.14 – Flexion/Extension of Shoulder, Hip & Wrist Joints [4]

In the elbow, knee, metacarpophalangeal, and interphalangeal (hand & foot) joints, flexion occurs when the joints bend and extension occurs when the joints straighten (Figure 2.15).

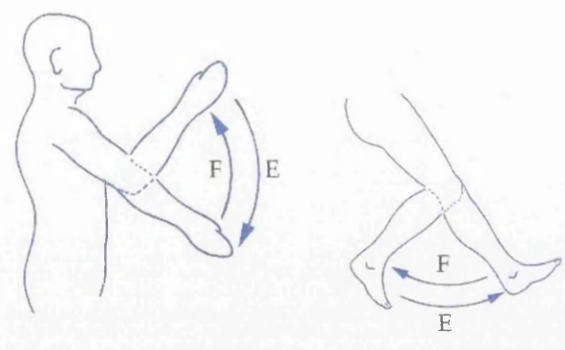


Figure 2.15 – Flexion/Extension of Elbow & Knee Joints [4]

In the trunk-the vertebral column as a whole-flexion refers to bending the trunk forward and extension refers to the reverse movement. Lateral flexion of the trunk occurs when the trunk bends to the side about an anteroposterior axis (Figure 2.16).

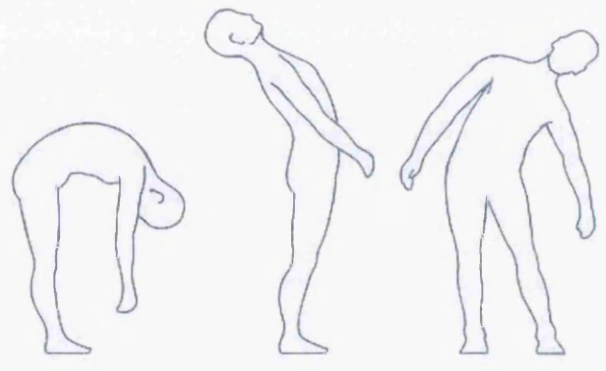


Figure 2.16 – Flexion/Extension of Trunk Joints [4]

2.3.3 Classification of Joints

- **Synarthrosis** is an **immoveable** joint. These joints are often called fixed joints and allow no movement between bones. These joints are interlocked and held together by connective tissue, or they are fused together. The places where the bones of the skull meet (sutures) each other (Figure 2.17) are examples of immoveable joints.

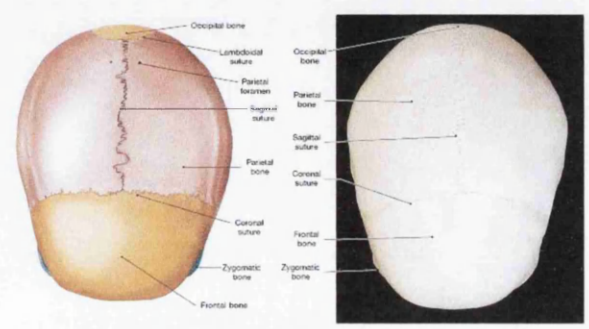


Figure 2.17 – Synarthrosis Joint [5]

- **Amphiarthrosis** is a **slightly moveable** joint. These joints permit a small amount of movement between the bones. These bones are farther apart from each other than immoveable joint bones. The joints between the two bones of the lower leg (tibia and fibula) and the joints of the vertebrae are examples of slightly movable joints (Figure 2.18).

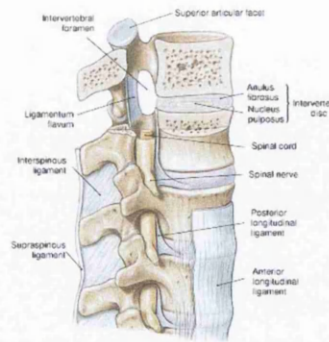


Figure 2.18 – Amphiarthrosis Joint [5]

- **Diarthrosis** is a **freely moveable** joint. Most of the joints of the body are freely moveable joints. These joints are grouped according to the shapes of surfaces of the adjacent bones. There are five types of freely movable joints:

a. **Ball & Socket Joint** – Movement is in three planes. Permits circular movement-the widest range of motion. This type of joint is the hip joint and the shoulder joint, which enables the arm to move up, down, forward, backward, as well as to rotate it in a complete circle (Figure 2.19).

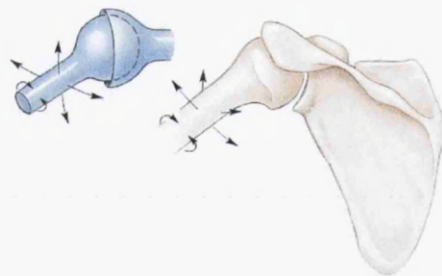


Figure 2.19 – Ball & Socket Joint [5]

b. **Hinge Joint** – Movement is in one plane only. Permits a back-and-forth motion. This type of joint is the knee joint which enables the leg to flex and extend and the elbow joint (Figure 2.20) which allows the forearm to move forward and backward.

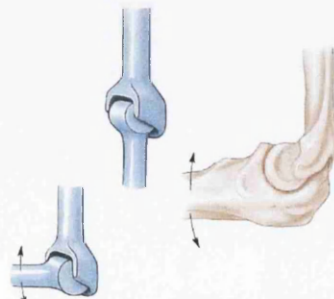


Figure 2.20 – Hinge Joint [5]

- c. **Pivot Joint** – Movement is in one plane only. Permits rotation of one bone around another. The elbow enables the hand to turn over. It also allows the head to turn from side to side and the motion between atlas & axis at the vertebrae (Figure 2.21).

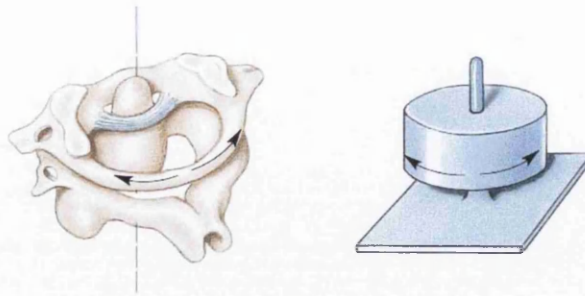


Figure 2.21 – Pivot Joint [5]

- d. **Gliding Joint** – Movement is in two planes only. Permits a sliding motion of one bone over another. Found at the ends of the collarbones, between wrist bones (Figure 2.22), and between anklebones.

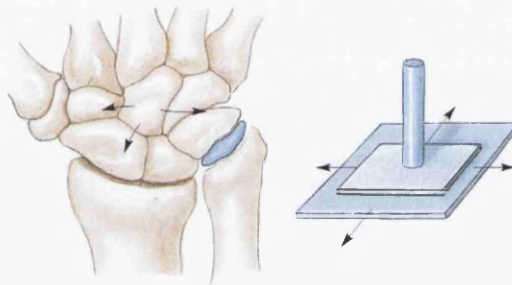


Figure 2.22 – Gliding Joint [5]

- e. **Saddle Joint** – Movement is in two planes. Articular surfaces of both bones are concave in one direction and convex in the other. This type of joint is found between 1st metacarpal of thumb & carpal (Figure 2.23).

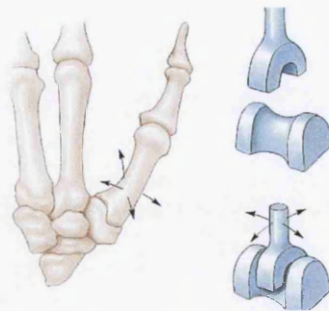


Figure 2.23 – Saddle Joint [5]

2.4 The Skeletal Muscles

2.4.1 Function of Muscles

The predominant function of muscles is contractibility. When a muscle contracts it tends to shorten. However, the muscle may shorten, lengthen, or stay in the same length depending on the external load on the muscle (the load tending to lengthen the muscle). Nearly all movement in the body is the result of muscle contraction. There are three types of muscular contraction:

- **Concentric** contraction-The length of the muscle will shorten if the muscle force is greater than the external load.
- **Eccentric** contraction-The length of the muscle will lengthen if the muscle force is less than the external load.
- **Isometric** contraction-The length of the muscle will not change if the muscle force is equal to the external load.

In addition to movement, muscle contraction also fulfills some other important functions in the body, such as posture, joint stability, and heat production. Posture, such as sitting and standing, is maintained as a result of muscle contraction. The skeletal muscles are continually making fine adjustments that hold the body in stationary positions. The tendons of many muscles extend over joints and in this way contribute to joint stability. This is particularly evident in the knee and shoulder joints, where muscle tendons are a major factor in stabilizing the joint. Heat production, to maintain body temperature, is an important by-product of muscle metabolism. Nearly 85 percent of the heat produced in the body is the result of muscle contraction.

2.4.2 Muscles of the Head & Neck

Humans have well-developed muscles in the face that permit a large variety of facial expressions. Because the muscles are used to show surprise, disgust, anger, fear, and other emotions, they are an important means of nonverbal communication. Muscles of facial expression include frontalis, orbicularis oris, laris oculi, buccinator, and zygomaticus. These muscles of facial expressions are identified in the illustration below (Figure 2.24).

There are four pairs of muscles that are responsible for chewing movements or mastication. All of these muscles connect to the mandible and they are some of the strongest muscles in the body. Two of the muscles, temporalis and masseter are identified in Figure 2.24.

There are numerous muscles associated with the throat, the hyoid bone and the vertebral column. Two of the more obvious and superficial neck muscles are identified in Figure 2.24. They are sternocleidomastoid and trapezius⁶.

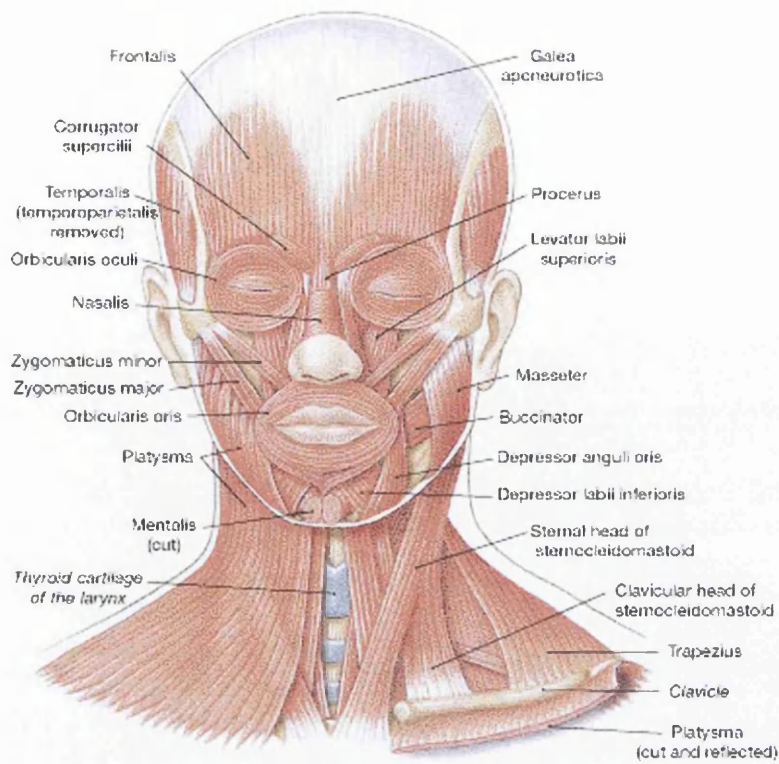


Figure 2.24 – Head & Neck Muscles [5]

2.4.3 Muscles of the Trunk

The muscles of the trunk include those that move the vertebral column, the muscles that form the thoracic and abdominal walls, and those that cover the pelvic outlet.

The erector spinae group (Figure 2.26) of muscles on each side of the vertebral column is a large muscle mass that extends from the sacrum to the skull. These muscles are primarily responsible for extending the vertebral column to maintain erect posture. The deep back muscles occupy the space between the spinous and transverse processes of adjacent vertebrae.

The muscles of the thoracic wall are involved primarily in the process of breathing. The intercostal muscles (Figure 2.25) are located in spaces between the ribs. They contract during forced expiration. External intercostal muscles (Figure 2.25) contract to elevate the ribs during the inspiration phase of breathing⁶.

The abdomen, unlike the thorax and pelvis, has no bony reinforcements or protection. Some of the abdomen muscles (Figure 2.24) are the following muscles: the internal and external abdominal oblique which with muscles of opposite side, abduct and rotate trunk, and the rectus abdominis which flexes trunk.

Some of the muscles of the pelvis are the gluteus maximus which extends and laterally rotates hip and the gluteus medius and minimus which abduct and medially rotate hip⁷.

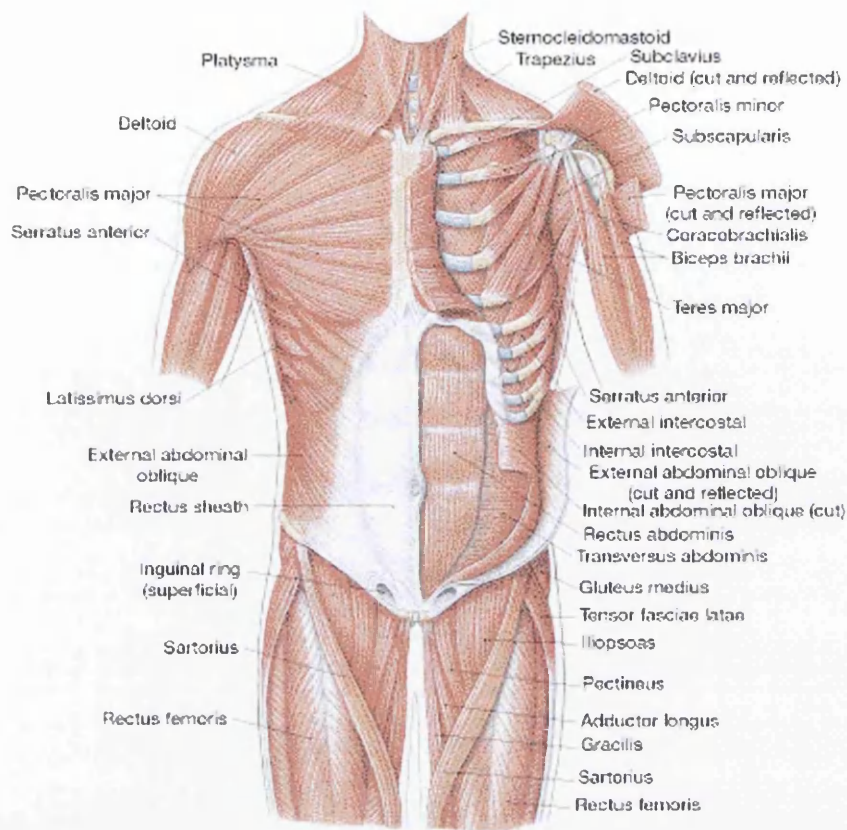


Figure 2.25 – Muscles of Trunk (Anterior View) [5]

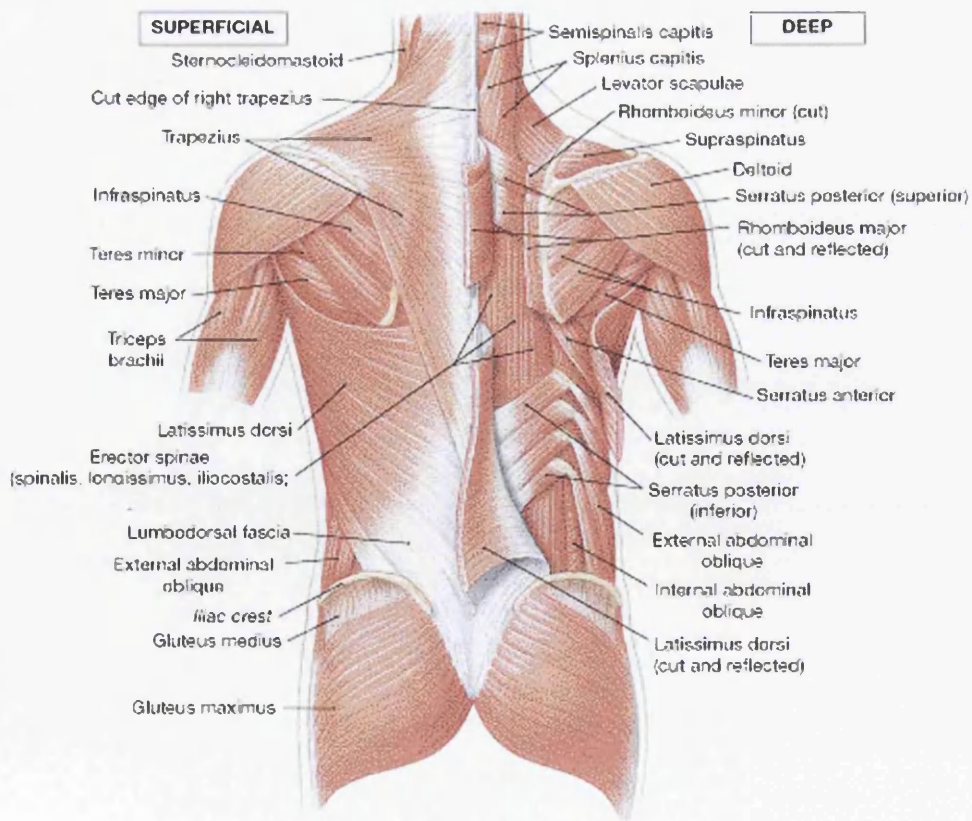


Figure 2.26 – Muscles of Trunk (Posterior View) [5]

2.4.4 Muscles of the Upper Extremity

The muscles of the upper extremity include those that attach the scapula to the thorax and generally move the scapula, those that attach the humerus to the scapula and generally move the arm, and those that are located in the arm or forearm that move the forearm, wrist, and hand. The illustration below shows some of the muscles of the upper extremity (Figure 2.27).

Muscles that move the shoulder and arm include the trapezius and serratus anterior. The pectoralis major, latissimus dorsi, deltoid, and rotator cuff muscles connect to the humerus and move the arm (Figures 2.25–2.26). The muscles that move the forearm are located along the humerus, which include the triceps brachii, biceps brachii, brachialis, and brachioradialis (Figure 2.27). The 20 or more muscles that cause most wrist, hand, and finger movements are located along the forearm⁶.

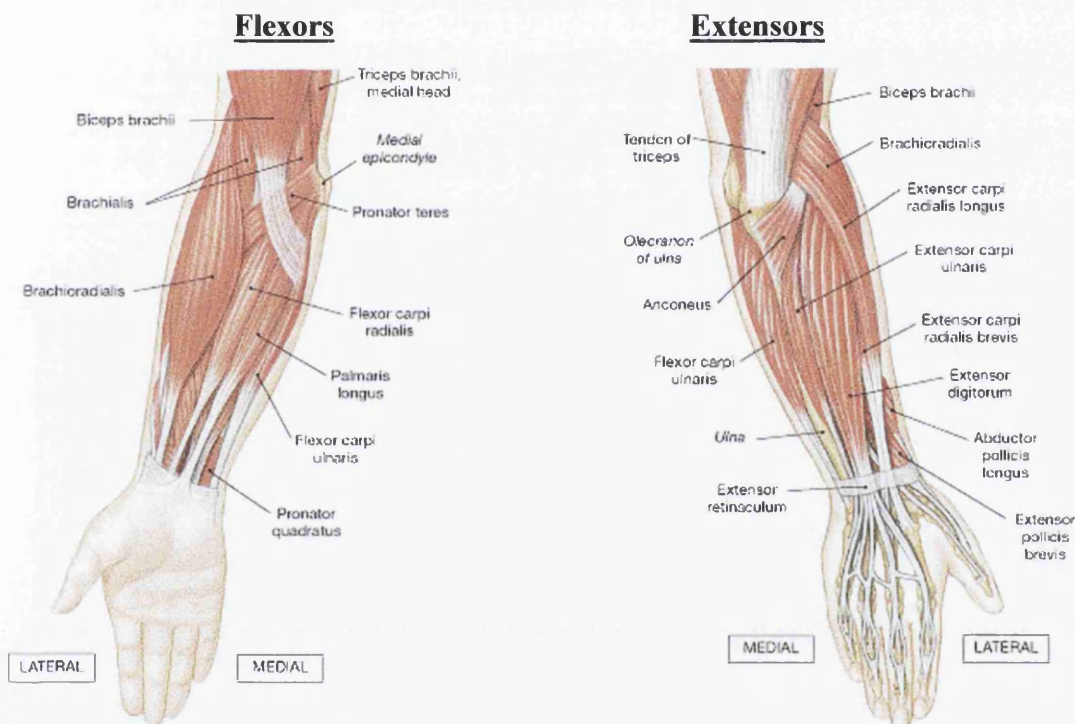


Figure 2.27 – Muscles of Forearm (Flexors/Extensors) [5]

2.4.5 Muscles of the Lower Extremity

The muscles that move the thigh have their origins (attachment to stationary bone) on some part of the pelvic girdle and their insertions (attachment to moveable bone) on the femur. The largest muscle mass belongs to the posterior group, the gluteal muscles (Figure 2.28), which, as a group, abduct the thigh. The iliopsoas (Figure 2.25), an anterior muscle, flexes the thigh. The muscles in the medial compartment adduct the thigh.

Muscles that move the leg are located in the thigh region. The quadriceps femoris muscle group (rectus femoris, vastus lateralis, vastus medialis) straightens the leg at the knee. The hamstrings (biceps femoris,

semitendinosus, semimembranosus) are antagonists (cause action opposite to the desired action) to the quadriceps femoris muscle group, which are used to flex the leg at the knee.

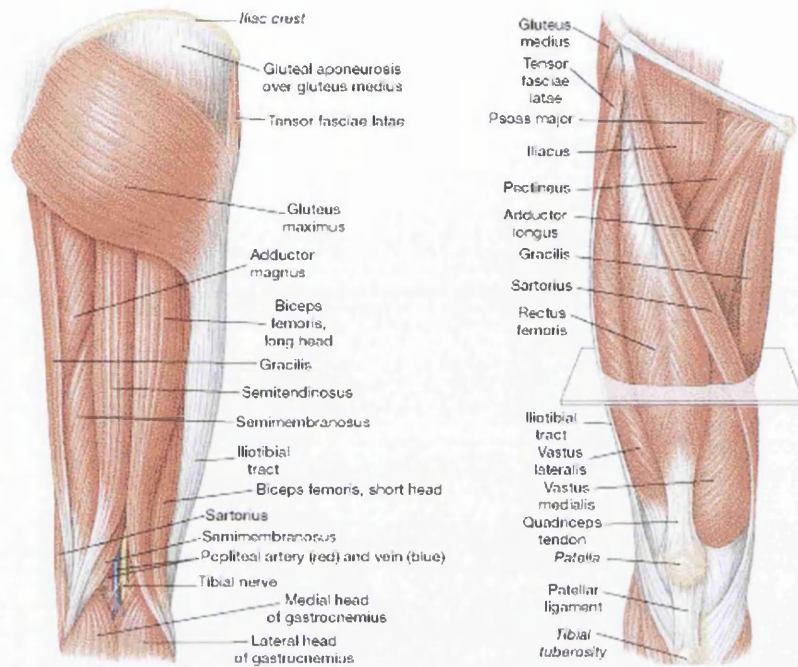


Figure 2.28 – Muscles of Thigh (Posterior/Anterior) [5]

The muscles located in the leg that move the ankle and foot are divided into anterior, posterior, and lateral compartments. The tibialis anterior, which dorsiflexes the foot, is antagonistic to the gastrocnemius and soleus muscles (Figure 2.29), which plantar flex the foot⁶.

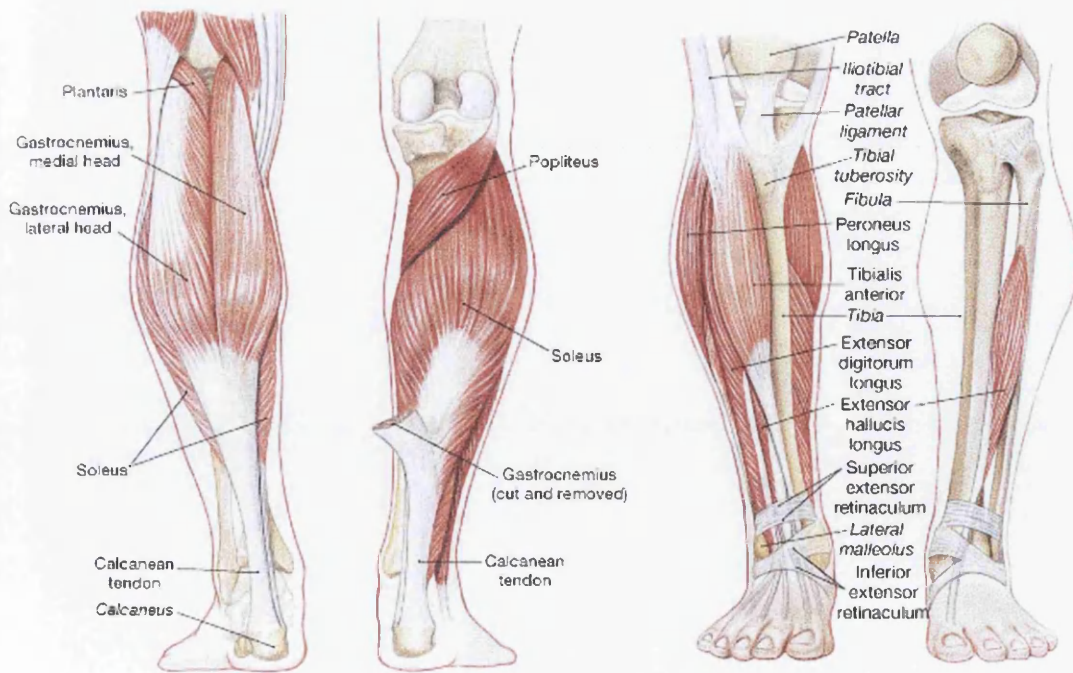


Figure 2.29 – Muscles of Leg (Posterior/Anterior) [5]

2.5 Conclusions

This chapter intends to be a well structured and concise introduction for someone who is interested in musculoskeletal modelling and has no background in the physiology of the human body. Combined with the proper illustrations, this chapter can be a great help for the design and interpretation of the skeletal, muscular and articular system and explain how these systems are related to each other.

CHAPTER 3

COMPUTER MODELLING & ANTHROPOMETRY

3.1 Introduction

There is no doubt, nowadays, that computer modelling is an integral part of the research action of scientists all over the world. Human movement is one of the sciences that widely uses simulation techniques.

This chapter begins with the reasons that prompt researchers to develop computer models and the limitations that go with these models. It continues with the introduction of the research method that is widely followed for the investigation of the human movement and how this method relates computer modelling and the science of anthropometry. In particular it deals with the subfield of anthropometry called body segment parameters, and presents background material on measuring and estimating these parameters.

3.2 Why Simulate Human Movement?

The best way to obtain human movement data, it seems, would be to collect it from human subjects. In other words, no data can be more accurate, more real than those collected from human subjects, so why would scientists want to generate data artificially?

The most general answer is that human subjects are extremely complex and have intrinsic limitations. They are randomly variable, subject to fatigue, have limits of strength and coordination, and must be treated with due consideration for safety and ethics.

Computer models also allow researchers to experiment with conditions that cannot be tested on human subjects because they exceed practical limits. For example, locomotion patterns under an increased and decreased gravitational constant ($9.81m/s^2$) could be tested to study the effects of gravity on human movement. The mass of model limbs could be changed to study the effect of limb mass on movement^{8,9}. A single parameter in a model can also be changed and immediately tested. For example, a researcher who wants to study how increasing elbow extensor strength might improve throwing performance can change the model thrower and test it immediately, whereas a human subject would have to undergo a training regimen in which strength might improve in other muscles as well. Another example is to test the effects of torso mass on ground reaction forces. This, too, is a matter of changing a single model parameter. However, if this study was attempted in human subjects, the results might be confounded by concomitant changes in walking velocity, stride length, and perhaps even footwear function.

Another important and commonly used feature of computer simulations is that they can be adapted to search for optimal solutions. This is very useful for athletic performance, clinical applications, and other large issues in human movement. For example, scientists could determine the optimal movement pattern for a football kick, the most economical walking cadence for individuals with different body types, and the optimum flexibility of a prosthetic foot for an amputee's body weight, age, and stride.

Models are often used simply to test biomechanist's understanding of movement. This is a key reason for the popularity of rigid-body models in biomechanics despite the fact that none of them is perfect. If biomechanists understood a movement perfectly, they could replicate it on a computer. However, models eventually fall short in their predictive capabilities, and they are left to assess why. The search for answers often develops into a new movement theory.

3.3 Limitations of Computer Models

No matter how sophisticated a computer model becomes, many limitations remain, and they arise from a number of factors:

- **The numerical imperfections that exist in the solution process.**

Numerical errors are often slight and can often be resolved. Nonetheless, models become increasingly sensitive as the number of components increases. In addition, small errors that arise early in a simulated movement can propagate and grow to measurable levels by the end of the simulation. Also, models can have parameters that differ by many orders of magnitude, a situation referred to as stiff system of equations. This can lead to a sensitive model that generates inconsistent results.

- **The difficulty of accurately modelling impacts.**

Impacts are very difficult to model. One reason for this is that they incur sharp changes in the accelerations of the body segments, which leads to numerical errors. Also, impacts generally violate the assumption that the human body consists of rigid segments. During walking, for example, the feet deform, storing and dissipating mechanical energy in a manner that is extremely difficult to model. When the human body runs or performs a drop landing, the impacts make computer simulation very difficult, indeed.

- **The approximation of complex human structures.**

Human structures themselves are much more complicated than our models of them. For example, most simulations use simple representations of joints and assume that segments are rigid; for most purposes, these work well. However, joints are not simple pivots, muscles are not simple torque actuators, and segments are not rigid. Thus, researchers must be alert to the effects

of such simplifications. For example, if a discrepancy was observed in the ankle moment data of a running model, a researcher might first question whether the problem stemmed from the assumption that the foot segment was rigid. A second question might be whether the model ankle reasonably approximated the orientation of the ankle joint.

- **The enormous adaptability of the human system.**

When modeling, we must remember that we are simulating a human system in which the sensory and control systems and their organization are extraordinary complex. A complete model of the moving human-even if we understood the entire human system to begin with-would be too much for the most sophisticated computers to handle. Therefore, every model of the human system is an oversimplification. This is an accepted part of the science. For example, virtually all walking simulations have been performed on level ground and researchers have difficulty of completely modelling the movement in even this simple situation.

In addition, humans can adapt to myriad surfaces, turns, steps, bumps, and other conditions while walking, and each of these would require a different-and likely complex-adjustment to the model. Models also neglect vast array of senses that influence the movements with their messages. Even if these concerns will be put aside, the issue of adaptability under simple test conditions is also pertinent. In short, replicating a human movement is generally straightforward, but predicting how that movement will change under different test conditions is difficult. Suppose, for example, that there is a highly accurate simulation of normal walking. What will happen when the torso mass increases, as it would if we had to account for a backpack? Will the stride lengthen or shorten? Will the torso angle change? Will the load be too heavy altogether?

3.4 Research Method

The primary technique that is widely used for the investigation of human movement is dynamic modelling. This is a three-stage process, firstly an input model is being constructed, then this model is run through a simulator and finally the results are analysed.

1. The Input Model

This requires two components: inertial properties for the construction of a link-segment model, and kinematics. Inertial properties are the basic physical characteristics of the body segments which are the mass, center of mass and moment of inertia. Kinematics are the motion characteristics of the body. They include linear and angular displacements, velocities and accelerations.

2. Simulation

The constructed input model is run through a dynamic analysis package. For rigid segment models this can be done using a package such as Anybody Technology.

The technique that is used to perform the simulation is the so called “inverse dynamics solution”. Inverse dynamics is the specialised branch of mechanics that bridges the areas of kinematics and kinetics. It is the process by which variables such as joint reaction forces and moments of force are computed from a combination of measured external forces, segment kinematics and segment inertial characteristics.

3. Analysis

Once a model is simulated, it is ready to make predictions about the human movement to answer a particular research question. Researchers again have to investigate if the data the model generates are reasonable, and perhaps compare them with experimental data. If the model data are not realistic, it must be determined if there is a problem with the model itself or with the way simulation runs. An entire series of studies may have to be performed to establish the model’s validity.

3.5 Anthropometry

3.5.1 Introduction

As it was mentioned at the previous section, the first stage for the investigation of human movement is the determination of the physical characteristics of the body segments. **Anthropometry** is the discipline concerned with the measurement of these physical characteristics of humans.

Biomechanists are mainly interested in the inertial properties of the body and its segments, a subfield of anthropometry called body segment parameters. Before a kinetic analysis of human movement is possible, each segment’s physical characteristics and inertial properties must be determined. The relevant characteristics are the segmental mass, locations of the segmental centers of gravity, and segmental mass moments of inertia.

A major concern for biomechanists is the assumption that body segments behave as rigid bodies during movement. This assumption obviously is not valid because bones bend, blood flows, ligaments stretch, and muscles contract. It is also common to model some body parts as single rigid bodies despite the fact that they consist of several segments. For instance, the foot is commonly considered to be one segment even though it clearly can bend at the metatarsal-phalangeal joints. Similarly, the trunk is often treated as one single rigid body, but sometimes as two or three. In reality, however, it is a series of interconnected rigid bodies including the many interconnected vertebrae, pelvis, and scapulae. These

assumptions simplify the otherwise complex musculoskeletal system and eliminate the necessity of quantifying the changes in mass distribution caused by tissue deformation and movement of bodily fluids. By also assuming that segmental mass distributions is similar among members of a particular population, an individual's segmental parameters can be estimated by applying equations based on the averages obtained from samples taken from the population.

Attempts to quantify body segment parameters generally fall into four categories: cadaver studies, mathematical modelling, scanning and imaging techniques, and kinematic measurements. Some of the literature that utilizes these categories are presented below in the following subsections. **It has to be mentioned that all of the work presented in anthropometry, is based on the literature of Robertson¹⁰ (2004) and the references belong to his work.**

3.5.2 Cadaver Studies

Inertial properties (mass, centre of mass, moment of inertia) are difficult to be determined for a particular living person. If scientists were to quantify these properties for a robot, each segment would be separated and analyzed individually by performing specific tests. Because this is not possible for living persons, indirect methods must be used. For example, the coefficient method uses tables of proportions that predict the body segment parameters from simple, noninvasive measures such as total body mass, height, and segment lengths. The earliest attempts at this procedure date to the works of Harles¹¹ (1860), Braune and Fischer¹² (1889), and Fischer¹³ (1906), but the most significant advancement was the work done by W.T. Dempster¹⁴, published in 1955 as the monograph *Space Requirements of the Seated Operator*. This document, which Dempster compiled while working for U.S. Air Force, not only outlined the procedures for measuring body segment parameters from cadaveric materials, but also included tables for proportionally determining the body segment parameters needed to biomechanically analyze human motion.

Dempster collected data from living persons, anatomical specimens, and most importantly (for biomechanists), from eight complete cadavers. First, he used the method of Reuleaux¹⁵ (1876) to determine the average center of rotation at each joint (Figure 3.1). As shown in Figure 3.1 Points A and B are fixed on the rigid body. After movement from Position 1 to position 2, dotted lines are drawn as right bisectors of the lines A_1-A_2 and B_1-B_2 . The center of rotation is at the intersection of the dotted lines. At some joints, particularly the shoulder, the center of rotation was difficult to identify. These locations became the endpoints for the various body segments. Table 3.1 describes these endpoints.

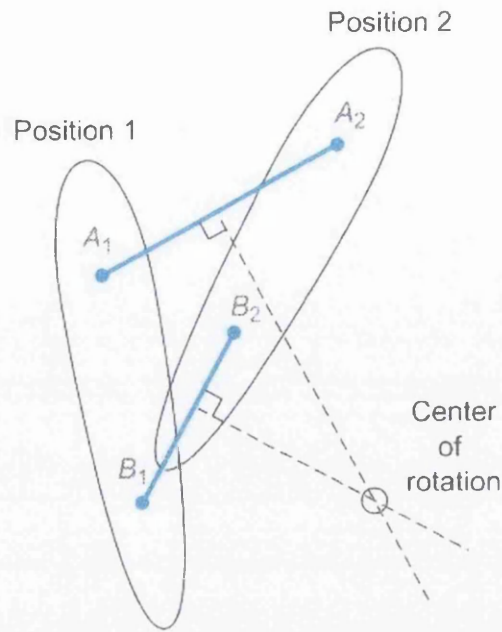


Figure 3.1 – Reuleaux's Method [10]

The cadavers were then segmented according to Dempster's own techniques and their lengths, masses, and volumes were carefully recorded. Dempster then calculated the location of the center of gravity (using a balancing technique) and the moment of inertia (using a pendulum technique) for each segment. Finally, Dempster created tables showing the segmental masses as proportions of the total body mass and the locations of the centers of gravity and lengths of the radii of gyration as proportions of the segment's lengths. These data in modified form (Miller and Nelson¹⁶ 1973; Plagenhoef 1971¹⁷; Winter¹⁸ 2005) appear in Table 3.2. Later Barter¹⁹ (1957), working with Dempster's data, performed stepwise regression analysis to derive regression equations that more accurately compute segmental masses.

Segment	Proximal End	Distal End
Clavicular link	Sternoclavicular joint center: midpoint of palpable junction between proximal end of clavicle and upper border of sternum	Claviscapular joint center: midpoint of line between coracoid tuberosity of clavicle and acromioclavicular articulation at lateral end of clavicle
Scapular link	Claviscapular joint center: see Clavicular link, Distal end	Glenohumeral joint center: midpoint of palpable bony mass of head and tuberosities of humerus
Humeral link	Glenohumeral joint center: see Scapular link, Distal end	Elbow joint center: midpoint between lowest palpable point of medial epicondyle of humerus and a point 8mm above the radiale (radiohumeral junction)

Radial link	Elbow joint center: see Humeral link, Distal end	Wrist joint center: distal wrist crease at palmaris longus tendon or palpable groove between lunate and capitate bone in line with metacarpal bone III
Hand link	Wrist joint center: see Radial link, Distal end	Center of mass of hand: midpoint between proximal transverse palmar crease and radial longitudinal crease in line with third digit
Femoral link	Hip joint center: tip of femoral trochanter 1 cm anterior to most laterally projecting part of greater trochanter	Knee joint center: midpoint between centers of posterior convexities of femoral condyles
Leg link	Knee joint center: see Femoral link, Distal end	Ankle joint center: level of a line between tip of lateral malleolus of fibula and a point 5mm distal of the tibial malleolus
Foot link	Ankle joint center: see Leg link, Distal end	Center of gravity of foot: midway between ankle joint center and ball of foot at head of metatarsal II

Table 3.1 - Dempster's Segment Endpoints Definitions [10]

Many other body segment parameter studies have been conducted since Dempster's groundbreaking work. Two in particular, by Clauser, McConville, and Young²⁰ (1969) and Chandler²¹ et al. (1975), are note-worthy because they defined body segments using palpable bony landmarks instead of estimated, averaged joint centers of rotation. They also segmented an additional 13 cadavers (only six were used by Chandler et al.) and presented the results in a table similar to Dempster's. The segment definitions appear in Table 3.3 and the values are presented in Table 3.4.

Segment	Endpoints (proximal to distal)	Segmental mass/total mass (P)	Center of mass/ Segment length		Radius of gyration/ Segment length			Density(gr/cm^3)
			(R_{proximal})	(R_{distal})	(K_{cg})	(K_{proximal})	(K_{distal})	
Hand	Wrist center to knuckle II of third finger	0.006	0.506	0.494	0.298	0.587	0.577	1.16
Forearm	Elbow to wrist center	0.016	0.430	0.570	0.303	0.526	0.647	1.13
Upper arm	Glenohumeral joint to elbow center	0.028	0.436	0.564	0.322	0.542	0.645	1.07
Forearm and hand	Elbow to wrist center	0.022	0.682	0.318	0.468	0.827	0.565	1.14
Upper extremity	Glenohumeral joint to wrist center	0.050	0.530	0.470	0.368	0.645	0.596	1.11
Foot	Ankle to ball of foot	0.0145	0.500	0.500	0.475	0.690	0.690	1.10
Leg	Knee to ankle center	0.0465	0.433	0.567	0.302	0.528	0.643	1.09
Thigh	Hip to knee center	0.100	0.433	0.567	0.323	0.540	0.653	1.05
Lower extremity	Hip to ankle center	0.161	0.447	0.553	0.326	0.560	0.650	1.06
Head	C7-T1 to ear canal	0.081	1.000	0.000	0.495	1.116	0.495	1.11
Shoulder	Sternoclavicular joint to glenohumeral joint center	0.0158	0.712	0.288	-	-	-	1.04
Thorax	C7-T1 to T12-L1	0.216	0.820	0.180	-	-	-	0.92
Abdomen	T12-L1 to L4-L5	0.139	0.440	0.560	-	-	-	-
Pelvis	L4-L5 to trochanter	0.142	0.105	0.895	-	-	-	-
Thorax and abdomen	C7-T1 to L4-L5	0.355	0.630	0.370	-	-	-	-
Abdomen and pelvis	T12-L1 to greater trochanter	0.281	0.270	0.730	-	-	-	1.01
Trunk	Greater trochanter to glenohumeral joint	0.497	0.495	0.505	0.406	0.640	0.648	1.03
Head, arms and trunk	Greater trochanter to glenohumeral joint	0.678	0.626	0.374	0.496	0.798	0.621	-
Head, arms and trunk	Greater trochanter to mid-rib	0.678	1.142	-0.142	0.903	1.456	0.914	-

Table 3.2 - Dempster's Body Segment Parameters [10]

Acromion	Point at the superior and external border of the acromion process of the scapula. This point is easier to find if the subject bends laterally at the trunk to relax the deltoid muscle.
Glabella	Bony ridge under the eyebrows.
Metacarpale III	Distal head of the third metacarpal (proximal knuckle of the middle finger).
Occiput	Occipital process.
Radiale	Point at the proximal and lateral border of the head of the radius. Palpate downward in the lateral dimple at the elbow. Have the subject pronate and supinate the forearm slowly so that the radius may be felt rotating beneath the skin.
Sphyrion	Most distal point of the medial malleolus of the tibia. Place the marker on the fibula at the level of the sphyrion. Do not use the sphyrion fibulare, as it is more distal than the sphyrion.
Stylian	Styloid process of the radius. The styloid process of the ulna may be used.
Tibiale	Most proximal point of the medial border of the head of the tibia. Use the tibiale externum, which is equivalent. Palpate the tendon of the quadriceps at the distal end of the patella and follow the border of the tibia laterally to the frontal border of the lateral collateral ligament. The point is easier to find if the subject unloads the leg and slightly flexes the knee.
Trochanter	Most superior point on the greater trochanter of the femur, which is difficult to locate in women and people with excessive subcutaneous fat. Have the subject abduct and move the leg forward and backward. Palpate the femur with the index finger, and move upward gradually until the most superior palpable point on the trochanter is found.

Table 3.3 - Clauser et al.'s Segment Endpoints Definition [10]

For their 3-D kinematic and kinetic investigation of the lower extremity during human gait, Vaughan, Davis, and O'Connor²² (1992) enhanced the methods used by Chandler et al. (1975). They created regression equations for the masses of the lower extremities that included various segmental anthropometrics, such as calf and midhigh circumference, in addition to segment length and body mass (Table 3.5). Hinrichs²³ (1985) also extended Chandler's work by applying regression equations for the calculation of segmental moments of inertia.

An average set of segment lengths expressed as a percentage of body height was prepared by Drillis and Contini²⁴ (1966) and is shown in Figure 3.2. These segment proportions serve as a good approximation in the absence of better data, preferably measured directly from the individual.

Segment	Endpoints (proximal to distal)	Segmental mass/ total mass (P)	Center of mass/ segment length		Radius of gyration/ Segment length	
			(R _{proximal})	(R _{distal})	(K _{cg}) [*]	(K _{proximal})
Hand	Styloid to metacarpale III	0.0065	0.1802	0.8198	0.6019	0.6283
Forearm	Radiale to styloid	0.0161	0.3896	0.6104	0.3182	0.5030
Upper arm	Acromion to radiale	0.0263	0.5130	0.4870	0.3012	0.5949
Forearm and hand	Radiale to styloid	0.0227	0.6258	0.3742		
Upper extremity	Regression equation	0.0490	0.4126	0.5874		
Foot	Heel to toe II	0.0147	0.4485	0.5515	0.4265	0.6189
Foot	Sphyrion to floor	0.0147	0.4622	0.5378		
Leg	Tibiale to sphyrion	0.0435	0.3705	0.6295	0.3567	0.5143
Thigh	Trochanter to tibiale	0.1027	0.3719	0.6281	0.3475	0.5090
Leg and foot	Tibiale to floor	0.0582	0.4747	0.5253		
Lower extremity	Trochanter to floor (sole)	0.1610	0.3821	0.6179		
Trunk	Chin-neck intersect. to trochanter	0.5070	0.3803	0.6197	0.4297	0.5738
Head	Top of head to chin-neck intersect.	0.0728	0.4642	0.5358	0.6330	0.7850
Trunk and head	Chin-neck intersect. To trochanter	0.5801	0.5921	0.4079		
Total body		1.0000	0.4119	0.5881	0.7430	0.8495

$$* K_{cg} = \sqrt{K_{proximal}^2 - R_{proximal}^2}$$

Table 3.4 - Clauser, McConville, and Young's Body Segment Parameters [10]

Segment	Axis	Regression equation
Foot	Flexion/extension	$I_{f/e} = 0.00023 \cdot m_{total} \cdot [4 \cdot h_m^2 + 3 \cdot l_{foot}^2] + 0.00022$ $h_m = \text{malleolus height}(m)$
	Abduction/adduction	$I_{a/a} = 0.00021 \cdot m_{total} \cdot [4 \cdot h_m^2 + 3 \cdot l_{foot}^2] + 0.00067$
	Internal/external rotation	$I_{i/e} = 0.00041 \cdot m_{total} \cdot [h_m^2 + w_{foot}^2] - 0.00008$ $w_{foot} = \text{foot width}(m)$
Leg	Flexion/extension	$I_{f/e} = 0.00347 \cdot m_{total} \cdot [l_{leg}^2 + 0.076 \cdot c_{leg}^2] + 0.00511$ $c_{leg} = \text{leg circumference}(m)$
	Abduction/adduction	$I_{a/a} = 0.00387 \cdot m_{total} \cdot [l_{leg}^2 + 0.076 \cdot c_{leg}^2] + 0.00138$
	Internal/external rotation	$I_{i/e} = 0.00041 \cdot m_{total} \cdot c_{leg}^2 + 0.00012$
Thigh	Flexion/extension	$I_{f/e} = 0.00762 \cdot m_{total} \cdot [l_{thigh}^2 + 0.076 \cdot c_{thigh}^2] + 0.01153$ $c_{thigh} = \text{thigh circumference}(m)$
	Abduction/adduction	$I_{a/a} = 0.00762 \cdot m_{total} \cdot [l_{thigh}^2 + 0.076 \cdot c_{thigh}^2] + 0.01186$
	Internal/external rotation	$I_{i/e} = 0.0051 \cdot m_{total} \cdot c_{thigh}^2 + 0.00305$

Table 3.5 – Vaughan, Davis, and O'Connor's Body Segment Parameters [10]

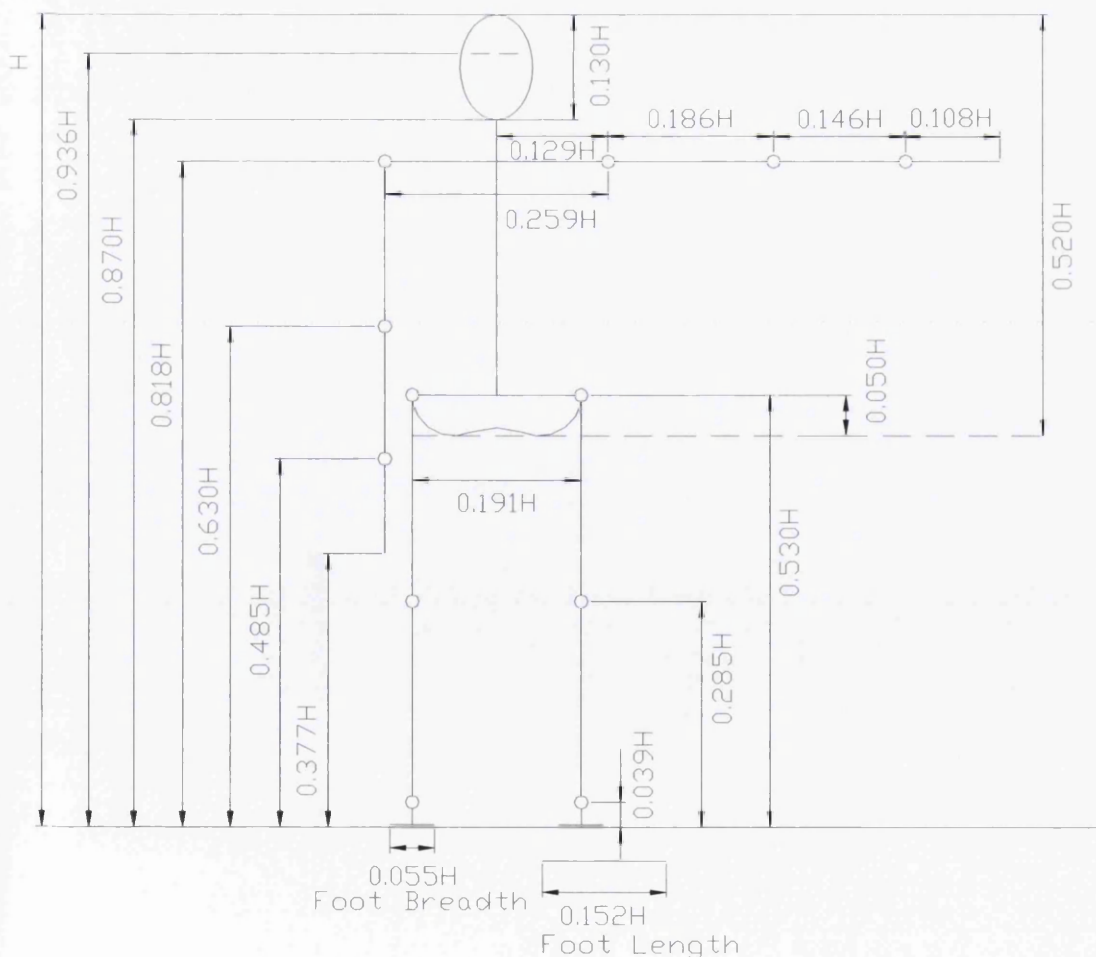


Figure 3.2 – Body Segment Lengths Expressed as a Fraction of Body Height H [18]

3.5.3 Mathematical Modelling

Mathematical modelling of the inertial properties of human body segments was pioneered by the work of Hanavan²⁵ (1964) when it became necessary to model the body for 3-D analyses. Hanavan made the assumption that mass was uniformly distributed within each segment, and that the segments were rigid bodies that could be presented by geometrical shapes. The body was divided in fifteen segments (Figure 3.3) and most of the segments were modeled as frusta of right-circular cones. The hands were modeled as spheres, the head as an ellipsoid, and the trunk segments as elliptical cylinders. Figure 3.4 contains equations used for the computation of the principal mass moments of inertia of the main uniform solids Hanavan used. By using the data provided by Dempster (1955) and taking additional anthropometric measures (25 in total), such as midthigh circumference, malleolus height, knee diameter, and biacromial breadth, equations were developed to compute the three principal moments of inertia. Hanavan's methods have since been enhanced to include more segments and more directly measured anthropometric measures.

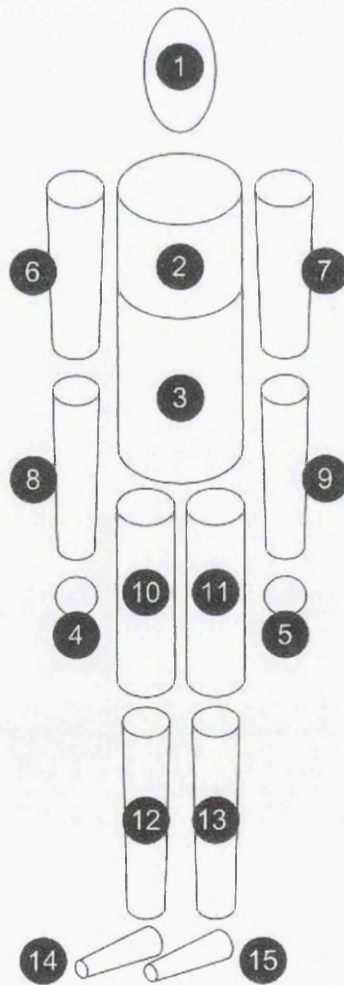
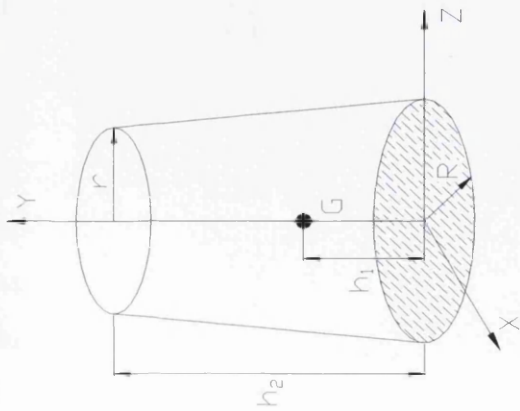


Figure 3.3 – Hanavan's Model [10]

Truncated Circular Cone



Centre of mass (G)

$$h_1 = \frac{h_2 \cdot (R^2 + 2 \cdot R \cdot r + 3 \cdot r^2)}{4 \cdot (R^2 + R \cdot r + r^2)}$$

Volume

$$V = \frac{\pi \cdot h_2}{3} \cdot (R^2 + R \cdot r + r^2)$$

Mass

$m = d \cdot V$ where d is density

Moment of Inertia (G)

$$I_x = \frac{m}{20} \left[3 \cdot (R^2 + r^2) + \frac{(2 \cdot R^2 + 6 \cdot R \cdot r + 12 \cdot r^2) \cdot h_2^2 - 3 \cdot r^2 \cdot R^2}{R^2 + R \cdot r + r^2} \right]$$

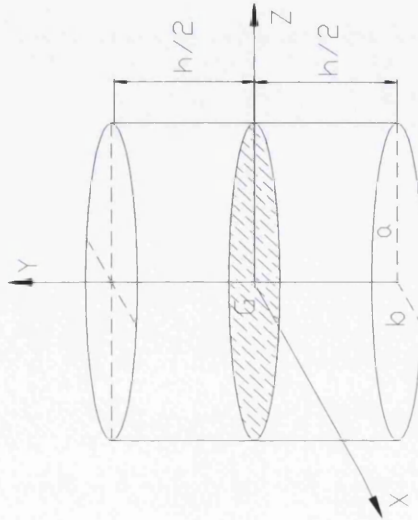
$$\Rightarrow I_{x_G} = \frac{m}{20} \left[3 \cdot (R^2 + r^2) + \frac{(2 \cdot R^2 + 6 \cdot R \cdot r + 12 \cdot r^2) \cdot h_2^2 - 3 \cdot r^2 \cdot R^2}{R^2 + R \cdot r + r^2} \right] - m \cdot h_1^2$$

$I_x = I_{x_G} + m \cdot h_1^2 \Rightarrow I_{x_G} = I_x - m \cdot h_1^2$

$I_{z_G} = I_{x_G}$

$I_{y_G} = \frac{3 \cdot m}{10} \cdot \frac{(R^5 - r^5)}{(R^3 - r^3)}$

Elliptical Cylinder



Centre of mass (G)

$G = \frac{h}{2}$

Volume

$V = \pi \cdot a \cdot b \cdot h$

Mass

$m = d \cdot V$ where d is density

Moment of Inertia (G)

$I_{x_G} = \frac{1}{4} \cdot m \cdot a^2 + \frac{1}{12} \cdot m \cdot h^2$

$I_{y_G} = \frac{1}{4} \cdot m \cdot (a^2 + b^2)$

$I_{z_G} = \frac{1}{4} \cdot m \cdot b^2 + \frac{1}{12} \cdot m \cdot h^2$

Figure 3.4 – Inertial Properties of Hanavan's Model Segment

Hatze²⁶ (1980) developed another model that uses 242 direct anthropometric measurements to define the segmental properties of a 42-DOF, 17 segment model of the body (Figure 3.5).

A model presented by Yeadon²⁷ (1990) divides the body into 40 solids. A total of 95 measurements are necessary. The solids representing the torso, hands, and feet are stadium solids (Figure 3.5).

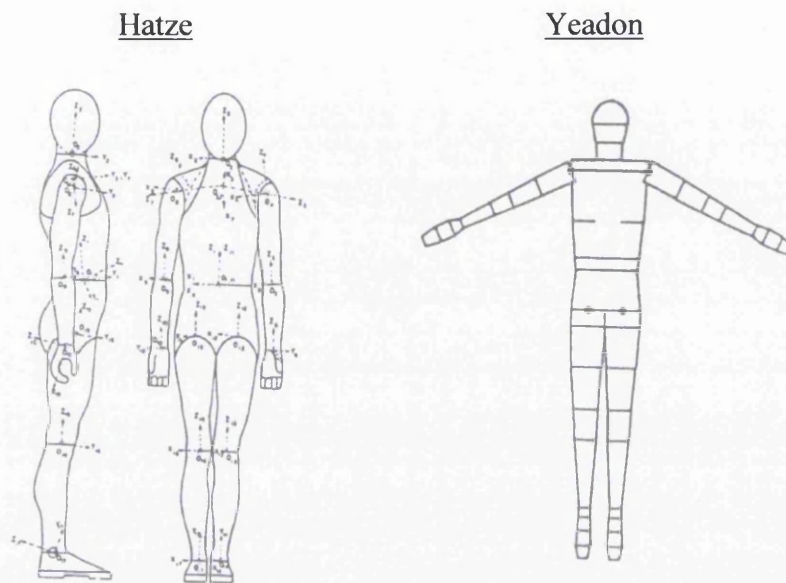


Figure 3.5 – Hatze’s & Yeadon’s Models [30]

3.5.4 Scanning And Imaging Techniques

Another approach to determining inertial properties and then estimating body segment parameters involves scanning the living body with various radiation techniques. For instance, Zatsiorsky and Seluyanov^{28, 29} (1983, 1985) presented data from an extensive study on the body segment parameters of both male (n=100) and female (n=15) living subjects. To compute mass distribution, gamma mass scanning was used to quantify the density of incremental slices of each segment. This method enabled estimations to be made of the mass, center of mass, and principal moments of inertia in 3-D for a total of 15 segments. Their subjects included individuals younger than the ages of the cadavers used in previous studies; furthermore, they applied regression equations to customize body segment parameters. These values and methods were recently documented and republished by Zatsiorsky³⁰ (2002).

In addition to gamma mass scanning, other techniques that have been developed to quantify human body segment parameters include photogrammetry (Jensen^{31, 32} 1976, 1978), magnetic resonance imaging (Mungiole and Martin³³ 1990), and dual-energy X-ray absorptiometry (DEXA; Durkin and Dowling³⁴ 2003; Durkin, Dowling, and Andrews³⁵ 2002).

Furthermore, some researchers have provided data from populations that are not well represented by the previously mentioned studies, which were based primarily on middle-aged male or young adult Caucasians. For example, Jensen^{36, 37, 38} and colleagues provided data for children (1986, 1989) and

pregnant women (1996), Schneider and Zernicke³⁹ (1992) quantified regression equations for the body segment parameters of infants, and Pavol, Owings, and Grabiner⁴⁰ (2002) estimated inertial properties for the elderly.

3.5.5 Kinematic Techniques

Kinematic techniques are methods that measure kinematic characteristics to indirectly determine the inertial properties of segments. Hatze⁴¹ (1975) developed an oscillation technique that derives the mass, center of mass, and moment of inertia of the segments of the extremities and the damping coefficients of joints. The technique, which cannot be used for the trunk segments, requires that a body part be set into oscillation with an instrumented spring. The muscles must be relaxed so that they do not influence the damped oscillations of the spring-limb system. Mathematically derived equations based on small oscillation theory are then used to estimate the properties of the joint and segment based on a passively damped reduction of the oscillations of the spring-limb system.

Another technique for estimating mass moment of inertia is called the quick release method (Drillis, Contini, and Bluestein⁴² 1964). This technique also assumes that the muscles are relaxed and that the acceleration of a rapidly accelerated segment is affected only by the segment's rotational inertia. Thus, by measuring the angular acceleration of a segment after release of a known force (F) or moment (M), the moment of inertia (I) of the segment can be derived from the following relationship:

$$I = M / a = (F \cdot d) / a \quad (3.1)$$

where M (or $F \cdot d$) is the moment of force immediately before release and a is the angular acceleration after release. Obviously, this method can be used only for terminal segments because applying it to other body parts would be difficult.

3.6 Summary

Some of the main reasons that researchers use more and more computer models, are due to the extreme complexity of the human body, the need of prediction, and the intrinsic limitations of humans as they are randomly variable, subject to fatigue, with limits of strength and coordination, and that they must be treated with due consideration for safety and ethics. Nevertheless, no matter how sophisticated a computer model becomes, many limitations remain, and they arise from a number of factors like the numerical imperfections that exist in the solution process, the difficulty of accurately modelling impacts, the approximation of complex human structures and the enormous adaptability of the human system. Of course, for the implementation of a simulation analysis of human movement, each segment's physical characteristics and inertial properties must be determined first. The relevant characteristics are

the segmental mass, locations of the segmental centers of gravity, and segmental mass moments of inertia. Attempts to quantify these characteristics generally fall into four categories. One of these categories is the study that uses cadavers. The origin of this method is due to the difficulty to determine the inertial properties of a particular living person, as it is impossible to separate and analyse each body segment individually. Another category is the one of mathematical modelling, in which segments can be represented by known geometrical shapes and in combination with some anthropometric measures can make the computation of the three principal moments of inertia possible. A third category is the one of scanning and imaging techniques like the gamma mass scanning technique, the photogrammetry, the magnetic resonance imaging and the dual-energy X-ray technique. And a fourth category is the one with the kinematic techniques such the *oscillation technique* that derives the inertial characteristics of all the body segments except those of the trunk and determines the damping coefficients of joints. Another technique that falls into the fourth category is the *quick-release method* which estimates mass moment of inertia.

Chapter 4 Page 57

CHAPTER 4 MUSCULOSKELETAL SOFTWARE SYSTEMS & APPLICATION AREAS

4.1 Introduction

The present chapter deals with the presentation of the basic features and some application areas of the most popular musculoskeletal software systems in the commerce.

The first part is dedicated to the basic characteristics of these simulation tools, starting with the *Anybody Technology*⁴³ system, it continues with the *SIMM*⁴⁴ software suite, the *LifeMOD*⁴⁵ software system, the *SimMechanics*⁴⁷ system and ends with the *MADYMO*⁴⁸ software system. The second part includes the application areas of the above musculoskeletal software systems and some representative examples for each of these areas. These areas are: *automotive, medical-rehabilitation, aerospace, occupational health, and sports.*

4.2 Musculoskeletal Software Systems

4.2.1 Anybody Technology⁴³

The Anybody Modelling System is a software system for the simulation of human movement (Figure 4.1). It can model smaller or larger subsets of the musculoskeletal system (or the entire body) and compute muscle forces, joint reactions, metabolism, mechanical work, etc. for given movements.



Figure 4.1 – The Anybody Modelling System [43]

Any property of the Anybody model is parametric, and the system can be used for optimization of movement patterns, working positions, anthropometric data, etc.

Technically, the system is characterized by the following properties:

- It uses an optimization technique to solve the muscle recruitment problem and reverses this by means of other optimization techniques so that forward dynamics problems can also be treated.
- It handles static and dynamic models.
- Models are fully three-dimensional.
- It runs on the MS Windows platform.
- It handles very large models on small computers. It is entirely feasible to analyze a model with several hundreds of muscles on an ordinary PC.
- Models are developed in the body modelling language AnyScript.
- The Anybody Modelling System is self-contained. It does not build on top of other platform or require any additional software to be installed.

Anybody Modelling System is the software that is used in this research thus a more detailed presentation of this system is included below.

4.2.2 SIMM Software Suite⁴⁴

4.2.2.1 SIMM

SIMM (Software for Interactive Musculoskeletal Modelling) is a graphics-based software package that enables the user to quickly develop and analyze musculoskeletal models (Figure 4.2). In SIMM, a musculoskeletal model consists of representations of bones, muscles, ligaments, and other structures.

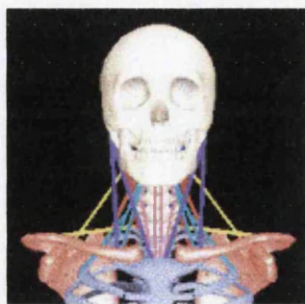


Figure 4.2 – The SIMM Modelling System [44]

Muscles span the joints and develop force, thus generating moments about the joints. SIMM enables an analysis of a musculoskeletal model by calculating the joint moments that each muscle can generate at any body position. By manipulating a model using the graphical interface, the user can quickly explore

the effects of changing musculoskeletal geometry and other model parameters on the muscle forces and joint moments. It has the following features:

- **Muscle Wrapping.** Users can interactively define spheres, ellipsoids, and cylinders for muscle-tendon actuators to wrap over. SIMM automatically calculates muscle paths over these wrapping objects. Muscle lengths, forces, and moment arms are all calculated correctly for the new, wrapped muscle.
- **Bone Deformations.** A powerful tool allows user to warp bones into new shapes to model various bony deformities. Deformations such as tibial torsion and femoral anteversion are straightforward to model and can be implemented with a range of severity of deformation.
- **Model Scaling.** A scaling utility is able to scale an existing model to match any size individual. Body segments (and corresponding joints kinematics) can be individually scaled by specifying X, Y, Z scale factors for each segment. Muscle paths are scaled with the segments, but muscle force-generating parameters are not scaled.
- **TIFF/movie Export.** Users can export a single image of a model, or an entire animation sequence to TIFF images that can then be combined into a movie file.
- **Muscle Display.** Muscles are drawn as smooth-shaded cylinders, making them look more realistic, especially in close-up views.
- **Muscle Point Editing.** A selection box in the model window helps users identify muscle points for selecting. There is also a feature for picking a polygon on a bone, which is helpful for cases in which users want to attach the muscle point to the bone surface.
- **Motion Objects.** SIMM has a class of objects that users can use to improve the animation of musculoskeletal models. Motion objects are polyhedra for which users can specify the position, orientation, size, and colour in each frame of data in a motion file.
- **Norm Options.** The utility program *norm* has several options for processing polyhedron files. It can fill in holes in polyhedra, subdivide edges, and has options that improve its ability to find correct vertex normals for polyhedra that are not closed objects.
- **GUI Tools.** Many user interface elements make it easy to interact with a model and to change the display properties of the bones, muscles, and other model components.

4.2.2.2 Dynamics Pipeline

Dynamics Pipeline connects SIMM to SD/FAST (Symbolic Dynamics, Mountain View, CA) to simulate the dynamics of rigid-body systems. The Pipeline is a general-purpose software package that enables users to perform forward and inverse dynamics simulations on musculoskeletal models. For forward dynamics, the user specifies the on/off times of muscles - the Pipeline computes the resulting

Chapter 4
1 FEB 12

motions of the body segments. For inverse dynamics, the user specifies the time histories of the joint angles during some activity - the Pipeline calculates the torques required to generate that motion.

The Dynamics Pipeline runs on Silicon Graphics workstations and requires Irix 5.3 or higher. It also runs on IBM compatible PCs running Windows.[®] The Dynamics Pipeline requires both SIMM and SD/FAST.

4.2.2.3 Generic Knee Model

The SIMM Generic Knee Model is a high resolution anatomical and kinematic model of the knee derived from plastic bone models and MRI data and designed specifically for SIMM users. The Knee Model consists of four bones (femur, tibia, fibula, and patella), five ligaments (anterior and posterior cruciate, medial and lateral collateral, and patellar ligaments), and the lateral and medial menisci. The bones and menisci are modelled as rigid polyhedra and the ligaments as deformable polyhedra.

The Generic Knee has two degrees of freedom that represent flexion-extension and varus-valgus motions. The femur is the fixed bone and the tibia moves relative to it, while the patella and the menisci move relative to the tibia.

Kinematic functions defining the varus-valgus motion of the tibia force it to rotate about the center of the lateral tibial plateau during valgus motion and about the center of the medial tibial plateau during varus motion. The kinematics of the knee can easily be modified in SIMM to include new degrees of freedom or to represent new experimental data.

4.2.2.4 Full Body Model

The Full-Body Model is a SIMM musculoskeletal model of an average, adult male. It contains 86 degrees of freedom, 117 joints, and 344 muscle-tendon actuators. The joints have anatomically accurate kinematics (e.g., the knee model accounts for the sliding and rolling of the tibia and patella on the femur). The muscle-tendon actuators include the force-generating properties so that SIMM can calculate the length, moment arm, force, and joint moment for any muscle in any body position. The model was jointly developed by biomechanics research labs at several universities. All of the model parameters are from published sources and a bibliography is provided. In addition, the model is fully customizable, so user can change any of its parameters or add additional muscles or degrees of freedom.

The Full-Body Model is an excellent resource for anyone who wants to perform a musculoskeletal analysis of any portion of the human body. The model can be animated by recorded motion data, and used to examine muscle strengths and joint moments during the activity. By using the model scaling utility that comes with SIMM, the model can be scaled to match any size individual. Workspace designers,

biomechanics researchers, and people involved in task analysis and ergonomics will find this comprehensive, accurate model especially useful.

4.2.2.5 Real Time Motion Module

The Real Time Motion Module imports motion files created by Motion Analysis systems. Motions imported into SIMM are mapped onto a pre-built, customizable, full-body model with 344 anatomically accurate muscles. The body segments, joint kinematics, and muscle attachment sites of the model are scaled to match the size of the subject whose motion is being recorded. To accompany a motion, SIMM can also import analogue data such as ground-reaction forces and muscle activation levels (EMG data). Additionally, SIMM can import kinetic data computed by OrthoTrak.

The Motion Analysis file formats that SIMM can import are: TRC, TRB, ANB, ANC, and XLS. The Real Time Motion Module makes an excellent post-processing tool for analyzing a recorded motion in detail. Once the motion is imported into SIMM, it can be played back in a 3D graphical display showing the skeleton, ground-reaction forces, and muscle activations. Ground-reaction forces are displayed as vectors superimposed on the model, and the muscle activation data is used to change the size and colours of the muscles. In addition, plots can be made of joint angles, ground forces, muscle activations, and muscle lengths.

The Real Time Motion Module has the ability to receive motion data in real-time, as it is being collected. It works by connecting to the machine that is running Motion Analysis software using a standard Ethernet connection. As motion data is collected and processed, it is sent out over the network to the machine running SIMM. SIMM can then animate the musculoskeletal model and display streaming plots of the data. Because the model being animated has accurate muscle-tendon actuators, user can even display real-time plots of the subject's muscle lengths as the motion is being captured.

The Real Time Motion Module is the ideal tool for clinicians and researchers who want to visualize the relationships between muscle activity, external forces, and the resulting body motion. It is particularly useful in task training and other applications that benefit from real-time feedback. A subject's motion can be recorded and instantly compared to a desired motion. The SIMM display can also be used as real-time visual biofeedback so that the subject can monitor his own activity. Gait analysis and sports performance centers can use SIMM and the Real Time Motion Module to analyze in detail an individual's motion and help diagnose problems and improve performance. Biomechanical research and industrial applications include product design, ergonomics analysis, and anthropology.

4.2.2.6 C3D Module

The C3D Module is an optional component to SIMM that has been developed to visualize results of gait analysis studies. This software is specifically designed to read in C3D files to create a three-dimensional animation showing body motion, muscle activity, and ground reaction forces. It is used by several leading motion analysis laboratories to interpret gait analysis data and to communicate results of these studies to others.

4.2.2.7 FIT Module

The FIT (**F**orward **I**nverse **D**ynamics **T**ool) Module allows users to perform forward and inverse dynamics simulations on the musculoskeletal models, without any programming. It works by loading pre-compiled simulations of generic models and then customizing them to match the particular model.

For inverse dynamics, users can load a motion into the FIT Module and specify which properties, such as joint torques and joint contact forces, users would like to calculate. SIMM uses a novel dynamics technique to calculate these properties, taking into account ground-reaction forces and muscle EMG data, if available. An upcoming version of the FIT Module will use an optimization algorithm to estimate the individual muscle forces that produce the calculated joint torques.

For forward dynamics, users can configure the model in any desired starting position, and specify the excitation patterns of the muscles. Users can also apply external forces and torques by defining them in an input file. SIMM then starts the simulation and integrates the equations of motion for the specified time period. The results are shown as they are calculated in the SIMM model window and plot windows. Using the graphical interface, it is quick and easy to run a simulation, examine the results, and then adjust the muscle excitations or other import variables and run another simulation.

In conjunction with the Real-Time Motion Module, the FIT Module is a powerful toolkit for performing dynamics analyses on virtually any recorded motion. The recorded marker data (TRB, TRC files) is imported into SIMM and mapped onto a full-body model (with masses, inertias, and muscles) that is scaled to fit the size of the subject. If desired, the imported motion can then be cropped, and events can be added to it (e.g., heelstrike, toe-off).

4.2.3 LifeMOD⁴⁵

LifeMOD is a plug in to the popular ADAMS⁴⁶ software to provide a complete modeling environment to refine the task of creating passive, and forward-dynamics biological models which interact with the environment, tools, equipment and each other (Figure 4.3).

ADAMS is a motion simulation software for analysing the complex behaviour of mechanical assemblies.

ADAMS allows users to test virtual prototypes and optimise designs for performance, safety, and comfort, without having to build and test numerous physical prototypes.

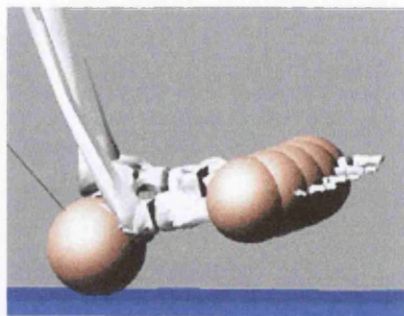


Figure 4.3 – The LifeMOD Modelling System [45]

Some of the features of LifeMOD are presented below:

- **Multiple Interacting Models.** As many as 6 models may be created in one environment. The models may interact with the environment, equipment, or each other to perform coordinated tasks. Models may be specified separately as passive or active.
- **Five Anthropometric Body Size Databases.** Databases included for GeBod data, and PeopleSize UK, USA, Japan and China databases. Segment sizes and mass properties may be scaled from data for male, female, median, 5%-ile and 95th-ile.
- **Several Types Human Body Representation.** Five types of human body representation include male skin, female skin, ellipsoid, crash dummy and stick. The investigator may switch between representations all through the simulation process.
- **Segments May Range in Bio-fidelity.** LifeMOD employs a "segment refine" process to create human models. A 19 segment model is created initially, with bones lumped into segments. The segments are scaled based on the anthropometric databases. Individual segments may then be refined or dispersed into the true number of bones for detailed representation. This allows the investigator to create a model with the most efficient fidelity.
- **Complete Scalable Skeleton.** All bone graphics representing the complete human skeleton are generated and scaled based on the anthropometric database which includes male/female adult and child populations.
- **Three Body Generation Input Methods.** 1) Simple: height/weight/gender 2) Body Segment Measurements: 50 body measurements 3) Joint Locations: 22 joint location measurement.
- **Body File Import/Export.** Body data may be imported/exported for cataloguing and automatic body generation capability.
- **Any Biological Entity May Be Simulated.** LifeMOD allows the import of body segment geometry from any biological organism.

- **Large Body Posture Library.** Large standard library of postures as well as a posture save capability to add to the library.
- **Human Joints Automatically Generated.** A standard set of 20 tri-axis joints is created automatically. Each joint DOF may be individually modified to include stops, forces, friction, torques, etc.
- **Joints May be Individually Refined.** Each joint DOF may be modified specifically to further replicated the intended human motion.
- **Hybrid III Crash Dummy Strength Characteristics.** Crash dummy strength characteristics are available for passive-type simulations. The strength model may be scaled to the particular body size and gender. The strength algorithm includes joint friction, non-linear stiffness/damping and limits with hysteresis using data derived from the physical dummy.
- **Force-based Joints.** More sophisticated joints which rely on bearing surface contact forces and ligament stabilization may be modelled
- **Joint Axes Manipulation.** To adjust the posture of the model each joint axes may be adjusted using control panels. The posture may be saved for later retrieval.
- **Automatic PD-servo controllers created at each joint axis.** Using joint motion patterns developed from inverse-dynamics solution, pd-servo controllers may be created at each joint DOF to allow the human model to track the recorded motion using joint torques.
- **Full Body Muscle/tendon Groups.** The joints may be stabilized using ligament forces and driven using muscle forces. The muscle forces are series spring/damper activation elements which can create forces derived from motion patterns from the inverse-dynamics simulation. Insertion points for the tissues may be relocated easily, with the cursor or control panels.
- **Tissue Contact Force Tools.** Tools are available to model the forces resulting from body segment contacting the environment, or each other.
- **Any MOCAP device maybe used to provide data.** LifeMOD contains a standard interface to motion capture products through a simple marker trajectory ASCII text file.
- **Complete marker sets automatically positioned.** Marker sets are automatically created on the model after import from the motion capture system. Marker trajectories, imported from a MOCAP system may be used to drive the model using a "Motion Agent" approach. Motion agents are created at the locations of the motion capture sensors and attached to the model using springs. The stiffness properties of the springs are based on the relative accuracy of the specific marker. Thus, the motion agents influence the motion of the model not govern, since the model is bound by the joint limits, environment, physics, etc.
- **Automatic Equilibration Process.** The human model is automatically fit into the marker cloud using an energy minimization principal.

- **Accommodates subject-model differences and other error sources.** Since motion agents are not rigidly attached to the model, the model can allow some relative motion thus, accommodating any model-subject discrepancies and/or other data errors (i.e., skin wobble).
- **Inverse Dynamics Simulation.** After the motion agents are positioned on the model, a simulation is performed to capture the motion of the model and/or muscles. The motion agents drive the model, yet the model is under the influence of the joint limits, external forces and gravity (i.e., the foot will not be forced through the floor). During this process, "learning" elements in the muscles record each muscle contraction history.
- **Forward Dynamics Simulation.** Learning elements are replaced with contractile elements based on the recorded contraction histories to drive the human model to match the recorded motion. A simulation is then performed with the model guided by the internal forces (joint torques and/or muscle forces) and influenced by the external forces (gravity, contact, etc.) From this simulation the investigator obtains insight into muscle function, timings, magnitudes, force action, etc.
- **Animations.** The simulation results may be used for detailed animations with dynamic plot display. Also, move (avi) files may be created instantly.
- **Kinematics.** Kinematic data including joint/segment displacements, velocities, accelerations may be recorded in any coordinate frame. Any location on the model may be tracked and reported. Kinematic data from any interfacing mechanical system may be reported as well.
- **Kinetics.** The model kinetics include torques and forces such as joint torques, soft tissue forces and contact forces. Internal reactions such as mechanical system forces and interface forces may also be reported.

4.2.4 SimMechanics⁴⁷

SimMechanics allows engineers to model and simulate mechanical systems in the Simulink environment (Figure 4.4). With SimMechanics, user can directly model mechanical components

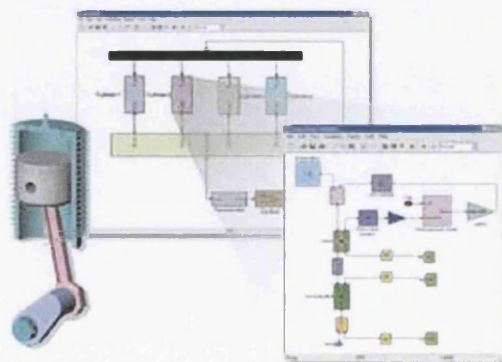


Figure 4.4 – The SimMechanics Modelling System [47]

(bodies and joints), simulate their motion, and analyze the results without having to derive the mathematical equations governing the system.

Technically the system is characterized by the following properties:

- Models and simulates rigid body mechanical systems within Simulink.
- Provides full modelling hierarchy allowing mechanical blocks to be grouped with other block types.
- Uses Simulink's solvers for accurate three-dimensional analysis and simulation.
- Contains comprehensive joint and constraint libraries.
- Models linear and angular motions, forces and torques.
- Provides trimming and linearization capabilities for control system design.
- Enables visualization and animation of mechanical systems using either Virtual Reality Toolbox or Matlab graphics.
- Performs kinematic, forward, and inverse dynamic analysis of mechanical systems.
- Uses $O(n)$ recursive algorithms to solve equations of motion for multibody systems.
- Provides multiple local coordinate systems for model definition.
- Ability to generate stand-alone C code using Real-Time Workshop.
- Provides a SolidWorks translator to enable the use of CAD tools to define mechanical models.

4.2.5 MADYMO⁴⁸

TNO Automotive has applied its expertise in biomechanics research, crash tests and simulation techniques to improve safety design by developing multi-directional human models in MADYMO. MADYMO provides seamless integration of multibody (MB) and finite element (FE) capabilities.



Figure 4.5 – The MADYMO Modelling System [48]

From fast and efficient MB modelling for the concept stage to detailed FE modelling for the design and prototype stage: It can all be done with this code. MADYMO's multibody capabilities form the mathematical foundation of both MADYMO and the TNO developed MADYMO occupant models. Easy-to-generate multibody systems can be simply described as rigid and flexible bodies (or masses), interconnected by various types of kinematic joints.

The human models developed by TNO offer improved biofidelity and can be used in many applications in the area of crash safety, personal protection, sports injuries and seat comfort. For the various applications several full body models, detailed segment models and combinations are available.

The Madymo human models offer:

- Improved biofidelity over crash dummies.
- Improved crash protection for various body sizes.
- Study of injury mechanisms.
- Injury criteria based on tissue failure (stress/strain).
- Modelling of post-failure (e.g. fracture) response.
- Rapid incorporation of biomechanical data.
- Inclusion of muscle activity.
- Posture analysis.
- Evaluation of comfort.
- Accident reconstruction.

4.3 Application Areas

4.3.1 Automotive

- **Egress**⁴⁹. With the ageing population, ingress/egress becomes an increasingly important topic, this egress model (Figure 4.6) demonstrates how the position of an assistive handle on the window frame influences the knee joint forces as well as the muscular effort of egress. For an elderly individual with arthritic knees, this can be of tremendous importance to the usability of the vehicle. The idea behind the study is that placement of a handle on the window frame might allow car drivers to divide the load of egress between the legs and the arms, thus reducing the maximum load on any of the elements. In such a situation, the AnyBody Modelling System automatically recruits muscles such that the effort is distributed between the body parts and indeed between the individual muscles to share the load between the elements according to their individual strengths.



Figure 4.6 – Egress Model [49]

- **Pedal Design⁴⁹**. What is a good pedal? In some sense it should provide effortless and yet precise operation. This model example (Figure 4.7) demonstrates how the pedal stiffness and the distance between the seat and the pedal influences the muscle effort of operating the pedal. Consider a pedal hinged at one end and equipped with a torsional spring that stretches when the pedal is

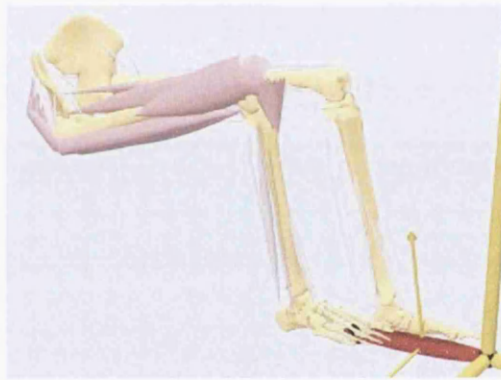


Figure 4.7 – Pedal Design Model [49]

depressed. The dilemma is the following: If the spring is too weak, then the pedal will not provide much support for the leg, and the operator consequently has to extend the leg and hold it up against gravity. This will become very tiring. On the other hand, if the spring is too rigid, then the muscular effort of depressing it will become too large and repetitive operation of the pedal or static maintenance of a particular pedal position will cause fatigue. Similarly, different seat positions influence the muscular effort.

Even a simple model problem like this is much too complicated to be thoroughly investigated by experimental methods. With musculoskeletal modelling it is possible to perform a systematic variation of the model's parameters and obtain a complete overview of the problem.

- **Active Human Surrogate Control of a Motorcycle: Stabilizing and De-Stabilizing⁵⁰**.

Two aspects of a motorcycle injury crash were studied in this project. 1) What were the riders

actions which led to unsafe handling of the motorcycle? 2) What were the injury-producing mechanisms present during the crash?



Figure 4.8 – Rider Model [50]

This inquiry illustrates methods and procedures which are useful for motorcycle safety design as well as reliability analysis. Human models were created of the rider (Figure 4.8) under rider control to simulate a mild lane change manoeuvre. Manual control and vehicle response characteristics were evaluated for cases involving a system disturbance such as an encounter with a road pothole during the turning phase in the lane change. For the case when the rider/cycle system becomes unstable resulting in a crash, the 3-dimensional joint strength model for the human surrogate rider mode switches from an active, motorcycle control model (pre-crash phase) to a passive, rebound model (crash phase), derived from the Hybrid III crash dummy.

- **Safety Equipment Evaluation System⁵⁰**. Over the years many types of vehicular injury scenarios have been simulated. From this experience, software systems for the evaluation of vehicle safety systems have been developed. The simulations performed by these systems include mechanical models of the safety equipment and human models (Figure 4.9), providing information on both the internal and external forces the human is subjected to during various types of collisions.

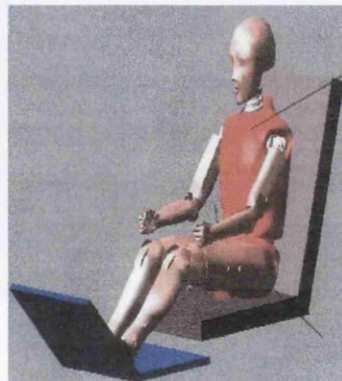


Figure 4.9 – Crash Test Model [50]

4.3.2 Medical/Rehabilitation

- **Gait Analysis**⁴⁹. Gait analysis is one of the real success stories of biomechanics. Today, gait laboratories are used routinely for investigation of a variety of conditions, for diagnostics and planning of surgery, and for rehabilitation. However, traditional gait analysis is almost exclusively



Figure 4.10 – Gait Analysis Model [49]

about kinematics. It is possible to compute joint moments in the open-chain mechanisms formed by the lower extremities, but traditional gait analysis reveals little if anything about the internal forces in the body.

- **Wheelchair**⁴⁹. A very large proportion of wheelchair users (Figure 4.11) experience load-induced shoulder pain. This happens after several years' use, and it can be a very serious condition for an individual relying entirely on the arms for ambulation. The wheelchair parameters such as wheel diameter, pushrim position, axle position, and camber influence the shoulder forces during use.

But precisely how?

This is a field where the consequences of a design change do not appear until several years of use. This alone makes experimentation impossible. But it is also ethically unacceptable to fit wheelchairs to humans in any other way than following best practice. In other words, computer simulation is the only way to gain knowledge about the influence of the wheelchair design on body loads. Once design techniques have been established it also allows to tailor the wheelchairs for optimal operation by individuals.

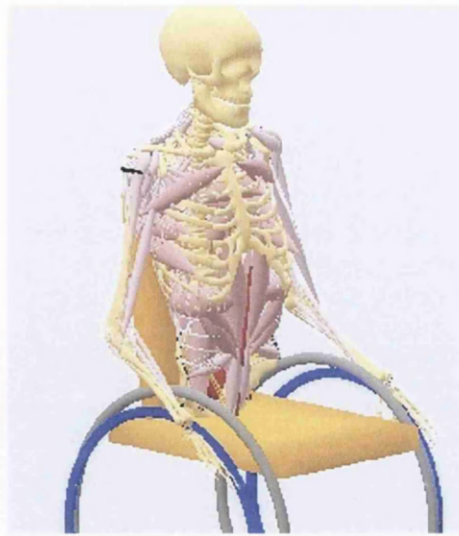


Figure 4.11 – Wheelchair Model [49]

- **Hip Replacement**⁵⁰. This example involves the usage of a human model to evaluate the functional performance of a new hip replacement system (Figure 4.12). A musculoskeletal lower body model is created and the hip system is implanted in the model. Motion capture data is used to drive the model for the inverse dynamics analysis to "train" the muscles to perform the task. A forward dynamics analysis is then performed with the muscles driving the model, to evaluate the forces at the stem/bone interface and at the cup and to determine if there is impingement between the stem and cup during the walk-sit-run activity.



Figure 4.12 – Hip Replacement Model [50]

- **Bed Settling**⁵⁰. This example involves the usage of a human model to determine the resting configuration for a patient in a hospital bed. The simulation is performed for beds with varying cushioning characteristics for patients with varying joint stiffness properties.

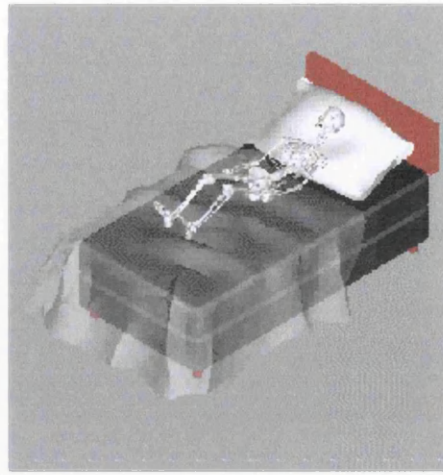


Figure 4.13 – Bet Settling Model [50]

4.3.3 Aerospace

- **Cockpit Design**⁴⁹. The performance of a pilot is crucial to the safety of the passengers and payload and, in military applications, to the success of the mission. Even with very careful selection of pilots, the human performance is now the bottleneck in many aircrafts. This makes the design of the cockpit environment a key factor (Figure 4.14).

With musculoskeletal simulation it is possible to assess the influence of the design parameters on pilot fatigue and aircraft operability. Modeller can evaluate whether the pilot will be able to move the hand from point A to point B when pulling 7g, and can assess the possible accuracy of such a handle operation.

For instance, flying a rescue helicopter in difficult weather conditions requires the utmost precision, and the placement and feedback of pedals and handles play an important role.



Figure 4.14 – Cockpit Design Model [49]

- **Passenger Spaces⁴⁹**. Space and weight are crucial factors in the design of any aircraft. Both of them translate directly into cost, and any passenger in the coach section of an international flight can vouch for the fact that the shortage of space can make flying a taxing experience in terms of comfort and fatigue (Figure 4.15). Getting in and out of seats, sitting in cramped spaces for many hours, and loading heavy bags into overhead bins are just a few of the cases that are difficult or fatiguing for young and fit people and may be downright impossible for elderly or impaired individuals. The good news is that with musculoskeletal modelling the consequences of the design of the environment can be assessed and indeed optimised: the dimensions of hinge mechanisms in overhead bins for easy closing; the placement of assistive handles and support points for entering and exiting a seat row; or the support profile of a seat and the kinematic compatibility of its adjustments with the human body.

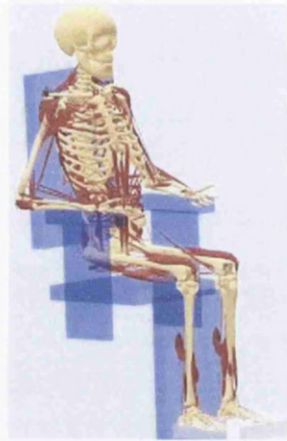


Figure 4.15 – Passengers Spaces Model [49]

- **Galileo space probe high-gain antenna deployment⁵⁰**. The Galileo space program suffered a setback when the high-gain antenna would not deploy while in space. From the deployment motor torque signal, engineers were able to ascertain that the antenna only partially deployed. Since the power necessary to run the motor was limited, the engineers did not want to try any efforts to jog free the antenna without knowing to what extent the antenna did deploy.

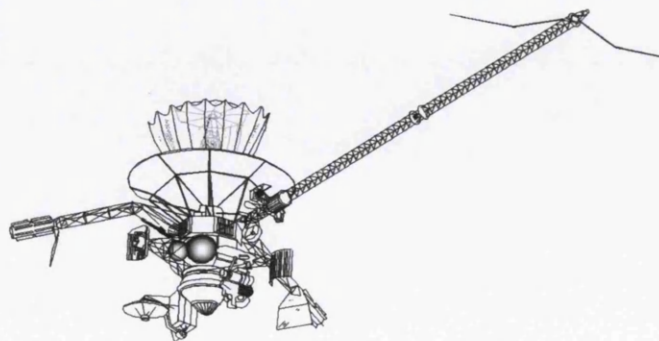


Figure 4.16 – Galileo Model [50]

Engineers were called to help determine the antenna configuration. The detailed model of the

antenna included the flexible spines, the effects of the antenna dish metal meshwork, the screw-jack mechanism, and all the locking and guiding mechanisms. The idea was to use this model to see where the antenna was "sticking" by comparing the torque signal generated by the actual spacecraft to the model. By successively fixing the flexible spines at various locations the torque curves generated by the model were consistent with the curves from the spacecraft. Torque curve profiles depended on how a stuck spine or spines would cause warping and therefore a torque change in the screw-jack mechanism. Now with a visual model of the state of the antenna, engineers were able to make better informed decisions on how to free the antenna.

- **Microgravity Environments**⁴⁹. While space flight and the experience of weightless conditions remains an exciting dream for most people, it is no easy task even for an astronaut. Experiments have shown that the human body can lose as much as two percent of its bone mass per month in microgravity conditions. Similarly, studies have shown a reduction of up to 20 percent of the mass of antigravity muscles in space missions lasting only 11 days. There is a good reason why we see astronauts being carried out of the capsule when they return after an extended space mission. It is believed that the loss of bone and muscle mass is related to the lack of gravitational forces on the body. So one of the countermeasures is to subject the astronauts during the mission to exercise of the structures that normally carry gravity. But precisely how can we fool the body into believing that gravity is still active? It is impossible or very difficult to measure the forces inside the human body, but musculoskeletal simulation can predict these forces with good accuracy and can in fact be used to devise exercises and equipment loading for the bone and muscle structures as closely as possible to the gravity of our natural environment.
- **Mars Rover Concept Development**⁵⁰. While in the concept stage, various vehicle configurations were considered. Engineers assisted with the development from initial concept to final model (Figure 4.17). With a virtual prototype, the gravity and soil conditions of Mars may be replicated in order to evaluate the terrain negotiation capability of the vehicle.



Figure 4.17 – Mars Rover Model [50]

4.3.4 Occupational Health

- **Workplace design**⁴⁹. Workplaces with elements of materials handling or lifting (Figure 4.18) are among those with the highest propensity for injury in back, shoulders and arms. The overall explanation is that any repetitive work task that degenerates tissue faster than it can be repaired will eventually traumatize the human body. Our muscles typically have much smaller moment



Figure 4.18 – Workplace Design Model [49]

arms than the exterior loads, and therefore small differences in the working postures or loading parameters can cause large differences in the tissue loads.

A musculoskeletal model allows for detailed analysis of a particular work task and for assessment of the consequences of a change in the workplace design. For example, it can give answer to the following question; how will the sideways movement of a handled material influence the joint forces, and is this likely to influence the shoulder pain the worker is experiencing?

- **Pushing**⁴⁹. Pushing is one of the most common manual work tasks. In this particular example (Figure 4.19) a hospital employee is pushing a bed with a total mass of 200 kg forward at an acceleration of $0.3m/s^2$. The duration of the push is 1.5 seconds. It is possible to choose three different heights of the horizontal handle bar: 0.9m, 1.1m and 1.3m. The question is: which handle bar height results in the minimal load on the body? A musculoskeletal model enables simple investigations of muscle efforts and joint forces depending on postures and other workplace parameters.



Figure 4.19 – Pushing Model [49]

- **Equipment design**⁴⁹. The design of portable equipment such as backpacks (Figure 4.20) has a large influence on comfort and fatigue. A musculoskeletal model can investigate the variation of

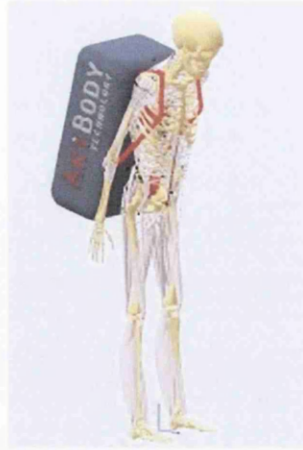


Figure 4.20 – Equipment Design Model [49]

the center of gravity which changes the posture and muscle activation in a complicated fashion, and the same can be said for the way the straps put their forces on the bones of the body.

4.3.5 Sports

- **Bicycling**⁴⁹. Bicycling is a fascinating sport in which small advantages between riders make all the difference. Fine adjustments of the bicycles or exercise strategies can mean the difference between victory and defeat. With the complexity of the mechanics of the human body, the road to improved performance remains any man's guess.

It is a highly repetitive task, and it is a good assumption that the rider is capable of recruiting muscles optimally for the task. The model (Figure 4.21) can be used to predict the parameter changes (mechanical design) that can improve performance.



Figure 4.21 – Bicycling Model [49]

- **Fitness Equipment⁴⁹**. Most strength-based sports performances are not only about strength. They are about the right strength. This is because muscles are heavy, and there is no need to add muscle

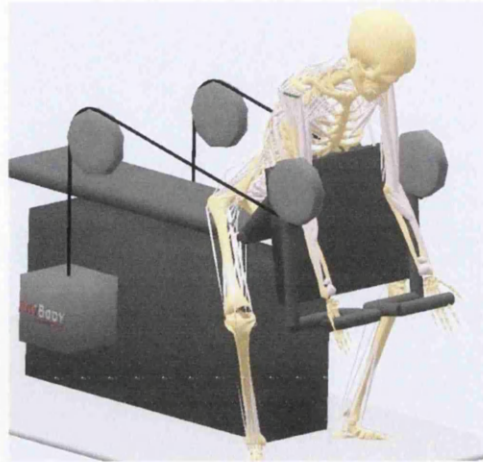


Figure 4.22 – Fitness Equipment Model [49]

mass that does not add to the performance. A musculoskeletal model can help identify the muscles that are currently the bottleneck for performance improvement, and it can subsequently design exercises for improvement of precisely these muscles.

Fitness equipment (Figure 4.22) is a major field of application of a musculoskeletal modelling system. The system's ability to model the mechanism of the machine and its interaction with the human body creates the opportunity of optimizing the machine's load of the muscular system.

- **The development of a virtual golfer⁵⁰**. This project resulted in the development of a human model capable of swinging a flexible golf club (Figure 4.23). The sophisticated human model was a personalizable model, capable of actually adapting to the mechanics of the golf club. For example, if a stiffer club was introduced into the model, the human model would adapt or change the swing mechanics to produce a "good" shot. Where a good shot was defined as an acceptable club head orientation and head speed at ball contact.

This capability is leading to the ability to understand the "human variables" when developing new golf club parameters, before actually building the club.



Figure 4.23 – Virtual Golfer Model [50]

4.4 Conclusions

One of the intentions of this chapter was to introduce the most popular software systems for musculoskeletal modelling and state the basic features of each of these systems. Of course from a first view, a potential user may decide which software suits to his needs, nevertheless a “hands on” experience could give a better insight of the capabilities and the user-friendliness of each of these simulation tools. Another intention of the present chapter was to show the grade in which these modelling systems could improve aspects of our everyday life. This was evident through the presentation of examples categorised in five application areas.

CHAPTER 5

THE ANYBODY MODELLING SYSTEM

5.1 Introduction

As it was mentioned in the Chapter 4, Anybody Technology is the simulation tool that is going to be used for the development of musculoskeletal models below. So, the present Chapter 5 aims to give a more detailed description than the one given in Chapter 4, of the main characteristics of the Anybody modelling system.

The presentation starts with a discussion on the model definition language of Anybody Technology called AnyScript, followed by an overview of the overall structure and the basic programming elements of an AnyScript based musculoskeletal model. The chapter continues with the presentation of the mechanical elements that constitute a model. Then it goes further with a section for the simulation technique that is used, and it ends with a discussion on the muscle recruitment solver this system provides.

5.2 The AnyScript Language

5.2.1 Definition

AnyScript is the model definition language of the AnyBody body modelling system.

AnyScript is actually an object-oriented programming language developed bottom-up for the purpose of describing the construction and behaviour of bodies of living creatures. It can also model different artifacts that the body happens to be connected to. Typical examples would be bicycles, furniture, sports equipment, hand tools, and workplaces.

AnyScript contains facilities for definition of bones (called segments), their connections by joints, muscles, movements, constraints, and exterior forces.

One of the ideas behind AnyScript is that its text-based format and object-oriented structure makes it easy to transfer elements between models. Researchers can build a library of body segments for use in their different analysis projects, and they can easily exchange models with other users and collaborate with them on complex modelling tasks⁵¹.

In many ways AnyScript looks syntactically like C++ or Java and this is a deliberate choice. In fact if AnyScript is compared to popular programming languages, it can be found that it is similar to JavaScript. The syntax is much the same, and AnyScript shares the property with JavaScript that it works on a set of predefined classes and does not let the user define new classes. Some of the syntactic similarities are the uses of braces to delimit different sections of the model from each other and the fact that each declaration is terminated by semicolon. The main difference is that AnyScript does not contain any executable

statements, so user cannot specify a sequence of events, i.e., write algorithms; AnyScript is purely declarative contrarily to all other mentioned languages⁵².

5.2.2 Basic Concepts⁵¹

An AnyScript model is roughly divided into two main sections:

- The body model containing the definition of the mechanical system.
- The study section containing lists of analyses and other operations to perform on the model.

An example of an AnyScript model is the one that is shown in Figure 5.1 below:

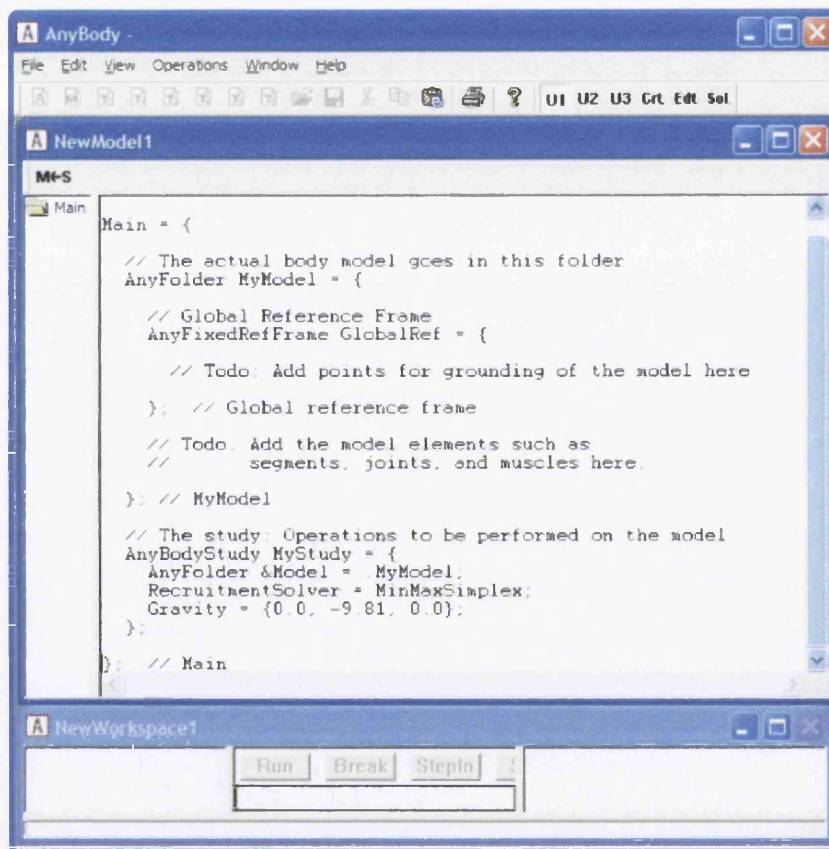


Figure 5.1 – AnyScript Model

Concentrating on the main features, the overall structure of the model looks like this (Figure 5.2):

```
Main = {  
  AnyFolder MyModel = {  
  }; // MyModel  
  AnyBodyStudy MyStudy = {  
  };  
}; // Main
```

Figure 5.2 – AnyScript Model's Overall Structure

What it is seen now is a hierarchy of braces - just like in a C, C++, or Java computer program. The outermost pair of braces is named "Main". Everything else in the model goes between these braces. Right now, there are two other sections inside the Main braces: The "AnyFolder MyModel" and the "AnyBodyStudy MyStudy". These are the two basic elements of most AnyBody models. The term "AnyFolder" is very general. In fact, any pair of braces in AnyScript is a folder. Users can think of a folder as a directory on the hard disk. A directory can contain other directories and files. It's exactly the same with folders. They can contain other folders and elements of the model. The "AnyFolder MyModel" is the folder containing the entire model a user is going to build. The name "MyModel" can be changed by the users to anything they like. For example the name "MyModel" of Figure 5.2 can be changed with the name "ArmModel" and this change is highlighted by red as shown in Figure 5.3.

It has also to be noticed the prefix "Any". All reserved words in AnyScript begin with "Any". This way the elements that belong to the system can be distinguished from what the user defines.

Of course there is more to an AnyScript file than the model. Once a model is created, a user can perform various operations on it. These operations are collected in "studies", and the "AnyBodyStudy MyStudy" is indeed such a study. A user can think of a study as the definition of a task or set of tasks to perform. The study also contains methods to perform the tasks.

```
// The actual body model goes in this folder
AnyFolder ArmModel = {
  // Global Reference Frame
  AnyFixedRefFrame GlobalRef = {
    // Todo: Add points for grounding
    // of the model here
  }; // Global reference frame
  // Todo. Add the model elements such as
  // segments, joints, and muscles here.
}; // ArmModel
```

Figure 5.3 – AnyScript Model's Actual Body

Looking closer at the contents of what is now the ArmModel folder (Figure 5.3), most of what is seen, it is just comments. It is always useful to add many comments to the models. It has to be noticed that one-line comments are started with "/*". Multiple-line comments are enclosed in "/* */" just like in the C++ or the Java language. It should also be noticed that lines are terminated with a semicolon ";". Even the lines with closing braces must be terminated with a semicolon. If a user does not terminate with a semicolon, then the statement continues on the next line.

As can be seen in Figure 5.3, the only actual model element in the ArmModel is the declaration of the "AnyFixedRefFrame GlobalRef". All models need a reference frame - a coordinate system - to work in. An AnyFixedRefFrame is a predefined data type a user can use when it is needed. What is there, is the

definition of an object of that type. The object gets the name "GlobalRef", and a user can subsequently refer to it anywhere in the ArmModel by that name.

5.3 Mechanical Elements⁵²

5.3.1 Segments

Segments are defined by the *AnySeg* keyword in AnyScript. They are the rigid bodies on which the system bases its analysis. In a body model, segments are usually bones, but since an AnyBody model often comprises various equipment and other items, segments are also used to model cranks, pedals, handles, tools, sports equipment, tables, chairs, and all the other environmental objects a body may be connected to.

In fact, an AnyBody model does not have to entail a living body. Users can easily create an AnyBody model of a machine in which no biological elements take part.

Segments in AnyBody are basically a set of mass properties. They do not have any particular shape associated with them. A segment is originated in its center of mass (CoM), and the mass properties are defined by means of a mass and an inertia tensor.

Another important property of a segment is that it can have nodes related to it. A segment has a local reference frame or coordinate system, and any number of nodes can be defined in this reference frame. Since they are locally defined, they follow with the movement of the segment as they should when they belong to a rigid body.

5.3.2 Joints

Joints are used to connect segments and allow them to articulate with respect to each other. A segment without joints is basically floating free in space. When users connect the segments by joints, user binds them together in some sense. But the mechanism as a whole can still move around in space.

There are several types of joints within Anybody modelling system. *AnySphericalJoint* is a point constraint also known as ball-socket joint, and allows rotation around all three axes with no linear translation. *AnyCylindricalJoint* allows rotation about a given axis and translation along the axis, i.e. a combined hinge and translational joint. *AnyRevoluteJoint* is an ideal hinge that only allows rotation about one axis. *AnyPrismaticJoint* is a translational joint that does not allow rotation about the axis.

AnyUniversalJoint is a joint that allows rotation about two perpendicular axes.

5.3.3 Drivers

Drivers create movement in a model. They are functions of time determining the position of a joint or the distance between two points or some other kinematic measure at any given time through the simulation period.

There are various drivers available to create different types of time dependency. Some of them are the following: *AnyKinEqSimpleDriver* provides motion with constant acceleration, *AnyKinEqFourierDriver* provides a Fourier expansion as a driver function and *AnyKinEqInterPolDriver* provides a driver function that interpolates between data points spaced in time.

5.3.4 Kinematic Measures

The concept was invented by AnyBody Technology as a way of describing dimensions in a kinematic model that users might want to get information about or control with drivers. A joint angle or a distance between two points are examples of kinematic measures. The position of the center of gravity of the entire model or a subset of its segments are other examples. There are various types of “kinematic measures” in Anybody Technology. *AnyKinLinear* is “kinematic measure” that measures translation, i.e., a 3D vector. It is either the position of one reference frame with respect to the global frame or the relative vector between two frames. *AnyKinCOM* measures the center of mass (CoM) of a collection of rigid bodies. *AnyKinPLine* measures the distance between points. *AnyKinSPLine* measures the length of the shortest path of a string that may be in contact with surfaces. *AnyKinRotational* measures 3D orientation/rotations. It measures either the rotation of a single reference frame with respect to the global frame or the relative rotation of one frame with respect to another. *AnyKinMeasureLinComb* is a composer of new “kinematic measures” as a linear combination of existing ones.

5.3.5 Forces

There are several types of forces in an AnyBody model; forces in joints, forces in muscles, and gravity forces working on the segments. This subsection, however, deals only with the application of external forces. *AnyForce* is the base class for all applied forces, i.e., forces that are explicit functions of the model state (position and velocity) and time. It is a generalised force in the sense that it can be applied to any kinematic measure or even an assembly of such. *AnyForce3D* is a three-dimensional concentrated force (a vector). The force can be specified in both global and local coordinates but not simultaneously. *AnyMoment3D* is a three dimensional moment applied to a reference frame. The moment can be specified in both global and local coordinates but not simultaneously.

5.3.6 Muscles

Muscles are the actuators of living bodies. They are activated by the central nervous system (CNS) by a complicated electro-chemical process. Determining the activation that realizes a desired movement requires an extremely intricate control algorithm. The CNS is superior to any computer man has made in this respect. AnyBody mimics the workings of the CNS by computing backwards from the movement and load specified by the user to the necessary muscle forces in a process known as inverse dynamics. To do so, the system must know the properties of the muscles involved.

AnyBody contains three different muscle models ranging from simple to more complicated physiological behaviour. The simplest model just assumes a constant strength of the muscle regardless of its working conditions. The more complicated models take such conditions as current length, contraction velocity, fiber length, pennation angle, tendon elasticity, and stiffness of passive tissues into account.

AnyMuscleModel assumes constant strength of the muscle. *AnyMuscleModel3E* is a three-element model taking serial and parallel elastic elements into account along with fiber length and contraction velocity.

AnyMuscleModel2ELin is a bilinear model taking length and contraction velocity into account.

The muscle models can be linked to different types of muscles. *AnyViaPointMuscle* is a muscle that passes through any number of nodes on segments on its way from origin to insertion.

AnyShortestPathMuscle is a muscle that can wrap over geometries such as cylinders and ellipsoids and CAD-defined surfaces. This muscle type is very computationally demanding and requires careful adjustment. *AnyGeneralMuscle* is a more standard actuator-type muscle that can be attached to a kinematic measure.

5.4 Inverse Dynamics

5.4.1 Introduction

The output of a movement is what is seen. It can be described by a large number of kinematic variables: displacements, joint angles, velocities, and accelerations. If researchers have an accurate model of the human body in terms of anthropometric variables, they can develop a reliable link-segment model. With this model and accurate kinematic data, they can predict the net forces and muscle moments that caused the movement we have just observed. Such an analysis technique is called an *Inverse Solution*. It is extremely valuable, as it allows to compute variables such as joint reaction forces and moments of force. Such variables are not measurable directly¹⁸. It is clear that resultant joint moments are mathematically associated with the (unknown) forces produced by the individual muscles. These unknown muscle forces can be estimated with a musculoskeletal model, like Anybody Technology provides, that uses a Hill model to represent each specific muscle.

5.4.2 Mathematical Formulation

The Inverse Dynamics Analysis for a segment (Figure 5.4) is based on the Newtonian and Euler three-dimensional equations of motion.

In this analysis, the ground reaction forces are measured in the GRS (global reference system) and the moments of inertia are known in the anatomical axes. All joint reaction forces are initially calculated in the GRS, and then both the reaction forces and the moments of the joints are calculated in the anatomical axes using the transformation matrices.

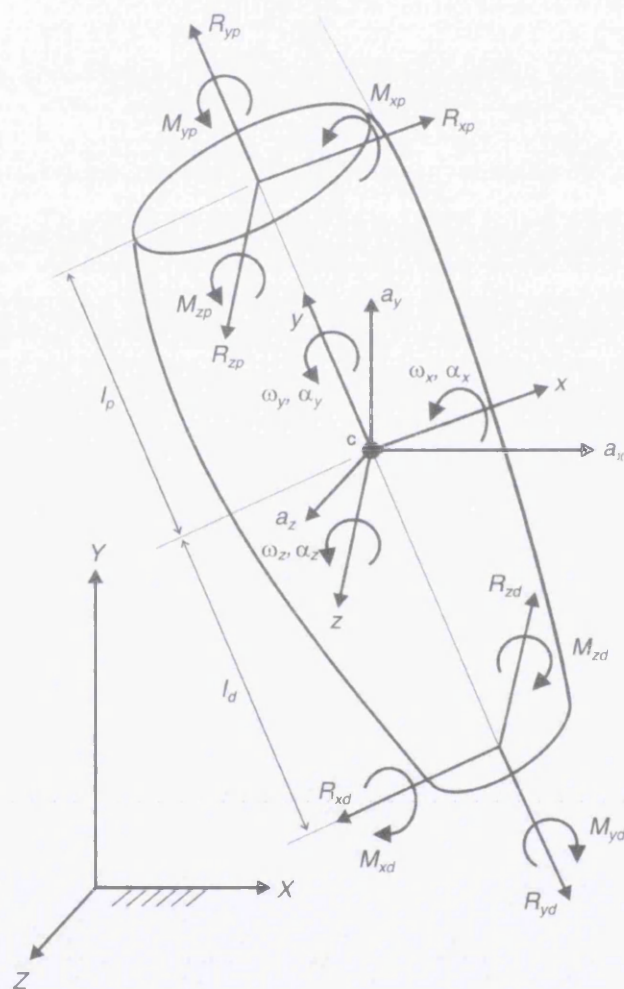


Figure 5.4 – 3D Free-Body Diagram for Solution of the Inverse Dynamics Equations [18]

In the analysis that will follow, the implementation of the Inverse Dynamics Method will be presented in simple steps. Known are the distal reaction forces and moments, the COM (center of mass) linear accelerations, and the segment's angular velocities and accelerations.

- **Step 1 - Calculation of the reaction forces at the proximal end of the segment in the GRS**

The calculation is implemented using the Newtonian three-dimensional equations of motion for a segment (equations 5.1, 5.2, 5.3)

$$\Sigma F_X = m \cdot a_X \quad \text{or} \quad R_{XP} - R_{XD} = m \cdot a_X \quad (5.1)$$

$$\Sigma F_Y = m \cdot a_Y \quad \text{or} \quad R_{YP} - R_{YD} - m \cdot g = m \cdot a_Y \quad (5.2)$$

$$\Sigma F_Z = m \cdot a_Z \quad \text{or} \quad R_{ZP} - R_{ZD} = m \cdot a_Z \quad (5.3)$$

where a_X , a_Y , a_Z are the segment COM accelerations in the X , Y , Z GRS directions and R_{XP} , R_{XD} , R_{YP} , R_{YD} , R_{ZP} , R_{ZD} are the proximal and distal reaction forces in the X , Y and Z axes.

- **Step 2 - Transformation of both proximal and distal reaction forces into the anatomical axes**

The transformation from the global to the anatomical axes is implemented using the following matrix R (5.4) that represents a *Cardan* X , y , z rotation sequence (to be described in section 6.6):

$$[R] = \begin{bmatrix} \cos \beta \cdot \cos \gamma & \cos \gamma \cdot \sin \beta \cdot \sin \alpha + \sin \gamma \cdot \cos \alpha & \sin \gamma \cdot \sin \alpha - \cos \gamma \sin \beta \cos \alpha \\ -\sin \gamma \cdot \cos \beta & \cos \alpha \cdot \cos \gamma - \sin \alpha \cdot \sin \beta \cdot \sin \gamma & \sin \gamma \cdot \sin \beta \cdot \cos \alpha + \cos \gamma \cdot \sin \alpha \\ \sin \beta & -\cos \beta \cdot \sin \alpha & \cos \alpha \cdot \cos \beta \end{bmatrix} \quad (5.4) \quad [10,18]$$

The reaction forces can then be transformed from the global to the anatomical axes system:

$$\begin{bmatrix} R_{xd} \\ R_{yd} \\ R_{zd} \end{bmatrix} = \begin{bmatrix} \cos \beta \cdot \cos \gamma & \cos \gamma \cdot \sin \beta \cdot \sin \alpha + \sin \gamma \cdot \cos \alpha & \sin \gamma \cdot \sin \alpha - \cos \gamma \sin \beta \cos \alpha \\ -\sin \gamma \cdot \cos \beta & \cos \alpha \cdot \cos \gamma - \sin \alpha \cdot \sin \beta \cdot \sin \gamma & \sin \gamma \cdot \sin \beta \cdot \cos \alpha + \cos \gamma \cdot \sin \alpha \\ \sin \beta & -\cos \beta \cdot \sin \alpha & \cos \alpha \cdot \cos \beta \end{bmatrix} \cdot \begin{bmatrix} R_{XD} \\ R_{YD} \\ R_{ZD} \end{bmatrix} \quad (5.5)$$

$$\begin{bmatrix} R_{xp} \\ R_{yp} \\ R_{zp} \end{bmatrix} = \begin{bmatrix} \cos \beta \cdot \cos \gamma & \cos \gamma \cdot \sin \beta \cdot \sin \alpha + \sin \gamma \cdot \cos \alpha & \sin \gamma \cdot \sin \alpha - \cos \gamma \sin \beta \cos \alpha \\ -\sin \gamma \cdot \cos \beta & \cos \alpha \cdot \cos \gamma - \sin \alpha \cdot \sin \beta \cdot \sin \gamma & \sin \gamma \cdot \sin \beta \cdot \cos \alpha + \cos \gamma \cdot \sin \alpha \\ \sin \beta & -\cos \beta \cdot \sin \alpha & \cos \alpha \cdot \cos \beta \end{bmatrix} \cdot \begin{bmatrix} R_{XP} \\ R_{YP} \\ R_{ZP} \end{bmatrix} \quad (5.6)$$

- **Step 3 - Transformation of the distal moments from the GRS into the anatomical axes**

The transformation from the global to the anatomical axes is implemented using the matrix R as shown in the previous step 2. The distal moments can then be transformed to the anatomical axes:

$$\begin{bmatrix} M_{xd} \\ M_{yd} \\ M_{zd} \end{bmatrix} = \begin{bmatrix} \cos \beta \cdot \cos \gamma & \cos \gamma \cdot \sin \beta \cdot \sin \alpha + \sin \gamma \cdot \cos \alpha & \sin \gamma \cdot \sin \alpha - \cos \gamma \sin \beta \cos \alpha \\ -\sin \gamma \cdot \cos \beta & \cos \alpha \cdot \cos \gamma - \sin \alpha \cdot \sin \beta \cdot \sin \gamma & \sin \gamma \cdot \sin \beta \cdot \cos \alpha + \cos \gamma \cdot \sin \alpha \\ \sin \beta & -\cos \beta \cdot \sin \alpha & \cos \alpha \cdot \cos \beta \end{bmatrix} \cdot \begin{bmatrix} M_{XD} \\ M_{YD} \\ M_{ZD} \end{bmatrix} \quad (5.7)$$

• **Step 4 - Calculation of proximal moments using Euler's equations**

The calculation of the proximal moments is implemented using Euler's equations:

$$I_x \cdot a_x + (I_z - I_y) \cdot \omega_y \cdot \omega_z = \Sigma M_x = R_{zd} \cdot l_d + R_{zp} \cdot l_p + M_{xp} - M_{xd} \quad (5.8)$$

$$I_y \cdot a_y + (I_x - I_z) \cdot \omega_x \cdot \omega_z = \Sigma M_y = M_{yp} - M_{yd} \quad (5.9)$$

$$I_z \cdot a_z + (I_y - I_x) \cdot \omega_x \cdot \omega_y = \Sigma M_z = -R_{xd} \cdot l_d - R_{xp} \cdot l_p + M_{zp} - M_{zd} \quad (5.10)$$

- where I_x, I_y, I_z = moments of inertia about x, y and z axes
- $\omega_x, \omega_y, \omega_z$ = components of angular velocity ω about x, y and z axes
- a_x, a_y, a_z = components of angular acceleration about x, y and z axes
- M_{xd}, M_{yd}, M_{zd} = previously transformed distal moments about x, y and z axes
- $R_{xd}, R_{xp}, R_{yd}, R_{yp}, R_{zd}, R_{zp}$ = previously transformed joint reaction forces about x, y and z axes
- l_p, l_d = distances from COM to proximal and distal joint

So, with the above components known, the only unknowns M_{xp}, M_{yp} and M_{zp} can now be solved.

The next stage of the analysis is to transform these proximal moments M_{xp}, M_{yp} and M_{zp} to the GRS so that inverse dynamics analysis can continue for the next segment. This transformation is accomplished by the anatomical to global transformation matrix ($[R]^T$ [18]) and yields a new set of distal moments M_{XD}, M_{YD} , and M_{ZD} for the next segment.

5.5 Muscle Recruitment Solver

5.5.1 Redundancy⁵³

Mathematical models of musculoskeletal systems typically consist of a linkage of rigid bodies and actuators to describe the muscles. A musculoskeletal system is usually redundant meaning that the number n_M of the muscles is greater than the number n_{DOF} of degrees of freedom of the system. As a consequence a desired motion can be achieved by an infinite number of activation patterns of the muscles. In nature the central nervous system (CNS) controls the activation of the muscles.

A solution to this problem can be obtained with the assumption that muscles are recruited according to an optimality criterion.

5.5.2 Optimisation

In the Anybody Modelling System the redundancy problem is treated using an optimisation technique, which is called the *min/max simplex recruitment solver*. This mathematical approach, embedded within Anybody Technology, assumes that minimising maximal muscle activity is physiologically reasonable as it corresponds to a minimum fatigue criterion⁵⁴. Such a strategy seems highly plausible for an organism evolved to survive in a competitive environment.

Mathematically optimisation means to find the minimum (or maximum) of an *objective function* G subject to equality and inequality constraint equations. The variables that the objective function depends on are called the *design variables*. In this case, the muscle forces and the muscular activities are the *design variables*.

Although the *min/max* criterion is highly non-linear, it is treated as a *linear programming* (LP) problem, which is solved using the *simplex* algorithm.

A linear programming problem is a problem in which it is asked to find the minimum (or maximum) of a linear *objective function* G , for example $G = 3x - 2y + z$, subject to one or more linear constraints of the form, for example, $x + y - 3z \leq 12$. The desired smallest value (or largest) value of the objective function is called *optimal value*, and a collection of values x, y, z , that gives the optimal value constitutes an *optimal solution*. The variables x, y, z are so called the *decision variables*.

In Anybody Technology, if G is the objective function then the mathematical definition of the problem becomes⁵⁵:

Minimise

$$G(f^{(M)}) \quad (5.11)$$

subject to $C \cdot f = d \quad (5.12)$

and

$$f_i^M \geq 0, i \in \{1, \dots, n^{(M)}\} \quad (5.13)$$

where $f^{(M)}$ is the vector of all muscles forces, $n^{(M)}$ is the number of muscles in the mechanism while $f_i^{(M)}$ represents the force generated by the i th muscle at a given moment. The non-negative constraints on the muscle forces (5.13), state that muscles can only pull, not push.

C is the coefficient matrix, which is derived from the equations necessary to relate the known loads of any particular system to the unknown muscle forces, f is the vector of all unknown forces, i.e. muscle as well as reaction forces, and d is a vector of all known applied loads and inertia forces arising due to the motion.

As it was previously stated, the Anybody system solves the optimisation problem using the *min/max simplex* algorithm, which distributes the collaborative muscle force in such a way that the maximum relative muscle activity is minimised:

$$G(f^{(M)}) = \max \left(\frac{f_i^{(M)}}{N_i} \right) \quad i \in \{1, \dots, n^{(M)}\} \quad (5.14)$$

The problem is solved by introducing an artificial variable, β , that functions as the upper bound on all muscle activities. The problem is then formulated as:

Minimise

β

$$\text{subject to} \quad \frac{f_i^{(M)}}{N_i} \leq \beta \quad i \in \{1, \dots, n^{(M)}\} \quad (5.15)$$

and

$$f_i^M \geq 0 \quad i \in \{1, \dots, n^{(M)}\} \quad (5.16)$$

where $f_i^{(M)}$ denotes the individual muscle force and N_i is the current muscle strength. Therefore

$\left(\frac{f_i^{(M)}}{N_i} \right)$ is the relative muscle activity. In this case is the maximum value of muscle activity for a

particular group of muscles that is optimised.

The way that the *simplex algorithm* is implemented, is presented using a simple example in Appendix A.

Figure 5.5 below, represents a flow chart of the iterative *min/max algorithm*. The algorithm stops if the solution is unique. It also terminates if there are no more muscles to be removed, if all muscles have been removed or the activity of the remaining muscles is very small. Another termination criterion is that the rank of the equality constraint equations is zero⁵³, which means that there is no work for the muscles to do.

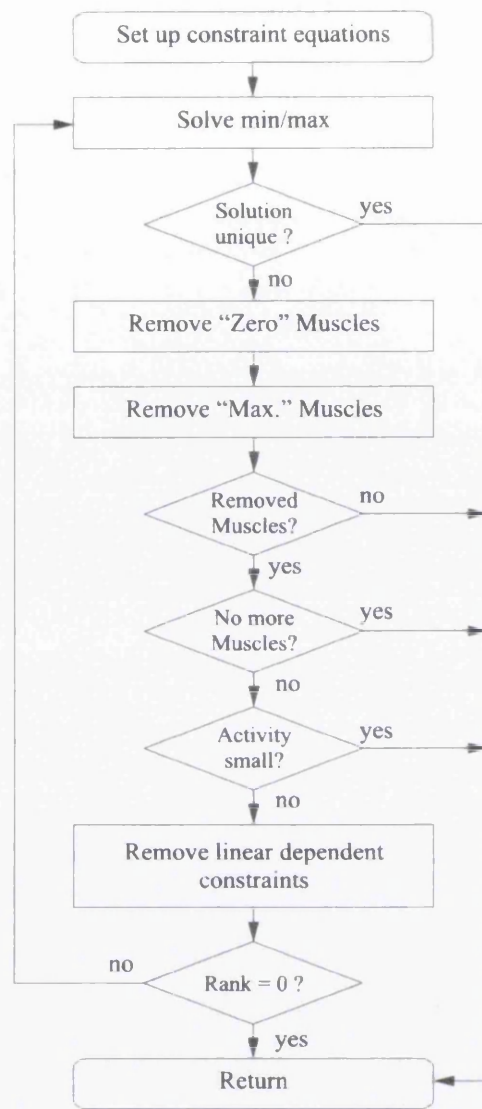


Figure 5.5 – Flow Chart of Min/Max Algorithm [53]

The overall process undertaken is outlined in the flow chart in Figure 5.6. The flow chart depicts the computational steps taken during analysis of the user-defined model. It shows how user-defined variables are used to carry out kinematic analysis and then inverse dynamic analysis of a particular model in order to calculate muscle forces.

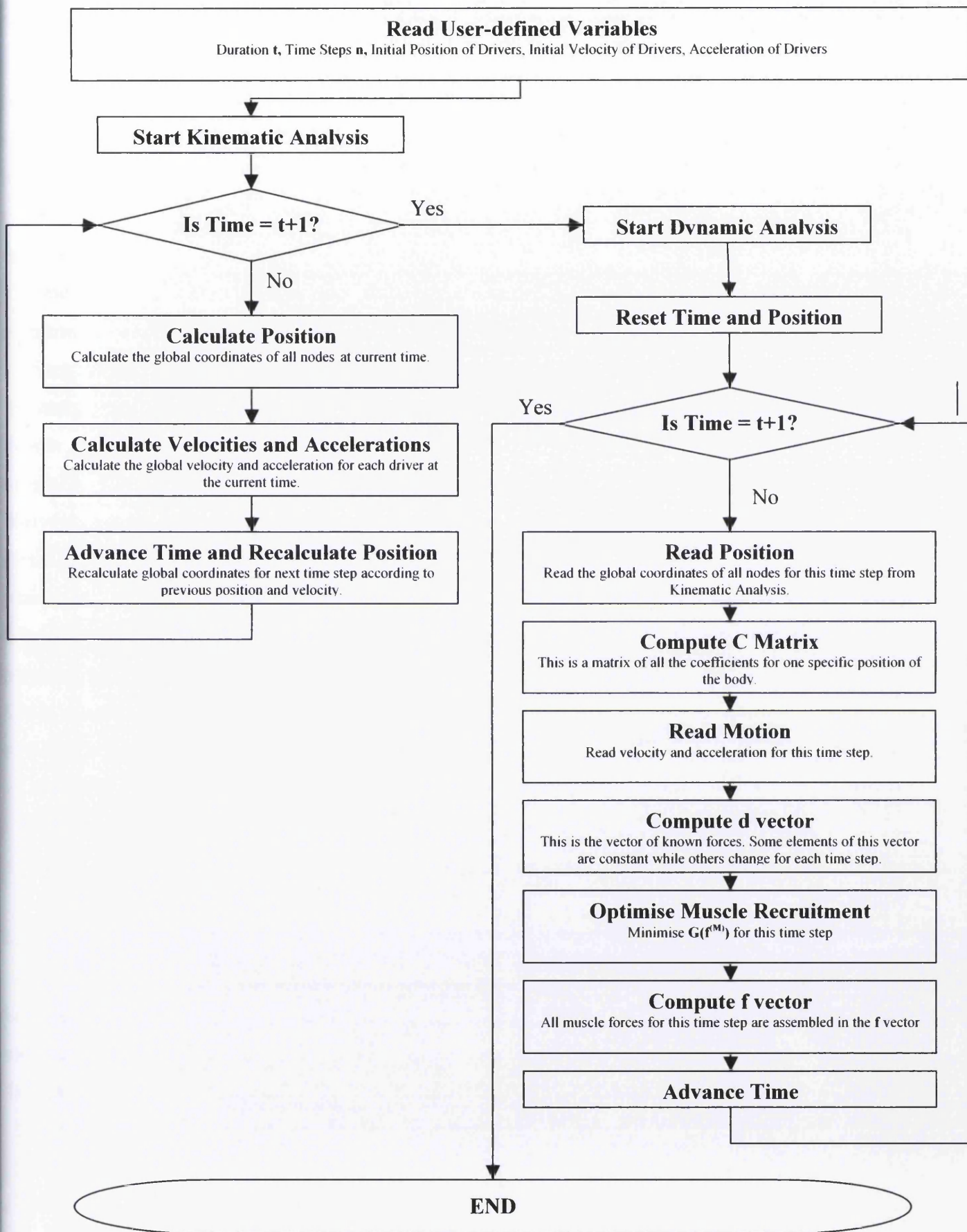


Figure 5.6 – The Process of Dynamic Analysis Within AnyBody

5.6 Summary

The model definition language of the Anybody modelling system is called AnyScript. AnyScript contains facilities for the definition of bones, their connections by joints, muscles, movements, constraints and exterior forces. It works on a set of predefined classes, does not let users define new classes and all reserved words in AnyScript begin with the prefix “Any”. An AnyScript model is roughly divided into two main sections; the body model, containing the definition of the mechanical system, and the study section containing lists of analyses and other operations to be performed on the model.

The mechanical system of the body model is constituted of the segments, joints, drivers, kinematic measures, forces and muscles.

The segments, which are usually bones, are basically a set of mass properties. A segment is originated in its center of mass, and the mass properties are defined by means of a mass and an inertia tensor.

Joints are used to connect segments and allow them to articulate with respect to each other. There are several types of joints.

Drivers create movement in a model. They are functions of time determining the position of a joint or the distance between two points or some other kinematic measure at any given time through the simulation period.

Kinematic measures are a way of describing dimensions in a kinematic model. They can measure translation, 3D rotations, the center of mass of a collection of rigid bodies, etc.

There are several types of forces (forces in joints, muscles, gravitational), but the forces that were presented in this chapter are the external forces. These forces can be applied to any kinematic measure or an assembly.

Muscles are the actuators of living bodies. Anybody contains three different muscle models and provides a facility that permits muscles to wrap over bones.

Inverse Solution is the analysis technique that is provided by Anybody Technology. This technique, with the accurate kinematic data of a desired motion, can predict the net forces and muscle moments that caused that motion. However, for the purpose of computing individual muscle forces, inverse dynamics is haunted by the so-called redundancy problem. Infinitely many different sets of muscle forces, of which the central nervous system (CNS) instantly chooses one, can therefore produce the resulted joint moments. This is due to the fact that there are more muscles than strictly necessary to drive most motions. For this purpose Anybody Technology introduced a “muscle recruitment solver” that solves the problem with the assumption that muscles are recruited according to an optimality criterion. In a few words AnyBody Technology assumes that minimizing maximal muscle activity is physiologically reasonable as it corresponds to a minimum fatigue criterion.

CHAPTER 6

DEVELOPMENT OF A TRIPLE INVERTED PENDULUM

6.1 Introduction

The previous Chapter 5 approached Anybody Technology theoretically describing the main characteristics of this system. The present Chapter 6 is intending to give a smooth introduction to the AnyBody modelling system from a practical point of view. For this purpose a triple inverted pendulum will be developed with a “step by step” progress. The reason of choosing this kind of model is because the human body in its simplest form can be assumed to be a triple inverted pendulum. The lower legs may represent the lower segment of the pendulum, the upper legs the middle segment, while the rest of the body the third segment.

The investigation starts with the presentation of the file structure and the Main.any file of the triple inverted pendulum in order to have an overview of the whole model. It continues with a “step by step” implementation of this model developing a single inverted pendulum firstly and then it goes further by adding one pendulum at a time. It has to be also noticed that the present model and the models that will follow are being developed using the version 1.0 of the Anybody Technology.

6.2 Structure

All the necessary AnyScript files for the implementation of this project will be contained in a Windows folder with name “Triple_Inverted_Pendulum”. This folder at its final stage will contain the “Anyfiles” shown in Figure 6.1.

Of course the structure in Figure 6.1 could be different if for example the anyfiles “Seg1, Seg2 and Seg3” were included in one folder with the name “Segments”. The same could happen for the anyfiles, “Driver1 and Driver2” which could be included in a “Driver” folder. All these possible changes do not make any difference except for a better-organized structure. And this is because as the model gets more complicated the number of files increases, as a result it becomes more difficult to find the necessary files.

It has to be also mentioned that the above structure has nothing to do with the file structure in the Anybody Software. The file Main.any is responsible for this structure as it is, let’s say, the “CPU” of the model. The Main.any file contains both the body model and the study section as it was mentioned in the previous chapter. All the anyfiles incorporated in the study can be grouped in different folders called “AnyFolders”. The final structure of the Main.any file will be in Figure 6.2 below.

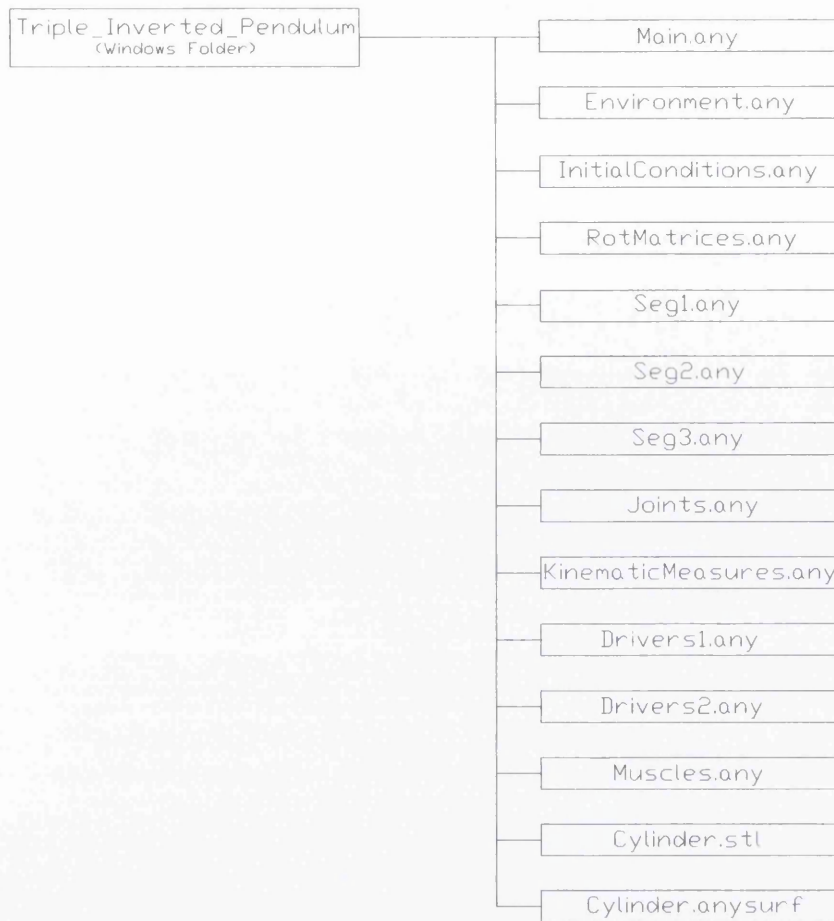


Figure 6.1 – Structure of Triple Inverted Pendulum Windows Folder

```

Main = {
  // The actual body model goes in this folder
  AnyFolder TripleInvPend = {
    // Global Reference Frame
    AnyFixedRefFrame GlobalRef = {
      #include "Environment.any"
    }; // Global reference frame
    //-----SEGMENTS-----
    AnyFolder Segs = {
      #include "Seg1.any"
      #include "Seg2.any"
      #include "Seg3.any"
    }; //Segments
  }
}

```

```

//-----INITIAL CONDITIONS-----

AnyFolder InitialConditions = {

    #include "InitialConditions.any"
    #include "RotMatrices.any"

}; //Initial Conditions

//-----JOINTS-----

AnyFolder Joints = {

    #include "Joints.any"

}; //Joints

//-----KINEMATIC MEASURES-----

AnyFolder KinematicMeasures = {

    #include "KinematicMeasures.any"

}; //Kinematic Measures

//-----DRIVERS-----

AnyFolder Drivers = {

    #include "Drivers1.any"
    // #include "Drivers2.any"

}; //Drivers

//-----MUSCLES-----

AnyFolder Muscles = {

    #include "Muscles.any"

}; //Muscles

}; // Triple Inverted Pendulum

// The study: Operations to be performed on the model
AnyBodyStudy MyStudy = {
    AnyFolder &Model = .TripleInvPend;
    RecruitmentSolver = MinMaxSimplex;
    Gravity = {0.0,-9.81,0.0};
};
}; // Main

```

Figure 6.2 – Structure of Main.any File

In the following sections, as mentioned before, the triple inverted pendulum will be developed “step by step” with a single inverted pendulum initially, followed by a second and a triple at the end. The function of each anyfile included, will be explained, and the contents of the Main.any file will be updated at each stage until it reaches its final form as shown before.

6.3 Single Inverted Pendulum

6.3.1 Defining Global Reference Frame

For the implementation of this stage, the first thing that has to be done is a mechanical drawing of the pendulum and its environment. The software package AutoCAD has been used and the final result is shown below (Figure 6.3):

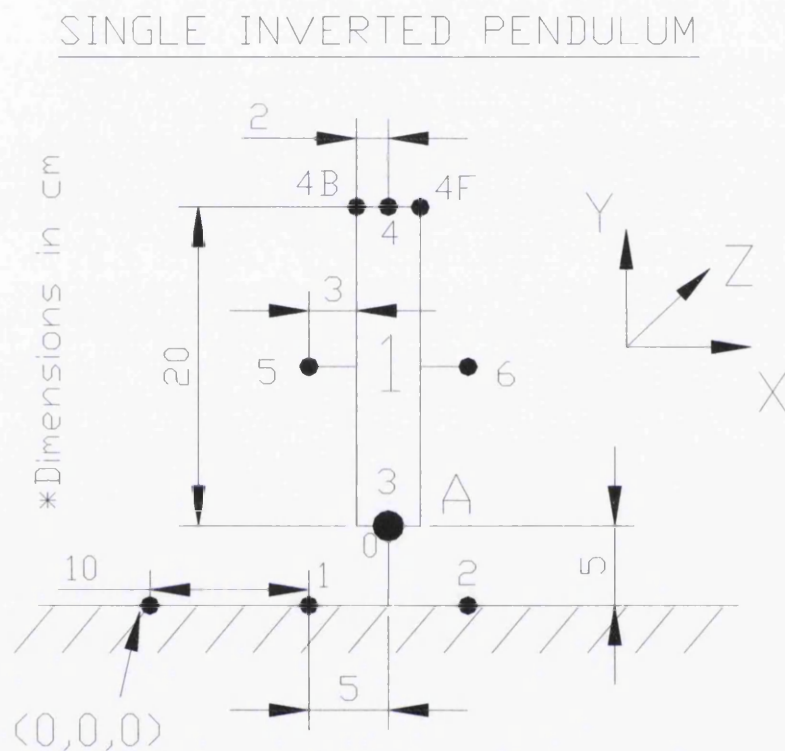


Figure 6.3 – Mechanical Drawing of Single Inverted Pendulum

In the above figure, the “ground nodes” 0,1 and 2 are part of the Environment.any file and their coordinates in meters are $(0.15,0.05,0)$, $(0.1,0,0)$ and $(0.2,0,0)$ respectively. Node 0 is used for the connection of the pendulum (segment1) with the ground via its “local node”, node 3. The other two ground nodes 1 and 2 will be connected to the local nodes of segment1, 5 and 6 respectively, with the muscles. The term “local nodes” is used because these nodes refer to the coordinate system of the segment they belong to. For example the local coordinates in meters of nodes 3 and 4 are $(0,-0.1,0)$ and $(0,0.1,0)$ respectively.

The next thing that has to be done, is the construction of the environment of the pendulum defining the Environment.any file. This file includes the three nodes as mentioned before and it is included in the statement *AnyFixedRefFrame* which describes a fixed reference frame which can be used for modeling

the “ground” of a mechanical system. The following Table 6.1 shows the Main.any and the Environment.any files, while Figure 6.4 shows the present model’s view.

<u>Main.any File</u>	<u>Environment.any File</u>
<pre>// Todo: Write a small description of your model here Main = { // The actual body model goes in this folder AnyFolder TripleInvPend = { // Global Reference Frame AnyFixedRefFrame GlobalRef = { #include "Environment.any" }; // Global reference frame }; // Triple Inverted Pendulum // The study: Operations to be performed on the model AnyBodyStudy MyStudy = { AnyFolder &Model = .TripleInvPend; RecruitmentSolver = MinMaxSimplex; Gravity = {0.0,-9.81,0.0}; }; }; // Main</pre>	<pre>AnyDrawRefFrame DrwGlobalRef = { ScaleXYZ = {0.5,0.5,0.5}; }; AnyRefNode Ground0 = { sRel = {0.15,0.05,0}; AnyDrawNode DrwNode0 = { RGB = {82/256,85/256,111/256}; //Blue ScaleXYZ = {0.01,0.01,0.01}; }; }; AnyRefNode Ground1 = { sRel = {0.1,0,0}; AnyDrawNode DrwNode1 = { RGB = {82/256,85/256,111/256}; //Blue ScaleXYZ = {0.02,0.02,0.02}; }; }; AnyRefNode Ground2 = { sRel = {0.2,0,0}; AnyDrawNode DrwNode2 = { RGB = {82/256,85/256,111/256}; //Blue ScaleXYZ = {0.02,0.02,0.02}; }; };</pre>

Table 6.1 – Main.any/Enironment.any Files

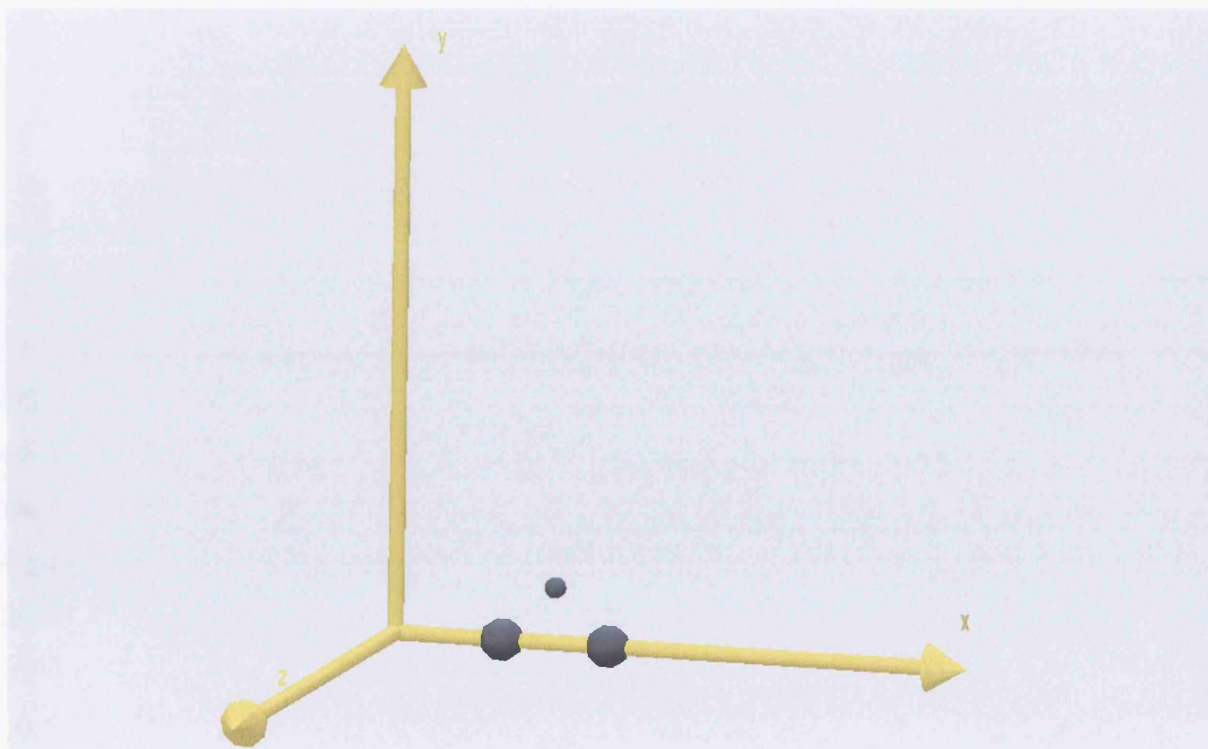


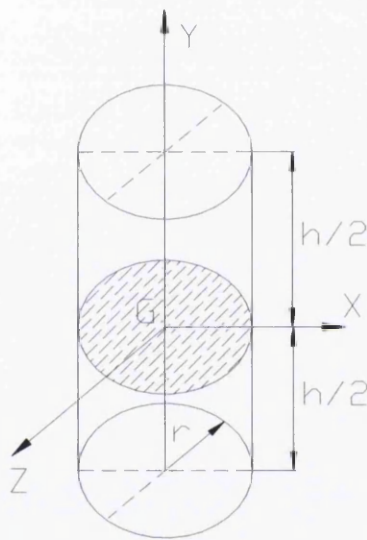
Figure 6.4 – Environment’s Model View

As it is shown in the Environment.any file, a node can be defined with the statement *AnyRefNode* followed by a name that has to be given to this node. The local or global coordinates of each node are included in the *sRel* statement, while these nodes are visible with the usage of the *AnyDrawNode* statement. It is possible to colourize a node (*RGB*) and scale to it (*ScaleXYZ*) as well. The global reference frame becomes visible with the *AnyDrawRefFrame* statement.

6.3.2 Defining Segments

As soon as the global reference frame and the ground nodes are defined the next step is the definition of the single inverted pendulum. This pendulum (*Seg1*) is a circular cylinder with the following geometrical properties (Figure 6.5):

Circular Cylinder



Centre of mass (G): $G = \frac{h}{2}$

Mass: $m = 2\text{ kg}$

Moment of Inertia (G)

$h = 0.2\text{ m}, r = 0.02\text{ m}$

$I_{x_G} = \frac{1}{12} \cdot m \cdot (3r^2 + h^2) = 0.0069\text{ kgm}^2$

$I_{y_G} = \frac{1}{2} \cdot m \cdot r^2 = 0.0004\text{ kgm}^2$

$I_{z_G} = I_{x_G}$

Figure 6.5 – Geometrical Properties of Pendulum

A segment or rigid body is defined in Anybody Technology with the statement *AnySeg* followed by a name for this segment. *AnySeg* may include the following statements: *Mass* (obligatory) for the total mass of the segment, *Jii* (obligatory) for the principal moments of inertia in a vector $\{I_{xx}, I_{yy}, I_{zz}\}$, *r* for the position vector of the origin with respect to the global reference system, *AnyRefNode* for the definition of the nodes and *AnyDrawSeg* for the visualization of the segment. Of course there are some more statements that may be included in *AnySeg* but this is going to be investigated later in this chapter.

In this particular case the name of the inverted pendulum is *Seg1* where: $r = \{0.15, 0.15, 0\}$, $\text{Mass} = 2$; $Jii = \{0.0069, 0.0004, 0.0069\}$. It contains six nodes with names *Node3*, *Node4*, *Node5*, *Node6*, *Node4B*, *Node4F* with local coordinates in meters $\{0, -0.1, 0\}$, $\{0, 0.1, 0\}$, $\{-0.05, 0, 0\}$, $\{0.05, 0, 0\}$, $\{-0.02, 0.1, 0\}$ and $\{0.02, 0.1, 0\}$ respectively.

Table 6.2 below shows the *Seg1.any* file with the updated *Main.any* file followed by the model's view (Figure 6.6).

Main.any File

```
// Todo: Write a small description of your
model here

Main = {

// The actual body model goes in this folder
AnyFolder TripleInvPend = {

// Global Reference Frame
AnyFixedRefFrame GlobalRef = {

#include "Environment.any"

}; // Global reference frame

//-----SEGMENTS -----

AnyFolder Segs = {

#include "Seg1.any"

}; // Segments

}; // Triple Inverted Pendulum

// The study: Operations to be performed on
the model
AnyBodyStudy MyStudy = {
AnyFolder &Model = .TripleInvPend;
RecruitmentSolver = MinMaxSimplex;
Gravity = { 0.0,-9.81,0.0};
};

}; // Main
```

Seg1.any File

```
AnySeg Seg1 = {
r = {0.15,0.15,0};
Mass = 2;
Jii = {0.0069,0.0004,0.0069};
AnyDrawSeg DrwSeg = {};
AnyRefNode Node3 = {
sRel = {0,-0.1,0};
};
AnyRefNode Node4 = {
sRel = {0,0.1,0};
};
AnyRefNode Node5 = {
sRel = {-0.05,0,0};
};
AnyRefNode Node6 = {
sRel = {0.05,0,0};
};
AnyRefNode Node4B = {
sRel = {-0.02,0.1,0};
};
AnyRefNode Node4F = {
sRel = {0.02,0.1,0};
};
}; // Seg1
```

Table 6.2 – Main.any/Seg1.any Files

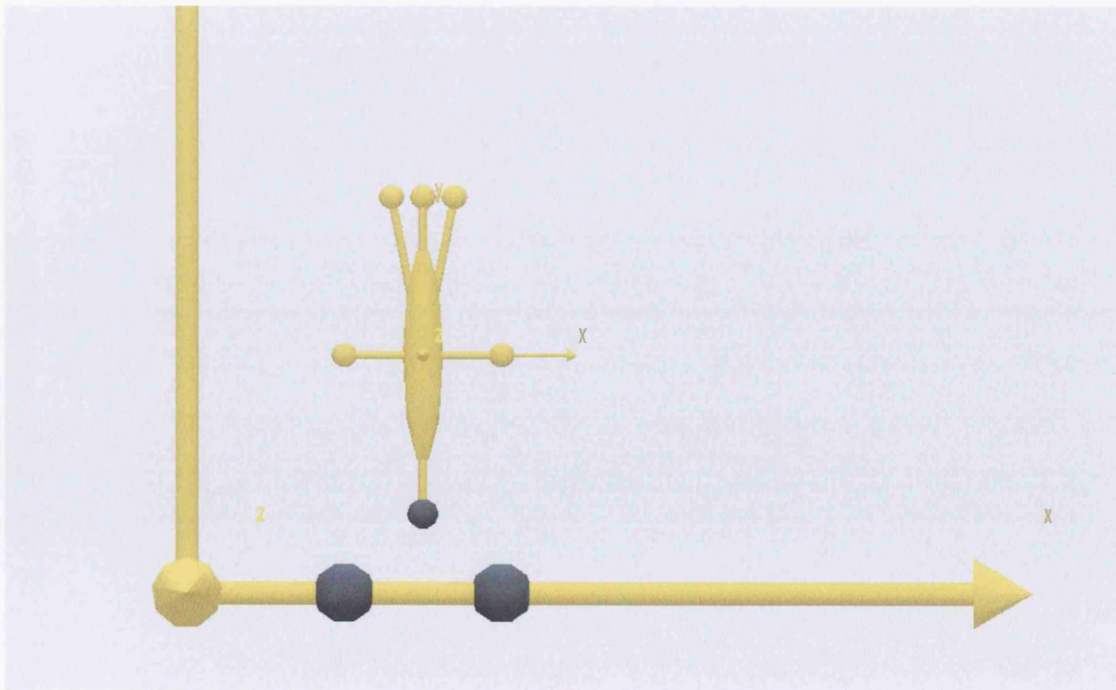


Figure 6.6 – Segment1's Model View

6.3.3 Connecting Segments By Joints

As it is seen in the above figure, although *Seg1* seems to be connected with the ground node *Ground0*, physically it's not yet connected. It is positioned at the correct position because of the statement $r = \{0.15, 0.15, 0\}$. Thus, a joint is needed and the Anybody modelling system provides a variety of joints. In this model, a revolute joint is going to be used which is an ideal hinge that only allows rotation about one axis which will be the z-axis in this particular case. Such a joint is defined by the statement *AnyRevoluteJoint*, followed by the name of the joint, and includes the axis of rotation (*Axis*) and the nodes that will be connected to each other. At the present model, the axis of rotation is the z-axis and the nodes are the *Ground0* of the global reference frame and the *Node3* of *Seg1*. Table 6.3 below, presents the *Joints.any* file (with the joint *A* that connects *Seg1* with ground) and the updated *Main.any* file as well.

<u>Main.any File</u>	<u>Joints.any File</u>
<pre> Main = { // The actual body model goes in this folder AnyFolder TripleInvPend = { // Global Reference Frame AnyFixedRefFrame GlobalRef = { #include "Environment.any" }; // Global reference frame //-----SEGMENTS ----- AnyFolder Segs = { #include "Seg1.any" }; //Segments //-----JOINTS----- AnyFolder Joints = { #include "Joints.any" }; //Joints }; // Triple Inverted Pendulum // The study: Operations to be performed on the model AnyBodyStudy MyStudy = { AnyFolder &Model = .TripleInvPend; RecruitmentSolver = MinMaxSimplex; Gravity = { 0.0,-9.81,0.0}; }; }; // Main </pre>	<pre> AnyRevoluteJoint A = { Axis=z; AnyRefNode &Node0 = Main.TripleInvPend.GlobalRef.Ground0; AnyRefNode &Node3 = Main.TripleInvPend.Segs.Seg1.Node3; }; //Joint A </pre>

Table 6.3 – Main.any/Joints.any Files

6.3.4 Definition of Movement

Since the body is connected to the ground it is now possible to give a motion to it. Anybody Technology provides a variety of drivers. A simple driver will be used that provides motion with constant acceleration. Table 6.4 below, presents the Drivers1.any file with the driver implementation and the updated Main.any file. The statement for this driver is *AnyKinEqSimpleDriver* followed by the name of this driver (*Seg1Motion*) and consists of the type of joint (*AnyRevoluteJoint*) and the joint that drives (*A*), the driver's position (*DriverPos*) in radians, the driver's velocity (*DriverVel*) in rad/sec, the acceleration (*DriverAcc*) in rad/sec² and the *Reaction.Type* which is 1 when motion is provided without muscles and 0 when muscles work.

<u>Main.any File</u>	<u>Drivers1.any File</u>
<pre> Main = { // The actual body model goes in this folder AnyFolder TripleInvPend = { // Global Reference Frame AnyFixedRefFrame GlobalRef = { #include "Environment.any" }; // Global reference frame }; //-----SEGMENTS ----- AnyFolder Segs = { #include "Seg1.any" }; //Segments }; //-----JOINTS----- AnyFolder Joints = { #include "Joints.any" }; //Joints }; //-----DRIVERS----- AnyFolder Drivers = { #include "Drivers1.any" }; //Drivers }; // Triple Inverted Pendulum // The study: Operations to be performed on the model AnyBodyStudy MyStudy = { AnyFolder &Model = .TripleInvPend; RecruitmentSolver = MinMaxSimplex; Gravity = { 0.0,-9.81,0.0}; }; }; // Main </pre>	<pre> //-----SEGMENT1 DRIVER----- AnyKinEqSimpleDriver Seg1Motion = { AnyRevoluteJoint &Jnt = ..Joints.A; DriverPos = {0 * pi/180}; //Rotation in degrees DriverVel = {0}; // rad/sec DriverAcc = {0}; Reaction.Type = {1}; }; //Seg1 Driver </pre>

Table 6.4 – Main.any/Drivers1.any Files

For instance if the *DriverPos* is 45 degrees and the *DriverVel* is -1 rad/sec then a sequence of time steps of segment1's motion is as follows (Figure 6.7). Note that the analysis has 101 time steps corresponding

to a division of the total analysis time into 100 equal pieces. The total time span simulated in the analysis is 1 sec.

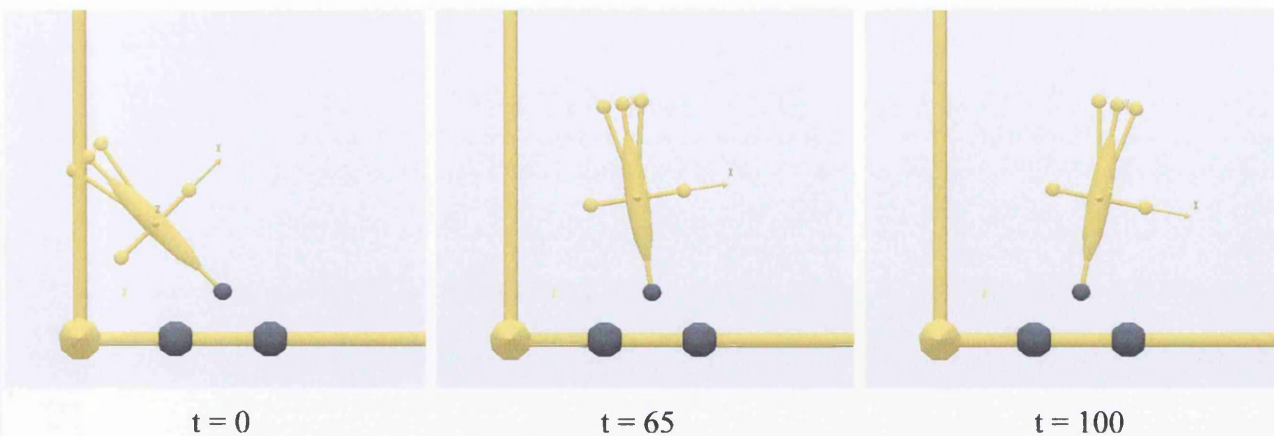


Figure 6.7 – Motion of Single Inverted Pendulum

6.3.5 Definition of Muscles

The development of the single inverted pendulum will continue with the implementation of motion given by muscles. The muscles that will be used will be muscles that connect two or more so-called “via points” (*AnyViaPointMuscle*) and contain the simplest possible model (*AnyMuscleModel*) having constant strength and no passive element properties. In this particular case two muscles will drive the model as the muscle cannot both pull and push but only pull. The first muscle with the name *Seg1Back* connects nodes *Ground1* and *Node5* and the second muscle with name *Seg1Front* connects the nodes *Ground2* and *Node6*. Both of these muscles have constant strength $F_0 = 100N$. Table 6.5 below includes the Muscles.any file with the muscles implementation and the updated Main.any file.

<u>Main.any File</u>	<u>Muscles.any File</u>
<pre> Main = { // The actual body model goes in this folder AnyFolder TripleInvPend = { // Global Reference Frame AnyFixedRefFrame GlobalRef = { #include "Environment.any" }; // Global reference frame //-----SEGMENTS ----- AnyFolder Segs = { #include "Seg1.any" }; //Segments </pre>	<pre> //---Simple muscle model with constant strength 100N----- AnyMuscleModel MusMd11 = { FO = 100; }; //Muscle Strength //-----SEGMENT1 BACK----- AnyViaPointMuscle Seg1Back = { AnyMuscleModel &MusMd11 = .MusMd11; AnyRefNode &Node1 = ..GlobalRef.Ground1; AnyRefNode &Node5 = ..Segs.Seg1.Node5; AnyDrawMuscle DrwMus = {}; }; //Seg1_Back </pre>

```

//-----JOINTS-----
AnyFolder Joints = {
    #include "Joints.any"
}; // Joints

//-----DRIVERS-----
AnyFolder Drivers = {
    #include "Drivers1.any"
}; // Drivers

//-----MUSCLES-----
AnyFolder Muscles = {
    #include "Muscles.any"
}; // Muscles

}; // Triple Inverted Pendulum

// The study: Operations to be performed on
the model
AnyBodyStudy MyStudy = {
    AnyFolder &Model = .TripleInvPend;
    RecruitmentSolver = MinMaxSimplex;
    Gravity = { 0.0, -9.81, 0.0 };
};

}; // Main

```

```

//-----SEGMENT1 FRONT-----
AnyViaPointMuscle Seg1Front = {
    AnyMuscleModel &MusMdl1 = .MusMdl1;
    AnyRefNode &Node2 = ..GlobalRef.Ground2;
    AnyRefNode &Node6 = ..Segs.Seg1.Node6;
    AnyDrawMuscle DrwMus = {};
}; // Seg1_Front

```

Table 6.5 - Main.any/Muscles.any Files

Now, if the *DriverPos* is 40 degrees, the *DriverVel* is -1 rad/sec and the *Reaction.Type* is 0 then a sequence of time steps of segment1's muscle driven motion is as follows (Figure 6.8):

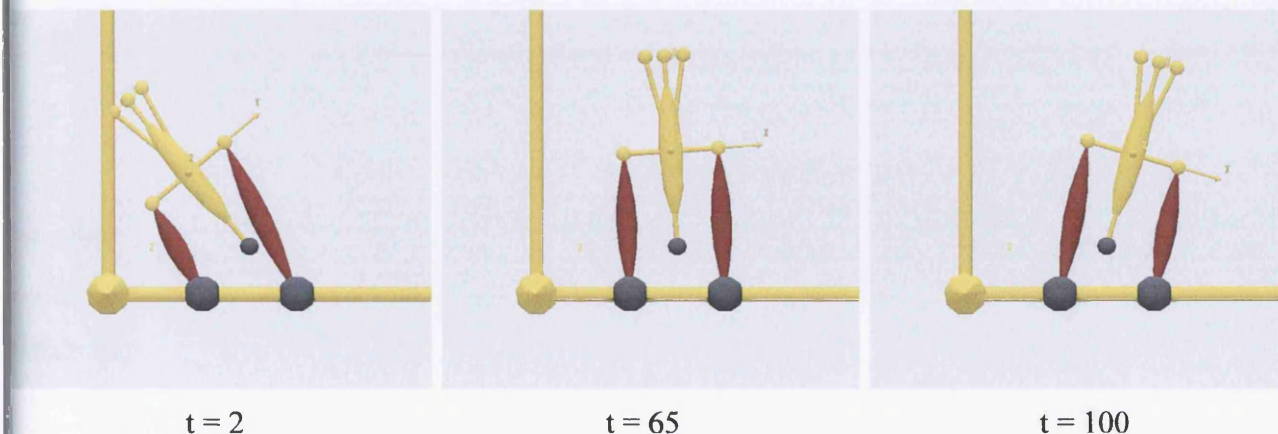


Figure 6.8 – Motion of Muscle Driven Single Inverted Pendulum

As mentioned before, the constant strength of the muscles is $F_0=100N$. If F_0 is changed to $F_0 = 50N$ then a sequence of time steps of the new motion is shown below (Figure 6.9):

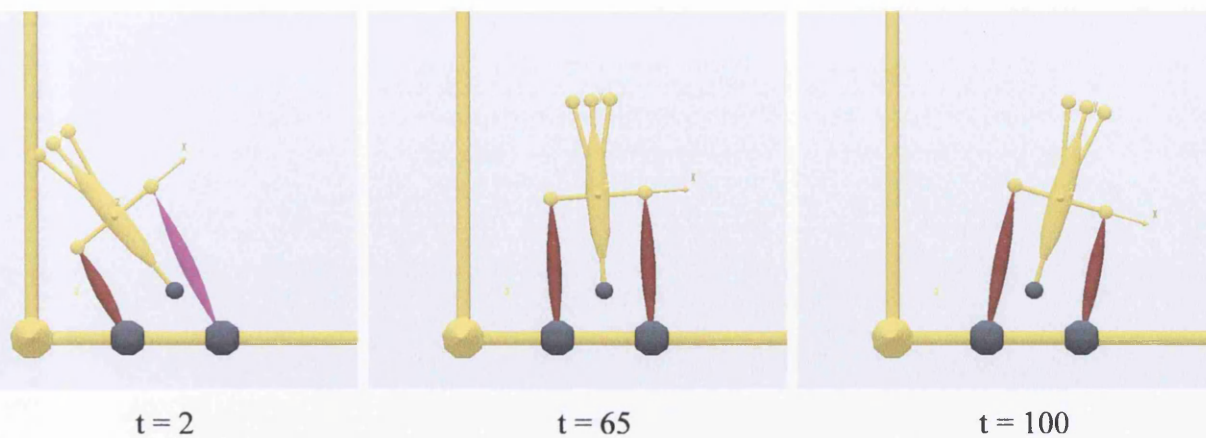


Figure 6.9 – Motion of Muscle Driven Single Inverted Pendulum ($F_0 = 50N$)

If now, the image of Figure 6.8 at time step 2 is compared with the corresponding one of Figure 6.9, what is observed is a difference at the colour of the muscle. This happens because the constant muscle strength F_0 was changed from $100N$ to $50N$ while the minimum force (F_m) required in the muscle unit of *Seg1Front* muscle for both motions is $54N$ (Figure 6.10).

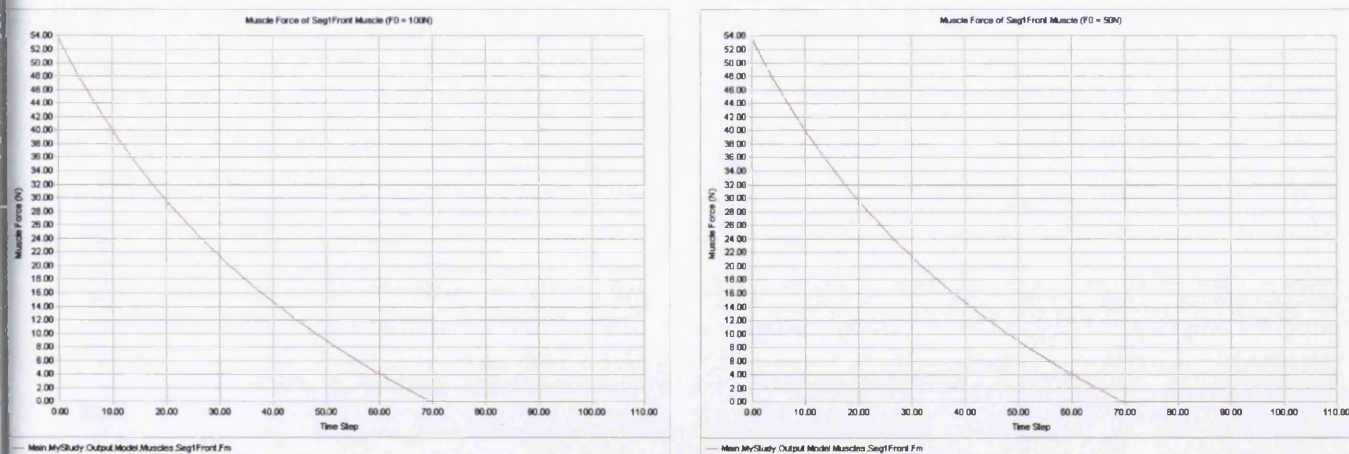


Figure 6.10 – F_m of *Seg1Front* Muscle

In a few words the *Seg1Front* muscle with $F_0 = 50N$ was over activated at the first time steps of the motion and this is shown in Figure 6.11. At time step 0, the muscle with $F_0 = 100N$ had 54% activity while the muscle with $F_0 = 50N$ had 108% activity.

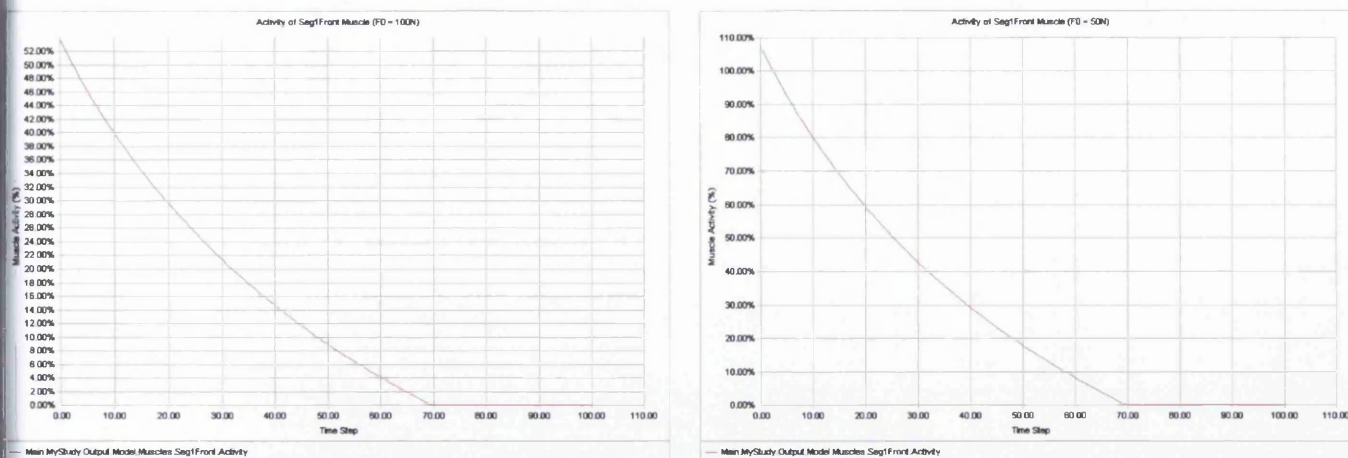


Figure 6.11 – *Activity of Seg1Front Muscle*

6.3.6 Adding Real Geometries

Finally, this section will end with the addition of a real geometric model of the cylinder that represents *Seg1*. All that is needed, is a file with a 3-D graphical representation of the cylinder. The format of the file must be STL ascii. STL is a very simple graphical file format that basically contains triangles. Virtually any CAD system can save geometry in STL format. In this particular case AutoCAD has been used, enhanced with the AutoCAD Plug-Ins “STL Import & Export” provided by the software development company “Floating Point Solutions Private Limited”. And that is necessary because AutoCAD’s own solution has some limitations.

“STL Import” for AutoCAD is a STL file import plug-in for AutoCAD, as AutoCAD does not provide such a facility at all. It gives AutoCAD the ability to import geometric data from ascii as well as binary stereolithography (STL) files.

“STL Export” for AutoCAD, gives AutoCAD the ability to export geometric data to ascii as well as binary stereolithography (STL) files. AutoCAD already has a “STLOUT” command which exports STL files, but unfortunately it exports only 3D Solids, and only one at a time. It does not export 3D Face, Polygon Mesh and Polyface Mesh entities which can very well be a part of a STL model. In addition, the solid must lie in the positive *XYZ* octant otherwise the “STLOUT” command is cancelled.

The first thing that has to be done is the drawing of a solid version of a cylinder in the AutoCAD environment. The next step is the “export” of this file in ascii form (saved as *Cylinder*) in the *Triple_Inverted_Pendulum Windows* folder, and with the inclusion of the *AnyDrawSTL* statement in *Seg1.any* file (Figure 6.12) the visualisation of the geometric model is done. It has to be mentioned that the position of the STL cylinder in the segment, depends on the position of the centroid of the cylinder with respect to the origin (0,0,0) in the AutoCAD drawing before the conversion (Figures 6.13,6.14).

Figure 6.13 shows the cylinder with the centroid coincident to the origin in the AutoCAD drawing while

Figure 6.14 the same cylinder with the centroid above the origin in the drawing.

Except that, it is possible for the STL cylinder to be attached on a node including the *AnyDrawSTL* statement in the *AnyRefNode* statement.

```


Seg1.any File



```
AnySeg Seg1 = {
 r = {0.15,0.15,0};
 Mass = 2;
 Jii = {0.0069,0.0004,0.0069};
 AnyDrawSeg DrwSeg = {};
 AnyRefNode Node3 = {
 sRel = {0,-0.1,0};
 };
 AnyRefNode Node4 = {
 sRel = {0,0.1,0};
 };
 AnyRefNode Node5 = {
 sRel = {-0.05,0,0};
 };
 AnyRefNode Node6 = {
 sRel = {0.05,0,0};
 };
 AnyRefNode Node4B = {
 sRel = {-0.02,0.1,0};
 };
 AnyRefNode Node4F = {
 sRel = {0.02,0.1,0};
 };
 AnyDrawSTL DrwSTL = {
 FileName = "Cylinder.stl";
 ScaleXYZ = {0.001,0.001,0.001};
 };
 /*
 AnySurfSTL Cylinder = {
 FileName = "Cylinder.stl";
 };
 */
 AnyDrawSurf DrwSurf = {
 FileName = "Cylinder.anysurf";
 ScaleXYZ = {0.001,0.001,0.001};
 };
 */
```


```

Figure 6.12 – Seg1.any File with STL/AnySurf Files

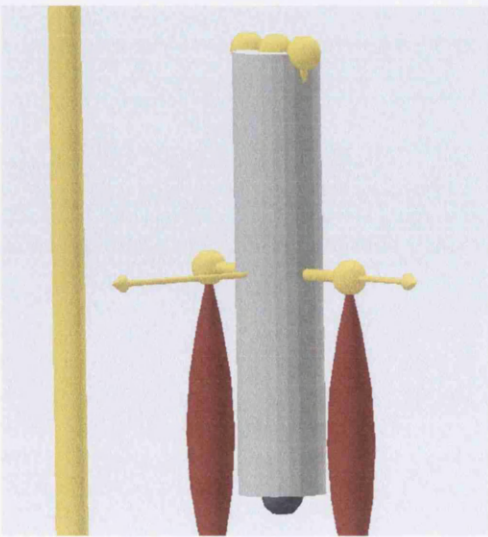
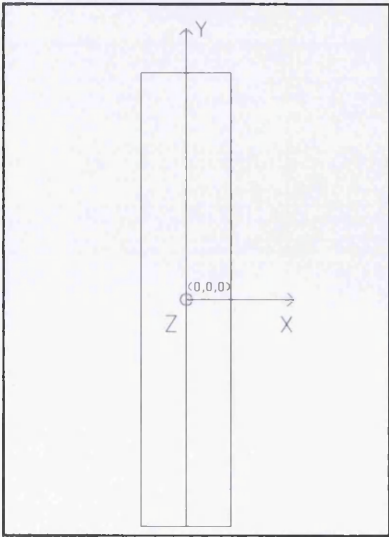


Figure 6.13 – AutoCAD/STL Model

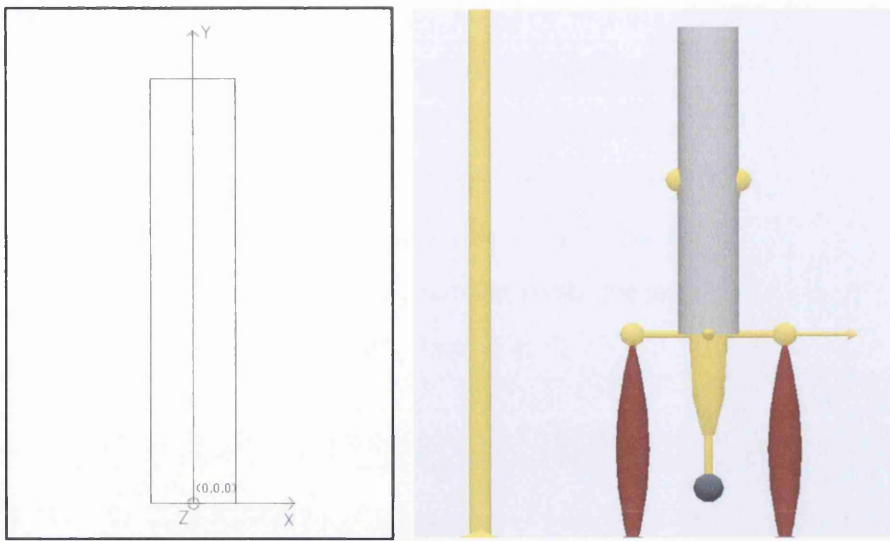


Figure 6.14 – AutoCAD/STL Model

Anybody Technology provides one more statement that can be used for adding real geometrical models instead of the *AnyDrawSTL*. That is the *AnyDrawSurf* statement (Figure 6.12) which can be used after the conversion of the STL format of the model to a new AnySurf format with the statement *AnySurfSTL*. The result of such a conversion is shown below (Figure 6.15) with the AnySurf model to be smoother than the STL one.

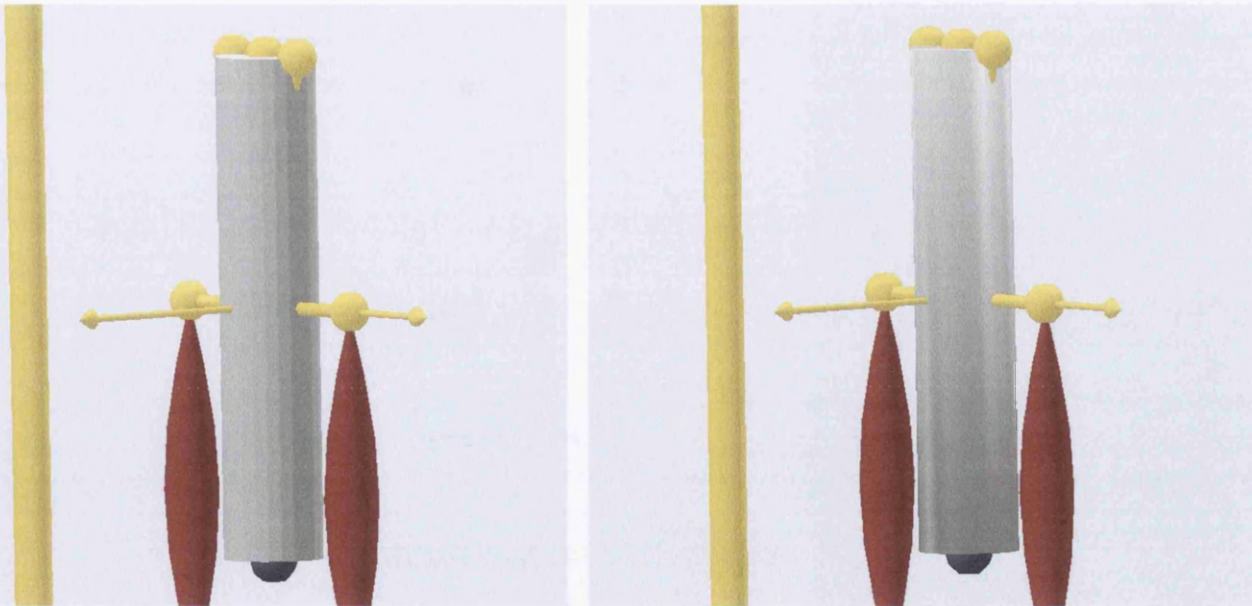


Figure 6.15 – STL/AnySurf Format

6.4 Double Inverted Pendulum

In this section the development of a double inverted pendulum will be investigated. This study is an extension of the previous one and the only difference is the addition of one more segment (segment2) upon segment1 (*Seg1*). The new segment2 (*Seg2*) has the same geometrical properties as *Seg1* and the same nodes. These nodes have the same local coordinates but they are numbered in a different way.

Figure 6.16 below represents the double inverted pendulum with the new *Seg2* and its nodes. *Seg1* and *Seg2* are connected to each other with the *B* joint (Node 4 & 7).

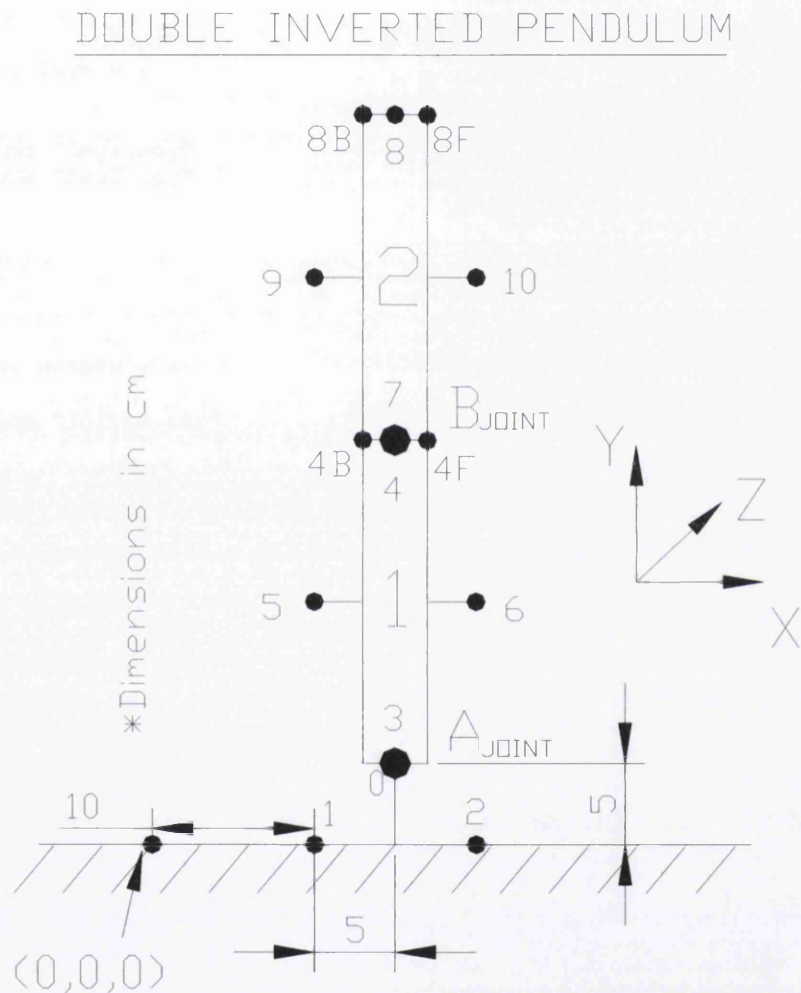


Figure 6.16 - Mechanical Drawing of Double Inverted Pendulum

Table 6.6 below includes the updated Main.any file, the new Seg2.any file and the updated Joints.any, Drivers1.any and Muscles.any files. It has to be noticed that the *DriverPos* in the new driver is -90 degrees while *DriverVel* is 3 rad/sec.

Main.any File

```
Main = {
// The actual body model goes in this folder
AnyFolder TripleInvPend = {

// Global Reference Frame
AnyFixedRefFrame GlobalRef = {

#include "Environment.any"

}; // Global reference frame

//-----SEGMENTS-----

AnyFolder Segs = {

#include "Seg1.any"
#include "Seg2.any"

}; //Segments

//-----JOINTS-----

AnyFolder Joints = {

#include "Joints.any"

}; //Joints

//-----DRIVERS-----

AnyFolder Drivers = {

#include "Drivers1.any"

}; //Drivers

//-----MUSCLES-----

AnyFolder Muscles = {

#include "Muscles.any"

}; //Muscles

}; // Triple Inverted Pendulum

// The study: Operations to be performed on
the model
AnyBodyStudy MyStudy = {
AnyFolder &Model = .TripleInvPend;
RecruitmentSolver = MinMaxSimplex;
Gravity = { 0.0,-9.81,0.0};

};

}; // Main
```

Seg2.any File

```
AnySeg Seg2 = {
r = {0.15,0.35,0};
Mass = 2;
Jii = {0.0069,0.0004,0.0069};
AnyDrawSeg DrwSeg = {};
AnyRefNode Node7 = {
sRel = {0,-0.1,0};
};
AnyRefNode Node8 = {
sRel = {0,0.1,0};
};
AnyRefNode Node9 = {
sRel = {-0.05,0,0};
};
AnyRefNode Node10 = {
sRel = {0.05,0,0};
};
AnyRefNode Node8B = {
sRel = {-0.02,0.1,0};
};
AnyRefNode Node8F = {
sRel = {0.02,0.1,0};
};
AnyDrawSurf DrwSurf = {
FileName = "Cylinder.anysurf";
ScaleXYZ = {0.001,0.001,0.001};
};
}; //Seg2
```

Joints.any File

```
AnyRevoluteJoint A = {
Axis=z;
AnyRefNode &Node0
=Main.TripleInvPend.GlobalRef.Ground0;
AnyRefNode &Node3
=Main.TripleInvPend.Segs.Seg1.Node3;
}; //Joint A

AnyRevoluteJoint B = {
Axis=z;
AnyRefNode &Node4
=Main.TripleInvPend.Segs.Seg1.Node4;
AnyRefNode &Node7
=Main.TripleInvPend.Segs.Seg2.Node7;
}; //Joint B
```

Drivers1.any File

```
//-----SEGMENT1 DRIVER-----
AnyKinEqSimpleDriver Seg1Motion = {
AnyRevoluteJoint &Jnt = ..Joints.A;
DriverPos = {40 * pi/180}; //Rotation in
degrees
DriverVel = {-1}; // rad/sec
DriverAcc = {0};
Reaction.Type = {0};
}; //Seg1 Driver
```

Muscles.any File

```
//-----Simple muscle model with constant strength 300N-----
```

```
AnyMuscleModel MusMdl1 = {  
    FO = 100;  
}; //Muscle Strength
```

```
//-----SEGMENT1 BACK-----
```

```
AnyViaPointMuscle Seg1Back = {  
    AnyMuscleModel &MusMdl1 = .MusMdl1;  
    AnyRefNode &Node1 = ..GlobalRef.Ground1;  
    AnyRefNode &Node5 = ..Segs.Seg1.Node5;  
    AnyDrawMuscle DrwMus = {};  
}; //Seg1_Back
```

```
//-----SEGMENT1 FRONT-----
```

```
AnyViaPointMuscle Seg1Front = {  
    AnyMuscleModel &MusMdl1 = .MusMdl1;  
    AnyRefNode &Node2 = ..GlobalRef.Ground2;  
    AnyRefNode &Node6 = ..Segs.Seg1.Node6;  
    AnyDrawMuscle DrwMus = {};  
}; //Seg1_Front
```

```
//-----SEGMENT2 BACK-----
```

```
AnyViaPointMuscle Seg2Back = {  
    AnyMuscleModel &MusMdl1 = .MusMdl1;  
    AnyRefNode &Node5 = ..Segs.Seg1.Node5;  
    AnyRefNode &Node4B = ..Segs.Seg1.Node4B;  
    AnyRefNode &Node9 = ..Segs.Seg2.Node9;  
    AnyDrawMuscle DrwMus = {};  
}; //Seg2_Back
```

```
//-----SEGMENT2 FRONT-----
```

```
AnyViaPointMuscle Seg2Front = {  
    AnyMuscleModel &MusMdl1 = .MusMdl1;  
    AnyRefNode &Node6 = ..Segs.Seg1.Node6;  
    AnyRefNode &Node4F = ..Segs.Seg1.Node4F;  
    AnyRefNode &Node10 = ..Segs.Seg2.Node10;  
    AnyDrawMuscle DrwMus = {};  
}; //Seg2_Front
```

```
//-----SEGMENT2 DRIVER-----
```

```
AnyKinEqSimpleDriver Seg2Motion = {  
    AnyRevoluteJoint &Jnt = ..Joints.B;  
    DriverPos = {-90 * pi/180}; //Rotation in degrees  
    DriverVel = {3}; // rad/sec  
    DriverAcc = {0};  
    Reaction.Type = {0};  
}; //Seg2 Driver
```

Table 6.6 – Double Inverted Pendulum Anyfiles

Figure 6.17 below, shows a sequence of images of double inverted pendulum's motion.

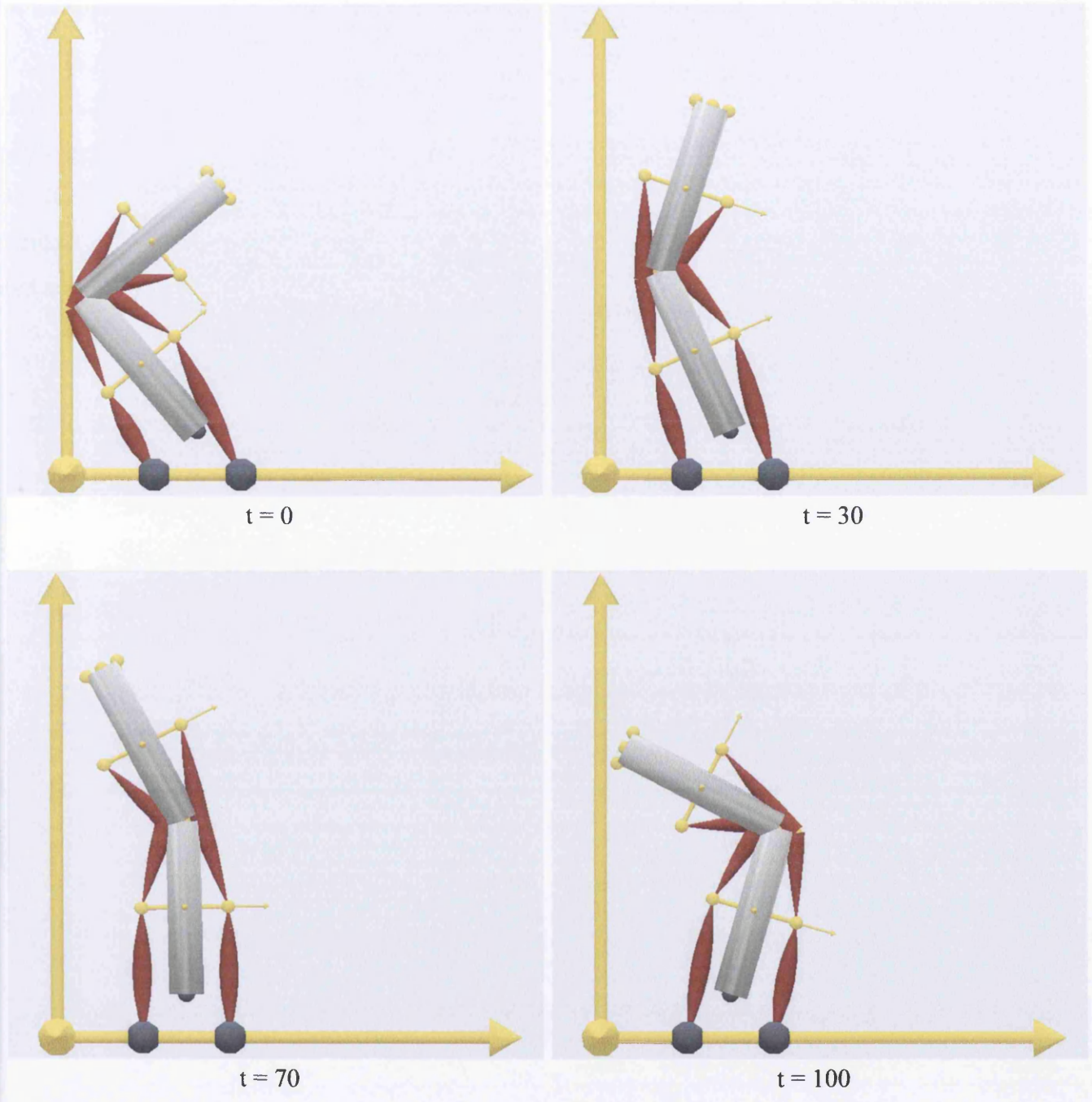


Figure 6.17 – Motion of Muscle Driven Double Inverted Pendulum

6.5 Triple Inverted Pendulum

In this section the development of a triple inverted pendulum will be investigated. This study is an extension of the previous one and the only difference is the addition of one more segment (segment3) upon segment2 (*Seg2*) and segment1 (*Seg1*). The new segment3 (*Seg3*) has the same geometrical properties as *Seg2* and *Seg1* and the same nodes. These nodes have the same local coordinates but they are numbered in a different way. Figure 6.18 below represents the triple inverted pendulum with the new *Seg3* and its nodes. *Seg2* and *Seg3* are connected to each other with the *C* joint (*Node 8 & 11*).

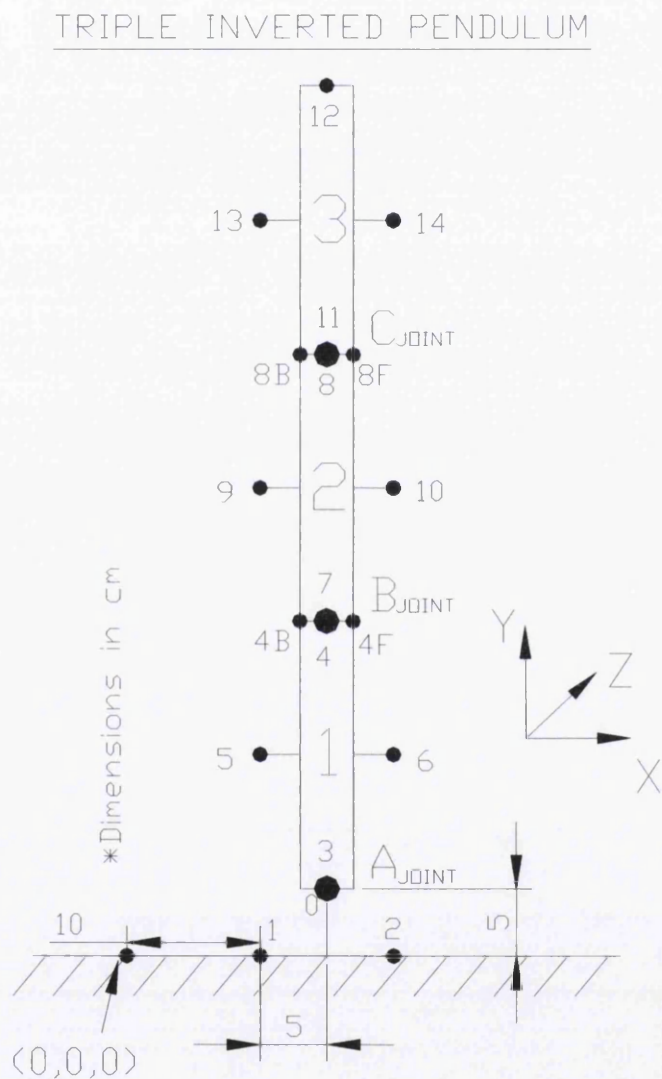


Figure 6.18 - Mechanical Drawing of Triple Inverted Pendulum

Table 6.7 below includes the updated Main.any file, the new Seg3.any file and the updated Joints.any, Drivers1.any and Muscles.any files. It has to be noticed that the *DriverPos* in the new driver is 90 degrees while *DriverVel* is -3 rad/sec.

Main.any File

```
Main = {  
  
  // The actual body model goes in this folder  
  AnyFolder TripleInvPend = {  
  
    // Global Reference Frame  
    AnyFixedRefFrame GlobalRef = {  
  
      #include "Environment.any"  
  
    }; // Global reference frame  
  
    //-----SEGMENTS -----  
  
    AnyFolder Segs = {  
  
      #include "Seg1.any"  
      #include "Seg2.any"  
      #include "Seg3.any"  
  
    }; //Segments  
  
    //-----JOINTS-----  
  
    AnyFolder Joints = {  
  
      #include "Joints.any"  
  
    }; //Joints  
  
    //-----DRIVERS-----  
  
    AnyFolder Drivers = {  
  
      #include "Drivers1.any"  
  
    }; //Drivers  
  
    //-----MUSCLES-----  
  
    AnyFolder Muscles = {  
  
      #include "Muscles.any"  
  
    }; //Muscles  
  
  }; // Triple Inverted Pendulum  
  
  // The study: Operations to be performed on  
  the model  
  
  AnyBodyStudy MyStudy = {  
    AnyFolder &Model = .TripleInvPend;  
    RecruitmentSolver = MinMaxSimplex;  
    Gravity = { 0.0,-9.81,0.0};  
  
  };  
  
}; // Main
```

Seg3.any File

```
AnySeg Seg3 = {  
  
  r = {0.15,0.55,0};  
  Mass = 2;  
  Jii = {0.0069,0.0004,0.0069};  
  AnyRefNode Node11 = {  
    sRel = {0,-0.1,0};  
  };  
  AnyRefNode Node12 = {  
    sRel = {0,0.1,0};  
  };  
  AnyRefNode Node13 = {  
    sRel = {-0.05,0,0};  
  };  
  AnyRefNode Node14 = {  
    sRel = {0.05,0,0};  
  };  
  AnyDrawSeg DrwSeg = {};  
  AnyDrawSurf DrwSurf = {  
    FileName = "Cylinder.anysurf";  
    ScaleXYZ = {0.001,0.001,0.001};  
  };  
  
}; //Seg3
```

Joints.any File

```
AnyRevoluteJoint A = {  
  Axis=z;  
  AnyRefNode &Node0  
=Main.TripleInvPend.GlobalRef.Ground0;  
  AnyRefNode &Node3  
=Main.TripleInvPend.Segs.Seg1.Node3;  
}; //Joint A  
  
AnyRevoluteJoint B = {  
  Axis=z;  
  AnyRefNode &Node4  
=Main.TripleInvPend.Segs.Seg1.Node4;  
  AnyRefNode &Node7  
=Main.TripleInvPend.Segs.Seg2.Node7;  
}; //Joint B  
  
AnyRevoluteJoint C = {  
  Axis=z;  
  AnyRefNode &Node8  
=Main.TripleInvPend.Segs.Seg2.Node8;  
  AnyRefNode &Node11  
=Main.TripleInvPend.Segs.Seg3.Node11;  
}; //Joint C
```

Muscles.any File

```
//-----Simple muscle model with constant
strength 300N-----
AnyMuscleModel MusMdl1 = {
  FO = 100;
};//Muscle Strength

//-----SEGMENT1 BACK-----
AnyViaPointMuscle Seg1Back = {
  AnyMuscleModel &MusMdl1 = .MusMdl1;
  AnyRefNode &Node1 = ..GlobalRef.Ground1;
  AnyRefNode &Node5 = ..Segs.Seg1.Node5;
  AnyDrawMuscle DrwMus = {};
};//Seg1_Back

//-----SEGMENT1 FRONT-----
AnyViaPointMuscle Seg1Front = {
  AnyMuscleModel &MusMdl1 = .MusMdl1;
  AnyRefNode &Node2 = ..GlobalRef.Ground2;
  AnyRefNode &Node6 = ..Segs.Seg1.Node6;
  AnyDrawMuscle DrwMus = {};
};//Seg1_Front

//-----SEGMENT2 BACK-----
AnyViaPointMuscle Seg2Back = {
  AnyMuscleModel &MusMdl1 = .MusMdl1;
  AnyRefNode &Node5 = ..Segs.Seg1.Node5;
  AnyRefNode &Node4B = ..Segs.Seg1.Node4B;
  AnyRefNode &Node9 = ..Segs.Seg2.Node9;
  AnyDrawMuscle DrwMus = {};
};//Seg2_Back

//-----SEGMENT2 FRONT-----
AnyViaPointMuscle Seg2Front = {
  AnyMuscleModel &MusMdl1 = .MusMdl1;
  AnyRefNode &Node6 = ..Segs.Seg1.Node6;
  AnyRefNode &Node4F = ..Segs.Seg1.Node4F;
  AnyRefNode &Node10 = ..Segs.Seg2.Node10;
  AnyDrawMuscle DrwMus = {};
};//Seg2_Front

//-----SEGMENT3 BACK-----
AnyViaPointMuscle Seg3Back = {
  AnyMuscleModel &MusMdl1 = .MusMdl1;
  AnyRefNode &Node9 = ..Segs.Seg2.Node9;
  AnyRefNode &Node8B = ..Segs.Seg2.Node8B;
  AnyRefNode &Node13 = ..Segs.Seg3.Node13;
  AnyDrawMuscle DrwMus = {};
};//Seg3_Back

//-----SEGMENT3 FRONT-----
AnyViaPointMuscle Seg3Front = {
  AnyMuscleModel &MusMdl1 = .MusMdl1;
  AnyRefNode &Node10 = ..Segs.Seg2.Node10;
  AnyRefNode &Node8F = ..Segs.Seg2.Node8F;
  AnyRefNode &Node14 = ..Segs.Seg3.Node14;
  AnyDrawMuscle DrwMus = {};
};//Seg3_Front
```

Drivers1.any File

```
//-----SEGMENT1 DRIVER-----
AnyKinEqSimpleDriver Seg1Motion = {
  AnyRevoluteJoint &Jnt = ..Joints.A;
  DriverPos = {40 * pi/180};//Rotation in
degrees
  DriverVel = {-1}; // rad/sec
  DriverAcc = {0};
  Reaction.Type = {0};
};//Seg1 Driver

//-----SEGMENT2 DRIVER-----
AnyKinEqSimpleDriver Seg2Motion = {
  AnyRevoluteJoint &Jnt = ..Joints.B;
  DriverPos = {-90 * pi/180};//Rotation in
degrees
  DriverVel = {3}; // rad/sec
  DriverAcc = {0};
  Reaction.Type = {0};
};//Seg2 Driver

//-----SEGMENT3 DRIVER-----
AnyKinEqSimpleDriver Seg3Motion = {
  AnyRevoluteJoint &Jnt = ..Joints.C;
  DriverPos = {90 * pi/180};//Rotation in
degrees
  DriverVel = {-3}; // rad/sec
  DriverAcc = {0};
  Reaction.Type = {0};
};//Seg3 Driver
```

Table 6.7 – Triple Inverted Pendulum Anyfiles

Figure 6.19 below, shows a sequence of images of triple inverted pendulum's motion.

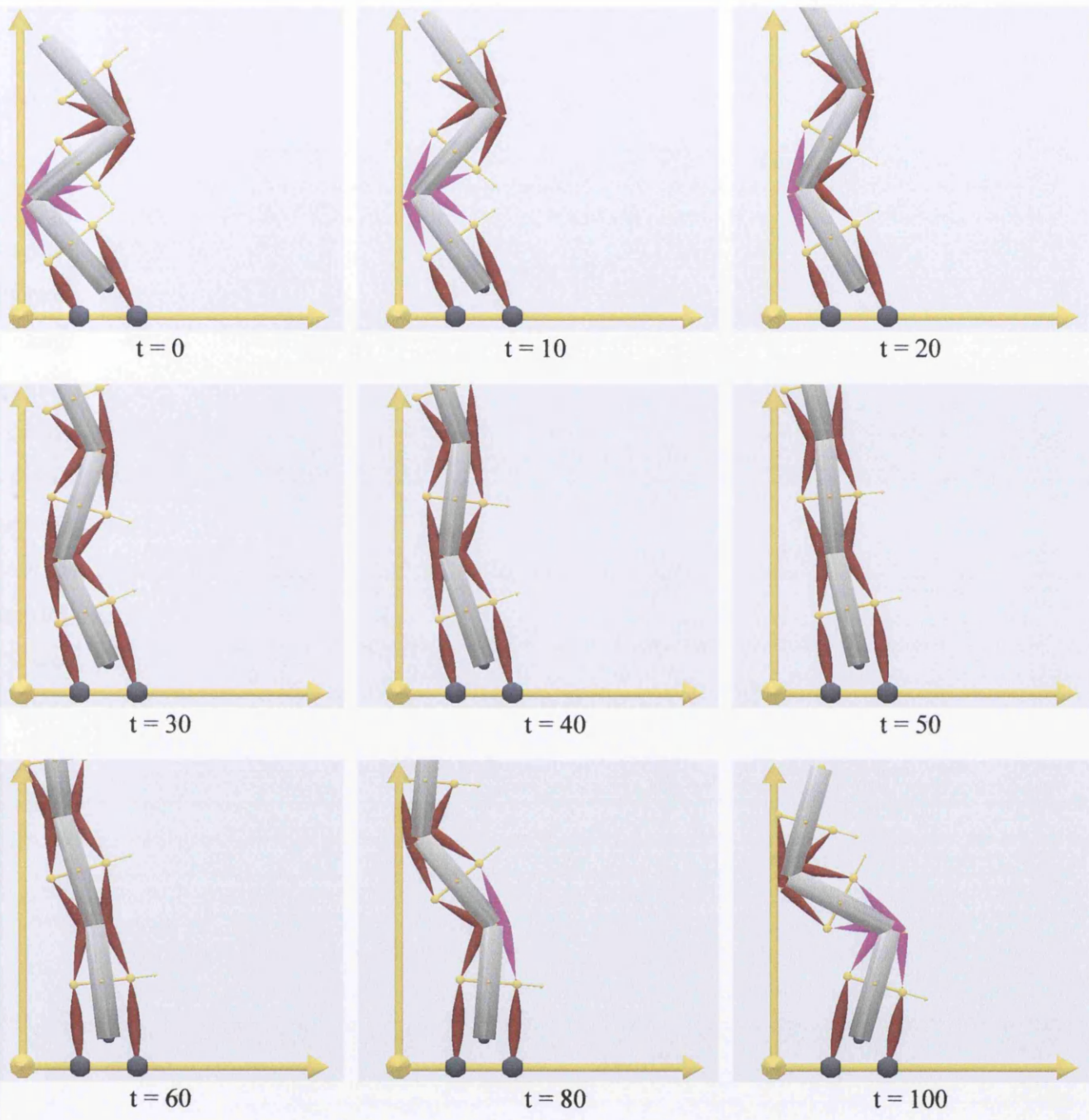


Figure 6.19 – Motion of Muscle Driven Triple Inverted Pendulum

6.6 Advanced Study

6.6.1 Initial Conditions

A very important aspect of the modeling procedure is the definition of the orientation of a segment without the need of a driver. And that's because Anybody Technology has a facility (as mentioned in Chapter 5) called "Kinematic Measure" which gives the ability to drive only one segment and make the others connected to it, follow its motion. For example, at *Node12* of *Seg3* (see Figure 6.18) a driver could be assigned and make *Seg1* and *Seg2* follow its motion without their drivers. But, in order to make *Node12* start from a specific position, it is very important to set the precise orientation of the segments. This will be analyzed later in this chapter.

In the Anybody modeling system the statement *Axes* which is included in the *AnySeg* statement is responsible for the orientation of a segment. *Axes*, contains a rotational matrix which performs rotation about all three axes. The type of rotation is a *Cardan* rotation sequence and the most common used in biomechanics is a *Xyz sequence*. This sequence involves, first, rotation about the medially directed axis (*X*); second, rotation about the anteriorly directed axis (*y*); and third, about the vertical axis (*z*). These rotations are diagrammed in Figure 6.20 below.

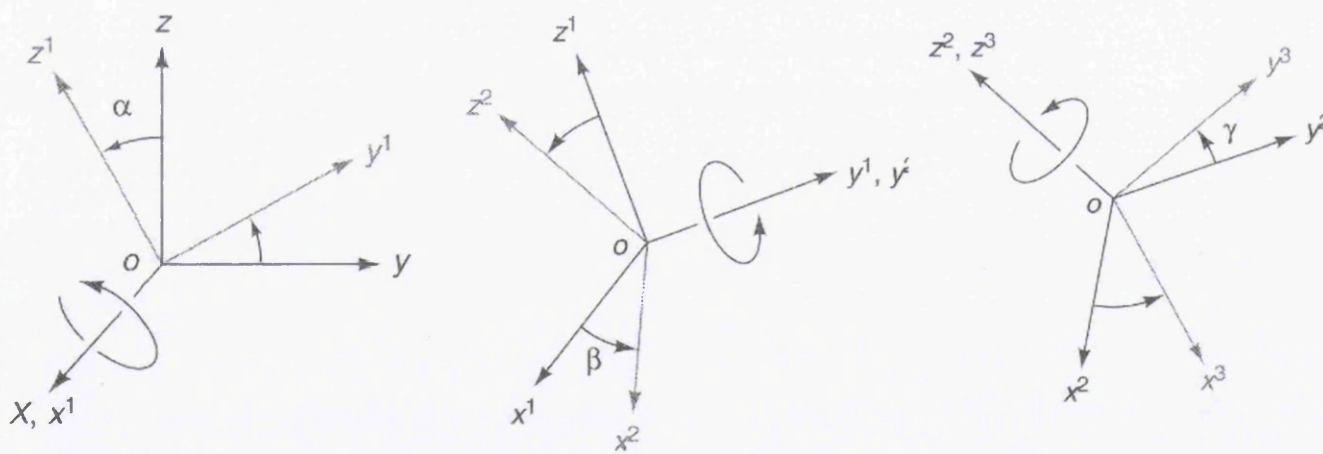


Figure 6.20 – *Xyz* Cardan Sequence of Rotations

The above figure shows that the first rotation takes place about the *X*-axis and leads to the new orientations of the *y*- and *z*-axes (y^1 and z^1), with the *X*-axis remaining in the same orientation, but now it is labeled x^1 . The second rotation about the y^1 axis leads to new positions of the x^1 and z^1 axis (x^2 and z^2). For the third rotation about the z^2 axis, the x^2 and y^2 axes assume the new orientation of x^3 and y^3 . For illustrative purposes, the angles for the *Xyz* sequence are designated α (alpha) for the first rotation, β (beta) for the second rotation, and γ (gamma) for the third rotation. The rotation matrix $[R]$ for an *Xyz* rotation sequence then can be described as:

$$[R] = [R_z] \cdot [R_y] \cdot [R_x] \quad (6.1)$$

where

$$[R_x] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha \\ 0 & -\sin \alpha & \cos \alpha \end{bmatrix} \quad (6.2)$$

$$[R_y] = \begin{bmatrix} \cos \beta & 0 & -\sin \beta \\ 0 & 1 & 0 \\ \sin \beta & 0 & \cos \beta \end{bmatrix} \quad (6.3)$$

$$[R_z] = \begin{bmatrix} \cos \gamma & \sin \gamma & 0 \\ -\sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (6.4)$$

The rotation can then be expressed as the successive rotation of these matrices to produce the combined rotation matrix:

$$[R] = \begin{bmatrix} \cos \beta \cdot \cos \gamma & \cos \gamma \cdot \sin \beta \cdot \sin \alpha + \sin \gamma \cdot \cos \alpha & \sin \gamma \cdot \sin \alpha - \cos \gamma \sin \beta \cos \alpha \\ -\sin \gamma \cdot \cos \beta & \cos \alpha \cdot \cos \gamma - \sin \alpha \cdot \sin \beta \cdot \sin \gamma & \sin \gamma \cdot \sin \beta \cdot \cos \alpha + \cos \gamma \cdot \sin \alpha \\ \sin \beta & -\cos \beta \cdot \sin \alpha & \cos \alpha \cdot \cos \beta \end{bmatrix} \quad (6.5) [10]$$

The updated Main.any file and the new files, InitialConditions.any and RotMatrices.any are shown in Table 6.8 below:

<u>Main.any File</u>	<u>InitialConditions.any File</u>
<pre>Main = { // The actual body model goes in this folder AnyFolder TripleInvPend = { // Global Reference Frame AnyFixedRefFrame GlobalRef = { #include "Environment.any" }; // Global reference frame</pre>	<pre>//-----ROTATION X-AXIS----- //-----Segment 1----- AnyVar alpha1 = 0; // rad //-----Segment 2----- AnyVar alpha2 = 0; // rad //-----Segment 3----- AnyVar alpha3 = 0; // rad //-----ROTATION Y-AXIS----- //-----Segment 1----- AnyVar beta1 = 0; // rad //-----Segment 2----- AnyVar beta2 = 0; // rad //-----Segment 3----- AnyVar beta3 = 0; // rad</pre>

```

//-----SEGMENTS-----
AnyFolder Segs = {
    #include "Seg1.any"
    #include "Seg2.any"
    #include "Seg3.any"
}; //Segments

//-----INITIAL CONDITIONS-----
AnyFolder InitialConditions = {
    #include "InitialConditions.any"
    #include "RotMatrices.any"
}; //Initial Conditions

//-----JOINTS-----
AnyFolder Joints = {
    #include "Joints.any"
}; //Joints

//-----DRIVERS-----
AnyFolder Drivers = {
    #include "Drivers1.any"
}; //Drivers

//-----MUSCLES-----
AnyFolder Muscles = {
    #include "Muscles.any"
}; //Muscles

}; // Triple Inverted Pendulum

// The study: Operations to be performed on
the model
AnyBodyStudy MyStudy = {
    AnyFolder &Model = .TripleInvPend;
    RecruitmentSolver = MinMaxSimplex;
    Gravity = { 0.0, -9.81, 0.0 };
};

}; // Main

```

```

//-----ROTATION Z-AXIS-----
//-----Segment 1-----
AnyVar gamma1 = 0; // rad
//-----Segment 2-----
AnyVar gamma2 = 0; // rad
//-----Segment 3-----
AnyVar gamma3 = 0; // rad

```

RotMatrices.any File

```

//-----SEGMENT1 ROTATION VECTOR-----
AnyMat33 R1 = { {cos(beta1) * cos(gamma1) ,
                cos(gamma1) * sin(beta1) * sin(alpha1) +
                sin(gamma1) * cos(alpha1) ,
                sin(gamma1) * sin(alpha1) - cos(gamma1) *
                sin(beta1) * cos(alpha1)} ,
                {-sin(gamma1) * cos(beta1) ,
                cos(alpha1) * cos(gamma1) - sin(alpha1) *
                sin(beta1) * sin(gamma1) ,
                sin(gamma1) * sin(beta1) * cos(alpha1) +
                cos(gamma1) * sin(alpha1)} ,
                { sin(beta1) , -cos(beta1) * sin(alpha1) ,
                cos(alpha1) * cos(beta1) }};

//-----SEGMENT2 ROTATION VECTOR-----
AnyMat33 R2 = { {cos(beta2) * cos(gamma2) ,
                cos(gamma2) * sin(beta2) * sin(alpha2) +
                sin(gamma2) * cos(alpha2) ,
                sin(gamma2) * sin(alpha2) - cos(gamma2) *
                sin(beta2) * cos(alpha2)} ,
                {-sin(gamma2) * cos(beta2) ,
                cos(alpha2) * cos(gamma2) - sin(alpha2) *
                sin(beta2) * sin(gamma2) ,
                sin(gamma2) * sin(beta2) * cos(alpha2) +
                cos(gamma2) * sin(alpha2)} ,
                { sin(beta2) , -cos(beta2) * sin(alpha2) ,
                cos(alpha2) * cos(beta2) }};

//-----SEGMENT3 ROTATION VECTOR-----
AnyMat33 R3 = { {cos(beta3) * cos(gamma3) ,
                cos(gamma3) * sin(beta3) * sin(alpha3) +
                sin(gamma3) * cos(alpha3) ,
                sin(gamma3) * sin(alpha3) - cos(gamma3) *
                sin(beta3) * cos(alpha3)} ,
                {-sin(gamma3) * cos(beta3) ,
                cos(alpha3) * cos(gamma3) - sin(alpha3) *
                sin(beta3) * sin(gamma3) ,
                sin(gamma3) * sin(beta3) * cos(alpha3) +
                cos(gamma3) * sin(alpha3)} ,
                { sin(beta3) , -cos(beta3) * sin(alpha3) ,
                cos(alpha3) * cos(beta3) }};

```

Table 6.8 – Main.any/Initial Conditions.any Files

The orientation of the segments is then defined, adding at the *AnySeg* statements of *Seg1*, *Seg2* and *Seg3* the statements, *Axes = ..InitialConditions.R1*, *Axes = ..InitialConditions.R2*; and *Axes = ..InitialConditions.R3*; respectively.

6.6.2 Kinematic Measures & Motion

This chapter will end with the step-by-step procedure of how the triple inverted pendulum can be driven using only one driver, taking advantage of the “Kinematic Measures” and the *Axes* facilities, Anybody provides. Thus, through this investigation, the role of these facilities will be determined. The model that will be used is the already known triple inverted pendulum with one difference; Joint *A* is changed to a spherical joint (*AnySphericalJoint*) in order to produce a more complicated motion, a 3-D motion.

The first step is to decide the position of the whole assembly in space. *Seg1* can be rotated in 3-D while *Seg2* and *Seg3* only about z-axis. For convenience, the orientation will be defined by *Cartesian* rotation vector which measures a general 3-D orientation as a single rotation about a given axis. If let's say, the orientation of *Seg1* in degrees about *x*, *y* and *z* axes is 60° , 60° and 60° respectively, and the orientation about the *z*-axis of *Seg2* and *Seg3* is -60° and 60° respectively, how are all these orientations going to be implemented? The solution to this problem is employing the “Kinematic Measures”.

The statement *AnyKinRotational* is used, followed by a name, and includes the reference frames that will be measured, and the statement *Type* which determines the possible rotational quantities. In this particular case, the reference frames are the *GlobalRef* and one of the segments *Seg1*, *Seg2*, *Seg3* each time, while the *Type* is *RotVector*. Then, these “Kinematic Measures” are included to the corresponding driver where the angles of rotations are stated and the degrees of freedom as well. All the “Kinematic Measures” are included in the *KinematicMeasures.any* file while the drivers are in a new file, the *Drivers2.any* file.

Table 6.9 below, represents the updated *Main.any* file and *Joints.any* file, the *KinematicMeasures.any* file and the *Drivers2.any* file.

Main.any File

```
Main = {  
  
  // The actual body model goes in this folder  
  AnyFolder TripleInvPend = {  
  
    // Global Reference Frame  
    AnyFixedRefFrame GlobalRef = {  
  
      #include "Environment.any"  
  
    }; // Global reference frame  
  
    //-----SEGMENTS -----  
  
    AnyFolder Segs = {  
  
      #include "Seg1.any"  
      #include "Seg2.any"  
      #include "Seg3.any"  
  
    }; //Segments  
  
  };  
};
```

Joints.any File

```
AnySphericalJoint A = {  
  AnyRefNode &Node0  
  =Main.TripleInvPend.GlobalRef.Ground0;  
  AnyRefNode &Node3  
  =Main.TripleInvPend.Segs.Seg1.Node3;  
}; //Joint A  
  
AnyRevoluteJoint B = {  
  Axis=z;  
  AnyRefNode &Node4  
  =Main.TripleInvPend.Segs.Seg1.Node4;  
  AnyRefNode &Node7  
  =Main.TripleInvPend.Segs.Seg2.Node7;  
}; //Joint B  
  
AnyRevoluteJoint C = {  
  Axis=z;  
  AnyRefNode &Node8  
  =Main.TripleInvPend.Segs.Seg2.Node8;  
  AnyRefNode &Node11  
  =Main.TripleInvPend.Segs.Seg3.Node11;  
}; //Joint C
```



```

//-----INITIAL CONDITIONS-----
AnyFolder InitialConditions = {
    #include "InitialConditions.any"
    #include "RotMatrices.any"
}; //Initial Conditions
//-----JOINTS-----
AnyFolder Joints = {
    #include "Joints.any"
}; //Joints
//-----KINEMATIC MEASURES-----
AnyFolder KinematicMeasures = {
    #include "KinematicMeasures.any"
}; //Kinematic Measures
//-----DRIVERS-----
AnyFolder Drivers = {
    // #include "Drivers1.any"
    #include "Drivers2.any"
}; //Drivers
//-----MUSCLES-----
AnyFolder Muscles = {
    #include "Muscles.any"
}; //Muscles
}; // Triple Inverted Pendulum
// The study: Operations to be performed on
the model
AnyBodyStudy MyStudy = {
    AnyFolder &Model = .TripleInvPend;
    RecruitmentSolver = MinMaxSimplex;
    Gravity = { 0.0,-9.81,0.0};
};
}; // Main

```

KinematicMeasures.any File

```

//-----CARTESIAN ROTATION VECTORS-----
//-----SEG1 KINEMATIC ROTATIONAL-----
AnyKinRotational Seg1Rot1 = {
    AnyFixedRefFrame &Ground = ..GlobalRef;
    AnyRefFrame &Seg1 = ..Segs.Seg1;
    Type = RotVector;
};
//-----SEG2 KINEMATIC ROTATIONAL-----
AnyKinRotational Seg2Rot1 = {
    AnyFixedRefFrame &Ground = ..GlobalRef;
    AnyRefFrame &Seg2 = ..Segs.Seg2;
    Type = RotVector;
};
//-----SEG3 KINEMATIC ROTATIONAL-----
AnyKinRotational Seg3Rot1 = {
    AnyFixedRefFrame &Ground = ..GlobalRef;
    AnyRefFrame &Seg2 = ..Segs.Seg3;
    Type = RotVector;
};

```

Drivers2.any File

```

//-----SEG1 DRIVER-----
AnyKinEqSimpleDriver Seg1Driver = {
    AnyKinRotational &Jnt =
..KinematicMeasures.Seg1Rot1;
    MeasureOrganizer = {0,1,2};
    DriverPos =
{60*pi/180,60*pi/180,60*pi/180};
    DriverVel = {0,0,0};
    DriverAcc = {0,0,0};
    Reaction.Type = {1,1,1};
};
//-----SEG2 DRIVER-----
AnyKinEqSimpleDriver Seg2Driver = {
    AnyKinRotational &Jnt =
..KinematicMeasures.Seg2Rot1;
    MeasureOrganizer = {2};
    DriverPos = {-60*pi/180};
    DriverVel = {0};
    DriverAcc = {0};
    Reaction.Type = {1};
};
//-----SEG3 DRIVER-----
AnyKinEqSimpleDriver Seg3Driver = {
    AnyKinRotational &Jnt =
..KinematicMeasures.Seg3Rot1;
    MeasureOrganizer = {2};
    DriverPos = {60*pi/180};
    DriverVel = {0};
    DriverAcc = {0};
    Reaction.Type = {1};
};

```

Table 6.9 - Main.any/Joints.any/KinematicMeasures.any/Drivers2.any Files

Running the model without muscles this time (*Reaction.Type=1*), as such a motion needs more muscles to be implemented, and with *DriverVel=0* for all the drivers, the position of the whole assembly is as follows (Figure 6.21).

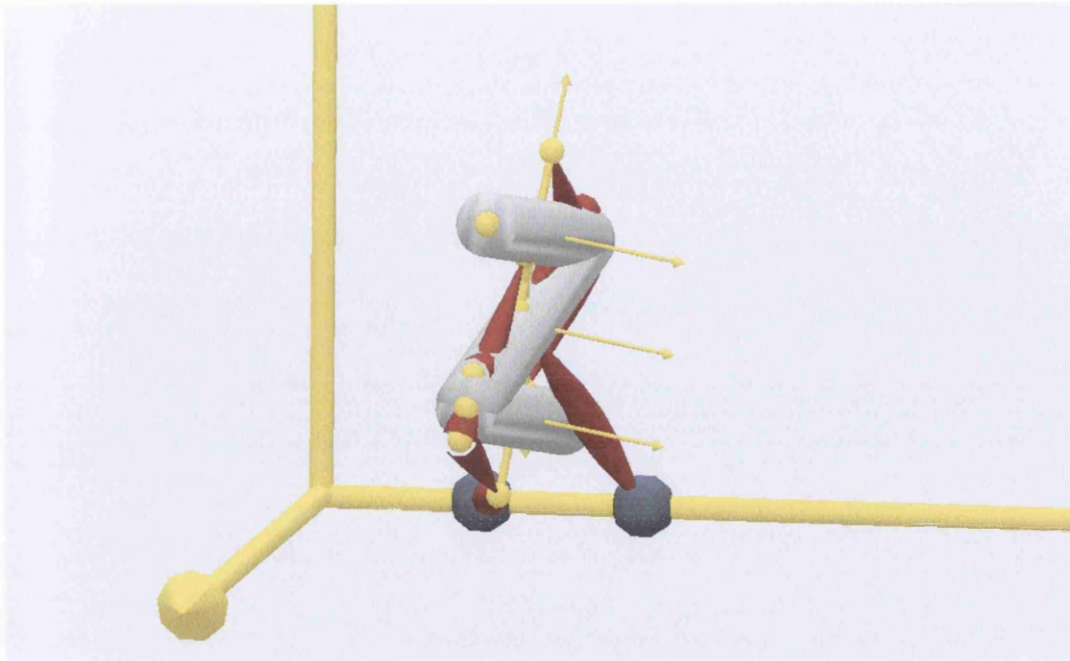


Figure 6.21 – 3-D Initial Position of Triple Inverted Pendulum

The second step, is to decide which segment or node should be chosen to drive the whole assembly. Of course more than one driver could be used. This depends on the degrees of freedom of the model. As mentioned before, *Node12* of *Seg3* could be used to drive the whole model. The motion that will be implemented here will be incorporating both translation and rotation.

The driver of *Node12* and the corresponding “Kinematic Measures” are defined as well. A new “Kinematic Measure” is obtained in this step, which measures translation. The statement for this “Measure” is *AnyKinLinear* followed by a name, and includes the reference frames that will be measured and the statement *Ref* that specify the reference system in which the vector is measured (-1=default/global system, 0 = first reference system, 1 = second reference system). In this particular case the reference frames are the *GlobalRef* and *Node12* while *Ref* is 0 (it could be -1 as well). An important thing that has to be mentioned, is that because of both translational and rotational motion of *Node12*, an organizer of these “Kinematic Measures” will be used. For this purpose the statement *AnyKinMeasureOrg* is used, followed by a name. It includes the “Kinematic Measures” and the statement *MeasureOrganizer* which indicates which measures will be produced. The updated *KinematicMeasures.any* file with the “Measures” of *Node12* is shown in Figure 6.22. The *MeasureOrganizer* will measure $x(1), y(2)$ and $z(3)$ translation and $y(4)$ and $z(5)$ rotation.

KinematicMeasures.any File

```
//-----CARTESIAN ROTATION VECTORS-----  
  
//-----SEG1 KINEMATIC ROTATIONAL-----  
AnyKinRotational Seg1Rot1 = {  
  AnyFixedRefFrame &Ground = ..GlobalRef;  
  AnyRefFrame &Seg1 = ..Segs.Seg1;  
  Type = RotVector;  
};  
  
//-----SEG2 KINEMATIC ROTATIONAL-----  
AnyKinRotational Seg2Rot1 = {  
  AnyFixedRefFrame &Ground = ..GlobalRef;  
  AnyRefFrame &Seg2 = ..Segs.Seg2;  
  Type = RotVector;  
};  
  
//-----SEG3 KINEMATIC ROTATIONAL-----  
AnyKinRotational Seg3Rot1 = {  
  AnyFixedRefFrame &Ground = ..GlobalRef;  
  AnyRefFrame &Seg2 = ..Segs.Seg3;  
  Type = RotVector;  
};  
  
//-----NODE12 KINEMATIC LINEAR-----  
AnyKinLinear Node12Lin = {  
  AnyFixedRefFrame &Ground = ..GlobalRef;  
  AnyRefNode &Node12 = ..Segs.Seg3.Node12;  
  Ref = 0;  
};  
//-----NODE12 KINEMATIC ROTATIONAL-----  
AnyKinRotational Node12Rot = {  
  AnyFixedRefFrame &Ground = ..GlobalRef;  
  AnyRefNode &Node12 = ..Segs.Seg3.Node12;  
  Type = RotVector;  
};  
//-----NODE 12 ORGANIZER-----  
AnyKinMeasureOrg Node12Pos = {  
  AnyKinLinear &ref1 = .Node12Lin;  
  AnyKinRotational &ref2 = .Node12Rot;  
  MeasureOrganizer = {0,1,2,4,5};  
};
```

Figure 6.22 – Updated KinematicMeasures.any File

The second step continues with running the model and recording the content of the *Node12Pos* in order to know the exact position of *Node12* before adding its driver in the *Driver2.any* file. After running the model, $Pos = \{0.135012, 0.200358, 0.212105, 1.047198, 1.047198\}$; with units *m*, *m*, *m*, *rad* and *rad* respectively, and is obtained from the left window next to the AnyScript model. Due to the fact that Anybody permits less than 10^{-6} constraint violations at the joints, it is recommended that the *Pos* elements must have 6-digit decimal accuracy. Now the *Node12* driver is added in the *Drivers2.any* file with the *Pos* elements in the *DriverPos* and all the velocities being equal to zero. An updated file of the

Drivers2.any file is shown in Figure 6.23. The model, with all the drivers, runs again in order to doublecheck that *DriverPos* of *Node12* is correct.

```
Drivers2.any File
```

```
//-----SEG1 DRIVER-----
AnyKinEqSimpleDriver Seg1Driver = {
  AnyKinRotational &Jnt = ..KinematicMeasures.Seg1Rot1;
  MeasureOrganizer = {0,1,2};
  DriverPos = {60*pi/180,60*pi/180,60*pi/180};
  DriverVel = {0,0,0};
  DriverAcc = {0,0,0};
  Reaction.Type = {1,1,1};
};
//-----SEG2 DRIVER-----
AnyKinEqSimpleDriver Seg2Driver = {
  AnyKinRotational &Jnt = ..KinematicMeasures.Seg2Rot1;
  MeasureOrganizer = {2};
  DriverPos = {-60*pi/180};
  DriverVel = {0};
  DriverAcc = {0};
  Reaction.Type = {1};
};
//-----SEG3 DRIVER-----
AnyKinEqSimpleDriver Seg3Driver = {
  AnyKinRotational &Jnt = ..KinematicMeasures.Seg3Rot1;
  MeasureOrganizer = {2};
  DriverPos = {60*pi/180};
  DriverVel = {0};
  DriverAcc = {0};
  Reaction.Type = {1};
};
//-----NODE 12 DRIVER-----
AnyKinEqSimpleDriver Node12Driver = {
  AnyKinMeasureOrg &ref = ..KinematicMeasures.Node12Pos;
  DriverPos = {0.135012, 0.200358, 0.212105, 1.047198, 1.047198};
  DriverVel = {0,0,0,0,0};
  Reaction.Type = {1,1,1,1,1};
};
```

Figure 6.23 – Updated Driver2.any File

The investigation continues with the third step in which the motion of the triple inverted pendulum will be implemented with just one driver, the *Node12Driver*. But, if the drivers of *Seg1* and *Seg2* will be eliminated, how are these segments going to be placed in the right position, i.e. the correct initial conditions? The solution to this problem is given by the rotational matrix facility (*Axes*) that Anybody provides, which can produce the desired rotation. All what is needed now, is the correct values of the variables of the rotation matrix, which are the angles *alpha*, *beta* and *gamma*. For this purpose a new set of “Kinematic Measures” is used that measure *Cardan* angles. The *AnyKinRotational* facility has to be used, with the difference that *Type* is *RotAxesAngles* now. Because of the *Xyz* sequence that rotational matrices represent, the same sequence have to be defined in this “Measure” as well. This can happen with the definition of the first, second and third axis of rotation, using the statements *Axis1*, *Axis2* and *Axis3*. The updated *KinematicMeasures.any* file with the new “Measures” is shown below (Figure 6.24).

KinematicMeasures.any File

```
//-----CARTESIAN ROTATION VECTORS-----
//-----SEG1 KINEMATIC ROTATIONAL-----
AnyKinRotational Seg1Rot1 = {
  AnyFixedRefFrame &Ground = ..GlobalRef;
  AnyRefFrame &Seg1 = ..Segs.Seg1;
  Type = RotVector;
};
//-----SEG2 KINEMATIC ROTATIONAL-----
AnyKinRotational Seg2Rot1 = {
  AnyFixedRefFrame &Ground = ..GlobalRef;
  AnyRefFrame &Seg2 = ..Segs.Seg2;
  Type = RotVector;
};
//-----SEG3 KINEMATIC ROTATIONAL-----
AnyKinRotational Seg3Rot1 = {
  AnyFixedRefFrame &Ground = ..GlobalRef;
  AnyRefFrame &Seg2 = ..Segs.Seg3;
  Type = RotVector;
};
//-----CARDAN ANGLES-----
//-----SEG1 KINEMATIC ROTATIONAL-----
AnyKinRotational Seg1Rot2 = {
  AnyRefFrame &Seg1 = ..Segs.Seg1;
  AnyFixedRefFrame &Ground = ..GlobalRef;
  Axis1=x;
  Axis2=y;
  Axis3=z;
  Type = RotAxesAngles;
};
//-----SEG2 KINEMATIC ROTATIONAL-----
AnyKinRotational Seg2Rot2 = {
  AnyRefFrame &Seg2 = ..Segs.Seg2;
  AnyFixedRefFrame &Ground = ..GlobalRef;
  Axis1=x;
  Axis2=y;
  Axis3=z;
  Type = RotAxesAngles;
};
//-----SEG3 KINEMATIC ROTATIONAL-----
AnyKinRotational Seg3Rot2 = {
  AnyRefFrame &Seg2 = ..Segs.Seg3;
  AnyFixedRefFrame &Ground = ..GlobalRef;
  Axis1=x;
  Axis2=y;
  Axis3=z;
  Type = RotAxesAngles;
};
//-----NODE12 KINEMATIC LINEAR-----
AnyKinLinear Node12Lin = {
  AnyFixedRefFrame &Ground = ..GlobalRef;
  AnyRefNode &Node12 = ..Segs.Seg3.Node12;
  Ref = 0;
};
//-----NODE12 KINEMATIC ROTATIONAL-----
AnyKinRotational Node12Rot = {
  AnyFixedRefFrame &Ground = ..GlobalRef;
  AnyRefNode &Node12 = ..Segs.Seg3.Node12;
  Type = RotVector;
};
//-----NODE 12 ORGANIZER-----
AnyKinMeasureOrg Node12Pos = {
  AnyKinLinear &ref1 = .Node12Lin;
  AnyKinRotational &ref2 = .Node12Rot;
  MeasureOrganizer = {0,1,2,4,5};
};
```

Figure 6.24 – Updated KinematicMeasures.any File

Now, the model has to be run again and the angles measured by the *AnyKinRotational* (*Type=RotAxesAngles*) facility are recorded. These measures have to be transferred to the *InitialConditions.any* file which becomes (Figure 6.25):

```
InitialConditions.any File  
  
//-----ROTATION X-AXIS-----  
//-----Segment 1-----  
AnyVar alpha1 = -1.395077; // rad  
//-----Segment 2-----  
AnyVar alpha2 = 1.378331; // rad  
//-----Segment 3-----  
AnyVar alpha3 = -1.395077; // rad  
  
//-----ROTATION Y-AXIS-----  
//-----Segment 1-----  
AnyVar beta1 = -0.147381; // rad  
//-----Segment 2-----  
AnyVar beta2 = -0.441711; // rad  
//-----Segment 3-----  
AnyVar beta3 = -0.147381; // rad  
  
//-----ROTATION Z-AXIS-----  
//-----Segment 1-----  
AnyVar gamma1 = -1.395077; // rad  
//-----Segment 2-----  
AnyVar gamma2 = 1.637332; // rad  
//-----Segment 3-----  
AnyVar gamma3 = -1.395077; // rad
```

Figure 6.25 – InitialConditions.any File

In order to check if these measures cause the desired rotation of the segments, the model has to be run again and checked at the time **“study operation”**: **Load time positions have been re-established** (Figure 6.26), the *Cartesian* “Measures” of the *AnyKinRotational* (*Type=RotVector*) facility. In this model indeed these are the correct “Measures”. Note that the model has to be reloaded before every “run” even if the model remains the same.

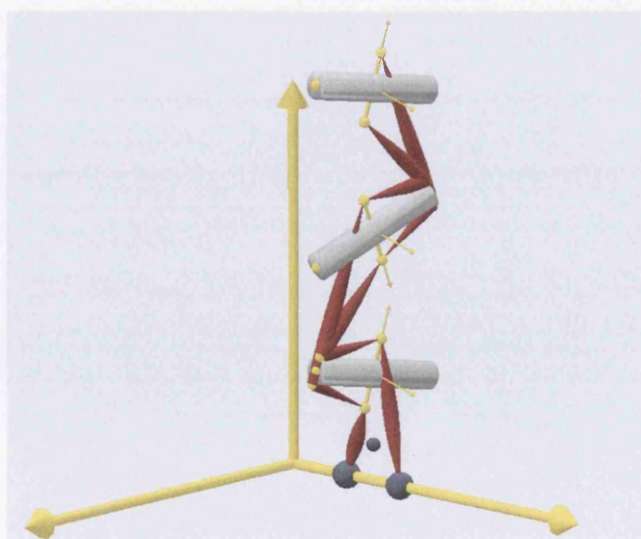


Figure 6.26 – Model with Initial Conditions at Time Step 0

Then, all the drivers of the Drivers2.any file are being eliminated and the only one that remains is the *Node12Driver*. The updated Driver2.any file with a selection of driver velocities (*DriverVel*) and the motion of the triple inverted pendulum are shown in Figures 6.27 and 6.28 respectively, while the whole procedure followed in this investigation, is concentrated in a flow chart in Figure 6.31. The linear displacement and velocity of Node12 are presented in Figure 6.29 while the angular displacement and velocity in Figure 6.30.

Driver2.any File

```

/*
//-----SEG1 DRIVER-----
AnyKinEqSimpleDriver Seg1Driver = {
    AnyKinRotational &Jnt = ..KinematicMeasures.Seg1Rot1;
    MeasureOrganizer = {0,1,2};
    DriverPos = {60*pi/180,60*pi/180,60*pi/180};
    DriverVel = {0,0,0};
    DriverAcc = {0,0,0};
    Reaction.Type = {1,1,1};
};

//-----SEG2 DRIVER-----
AnyKinEqSimpleDriver Seg2Driver = {
    AnyKinRotational &Jnt = ..KinematicMeasures.Seg2Rot1;
    MeasureOrganizer = {2};
    DriverPos = {-60*pi/180};
    DriverVel = {0};
    DriverAcc = {0};
    Reaction.Type = {1};
};

//-----SEG3 DRIVER-----
AnyKinEqSimpleDriver Seg3Driver = {
    AnyKinRotational &Jnt = ..KinematicMeasures.Seg3Rot1;
    MeasureOrganizer = {2};
    DriverPos = {60*pi/180};
    DriverVel = {0};
    DriverAcc = {0};
    Reaction.Type = {1};
};

*/

//-----NODE 12 DRIVER-----
AnyKinEqSimpleDriver Node12Driver = {
    AnyKinMeasureOrg &ref = ..KinematicMeasures.Node12Pos;
    DriverPos = {0.135012, 0.200358, 0.212105, 1.047198, 1.047198};
    DriverVel = {-0.1,0.4,-0.4,-1,-0.8};
    Reaction.Type = {1,1,1,1,1};
};

```

Figure 6.27 – Updated Drivers2.any File

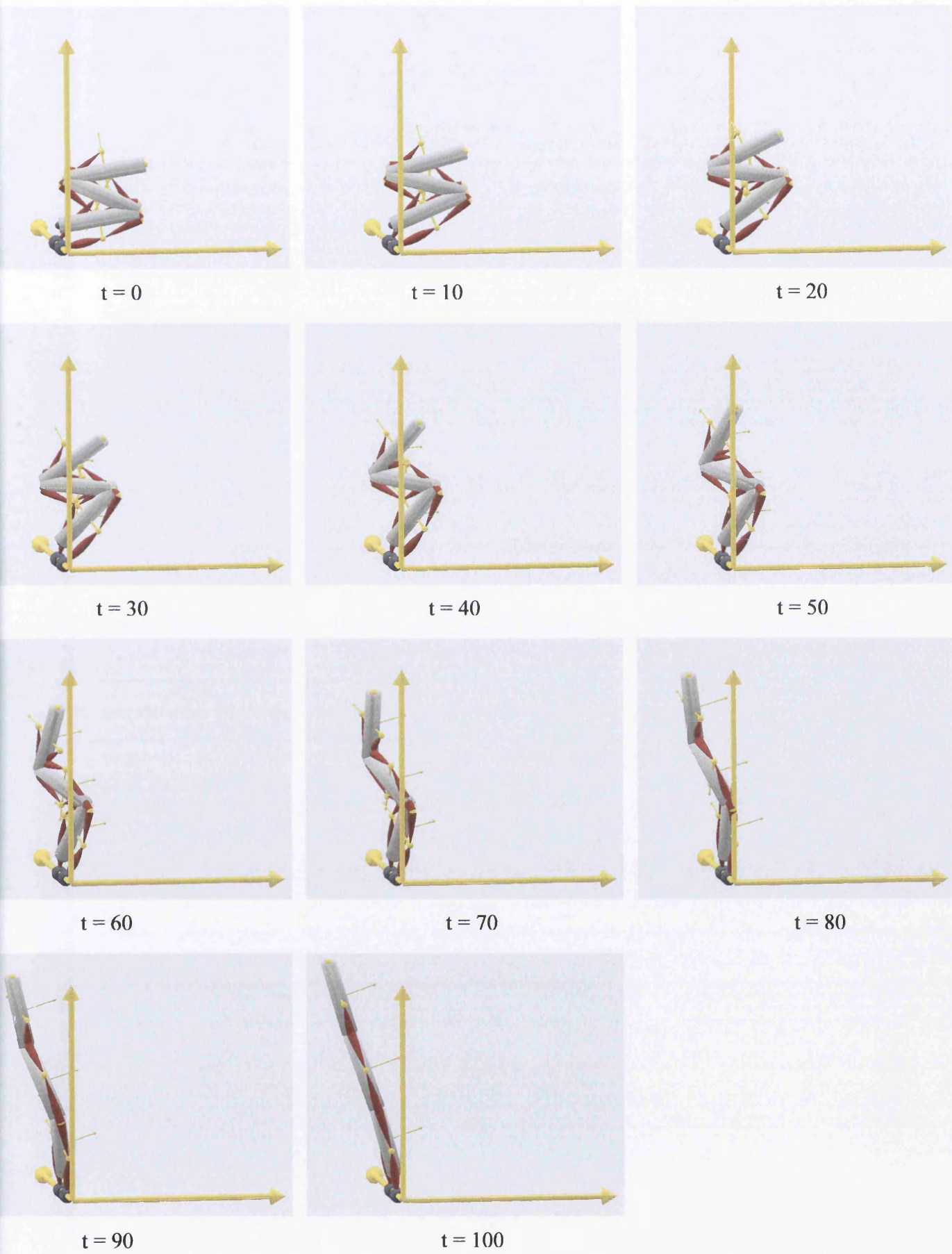


Figure 6.28 – Motion of Triple Inverted Pendulum with only One Driver

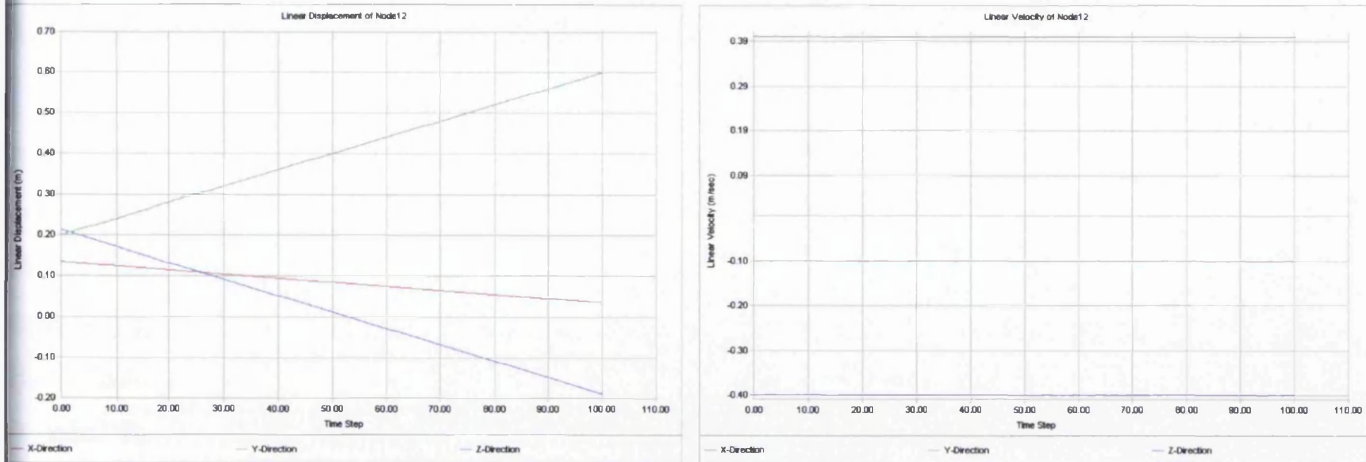


Figure 6.29 – Linear Displacement & Velocity of Node 12

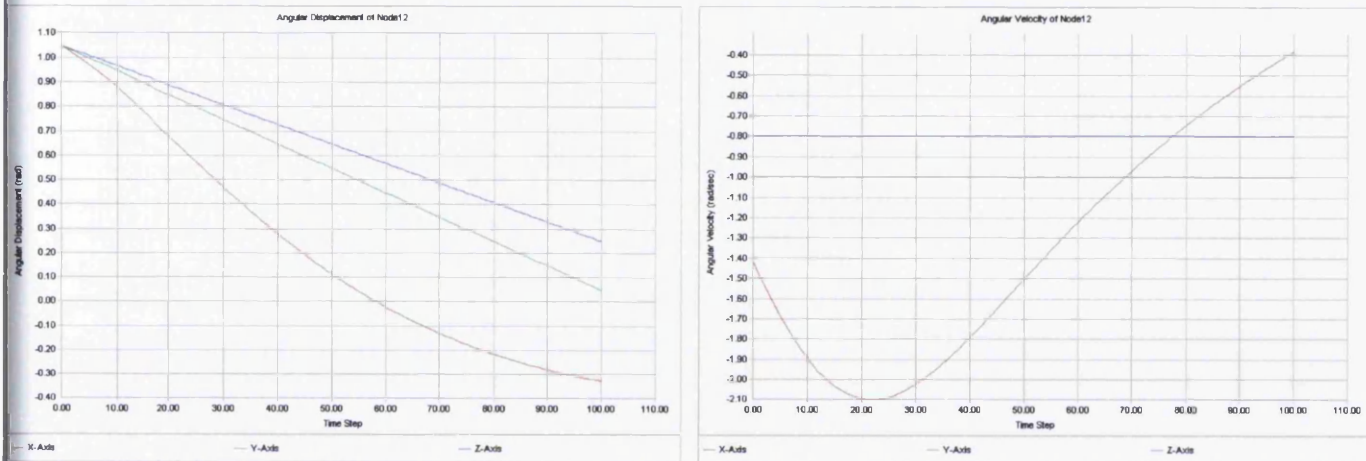


Figure 6.30 – Angular Displacement & Velocity of Node 12

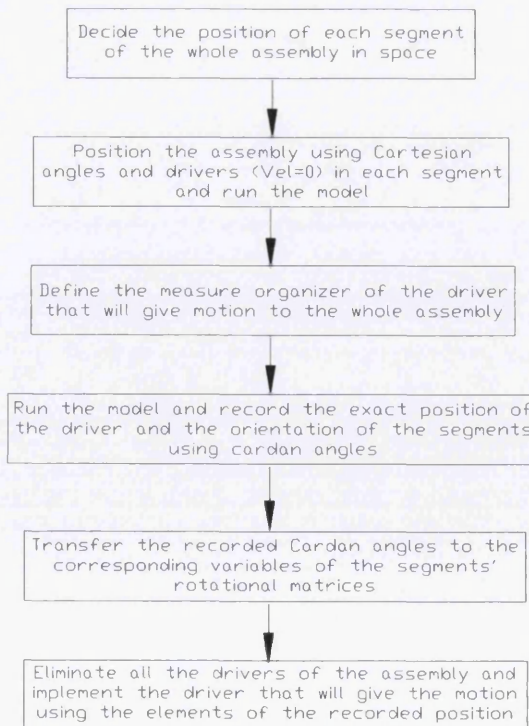


Figure 6.31 - Key Steps of the Triple Inverted Pendulum Motion Implementation

6.7 Conclusions

The research in the present chapter proved to be a worthy first “hands on” experience with the Anybody modelling system. The key factor of this experience was the implementation of a simple mechanical system as the triple inverted pendulum is, instead of going directly to the implementation of an extremely complex model as the human body is. The “step by step” development of the pendulum assembly gave confidence and provided a logical manner for the development of the whole assembly. Basic features of the modelling system that constitute a mechanical model, like the definition of the “Global Reference Frame”, the definition of a “Segment”, a “Joint”, a “Driver”, a “Kinematic Measure”, and a “Muscle”, were analysed. In addition, the way of importing real geometries and setting initial conditions in a model was investigated, while the different ways of giving motion to an assembly were clearly indicated.

CHAPTER 7

DEVELOPMENT OF A SIMPLIFIED HUMAN MODEL

7.1 Introduction

This study is intending to be the preparatory stage that will lead to the final implementation of the complicated parameterized model of the human body. A simplified model of the human body will be developed which can move in 3-D space and produce complex motions. In a way similar to this model being the base for the parameterized human model, the inverted pendulum model will be the basis for the current model. The final aim of this investigation will be the implementation of a lifting technique under an “extreme position” by the model. The reason that an “extreme position” has been chosen, is that if the model implements such a position successfully then this means that it can implement a wide of range of motions. The model will be developed in three stages; the lower extremities first, followed by the spine and the upper extremities at the end.

7.2 Structure

All the necessary AnyScript files for the implementation of this project will be contained in a Windows folder with name “Simplified_Human_Model”. This folder at its final stage will contain the following folders and “Anyfiles” (Figure 7.1):

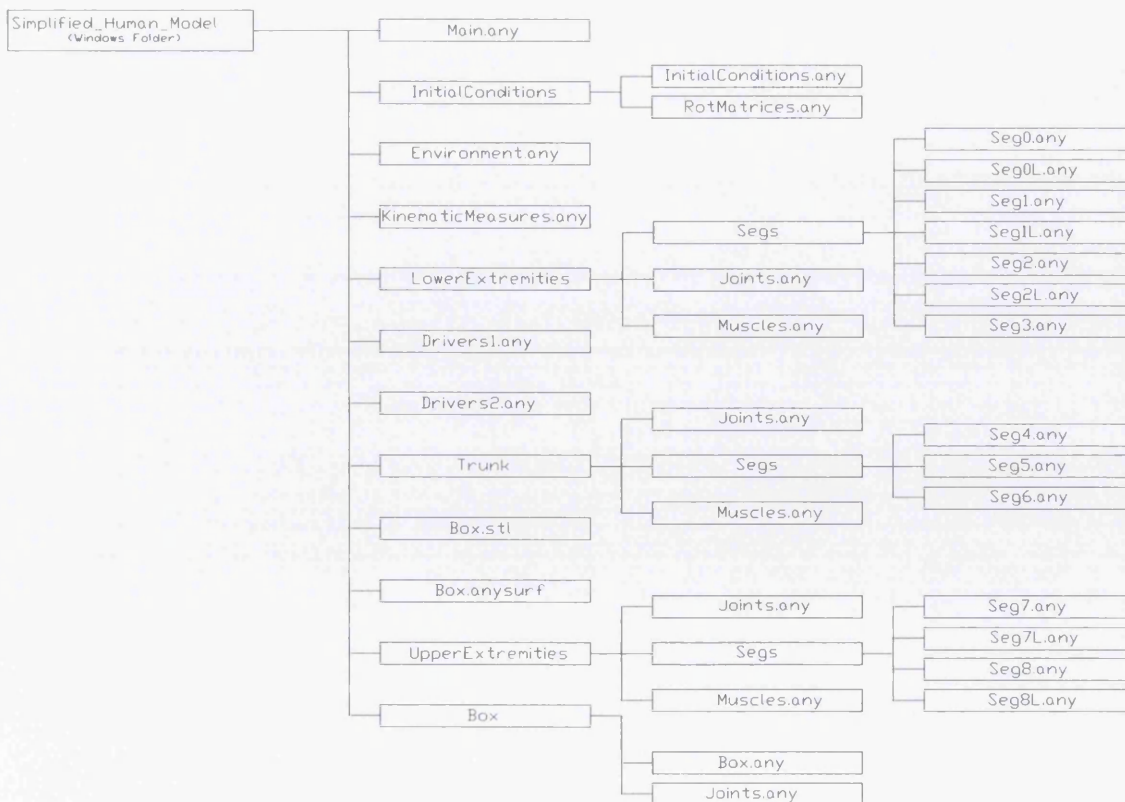


Figure 7.1 - Structure of the Simplified Human Model Windows Folder

The file Main.any is one more time, the “CPU” of the model. The Main.any file contains both the body model and the study section as it was mentioned in the previous chapter. The file Environment.any contains the following statement *AnyDrawRefFrame DrwGlobalRef = {};* for drawing the reference frame with the x, y and z axes. The final structure of the Main.any file will be as follows (Figure 7.2):

```

Main = {
  // The actual body model goes in this folder
  AnyFolder HumanModel = {

    // Global Reference Frame
    AnyFixedRefFrame GlobalRef = {
      #include "Environment.any"
    }; // Global reference frame

    //-----POSITION PARAMETERS-----
    AnyFolder Position = {

      //-----CENTRE OF MASS X COORDINATE-----
      AnyVar CoM_X = 0.244655; //Meters
      AnyVar CoM_XV = -0.03;

      //-----FEET-----
      //-----Right Foot XYZ Position (Heel)-----
      AnyVar RF_X = 0.1; //Meters
      AnyVar RF_XV = -0.05;

      AnyVar RF_Y = 0; //Meters
      AnyVar RF_YV = 0;

      AnyVar RF_Z = 0.3; //Meters
      AnyVar RF_ZV = 0;

      //-----Left Heel XYZ (Heel)-----
      AnyVar LF_X = 0; //Meters
      AnyVar LF_XV = 0;

      AnyVar LF_Y = 0.06; //Meters
      AnyVar LF_YV = -0.055;

      AnyVar LF_Z = 0.1; //Meters
      AnyVar LF_ZV = 0;

      //-----Right Heel XYZ Rotation (Heel)-----
      // AnyVar RFrot_X = 0 * pi/180; //Degrees
      // AnyVar RFrot_XV = 0 ;

      AnyVar RFrot_Y = -20 * pi/180; //Degrees
      AnyVar RFrot_YV = 0;

      AnyVar RFrot_Z = 0 * pi/180; //Degrees
      AnyVar RFrot_ZV = 0 ;

      //-----Left Heel XYZ Rotation (Heel)-----
      // AnyVar LFrot_X = 0 * pi/180; //Degrees
      // AnyVar LFrot_XV = 0 ;

      AnyVar LFrot_Y = 30 * pi/180; //Degrees
      AnyVar LFrot_YV = 0 ;

      AnyVar LFrot_Z = -20 * pi/180; //Degrees
      AnyVar LFrot_ZV = 0.3 ;

      //-----PELVIS (SEG3)-----
      //-----Seg3 X-Axis-----
      AnyVar P_X = 0.15 ; //Meters
      AnyVar P_XV = 0;
    }
  }
}

```

```

//-----Seg3 Y-Axis-----
AnyVar P_Y = 0.21004; //Meters !!!Original was 0.2
AnyVar P_YV = 0.15;
//-----Seg3 Z-Axis-----
AnyVar P_Z = 0.15 ; //Meters
AnyVar P_ZV = -0;

//-----Seg3 Rotation X-Axis-----
AnyVar PR_X = 0 * pi/180; //Degrees
AnyVar PR_XV = 0;
//-----Seg3 Rotation Y-Axis-----
AnyVar PR_Y = 0 * pi/180; //Degrees
AnyVar PR_YV = 0;
//-----Seg3 Rotation Z-Axis-----
AnyVar PR_Z = 0 * pi/180; //Degrees
AnyVar PR_ZV = 0;

}; //Position

```

//-----SEGMENTS & NODES-----

```

AnyFolder Segs = {

    AnyFolder LowerExtremities = {

        #include "LowerExtremities\Segs\Seg0.any"
        #include "LowerExtremities\Segs\Seg0L.any"
        #include "LowerExtremities\Segs\Seg1.any"
        #include "LowerExtremities\Segs\Seg1L.any"
        #include "LowerExtremities\Segs\Seg2.any"
        #include "LowerExtremities\Segs\Seg2L.any"
        #include "LowerExtremities\Segs\Seg3.any"
    };

    AnyFolder Trunk = {

        #include "Trunk\Segs\Seg4.any"
        #include "Trunk\Segs\Seg5.any"
        #include "Trunk\Segs\Seg6.any"
    };

    AnyFolder UpperExtremities = {

        #include "UpperExtremities\Segs\Seg7.any"
        #include "UpperExtremities\Segs\Seg7L.any"
        #include "UpperExtremities\Segs\Seg8.any"
        #include "UpperExtremities\Segs\Seg8L.any"
    };

    AnyFolder Box = {

        #include "Box\Box.any"
    };

}; //Segments

```

//-----INITIAL CONDITIONS-----

```

AnyFolder InitialConditions = {

    #include "InitialConditions\InitialConditions.any"
    #include "InitialConditions\RotMatrices.any"

}; //Initial Conditions

```

//-----JOINTS-----

```

AnyFolder Joints = {

    AnyFolder LowerExtremitiesJoints = {

```



```

    #include "LowerExtremities\Joints.any"
};

AnyFolder TrunkJoints = {

    #include "Trunk\Joints.any"
};

AnyFolder UpperExtremitiesJoints = {

    #include "UpperExtremities\Joints.any"
};

AnyFolder BoxJoints = {

    #include "Box\Joints.any"
};

}; // Joints

//-----KINEMATIC MEASURES-----

AnyFolder KinematicMeasures = {

    #include "KinematicMeasures.any"

}; // Kinematic Measures

//-----DRIVERS-----

AnyFolder Drivers = {

    #include "Drivers1.any"
    // #include "Drivers2.any"

}; // Drivers

//-----MUSCLES-----

AnyFolder Muscles = {

    AnyFolder LowerExtremitiesMuscles = {

        #include "LowerExtremities\Muscles.any"

    };

    AnyFolder TrunkMuscles = {

        #include "Trunk\Muscles.any"

    };

    AnyFolder UpperExtremitiesMuscles = {

        #include "UpperExtremities\Muscles.any"

    };

}; // Muscles
}; // HumanModel
// The study: Operations to be performed on the model
AnyBodyStudy MyStudy = {
    AnyFolder &Model = .HumanModel;
    RecruitmentSolver = MinMaxSimplex;
    Gravity = { 0.0, -9.81, 0.0 };
};
}; // Main

```

Figure 7.2 – Main.any File of Simplified Human Model

7.3 Lower Extremity

7.3.1 Segments

The segments that have been used are all cylinders with the same geometrical properties with the cylinders of the triple inverted pendulum (Figure 7.3). The only segment that is different is the segment3 which is a rectangular prism. The geometrical properties of segment3 are shown in Figure 7.4.

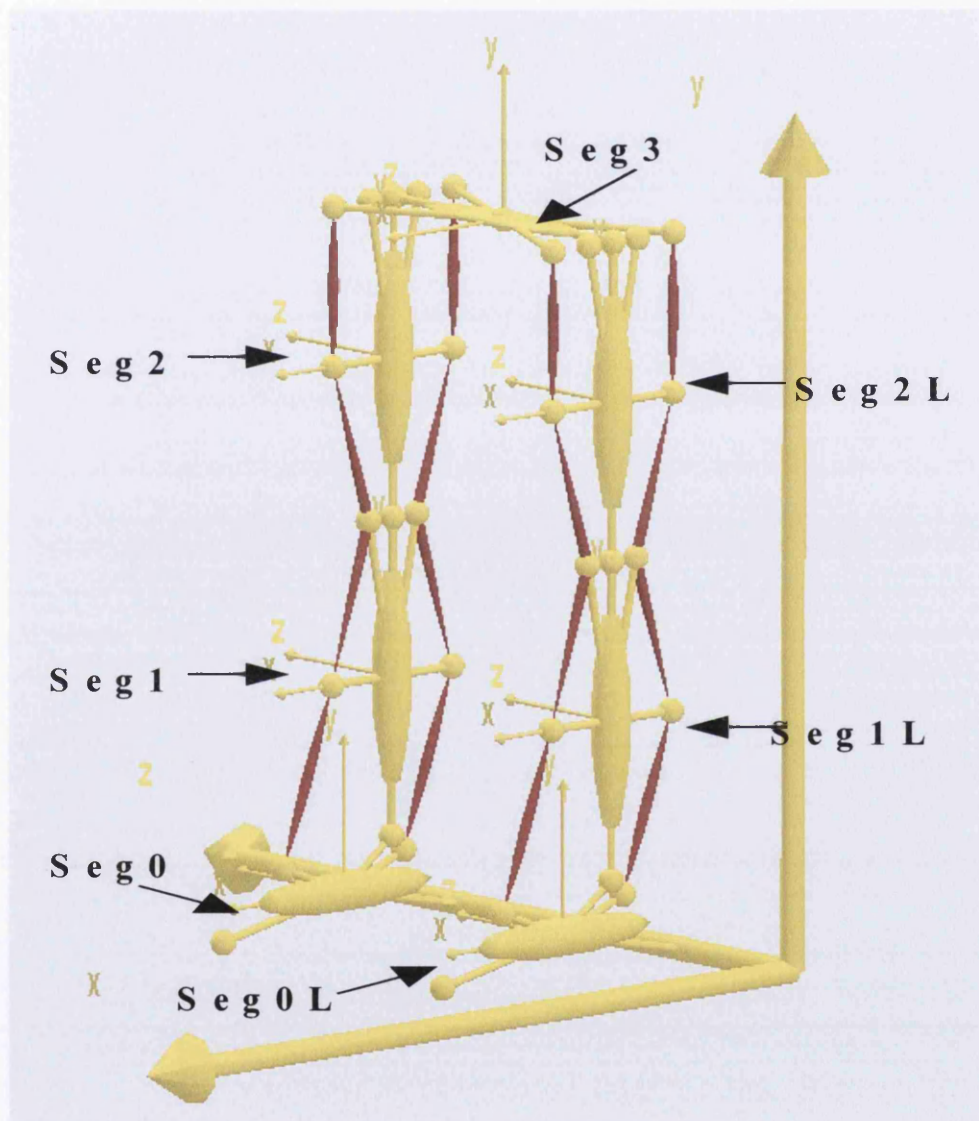
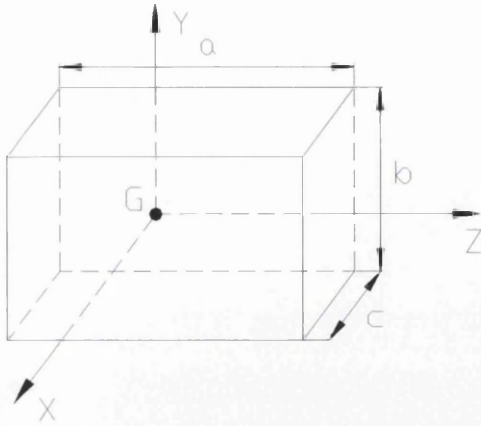


Figure 7.3 – Segments of Lower Extremity

Rectangular prism



Centre of mass (G): $G = \frac{b}{2}$

Mass: $m = 1\text{kg}$

Moment of Inertia (G)

$a = 0.2\text{m}, b = 0.05\text{m}, c = 0.1\text{m}$

$I_{x_G} = \frac{1}{12} \cdot m \cdot (a^2 + b^2) = 0.0035\text{kgm}^2$

$I_{y_G} = \frac{1}{12} \cdot m \cdot (a^2 + c^2) = 0.0042\text{kgm}^2$

$I_{z_G} = \frac{1}{12} \cdot m \cdot (b^2 + c^2) = 0.001\text{kgm}^2$

Figure 7.4 – Geometrical Properties of Segment 3

Table 7.1 below presents all the “anyfiles” of the segments that constitute the lower extremities.

<u>Seg0.any File</u>	<u>Seg0L.any File</u>
<pre>AnySeg Seg0 = { //r = {0.1,0.02,0.1}; Mass = 2; Jii = {0.0004,0.0069,0.0069}; //Axes = ...InitialConditions.R0; AnyRefNode Node0 = { sRel = {-0.04,0.03,0}; }; AnyRefNode Node1 = { sRel = {-0.05,0.02,0}; }; AnyRefNode Node2 = { sRel = {0.05,0.02,0}; }; AnyRefNode NodeHeel = { sRel = {-0.1,-0.02,0}; }; AnyRefNode NodeToeL = { sRel = {0.1,-0.02,0}; }; AnyDrawSeg DrwSeg = {}; };//Seg0</pre>	<pre>AnySeg Seg0L = { //r = {0.1,0.02,-0.1}; Mass = 2; Jii = {0.0004,0.0069,0.0069}; // Axes = ...InitialConditions.R0L; AnyRefNode Node0L = { sRel = {-0.04,0.03,0}; }; AnyRefNode Node1L = { sRel = {-0.05,0.02,0}; }; AnyRefNode Node2L = { sRel = {0.05,0.02,0}; }; AnyRefNode NodeHeelL = { sRel = {-0.1,-0.02,0}; }; AnyRefNode NodeToeL = { sRel = {0.1,-0.02,0}; }; AnyDrawSeg DrwSeg = {}; };//Seg0L</pre>
<u>Seg1.any File</u>	<u>Seg1L.any File</u>
<pre>AnySeg Seg1 = { //r = {0.1,0.15,0.1}; Axes = ...InitialConditions.R1; Mass = 2; Jii = {0.0069,0.0004,0.0069}; AnyRefNode Node3 = { sRel = {0,-0.1,0}; }; AnyRefNode Node4 = { sRel = {0,0.1,0}; }; AnyRefNode Node5 = { sRel = {-0.05,0,0}; }; AnyRefNode Node6 = { sRel = {0.05,0,0}; }; };</pre>	<pre>AnySeg Seg1L = { //r = {0.1,0.15,-0.1}; Axes = ...InitialConditions.R1L; Mass = 2; Jii = {0.0069,0.0004,0.0069}; AnyRefNode Node3L = { sRel = {0,-0.1,0}; }; AnyRefNode Node4L = { sRel = {0,0.1,0}; }; AnyRefNode Node5L = { sRel = {-0.05,0,0}; }; AnyRefNode Node6L = { sRel = {0.05,0,0}; }; };</pre>

```

AnyRefNode Node4B = {
  sRel = {-0.02,0.1,0};
};
AnyRefNode Node4F = {
  sRel = {0.02,0.1,0};
};
AnyDrawSeg DrwSeg = {};
}; //Seg1

```

```

AnyRefNode Node4BL = {
  sRel = {-0.02,0.1,0};
};
AnyRefNode Node4FL = {
  sRel = {0.02,0.1,0};
};
AnyDrawSeg DrwSeg = {};
}; //Seg1L

```

Seg2.any File

```

AnySeg Seg2 = {
// r = {0.1,0.35,0.1};
Axes = ...InitialConditions.R2;
Mass = 2;
Jii = {0.0069,0.0004,0.0069};
AnyRefNode Node7 = {
  sRel = {0,-0.1,0};
};
AnyRefNode Node8 = {
  sRel = {0,0.1,0};
};
AnyRefNode Node9 = {
  sRel = {-0.05,0,0};
};
AnyRefNode Node10 = {
  sRel = {0.05,0,0};
};
AnyRefNode Node8B = {
  sRel = {-0.02,0.1,0};
};
AnyRefNode Node8F = {
  sRel = {0.02,0.1,0};
};
AnyDrawSeg DrwSeg = {};
}; //Seg2

```

Seg2L.any File

```

AnySeg Seg2L = {
// r = {0.1,0.35,-0.1};
Axes = ...InitialConditions.R2L;
Mass = 2;
Jii = {0.0069,0.0004,0.0069};
AnyRefNode Node7L = {
  sRel = {0,-0.1,0};
};
AnyRefNode Node8L = {
  sRel = {0,0.1,0};
};
AnyRefNode Node9L = {
  sRel = {-0.05,0,0};
};
AnyRefNode Node10L = {
  sRel = {0.05,0,0};
};
AnyRefNode Node8BL = {
  sRel = {-0.02,0.1,0};
};
AnyRefNode Node8FL = {
  sRel = {0.02,0.1,0};
};
AnyDrawSeg DrwSeg = {};
}; //Seg2L

```

Seg3.any File

```

AnySeg Seg3 = {
Axes = ...InitialConditions.R3;
Mass = 1;
Jii = {0.0035, 0.0042, 0.001};
AnyRefNode NodeSeg3_0 = {
  sRel = {0,0,0};
};
AnyRefNode NodeSeg3_1 = {
  sRel = {0,0,0.1};
};
AnyRefNode NodeSeg3_2 = {
  sRel = {0,0,-0.1};
};
AnyRefNode NodeSeg3_3 = {
  sRel = {0.05,0,0.1};
};
AnyRefNode NodeSeg3_4 = {
  sRel = {0.05,0,-0.1};
};
AnyRefNode NodeSeg3_5 = {
  sRel = {-0.05,0,0.1};
};
AnyRefNode NodeSeg3_6 = {
  sRel = {-0.05,0,-0.1};
};
AnyDrawSeg DrwSeg = {};
}; //Seg3

```

Table 7.1 – Anyfiles of Lower Extremity Segments

7.3.2 Initial Conditions

The initial conditions have already been discussed in the previous chapter, so all the associated “anyfiles” are presented in the following table (Table 7.2).

<u>InitialConditions.any</u>	<u>RotMatrices.any File</u>
<pre>//-----ROTATION X-AXIS----- //-----Segment 1----- AnyVar alpha1 = 0; //-----Segment 1L----- AnyVar alpha1L = 0; //-----Segment 2----- AnyVar alpha2 = 0; //-----Segment 2L----- AnyVar alpha2L = 0; //-----Segment 3----- AnyVar alpha3 = 0; //-----ROTATION Y-AXIS----- //-----Segment 1----- AnyVar beta1 = 0; //-----Segment 1L----- AnyVar beta1L = 0; //-----Segment 2----- AnyVar beta2 = 0; //-----Segment 2L----- AnyVar beta2L = 0; //-----Segment 3----- AnyVar beta3 = 0; //-----ROTATION Z-AXIS----- //-----Segment 1----- AnyVar gamma1 = 100 * pi/180; // degrees //-----Segment 1L----- AnyVar gamma1L = 100 * pi/180; // degrees //-----Segment 2----- AnyVar gamma2 = -30 * pi/180; // degrees //-----Segment 2L----- AnyVar gamma2L = -30 * pi/180; // degrees //-----Segment 3----- AnyVar gamma3 = 0;</pre>	<pre>//-----SEGMENT1 ROTATION VECTOR----- AnyMat33 R1 = { {cos(beta1) * cos(gamma1) , cos(gamma1) * sin(beta1) * sin(alpha1) + sin(gamma1) * cos(alpha1) , sin(gamma1) * sin(alpha1) - cos(gamma1) * sin(beta1) * cos(alpha1)} , {-sin(gamma1) * cos(beta1) , cos(alpha1) * cos(gamma1) - sin(alpha1) * sin(beta1) * sin(gamma1) , sin(gamma1) * sin(beta1) * cos(alpha1) + cos(gamma1) * sin(alpha1)} , { sin(beta1) , -cos(beta1) * sin(alpha1) , cos(alpha1) * cos(beta1)}}; //-----SEGMENT1L ROTATION VECTOR----- AnyMat33 R1L = { {cos(beta1L) * cos(gamma1L) , cos(gamma1L) * sin(beta1L) * sin(alpha1L) + sin(gamma1L) * cos(alpha1L) , sin(gamma1L) * sin(alpha1L) - cos(gamma1L) * sin(beta1L) * cos(alpha1L)} , {-sin(gamma1L) * cos(beta1L) , cos(alpha1L) * cos(gamma1L) - sin(alpha1L) * sin(beta1L) * sin(gamma1L) , sin(gamma1L) * sin(beta1L) * cos(alpha1L) + cos(gamma1L) * sin(alpha1L)} , { sin(beta1L) , -cos(beta1L) * sin(alpha1L) , cos(alpha1L) * cos(beta1L)}}; //-----SEGMENT2 ROTATION VECTOR----- AnyMat33 R2 = { {cos(beta2) * cos(gamma2) , cos(gamma2) * sin(beta2) * sin(alpha2) + sin(gamma2) * cos(alpha2) , sin(gamma2) * sin(alpha2) - cos(gamma2) * sin(beta2) * cos(alpha2)} , {-sin(gamma2) * cos(beta2) , cos(alpha2) * cos(gamma2) - sin(alpha2) * sin(beta2) * sin(gamma2) , sin(gamma2) * sin(beta2) * cos(alpha2) + cos(gamma2) * sin(alpha2)} , { sin(beta2) , -cos(beta2) * sin(alpha2) , cos(alpha2) * cos(beta2)}};</pre>
<pre>//-----SEGMENT3 ROTATION VECTOR----- AnyMat33 R3 = { {cos(beta3) * cos(gamma3) , cos(gamma3) * sin(beta3) * sin(alpha3) + sin(gamma3) * cos(alpha3) , sin(gamma3) * sin(alpha3) - cos(gamma3) * sin(beta3) * cos(alpha3)} , {-sin(gamma3) * cos(beta3) , cos(alpha3) * cos(gamma3) - sin(alpha3) * sin(beta3) * sin(gamma3) , sin(gamma3) * sin(beta3) * cos(alpha3) + cos(gamma3) * sin(alpha3)} , { sin(beta3) , -cos(beta3) * sin(alpha3) , cos(alpha3) * cos(beta3)}};</pre>	<pre>//-----SEGMENT2L ROTATION VECTOR----- AnyMat33 R2L = { {cos(beta2L) * cos(gamma2L) , cos(gamma2L) * sin(beta2L) * sin(alpha2L) + sin(gamma2L) * cos(alpha2L) , sin(gamma2L) * sin(alpha2L) - cos(gamma2L) * sin(beta2L) * cos(alpha2L)} , {-sin(gamma2L) * cos(beta2L) , cos(alpha2L) * cos(gamma2L) - sin(alpha2L) * sin(beta2L) * sin(gamma2L) , sin(gamma2L) * sin(beta2L) * cos(alpha2L) + cos(gamma2L) * sin(alpha2L)} , { sin(beta2L) , -cos(beta2L) * sin(alpha2L) , cos(alpha2L) * cos(beta2L)}};</pre>

Table 7.2 – Initial Conditions Anyfiles

7.3.3 Joints

The joints that will be used in the lower extremity are shown in Figure 7.5 below, while the corresponding “anyfile” is depicted in Figure 7.6. It can be seen that there are no joints between the “feet” (*Seg0* & *Seg0L*) and the ground. This is done on purpose as the feet will have the ability to move in 3-D. This will be explained later in Section 7.2.

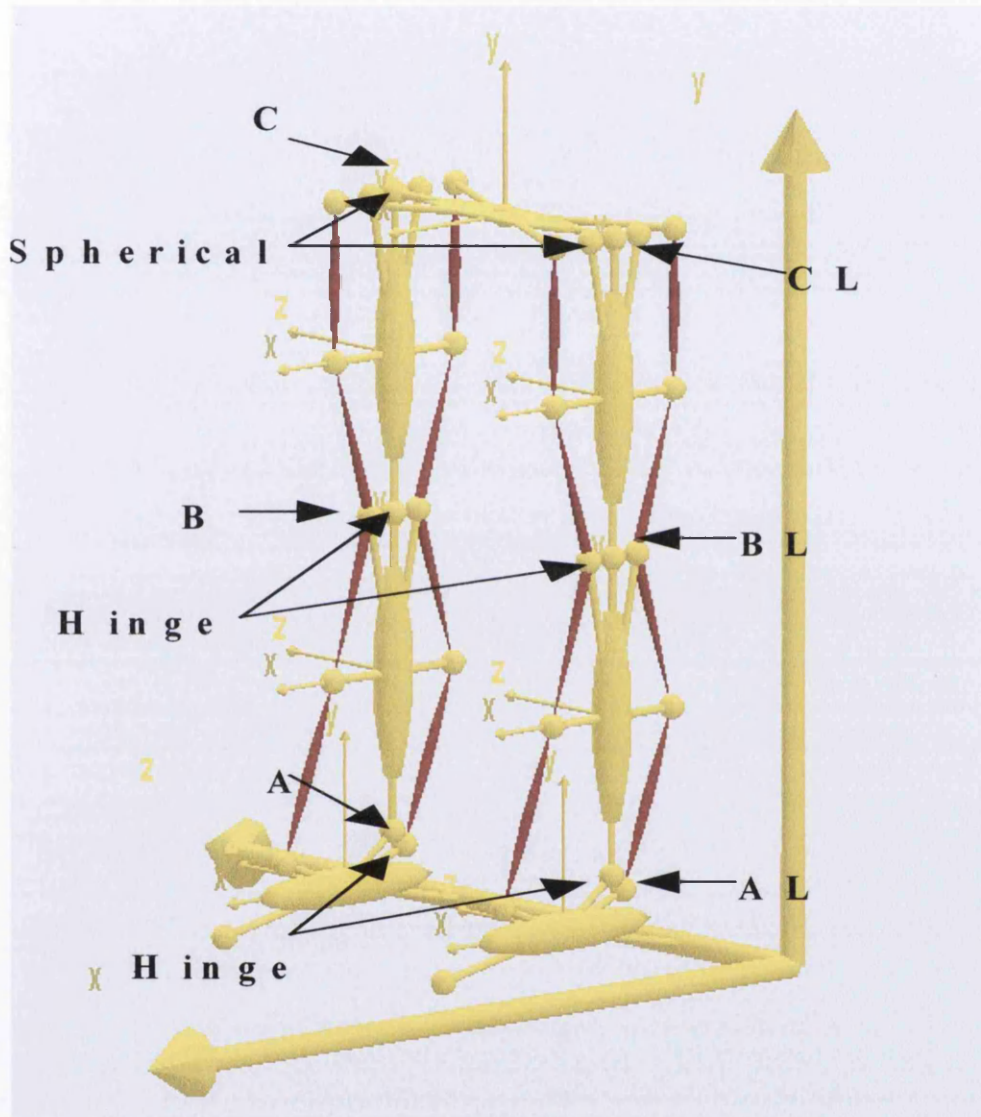


Figure 7.5 – Joints of Lower Extremity

Joints.any File

```
AnyRevoluteJoint A = {
  Axis=z;
  AnyRefNode &Node0 =...Segs.LowerExtremities.Seg0.Node0;
  AnyRefNode &Node3 =...Segs.LowerExtremities.Seg1.Node3;
}; //Joint A

AnyRevoluteJoint AL = {
  Axis=z;
  AnyRefNode &Node0L =...Segs.LowerExtremities.Seg0L.Node0L;
  AnyRefNode &Node3L =...Segs.LowerExtremities.Seg1L.Node3L;
}; //Joint AL

AnyRevoluteJoint B = {
  Axis = z;
  AnyRefNode &Node4 = ...Segs.LowerExtremities.Seg1.Node4;
  AnyRefNode &Node7 = ...Segs.LowerExtremities.Seg2.Node7;
}; //JointB

AnyRevoluteJoint BL = {
  Axis = z;
  AnyRefNode &Node4L = ...Segs.LowerExtremities.Seg1L.Node4L;
  AnyRefNode &Node7L = ...Segs.LowerExtremities.Seg2L.Node7L;
}; //JointBL

AnySphericalJoint C = {
  AnyRefNode &Node8 = ...Segs.LowerExtremities.Seg2.Node8;
  AnyRefNode &Node3_1 = ...Segs.LowerExtremities.Seg3.NodeSeg3_1;
}; //JointC

AnySphericalJoint CL = {
  AnyRefNode &Node8L = ...Segs.LowerExtremities.Seg2L.Node8L;
  AnyRefNode &Node3_2 = ...Segs.LowerExtremities.Seg3.NodeSeg3_2;
}; //JointCL
```

Figure 7.6 – Joints.any File of the Lower Extremity

7.3.4 Muscles

The muscles that have been used in this model are clearly displayed in Figure 7.7. One more time, the minimum number of muscles for the implementation of motion is used. The corresponding “anyfile” is shown below in Figure 7.8.

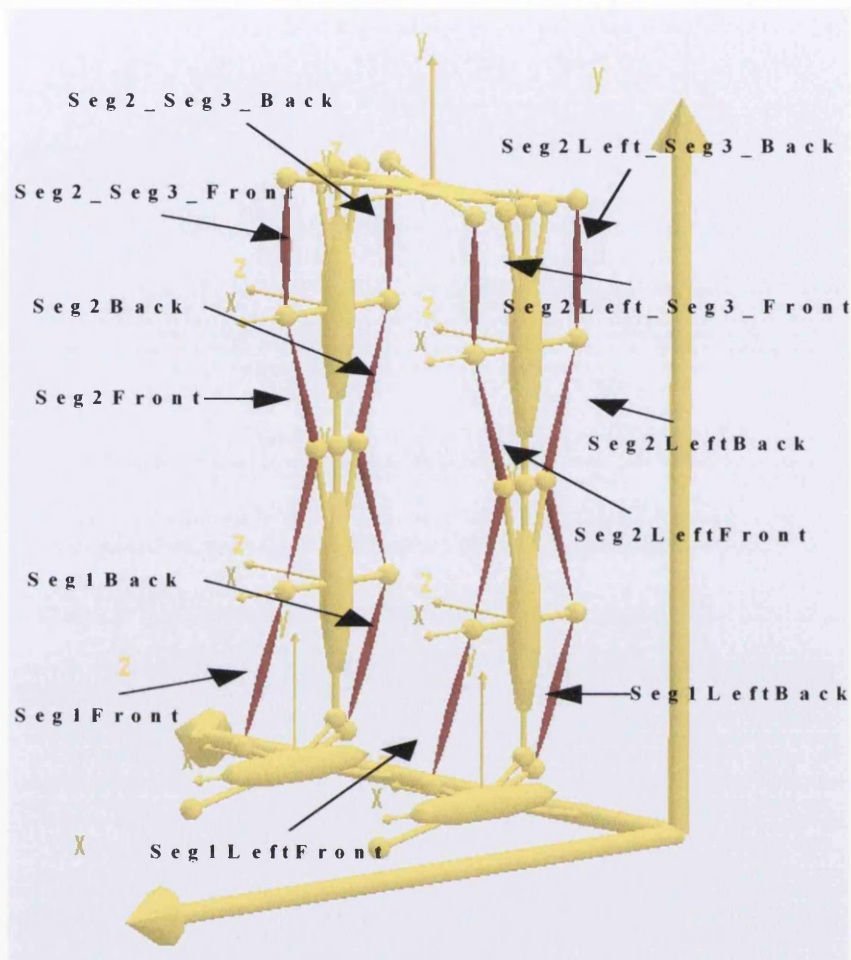


Figure 7.7 – Muscles of the Lower Extremity

Muscles.any File

```
//-----Simple muscle model with constant strength 300N-----  
AnyMuscleModel MusMd11 = {  
  FO = 10;  
}; //Muscle Strength  
//-----SEGMENT1 BACK-----  
AnyViaPointMuscle Seg1Back = {  
  AnyMuscleModel &MusMd11 = .MusMd11;  
  AnyRefNode &Node1 = ...Segs.LowerExtremities.Seg0.Node1;  
  AnyRefNode &Node5 = ...Segs.LowerExtremities.Seg1.Node5;  
  AnyDrawMuscle DrwMus = {};  
}; //Seg1_Back  
//-----LEFT SEGMENT1 BACK-----  
AnyViaPointMuscle Seg1LeftBack = {  
  AnyMuscleModel &MusMd11 = .MusMd11;  
  AnyRefNode &Node1L = ...Segs.LowerExtremities.Seg0L.Node1L;  
  AnyRefNode &Node5L = ...Segs.LowerExtremities.Seg1L.Node5L;  
  AnyDrawMuscle DrwMus = {};  
}; //Left Seg1_Back
```



```

//-----SEGMENT1 FRONT-----
AnyViaPointMuscle Seg1Front = {
    AnyMuscleModel &MusMdl1 = .MusMdl1;
    AnyRefNode &Node2 = ...Segs.LowerExtremities.Seg0.Node2;
    AnyRefNode &Node6 = ...Segs.LowerExtremities.Seg1.Node6;
    AnyDrawMuscle DrwMus = {};
}; //Seg1_Front

//-----LEFT SEGMENT1 FRONT-----
AnyViaPointMuscle Seg1LeftFront = {
    AnyMuscleModel &MusMdl1 = .MusMdl1;
    AnyRefNode &Node2L = ...Segs.LowerExtremities.Seg0L.Node2L;
    AnyRefNode &Node6L = ...Segs.LowerExtremities.Seg1L.Node6L;
    AnyDrawMuscle DrwMus = {};
}; //LeftSeg1_Front

//-----SEGMENT2 BACK-----
AnyViaPointMuscle Seg2Back = {
    AnyMuscleModel &MusMdl1 = .MusMdl1;
    AnyRefNode &Node5 = ...Segs.LowerExtremities.Seg1.Node5;
    AnyRefNode &Node4B = ...Segs.LowerExtremities.Seg1.Node4B;
    AnyRefNode &Node9 = ...Segs.LowerExtremities.Seg2.Node9;
    AnyDrawMuscle DrwMus = {};
}; //Seg2_Back

//-----LEFT SEGMENT2 BACK-----
AnyViaPointMuscle Seg2LeftBack = {
    AnyMuscleModel &MusMdl1 = .MusMdl1;
    AnyRefNode &Node5L = ...Segs.LowerExtremities.Seg1L.Node5L;
    AnyRefNode &Node4BL = ...Segs.LowerExtremities.Seg1L.Node4BL;
    AnyRefNode &Node9L = ...Segs.LowerExtremities.Seg2L.Node9L;
    AnyDrawMuscle DrwMus = {};
}; //Left Seg2_Back

//-----SEGMENT2 FRONT-----
AnyViaPointMuscle Seg2Front = {
    AnyMuscleModel &MusMdl1 = .MusMdl1;
    AnyRefNode &Node6 = ...Segs.LowerExtremities.Seg1.Node6;
    AnyRefNode &Node4F = ...Segs.LowerExtremities.Seg1.Node4F;
    AnyRefNode &Node10 = ...Segs.LowerExtremities.Seg2.Node10;
    AnyDrawMuscle DrwMus = {};
}; //Seg2_Front

//-----LEFT SEGMENT2 FRONT-----
AnyViaPointMuscle Seg2LeftFront = {
    AnyMuscleModel &MusMdl1 = .MusMdl1;
    AnyRefNode &Node6L = ...Segs.LowerExtremities.Seg1L.Node6L;
    AnyRefNode &Node4FL = ...Segs.LowerExtremities.Seg1L.Node4FL;
    AnyRefNode &Node10L = ...Segs.LowerExtremities.Seg2L.Node10L;
    AnyDrawMuscle DrwMus = {};
}; //Left Seg2_Front

//-----SEGMENT2_PELVIS BACK-----
AnyViaPointMuscle Seg2_Seg3_Back = {
    AnyMuscleModel &MusMdl1 = .MusMdl1;
    AnyRefNode &Node9 = ...Segs.LowerExtremities.Seg2.Node9;
    AnyRefNode &NodeSeg3_5 = ...Segs.LowerExtremities.Seg3.NodeSeg3_5;
    AnyDrawMuscle DrwMus = {};
}; //Seg2_Pelvis Back

//-----LEFT SEGMENT2_PELVIS BACK-----
AnyViaPointMuscle Seg2Left_Seg3_Back = {
    AnyMuscleModel &MusMdl1 = .MusMdl1;
    AnyRefNode &Node9L = ...Segs.LowerExtremities.Seg2L.Node9L;
    AnyRefNode &NodeSeg3_6 = ...Segs.LowerExtremities.Seg3.NodeSeg3_6;
    AnyDrawMuscle DrwMus = {};
}; //Left Seg2_Pelvis Back

//-----SEGMENT2_PELVIS FRONT-----
AnyViaPointMuscle Seg2_Seg3_Front = {
    AnyMuscleModel &MusMdl1 = .MusMdl1;
    AnyRefNode &Node10 = ...Segs.LowerExtremities.Seg2.Node10;
    AnyRefNode &NodeSeg3_3 = ...Segs.LowerExtremities.Seg3.NodeSeg3_3;
    AnyDrawMuscle DrwMus = {};
}; //Seg2_Pelvis Front

//-----LEFT SEGMENT2_PELVIS FRONT-----
AnyViaPointMuscle Seg2Left_Seg3_Front = {
    AnyMuscleModel &MusMdl1 = .MusMdl1;
    AnyRefNode &Node10L = ...Segs.LowerExtremities.Seg2L.Node10L;
    AnyRefNode &NodeSeg3_4 = ...Segs.LowerExtremities.Seg3.NodeSeg3_4;
    AnyDrawMuscle DrwMus = {};
}; //Seg2_Pelvis Front

```

Figure 7.8 – Muscles.any File of the Lower Extremity

7.3.5 Kinematic Measures & Drivers

In order to produce a wide range of movements by the lower extremity, one driver for each of the “feet” (*Seg0* & *Seg0L*) and one driver for the “pelvis” (*Seg3*) are needed. The feet will have the ability to move in 3-D and connected to the ground without the normal joints that have been used in the rest of the assembly. *Ground* is provided with the facility *AnyReacForce* which contains the “kinematic measures” that are associated with the motion of the particular segment. *AnyReacForce* provides a reaction force to the foot and in this particular case to the “heel” (*NodeHeel*, *NodeHeelL*) of both “feet”. The “kinematic measures” and the drivers that are associated with the right and left foot motion are presented in Tables 7.3 and 7.4 respectively. It has to be noted that the degree of freedom (3) which is for the rotation about x-axis is free, as this allows the production of extreme positions by the model. It has to also be mentioned that the various “input-variables” of the drivers can be defined at the *Main.any* file, as this file is the “CPU” of the model.

KinematicMeasures.any File

```
//-----RIGHT HEEL/GROUND KINEMATIC LINEAR-----  
AnyKinLinear RightHeelLin = {  
  AnyFixedRefFrame &Ground = ..GlobalRef;  
  AnyRefNode &Nodes = ..Segs.LowerExtremities.Seg0.NodeHeel;  
  Ref = 0;  
};  
//-----RIGHT HEEL/GROUND KINEMATIC ROTATIONAL-----  
AnyKinRotational RightHeelRot = {  
  AnyFixedRefFrame &Ground = ..GlobalRef;  
  AnyRefNode &Nodes = ..Segs.LowerExtremities.Seg0.NodeHeel;  
  Type = RotVector;  
};  
//-----RIGHT HEEL ORGANIZER-----  
AnyKinMeasureOrg RightHeelPos = {  
  AnyKinLinear &ref1 = .RightHeelLin;  
  AnyKinRotational &ref2 = .RightHeelRot;  
  MeasureOrganizer = {0,1,2,4,5};  
};
```

Drivers1.any File

```
//-----RIGHT HEEL DRIVER-----  
AnyKinEqSimpleDriver RightHeelDriver = {  
  AnyKinMeasureOrg &ref = ..KinematicMeasures.RightHeelPos;  
  DriverPos =  
{..Position.RF_X,..Position.RF_Y,..Position.RF_Z,..Position.RFrot_Y,..Position.RFrot_Z};  
  DriverVel =  
{..Position.RF_XV,..Position.RF_YV,..Position.RF_ZV,..Position.RFrot_YV,..Position.RFrot_ZV};  
  Reaction.Type = {0,0,0,0,0};  
};  
//-----RIGHT HEEL LIN/ROT REACTION-----  
AnyReacForce RightHeelToGroundReaction = {  
  AnyKinLinear &ref1 = ..KinematicMeasures.RightHeelLin;  
  AnyKinRotational &ref2 = ..KinematicMeasures.RightHeelRot;  
};
```

Table 7.3 – Kinematic Measures/Driver of Right Heel (*Seg0* - *NodeHeel*)

KinematicMeasures.any File

```
//-----LEFT HEEL/GROUND KINEMATIC LINEAR-----  
AnyKinLinear LeftHeelLLin = {  
  AnyFixedRefFrame &Ground = ..GlobalRef;  
  AnyRefNode &Nodes = ..Segs.LowerExtremities.Seg0L.NodeHeelL;  
  Ref = 0;  
};  
//-----LEFT HEEL/GROUND KINEMATIC ROTATIONAL-----  
AnyKinRotational LeftHeelLRot = {  
  AnyFixedRefFrame &Ground = ..GlobalRef;  
  AnyRefNode &Nodes = ..Segs.LowerExtremities.Seg0L.NodeHeelL;  
  Type = RotVector;  
};  
//-----LEFT HEEL ORGANIZER-----  
AnyKinMeasureOrg LeftHeelLPos = {  
  AnyKinLinear &ref1 = .LeftHeelLLin;  
  AnyKinRotational &ref2 = .LeftHeelLRot;  
  MeasureOrganizer = {0,1,2,4,5};  
};
```

Drivers1.any File

```
//-----LEFT HEEL DRIVER-----  
AnyKinEqSimpleDriver LeftHeelDriver = {  
  AnyKinMeasureOrg &ref = ..KinematicMeasures.LeftHeelLPos;  
  DriverPos =  
{..Position.LF_X,..Position.LF_Y,..Position.LF_Z,..Position.LFrot_Y,..Position.LFrot_Z};  
  DriverVel =  
{..Position.LF_XV,..Position.LF_YV,..Position.LF_ZV,..Position.LFrot_YV,..Position.LFrot_ZV};  
  Reaction.Type = {0,0,0,0,0};  
};  
//-----LEFT HEEL LIN/ROT REACTION-----  
AnyReacForce LeftHeelToGroundReactionL = {  
  AnyKinLinear &ref1 = ..KinematicMeasures.LeftHeelLLin;  
  AnyKinRotational &ref2 = ..KinematicMeasures.LeftHeelLRot;  
};
```

Table 7.4 – Kinematic Measures/Driver of Left Heel (*Seg0L - NodeHeelL*)

In addition to the above drivers, a full range of motions will be implemented with a driver that will drive the pelvis (*Seg3*). The *MeasureOrganizer* is {0,1,2,3,4,5} which means that both *x*, *y*, *z* translation and rotation are permitted. Table 7.5 shows the “kinematic measures” and the driver associated with *Seg3*.

KinematicMeasures.any File

```
//-----SEG3 KINEMATIC LINEAR-----  
AnyKinLinear Seg3Lin = {  
  AnyFixedRefFrame &Ground = ..GlobalRef;  
  AnyRefNode &Nodes = ..Segs.LowerExtremities.Seg3.NodeSeg3_0;  
};  
//-----SEG3 KINEMATIC ROTATIONAL-----  
AnyKinRotational Seg3Rot = {  
  AnyFixedRefFrame &Ground = ..GlobalRef;  
  AnyRefNode &Nodes = ..Segs.LowerExtremities.Seg3.NodeSeg3_0;  
  Type = RotVector;  
};  
//-----SEG3 ORGANIZER-----  
AnyKinMeasureOrg Seg3Pos = {  
  AnyKinLinear &ref1 = .Seg3Lin;  
  AnyKinRotational &ref2 = .Seg3Rot;  
  MeasureOrganizer = {0,1,2,3,4,5};  
};
```

Drivers1.any File

```
//-----SEG3 DRIVER-----  
AnyKinEqSimpleDriver Seg3Motion= {  
  AnyKinMeasureOrg &ref = ..KinematicMeasures.Seg3Pos;  
  DriverPos = {..Position.P_X,..Position.P_Y,..Position.P_Z,  
               ..Position.PR_X,..Position.PR_Y,..Position.PR_Z};  
  DriverVel = {..Position.P_XV,..Position.P_YV,..Position.P_ZV,  
               ..Position.PR_XV,..Position.PR_YV,..Position.PR_ZV};  
  Reaction.Type = {0,0,0,0,0,0};  
};
```

Table 7.5 – Kinematic Measures/Driver of Pelvis (Seg3 - NodeSeg3_0)

A very important feature of this way of motion implementation, is that the exact orientation (*Cardan* angles) of segments *Seg0*, *Seg0L*, *Seg1*, *Seg1L*, *Seg2*, *Seg2L* and *Seg3* is not needed. The only parameters in the *InitialConditions.any* file that have to be defined are those that are associated with the rotation about the z-axis of *Seg1/1L* and *Seg2/2L*. These are: *gamma1*, *gamma1L*, *gamma2* and *gamma2L*. *Gamma1* and *gamma1L* must have clockwise rotation while *gamma2* and *gamma2L* anticlockwise. The value of these parameters depends on the complexity of the motion and specifically the distance of the pelvis (*Seg3*) from the ground and the rotation of the feet (*Seg0* and *Seg0L*). For simple motions sometimes what is needed is only let's say 5 degrees rotation of just *gamma1* and *gamma1L*. In case that the model does not work with the current parameters, then, either these parameters have to be increased or the y-coordinate of the pelvis (*Seg3*) has to be increased. It has to also be noted that because of the actual size of the segments, sometimes the motion is not possible. Some of the values of *gamma1*, *gamma1L*, *gamma2* and *gamma2L* that permit a wide range of motion are 120, 120, -30 and -30 degrees respectively.

Finally, once all the initial conditions, kinematic measures and drivers are defined, it is time to define the inputs of the drivers which are all contained in the *Position* “anyfolder” in the *Main.any* file. Figure 7.9 below represents the *Position* “anyfolder” with the various inputs, while Figure 7.10 shows the resultant motion of these inputs. The coordinates of the right heel are (0.1m, 0m, 0.3m), and it is rotated about y-axis -20 degrees. The coordinates of the left heel are (0m, 0.06m, 0.1m), and it is rotated 30 degrees about y-axis and -20 about z-axis. The coordinates of the pelvis are (0.15m, 0.2m, 0.15m). The linear velocity of the pelvis along the y-axis is 0.15m/sec, while the linear velocity of the right heel along the x-axis is -0.05m/sec. The linear velocity of the left heel is -0.055m/sec along the y-axis, while its angular velocity about the z-axis is 0.3rad/sec.

Main.any File

```
//-----POSITION PARAMETERS-----  
AnyFolder Position = {  
  //-----FEET-----  
  //-----Right Foot XYZ Position (Heel)-----  
  AnyVar RF_X = 0.1; //Meters  
  AnyVar RF_XV = -0.05;  
  
  AnyVar RF_Y = 0; //Meters  
  AnyVar RF_YV = 0;  
  
  AnyVar RF_Z = 0.3; //Meters  
  AnyVar RF_ZV = 0;
```

```

//-----Left Foot XYZ (Heel)-----
AnyVar LF_X = 0; //Meters
AnyVar LF_XV = 0;

AnyVar LF_Y = 0.06; //Meters
AnyVar LF_YV = -0.055;

AnyVar LF_Z = 0.1; //Meters
AnyVar LF_ZV = 0;

//-----Right Foot XYZ Rotation (Heel)-----
// AnyVar RFrot_X = 0 * pi/180; //Degrees
// AnyVar RFrot_XV = 0 ;

AnyVar RFrot_Y = -20 * pi/180; //Degrees
AnyVar RFrot_YV = 0;

AnyVar RFrot_Z = 0 * pi/180; //Degrees
AnyVar RFrot_ZV = 0 ;

//-----Left Foot XYZ Rotation (Heel)-----
// AnyVar LFrot_X = 0 * pi/180; //Degrees
// AnyVar LFrot_XV = 0 ;

AnyVar LFrot_Y = 30 * pi/180; //Degrees
AnyVar LFrot_YV = 0 ;

AnyVar LFrot_Z = -20 * pi/180; //Degrees
AnyVar LFrot_ZV = 0.3 ;

//-----PELVIS (SEG3)-----
//-----Seg3 X-Axis-----
AnyVar P_X = 0.15 ; //Meters
AnyVar P_XV = 0;

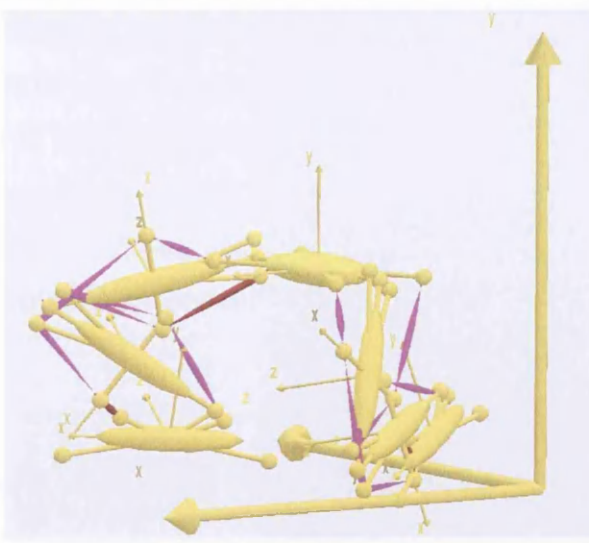
//-----Seg3 Y-Axis-----
AnyVar P_Y = 0.2; //Meters
AnyVar P_YV = 0.15;
//-----Seg3 Z-Axis-----
AnyVar P_Z = 0.15 ; //Meters
AnyVar P_ZV = -0;

//-----Seg3 Rotation X-Axis-----
AnyVar PR_X = 0 * pi/180; //Degrees
AnyVar PR_XV = 0;
//-----Seg3 Rotation Y-Axis-----
AnyVar PR_Y = 0 * pi/180; //Degrees
AnyVar PR_YV = 0;
//-----Seg3 Rotation Z-Axis-----
AnyVar PR_Z = 0 * pi/180; //Degrees
AnyVar PR_ZV = 0;

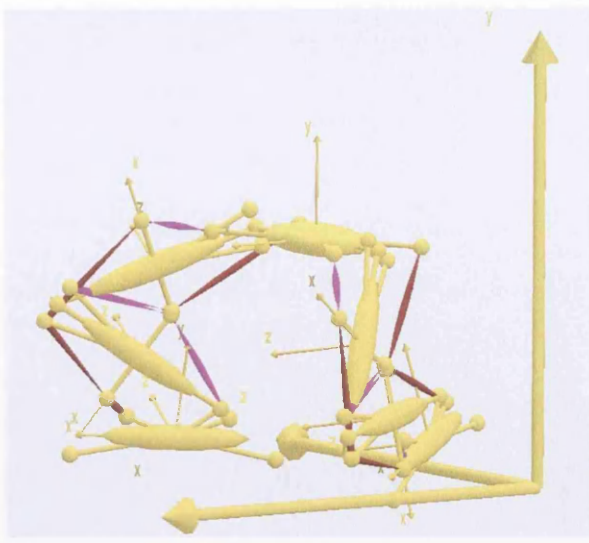
}; //Position

```

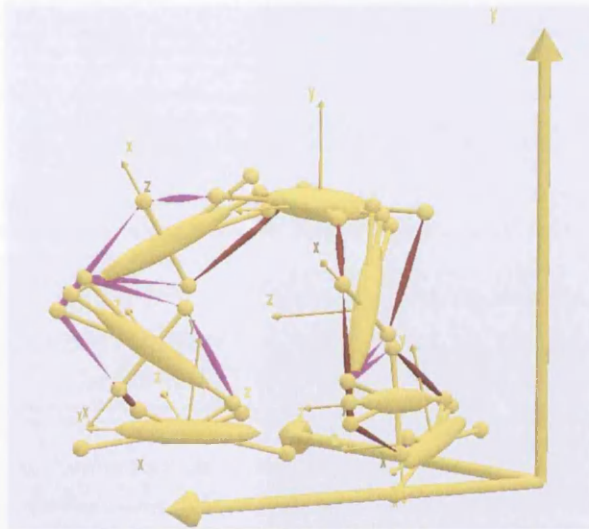
Figure 7.9– *Position Anyfolder of the Main.any file*



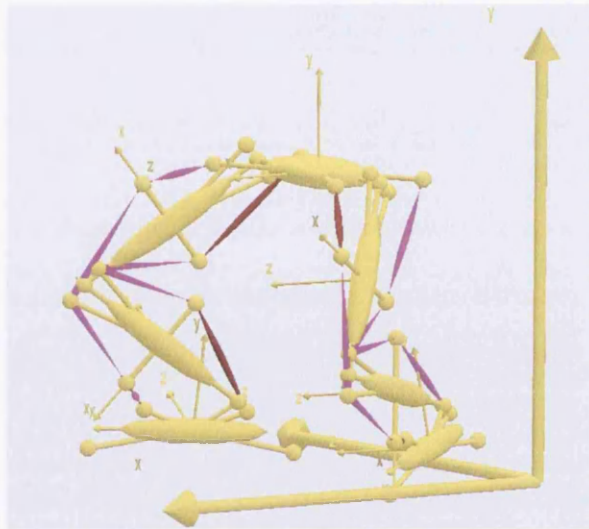
t = 0



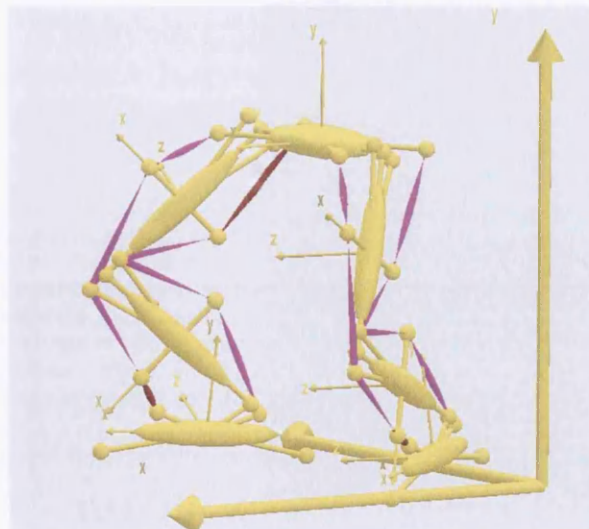
t = 20



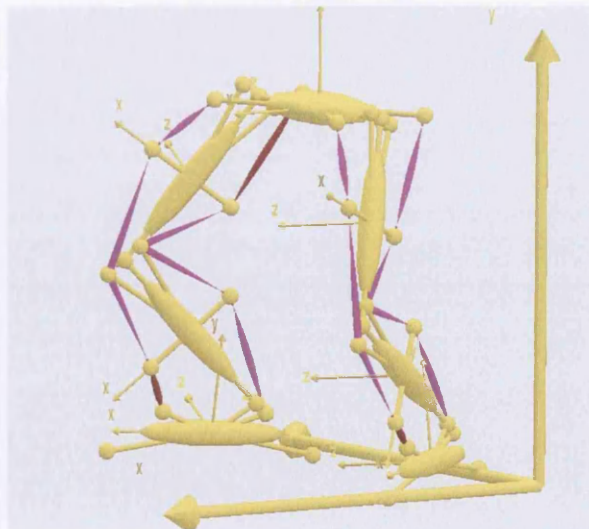
t = 40



t = 60



t = 80



t = 100

Figure 7.10 – Motion of Lower Extremity

This investigation will end with the addition of a very useful element that Anybody provides.

The model will be enhanced with a centre of mass (*CoM*) driver. The facility *AnyKinCoM* is used, which measures the centre of mass of a collection of rigid bodies. Then, with the addition of a *CoM* driver, the *CoM* of an assembly can be maintained in a specific position or can move in space, depending on the available degrees of freedom. In this particular case, one degree of freedom (translation/*x*-axis) is taken from the pelvis driver, and is given to the *CoM* driver. The corresponding “anyfiles” for such an implementation are presented in Table 7.6.

KinematicMeasures.any File

```
//-----SEG3 ORGANIZER-----
AnyKinMeasureOrg Seg3Pos = {
  AnyKinLinear &ref1 = .Seg3Lin;
  AnyKinRotational &ref2 = .Seg3Rot;
  MeasureOrganizer = {1,2,3,4,5};
};

//-----CENTRE OF MASS MEASURE-----
AnyKinCoM CoMMeasure = {
  AnySeg &seg0 = ..Segs.LowerExtremities.Seg0;
  AnySeg &seg0L = ..Segs.LowerExtremities.Seg0L;
  AnySeg &seg1 = ..Segs.LowerExtremities.Seg1;
  AnySeg &seg1L = ..Segs.LowerExtremities.Seg1L;
  AnySeg &seg2 = ..Segs.LowerExtremities.Seg2;
  AnySeg &seg2L = ..Segs.LowerExtremities.Seg2L;
  AnySeg &seg3 = ..Segs.LowerExtremities.Seg3;
}; //CentreOfMassMeasure
```

Drivers1.any File

```
//-----SEG3 DRIVER-----
AnyKinEqSimpleDriver Seg3Motion= {
  AnyKinMeasureOrg &ref = ..KinematicMeasures.Seg3Pos;
  DriverPos = {..Position.P_Y,..Position.P_Z,
    ..Position.PR_X,..Position.PR_Y,..Position.PR_Z};
  DriverVel = {..Position.P_YV,..Position.P_ZV,
    ..Position.PR_XV,..Position.PR_YV,..Position.PR_ZV};
  Reaction.Type = {0,0,0,0,0};
};

//-----CENTRE OF MASS DRIVER-----
AnyKinEqSimpleDriver CoMDriver = {
  AnyKinCoM &Segs = ..KinematicMeasures.CoMMeasure;
  MeasureOrganizer = {0};
  DriverPos = {..Position.CoM_X};
  DriverVel = {..Position.CoM_XV};
  Reaction.Type = {0};
}; //CentreOfMassDriver
```

Table 7.6 – Kinematic Measure/Driver of Centre of Mass of the Lower Extremity

Of course, before the elimination of the degree of freedom of the pelvis and the activation of the *CoM* driver a few steps have to be done. The first step is to place the model to the desired position. The second is to record the position (**accurate to six decimal places**) of the *CoM* (*AnyKinCoM* “kinematic measure” is activated), after the reloading of the model, at the time “**study operation: Initial conditions are fully**

updated". Then, the degree of freedom can be eliminated, transfer the recorded coordinate to the statement *DriverPos* of the *CoM* driver and activate the *CoM* driver. In this investigation the *CoM* x-coordinate is $0.181230m$ and in order to make the motion a bit more complicated, the *CoM* is moving with linear velocity, along x-axis, of $-0.03m/sec$. Figure 7.11 below, shows the inputs of the *CoM* driver that are included in the *Position* "anyfolder" of the *Main.any* file. Figure 7.12 presents time steps of the motion of the lower extremity with the *CoM* driver while Figure 7.13 shows the linear displacement of the centre of mass. Analysis of the muscle activity, the muscle forces and the joint loads (reaction forces) could be implemented as well. For illustrative purpose, Figure 7.14 represents the force and the activity of the *Seg2Front* muscle and Figure 7.15 the joint *C* reaction forces.

```

Main.any File
//-----CENTRE OF MASS X COORDINATE-----
AnyVar CoM_X = 0.181230; //Meters
AnyVar CoM_XV = -0.03;
```

Figure 7.11 – Inputs of CoM driver

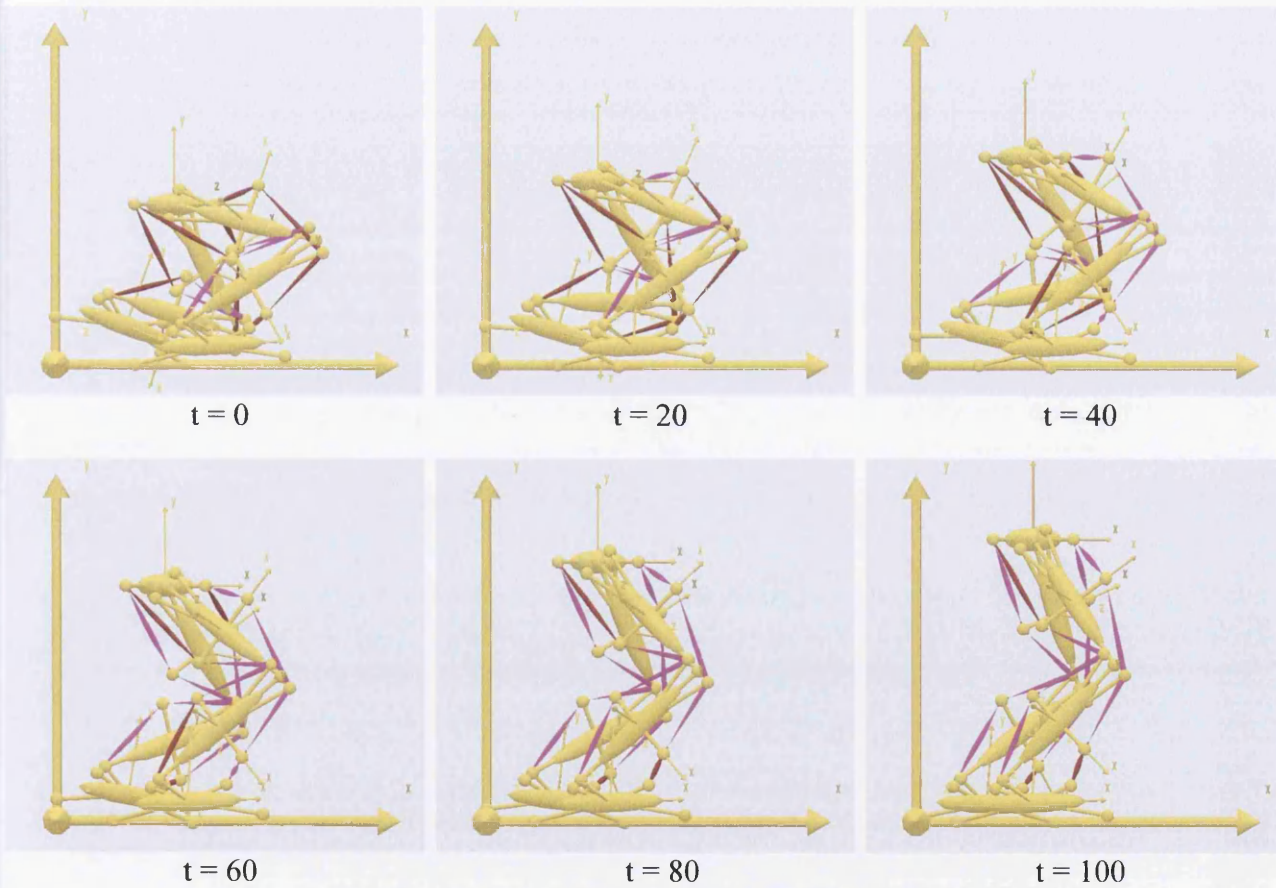


Figure 7.12 – Motion of Lower Extremity with CoM Drive

As it is shown in Figure 7.12 above, muscles are over-activated because of the low muscle force that has been chosen which is $F_0 = 10N$. This is clearly shown in Figure 7.14 below, which indicates that the *Seg2Front* muscle for the first 35 time steps does not work while after that it is over-activated with a peak value of 2400% as at that time step the muscle force needed to move the assembly is $240N$ while the

Chapter 7
 muscle force is $F_0=10N$. Figure 7.15 below indicates the reaction forces of joint C in x, y & z directions in which for instance at time step 14 the joint load at the x -direction is $-83N$, at the y -direction is $150N$ while at the z -direction is $-33N$.

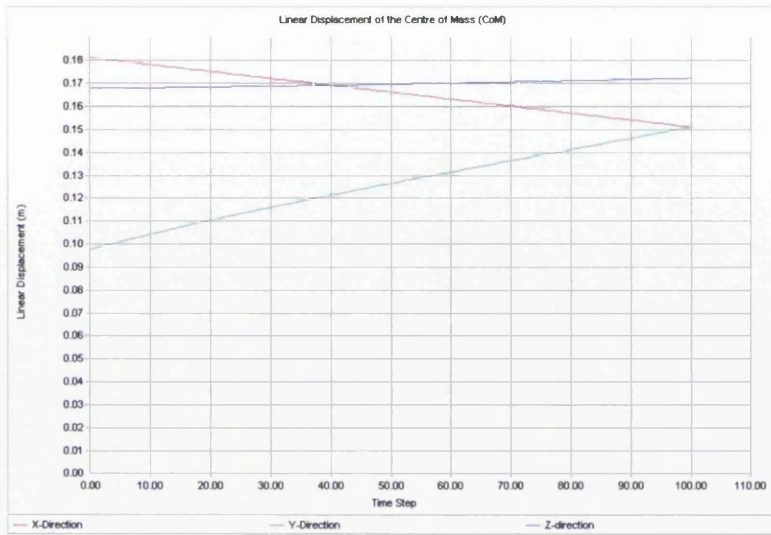


Figure 7.13 – Linear Displacement of CoM

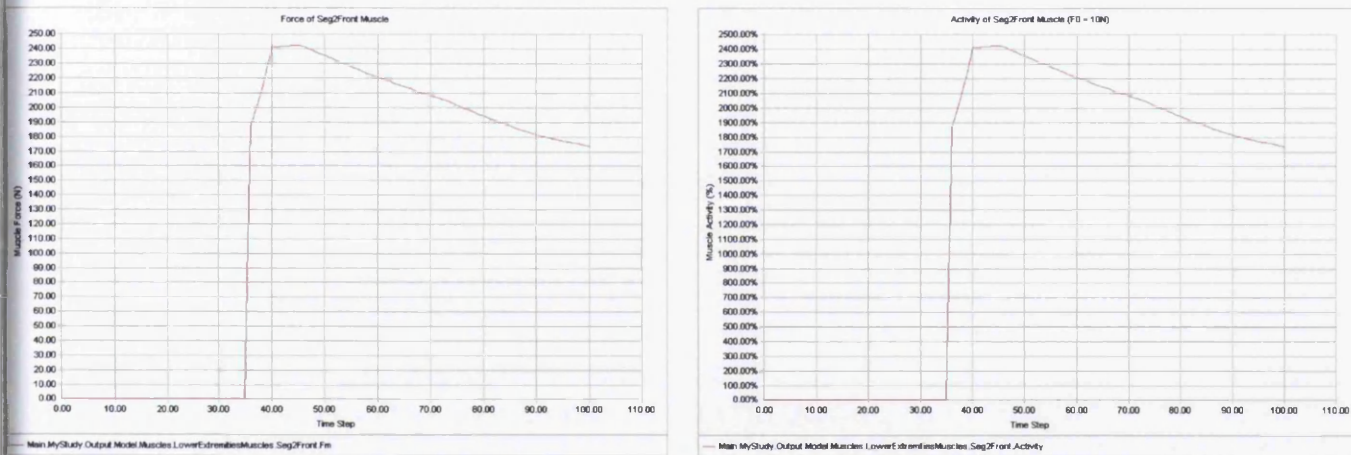


Figure 7.14 – F_m and Activity of $Seg2Front$ Muscle

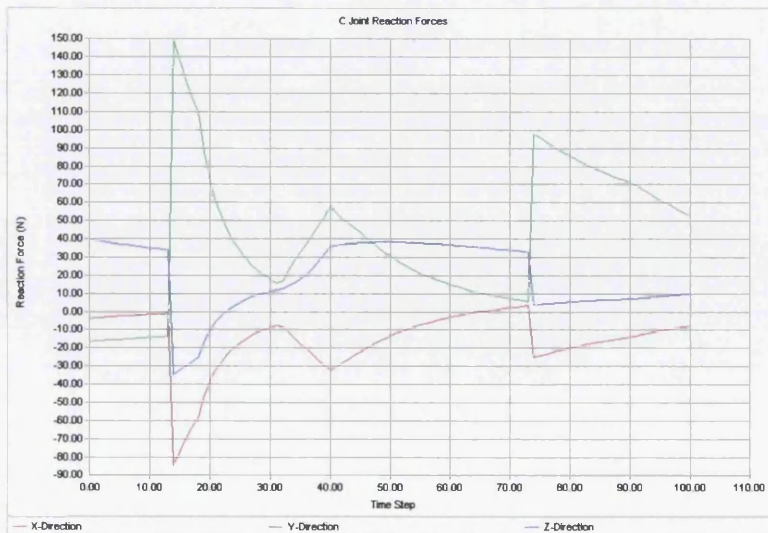


Figure 7.15 – C Joint Reaction Forces

7.4.1 Segments

Three segments constitute the trunk (Figure 7.16). These segments are all spheres and their geometrical properties are displayed in Figure 7.17.

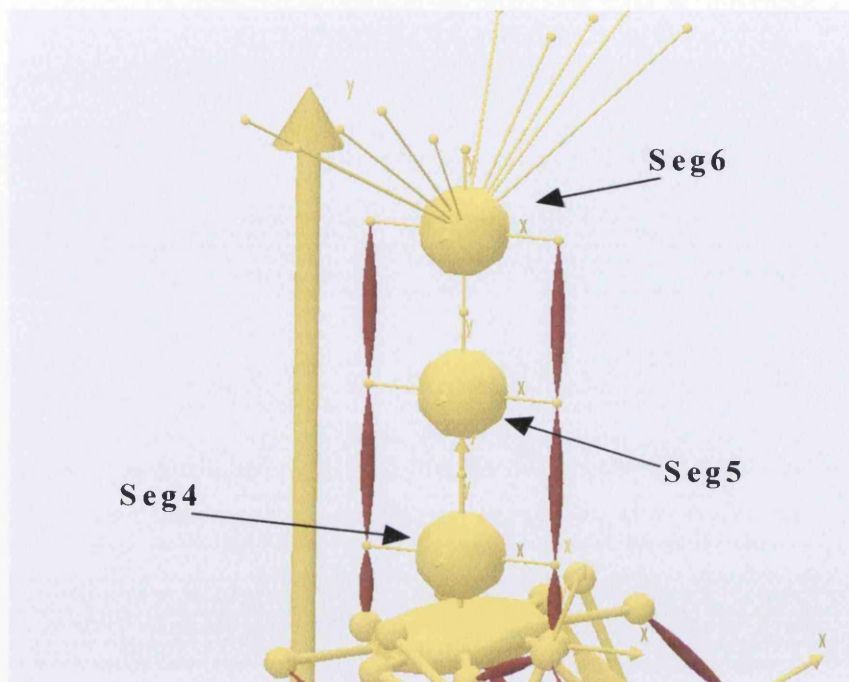


Figure 7.16 – Segments of Trunk

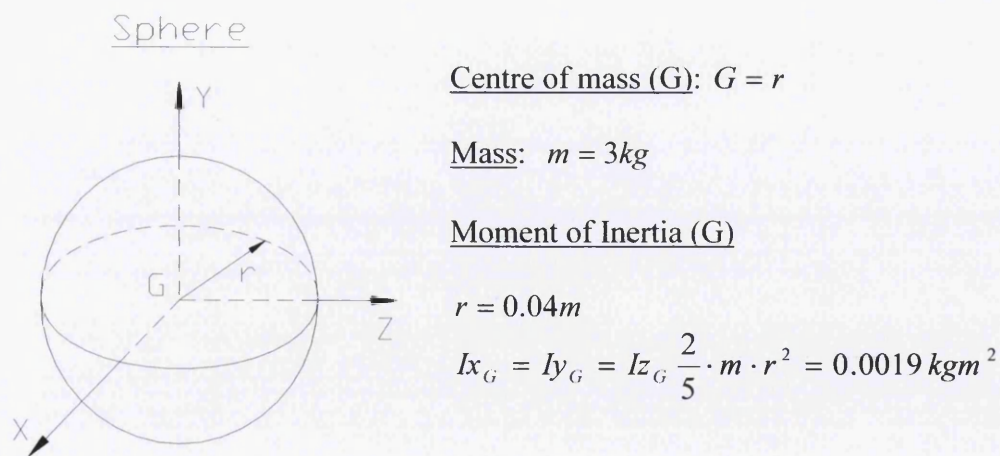


Figure 7.17 – Geometrical Properties of Trunk Segments

Table 7.7 below presents all the “anyfiles” of the segments that constitute the trunk.

Seg4.any File

```
//-----SEGMENT 4-----  
AnySeg Seg4 = {  
  //r= {0.1,0.51,0};  
  //Axes = ..InitialConditions.Seg4;  
  Mass = 3;  
  Jii = {0.0019,0.0019,0.0019};  
  AnyRefNode NodeSeg4_0 = {  
    sRel = {0,-0.04,0};  
  };  
  AnyRefNode NodeSeg4_1 = {  
    sRel = {0,0.04,0};  
  };  
  AnyRefNode NodeSeg4_2 = {  
    sRel = {-0.05,0,0};  
  };  
  AnyRefNode NodeSeg4_3 = {  
    sRel = {0.05,0,0};  
  };  
  AnyDrawSeg DrwSeg = { };  
}; //Seg4
```

Seg5.any File

```
//-----SEGMENT 5-----  
AnySeg Seg5 = {  
  //r= {0.1,0.59,0};  
  //Axes = ..InitialConditions.Seg5;  
  Mass = 3;  
  Jii = {0.0019,0.0019,0.0019};  
  AnyRefNode NodeSeg5_0 = {  
    sRel = {0,-0.04,0};  
  };  
  AnyRefNode NodeSeg5_1 = {  
    sRel = {0,0.04,0};  
  };  
  AnyRefNode NodeSeg5_2 = {  
    sRel = {-0.05,0,0};  
  };  
  AnyRefNode NodeSeg5_3 = {  
    sRel = {0.05,0,0};  
  };  
  AnyDrawSeg DrwSeg = { };  
}; //Seg5
```

Seg6.any File

```
//-----SEGMENT 6-----  
AnySeg Seg6 = {  
  //r= {0.1,0.7,0};  
  //Axes = ..InitialConditions.Spinel;  
  Mass = 3;  
  Jii = {0.0019,0.0019,0.0019};  
  AnyRefNode NodeSeg6_0 = {  
    sRel = {0,-0.04,0};  
  };  
  AnyRefNode NodeSeg6_1 = {  
    sRel = {0,0.04,0};  
  };  
  AnyRefNode NodeSeg6_2 = {  
    sRel = {-0.05,0,0};  
  };  
  AnyRefNode NodeSeg6_3 = {  
    sRel = {0.05,0,0};  
  };  
  AnyRefNode NodeSeg6_4 = {  
    sRel = {0,0.077,0.15};  
  };  
  AnyRefNode NodeSeg6_5 = {  
    sRel = {0,0.077,-0.15};  
  };  
  AnyRefNode NodeSeg6_6 = {  
    sRel = {0.05,0.077,0.15};  
  };  
};
```

```
AnyRefNode NodeSeg6_7 = {  
  sRel = {-0.05,0.077,0.15};  
};  
AnyRefNode NodeSeg6_8 = {  
  sRel = {0.05,0.077,-0.15};  
};  
AnyRefNode NodeSeg6_9 = {  
  sRel = {-0.05,0.077,-0.15};  
};  
AnyRefNode NodeSeg6_10 = {  
  sRel = {0,0.077,0.2};  
};  
AnyRefNode NodeSeg6_11 = {  
  sRel = {0,0.077,0.1};  
};  
AnyRefNode NodeSeg6_12 = {  
  sRel = {0,0.077,-0.2};  
};  
AnyRefNode NodeSeg6_13 = {  
  sRel = {0,0.077,-0.1};  
};  
AnyDrawSeg DrwSeg = { };  
}; //Seg6
```

Table 7.7 – Anyfiles of the Trunk Segments

7.4.2 Joints

The hinge joints (z-rotational axis) that will be used in the trunk are shown in Figure 7.18 below, while the corresponding “anyfile” in Figure 7.19.

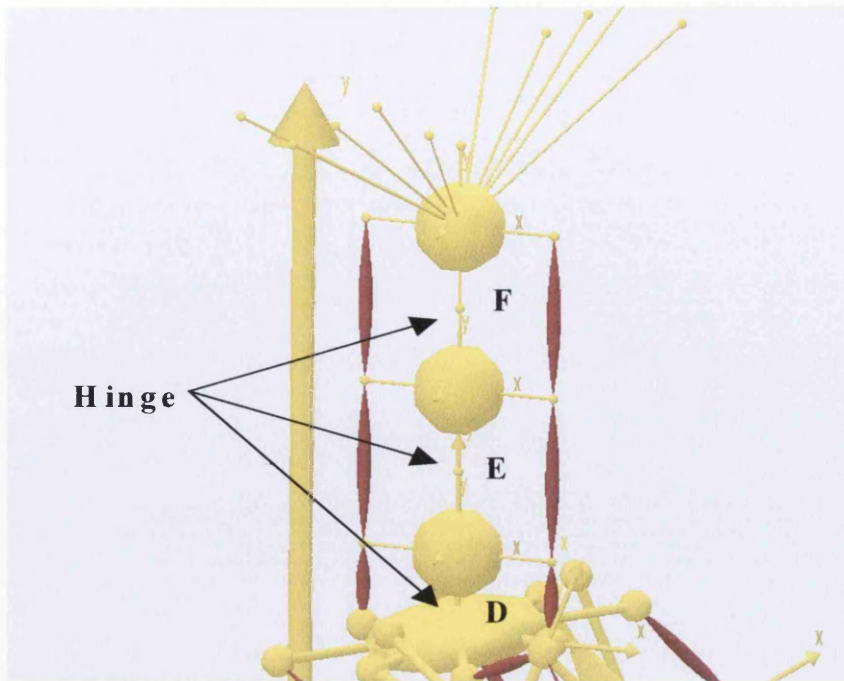


Figure 7.18 – Joints of the Trunk

Joints.any File

```

AnyRevoluteJoint D = {
  Axis=z;
  AnyRefNode &NodeS3_0 =...Segs.LowerExtremities.Seg3.NodeSeg3_0;
  AnyRefNode &NodeS4_0 =...Segs.Trunk.Seg4.NodeSeg4_0;
}; //Joint D

AnyRevoluteJoint E = {
  Axis=z;
  AnyRefNode &NodeS4_1 =...Segs.Trunk.Seg4.NodeSeg4_1;
  AnyRefNode &NodeS5_0 =...Segs.Trunk.Seg5.NodeSeg5_0;
}; //Joint E

AnyRevoluteJoint F = {
  Axis=z;
  AnyRefNode &NodeS5_1 =...Segs.Trunk.Seg5.NodeSeg5_1;
  AnyRefNode &NodeS6_0 =...Segs.Trunk.Seg6.NodeSeg6_0;
}; //Joint F

```

Figure 7.19 – Joints.any File of the Trunk

7.4.3 Muscles

The muscles that have been used in the trunk model are clearly displayed in Figure 7.20. One more time, the minimum number of muscles for the implementation of motion is used. The corresponding “anyfile” is shown below in Figure 7.21.

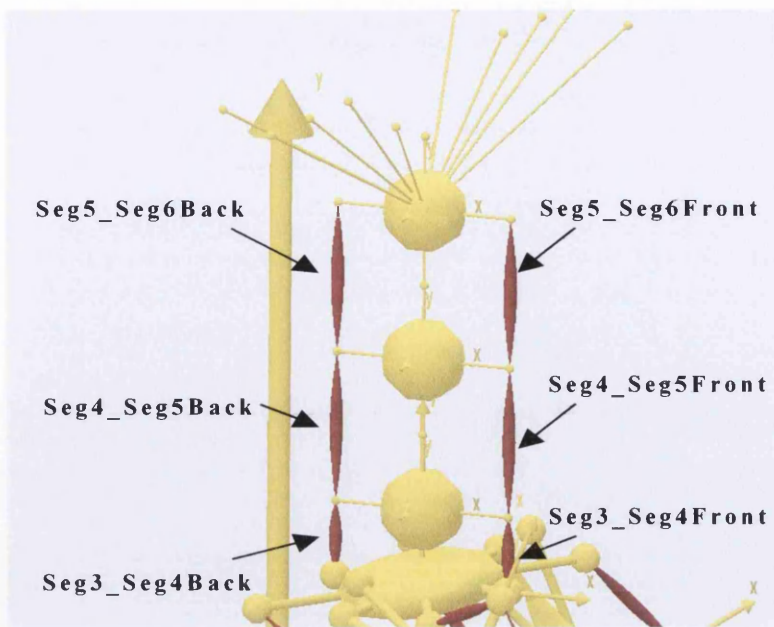


Figure 7.20 – Muscles of the Trunk

Muscles.any File

```

//-----Seg3-Seg4 BACK-----
AnyViaPointMuscle Seg3_Seg4Back = {
  AnyMuscleModel &MusMdl1 = .MusMdl1;
  AnyRefNode &NodeS3_7 = ...Segs.LowerExtremities.Seg3.NodeSeg3_7;
  AnyRefNode &NodeS4_2 = ...Segs.Trunk.Seg4.NodeSeg4_2;
  AnyDrawMuscle DrwMus = {};
}; //Seg3/Seg4_Back

//-----Seg3-Seg4 FRONT-----
AnyViaPointMuscle Seg3_Seg4Front = {
  AnyMuscleModel &MusMdl1 = .MusMdl1;
  AnyRefNode &NodeS3_8 = ...Segs.LowerExtremities.Seg3.NodeSeg3_8;
  AnyRefNode &NodeS4_3 = ...Segs.Trunk.Seg4.NodeSeg4_3;
  AnyDrawMuscle DrwMus = {};
}; //Seg3/Seg4_Front

//-----Seg4-Seg5 BACK-----
AnyViaPointMuscle Seg4_Seg5Back = {
  AnyMuscleModel &MusMdl1 = .MusMdl1;
  AnyRefNode &NodeS4_2 = ...Segs.Trunk.Seg4.NodeSeg4_2;
  AnyRefNode &NodeS5_2 = ...Segs.Trunk.Seg5.NodeSeg5_2;
  AnyDrawMuscle DrwMus = {};
}; //Seg4/Seg5_Back

//-----Seg4-Seg5 FRONT-----
AnyViaPointMuscle Seg4_Seg5Front = {
  AnyMuscleModel &MusMdl1 = .MusMdl1;
  AnyRefNode &NodeS4_3 = ...Segs.Trunk.Seg4.NodeSeg4_3;
  AnyRefNode &NodeS5_3 = ...Segs.Trunk.Seg5.NodeSeg5_3;
  AnyDrawMuscle DrwMus = {};
}; //Seg4/Seg5_Front

//-----Seg5-Seg6 BACK-----
AnyViaPointMuscle Seg5_Seg6Back = {
  AnyMuscleModel &MusMdl1 = .MusMdl1;
  AnyRefNode &NodeS5_2 = ...Segs.Trunk.Seg5.NodeSeg5_2;
  AnyRefNode &NodeS6_2 = ...Segs.Trunk.Seg6.NodeSeg6_2;
  AnyDrawMuscle DrwMus = {};
}; //Seg5/Seg6_Back

//-----Seg5-Seg6 FRONT-----
AnyViaPointMuscle Seg5_Seg6Front = {
  AnyMuscleModel &MusMdl1 = .MusMdl1;
  AnyRefNode &NodeS5_3 = ...Segs.Trunk.Seg5.NodeSeg5_3;
  AnyRefNode &NodeS6_3 = ...Segs.Trunk.Seg6.NodeSeg6_3;
  AnyDrawMuscle DrwMus = {};
}; //Seg5/Seg6_Front

```

Figure 7.21 – Muscles.any File of the Trunk

7.4.4 Kinematic Measures & Drivers

Because of the motion of the bones of the real human trunk is relative to each other, a new facility, that constrain the moments, with name *AnyKinMeasureLinComb* will be used. This is a composer of new kinematic measures as a linear combination of existing ones. To some extent, it is similar to *AnyKinMeasureOrg*, but the way it composes the new measure is different.

The linear combination is defined by a coefficient matrix, *Coef*, and constant terms in the vector *Const*. Each row in *Coef* and *Const* defines one component of the new kinematic measure. Each column of *Coef* contains the coefficients associated with one component of the existing measures that have been entered into this new measure. For example, in this particular case, the angular position and the angular velocity of *Joint D*, could be a linear combination of *Joint E*, and *Joint E* a linear combination of *Joint F*.

Let's say, that the whole motion is dependent on the motion characteristics of *Joint F*. Then the following mathematical expressions could relate the motion of each joint to each other:

$$\textbf{Joint F:} \quad \text{angular velocity } v_{\text{Joint F}} = v_{\text{Joint F}} \quad (7.1)$$

$$\text{angular position } \theta_{\text{Joint F}} = \theta_{\text{Joint F}} \quad (7.2)$$

$$\textbf{Joint E:} \quad \text{angular velocity } v_{\text{Joint E}} = v_{\text{Joint F}} + 0.5 \cdot v_{\text{Joint F}} \quad (7.3)$$

$$\text{angular position } \theta_{\text{Joint E}} = \theta_{\text{Joint F}} + 0.5 \cdot \theta_{\text{Joint F}} \quad (7.4)$$

$$\textbf{Joint D:} \quad \text{angular velocity } v_{\text{Joint D}} = v_{\text{Joint F}} + 0.5 \cdot v_{\text{Joint E}} \quad (7.5)$$

$$\text{angular position } \theta_{\text{Joint D}} = \theta_{\text{Joint F}} + 0.5 \cdot \theta_{\text{Joint E}} \quad (7.6)$$

So, if the angular displacement ($\theta_{\text{Joint F}}$) of *Joint F* is -0.2618 rad (-15 degrees) and the angular velocity ($v_{\text{Joint F}}$) is 0.25 rad/sec , then the associated “kinematic measures” and drivers that implement this particular motion are shown in Table 7.8 while in Figure 7.22 some related time steps are presented.

KinematicMeasures.any File

```
//-----TRUNK ORGANIZER-----  
AnyKinMeasureLinComb TrunkMeasure = {  
  AnyJoint &u1 = Main.HumanModel.Joints.TrunkJoints.D;  
  AnyJoint &u2 = Main.HumanModel.Joints.TrunkJoints.E;  
  AnyJoint &u3 = Main.HumanModel.Joints.TrunkJoints.F;  
  Coef = {{1, -0.5, 0},  
          {0, 1, -0.5},  
          {0, 0, 1}};  
  Const = {0, 0, 0};  
};
```

Drivers.any File

```
//-----TRUNK DRIVER-----  
AnyKinEqSimpleDriver TrunkDriver = {  
  AnyKinMeasureLinComb &ref = ..KinematicMeasures.TrunkMeasure;  
  DriverPos = {-15*pi/180, -15*pi/180, -15*pi/180};  
  DriverVel = {0.25, 0.25, 0.25};  
  Reaction.Type = {0, 0, 0};  
};
```

Table 7.8 – Kinematic Measure/Driver of Trunk

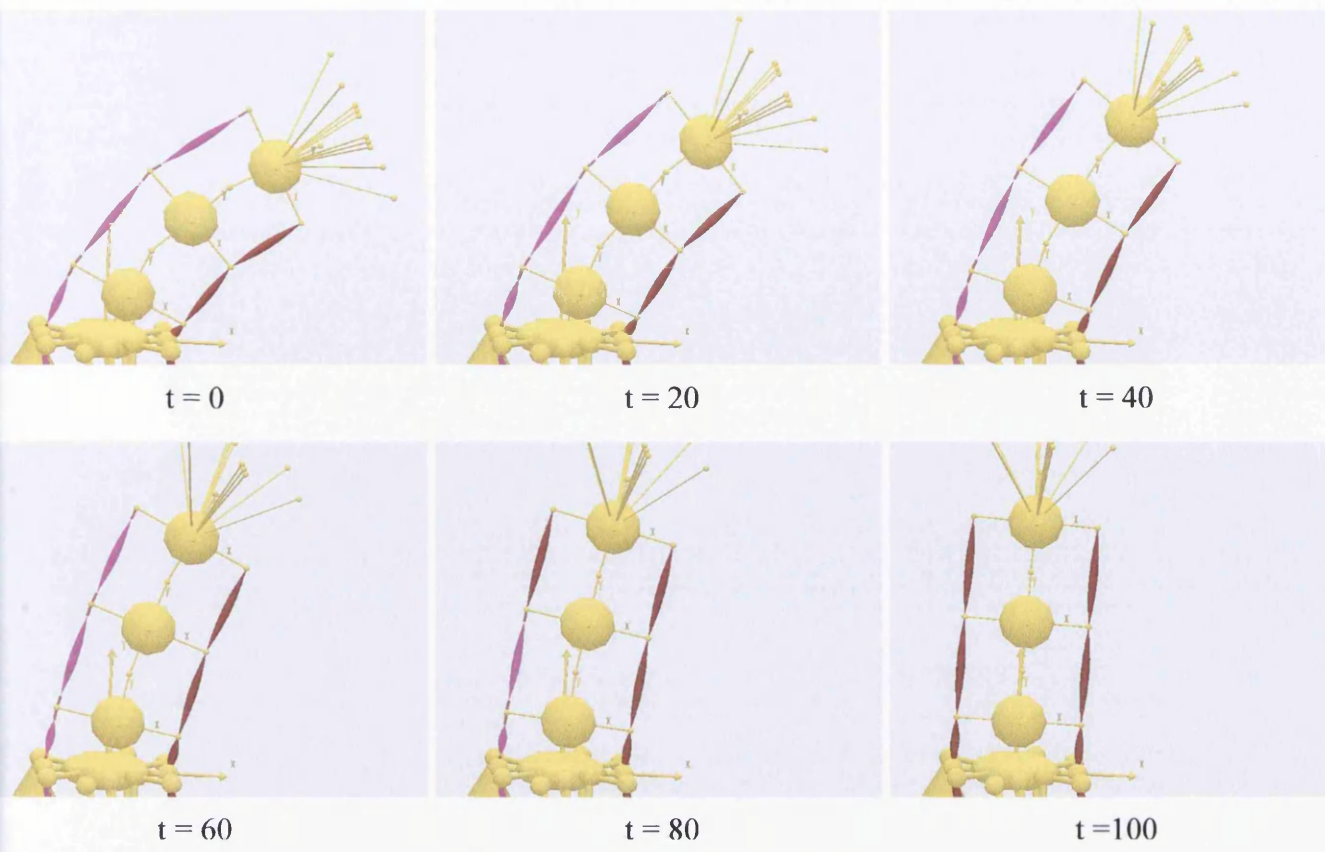


Figure 7.22 – Motion of Trunk

In the above Figure 7.22 the back trunk muscles which cause the motion are over-activated as the muscle force that has been chosen is $F_0=10N$. It has to be noted that γ_{m1} and γ_{m1L} in the InitialConditions.any file should be increased to 120 degrees because the addition of the trunk caused problem to the position of the lower extremities. For checking if the $Coef$ matrix and $Const$ vector are correct, the angular position and velocity of all the joints are shown in Figures 7.23 and 7.24 respectively.



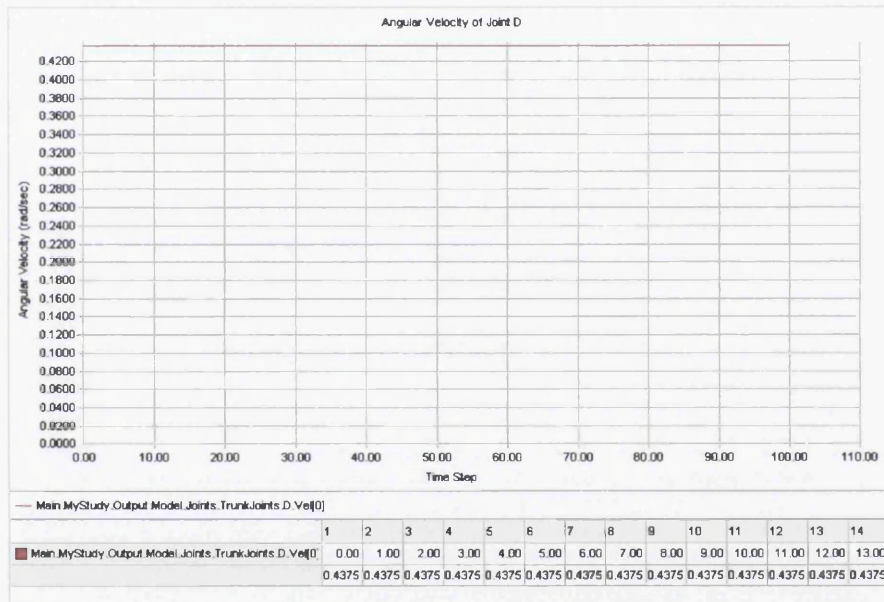
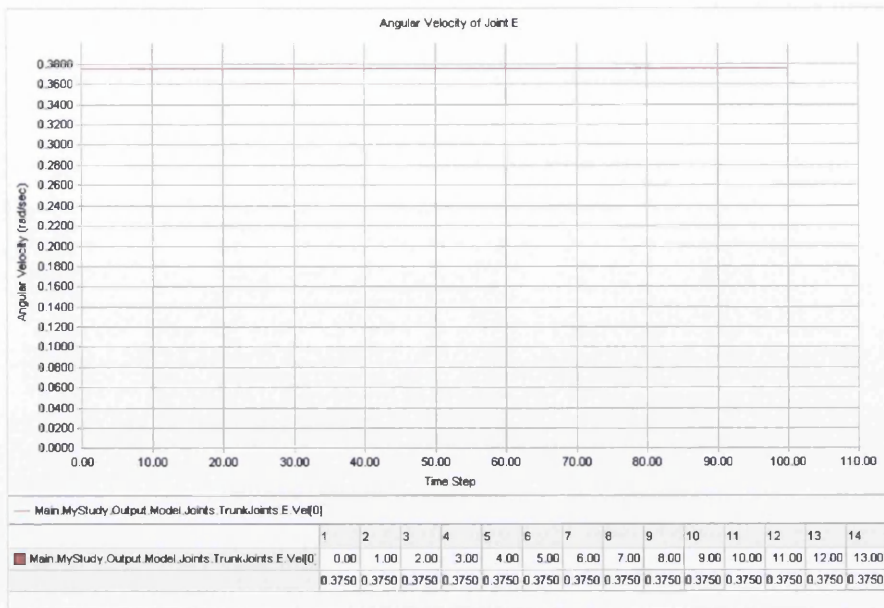


Figure 7.24 – Angular Velocity of Trunk Joints

Finally, the investigation of the trunk will end with the implementation of the motion of the whole assembly with the existence of the *CoM* driver as well. The first step to do this, is the elimination of the *CoM* driver first. The next step is to activate the *x*-translational component of the *Seg3* driver and add the new segments of the trunk at the *CoM* “kinematic measure” (Figure 7.25). It has to be noted that nothing has changed at the *Position* “anyfolder”. The next step is to run the model, and at the time “**study operation: Initial conditions are fully updated**” the position of *CoM* has to be recorded. Then, the new position of *CoM* has to be entered in the *Position* “anyfolder”, followed by the activation of the *CoM* driver and the elimination of the *x*-translational component of the *Seg3* driver. The new position of the

CoM driver is $0.197711m$. Figure 7.26 shows some time steps of the successful motion of the new assembly in which is reflected one more time the over-activation of the muscles that cause the motion.

```

KinematicMeasures.any File

//-----CENTRE OF MASS MEASURE-----

AnyKinCoM CoMeasure = {
  AnySeg &seg0 = ..Segs.LowerExtremities.Seg0;
  AnySeg &seg0L = ..Segs.LowerExtremities.Seg0L;
  AnySeg &seg1 = ..Segs.LowerExtremities.Seg1;
  AnySeg &seg1L = ..Segs.LowerExtremities.Seg1L;
  AnySeg &seg2 = ..Segs.LowerExtremities.Seg2;
  AnySeg &seg2L = ..Segs.LowerExtremities.Seg2L;
  AnySeg &seg3 = ..Segs.LowerExtremities.Seg3;
  AnySeg &seg4 = ..Segs.Trunk.Seg4;
  AnySeg &seg5 = ..Segs.Trunk.Seg5;
  AnySeg &seg6 = ..Segs.Trunk.Seg6;
}; //CentreOfMassMeasure

```

Figure 7.25 – Kinematic Measure of Center of Mass with the Trunk Segments

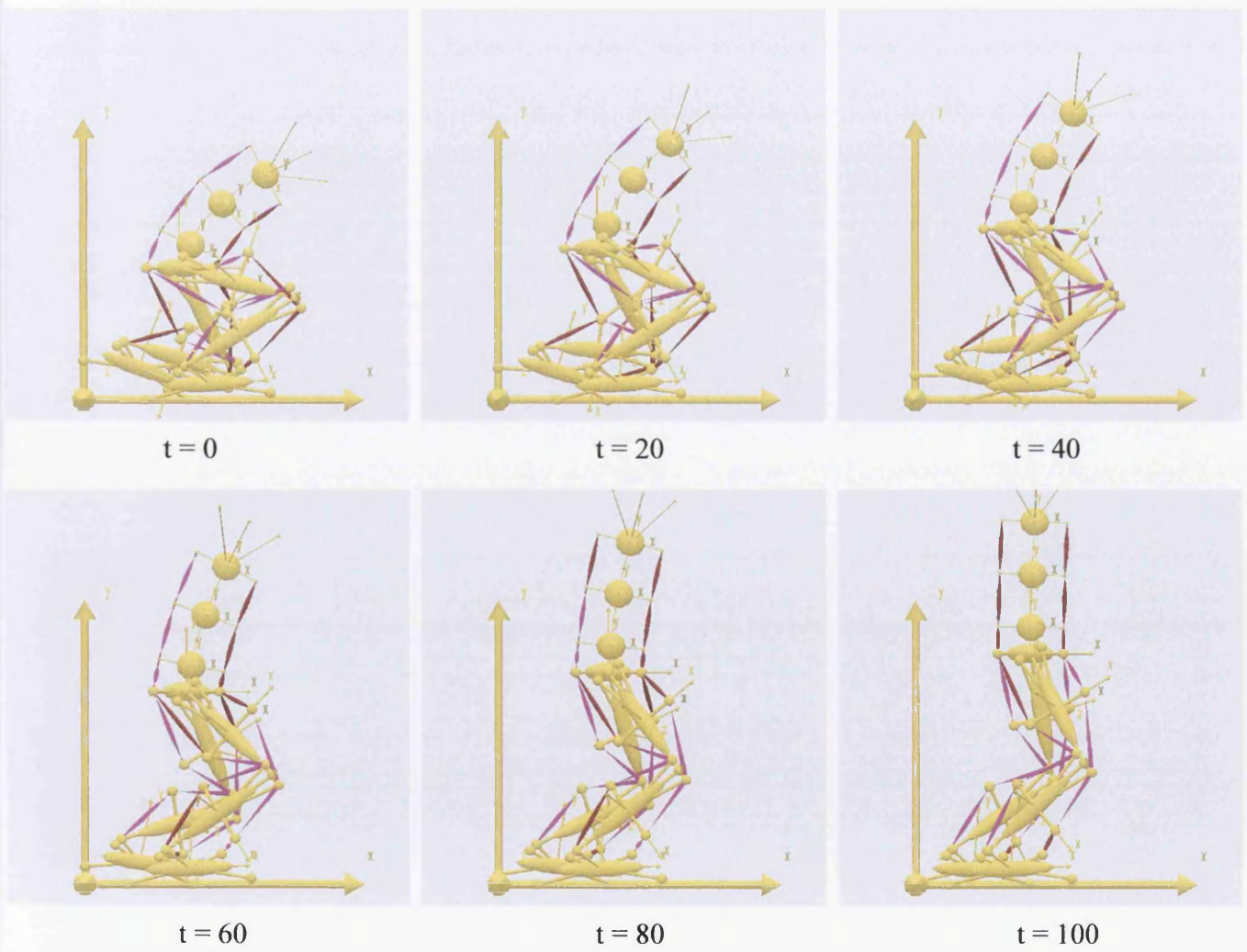


Figure 7.26 – Motion of Lower Extremity and Trunk with CoM Driver

7.5 Upper Extremity & Box

7.5.1 Segments

The segments that have been used are all cylinders with the same geometrical properties with the cylinders of the triple inverted pendulum (Figure 7.27). The only segment that is different, is the box segment which is a rectangular prism. The geometrical properties of the box are shown in Figure 7.28.

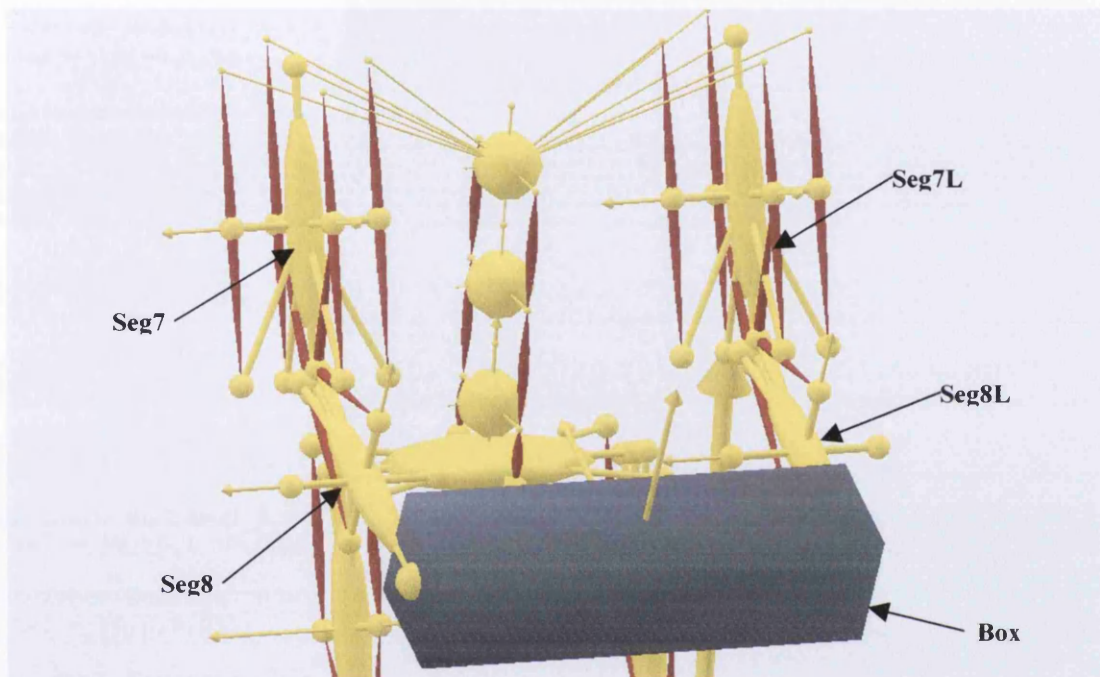
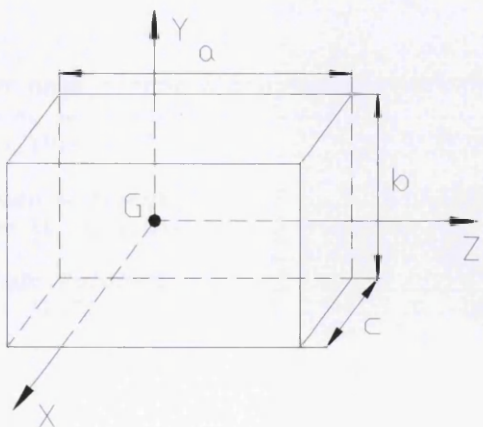


Figure 7.27 – Segments of UpperExtremity & Box

Rectangular prism



Centre of mass (G): $G = \frac{b}{2}$

Mass: $m = 3kg$

Moment of Inertia (G)

$a = 0.3m, b = 0.08m, c = 0.08m$

$$I_{x_G} = \frac{1}{12} \cdot m \cdot (a^2 + b^2) = 0.0241kgm^2$$

$$I_{y_G} = \frac{1}{12} \cdot m \cdot (a^2 + c^2) = 0.0241kgm^2$$

$$I_{z_G} = \frac{1}{12} \cdot m \cdot (b^2 + c^2) = 0.0032kgm^2$$

Figure 7.28 – Geometrical Properties of Box

Table 7.9 below presents all the “anyfiles” of the segments that constitute the upper extremity.

Seg7.any File

```
//-----SEGMENT 7-----
AnySeg Seg7 = {
  //r= {0.1,0.67,0.15};
  Axes = ...InitialConditions.R7;
  Mass = 2;
  Jii = {0.0069,0.0004,0.0069};
  AnyRefNode NodeSeg7_1 = {
    sRel = {0,0.1,0};
  };
  AnyRefNode NodeSeg7_2 = {
    sRel = {0,-0.1,0};
  };
  AnyRefNode NodeSeg7_3 = {
    sRel = {0.05,-0.1,0};
  };
  AnyRefNode NodeSeg7_4 = {
    sRel = {-0.05,-0.1,0};
  };
  AnyRefNode NodeSeg7_5 = {
    sRel = {0.05,0,0};
  };
  AnyRefNode NodeSeg7_6 = {
    sRel = {-0.05,0,0};
  };
  AnyRefNode NodeSeg7_7 = {
    sRel = {0,-0.1,0.05};
  };
  AnyRefNode NodeSeg7_8 = {
    sRel = {0,-0.1,-0.05};
  };
  AnyRefNode NodeSeg7_9 = {
    sRel = {0,0,0.05};
  };
  AnyRefNode NodeSeg7_10 = {
    sRel = {0,0,-0.05};
  };
  AnyDrawSeg DrwSeg = { };
}; //Segment7
```

Seg7L.any File

```
//-----SEGMENT 7L-----
AnySeg Seg7L = {
  //r= {0.1,0.67,-0.15};
  Axes = ...InitialConditions.R7L;
  Mass = 2;
  Jii = {0.0069,0.0004,0.0069};
  AnyRefNode NodeSeg7L_1 = {
    sRel = {0,0.1,0};
  };
  AnyRefNode NodeSeg7L_2 = {
    sRel = {0,-0.1,0};
  };
  AnyRefNode NodeSeg7L_3 = {
    sRel = {0.05,-0.1,0};
  };
  AnyRefNode NodeSeg7L_4 = {
    sRel = {-0.05,-0.1,0};
  };
  AnyRefNode NodeSeg7L_5 = {
    sRel = {0.05,0,0};
  };
  AnyRefNode NodeSeg7L_6 = {
    sRel = {-0.05,0,0};
  };
  AnyRefNode NodeSeg7L_7 = {
    sRel = {0,-0.1,0.05};
  };
  AnyRefNode NodeSeg7L_8 = {
    sRel = {0,-0.1,-0.05};
  };
  AnyRefNode NodeSeg7L_9 = {
    sRel = {0,0,0.05};
  };
  AnyRefNode NodeSeg7L_10 = {
    sRel = {0,0,-0.05};
  };
  AnyDrawSeg DrwSeg = { };
}; //Segment7L
```

Seg8.any File

```
//-----SEGMENT 8-----
AnySeg Seg8 = {
  //r= {0.1,0.47,0.15};
  Axes = ...InitialConditions.R8;
  Mass = 2;
  Jii = {0.0069,0.0004,0.0069};
  AnyRefNode NodeSeg8_1 = {
    sRel = {0,0.1,0};
  };
  AnyRefNode NodeSeg8_2 = {
    sRel = {0,-0.1,0};
  };
  AnyRefNode NodeSeg8_3 = {
    sRel = {0.05,0,0};
  };
  AnyRefNode NodeSeg8_4 = {
    sRel = {-0.05,0,0};
  };
  AnyRefNode NodeSeg8_5 = {
    sRel = {0,0,0.05};
  };
};
```

Seg8L.any File

```
//-----SEGMENT 8L-----
AnySeg Seg8L = {
  //r= {0.1,0.47,-0.15};
  Axes = ...InitialConditions.R8L;
  Mass = 2;
  Jii = {0.0069,0.0004,0.0069};
  AnyRefNode NodeSeg8L_1 = {
    sRel = {0,0.1,0};
  };
  AnyRefNode NodeSeg8L_2 = {
    sRel = {0,-0.1,0};
  };
  AnyRefNode NodeSeg8L_3 = {
    sRel = {0.05,0,0};
  };
  AnyRefNode NodeSeg8L_4 = {
    sRel = {-0.05,0,0};
  };
  AnyRefNode NodeSeg8L_5 = {
    sRel = {0,0,0.05};
  };
};
```

```

AnyRefNode NodeSeg8_6 = {
    sRel = {0,0,-0.05};
};
AnyRefNode NodeSeg8_7 = {
    sRel = {0.01,0.1,0};
};
AnyRefNode NodeSeg8_8 = {
    sRel = {-0.01,0.1,0};
};
AnyRefNode NodeSeg8_9 = {
    sRel = {0,0.1,0.01};
};
AnyRefNode NodeSeg8_10 = {
    sRel = {0,0.1,-0.01};
};
AnyDrawSeg DrwSeg = {};
}; //Segment8

```

```

AnyRefNode NodeSeg8L_6 = {
    sRel = {0,0,-0.05};
};
AnyRefNode NodeSeg8L_7 = {
    sRel = {0.01,0.1,0};
};
AnyRefNode NodeSeg8L_8 = {
    sRel = {-0.01,0.1,0};
};
AnyRefNode NodeSeg8L_9 = {
    sRel = {0,0.1,0.01};
};
AnyRefNode Node8L_10 = {
    sRel = {0,0.1,-0.01};
};
AnyDrawSeg DrwSeg = {};
}; //Segment8L

```

Box.any File

```

AnySeg Box = {
    Mass = 3;
    Jii = {0.0241,0.0241,0.0032};
    AnyRefNode NodeBox_0 = {
        sRel = {0,0,0};
    };
    AnyRefNode NodeBox_1 = {
        sRel = {0,0,0.15};
    };
    AnyRefNode NodeBox_2 = {
        sRel = {0,0,-0.15};
    };
    /*
    AnyDrawSTL DrwSTL = {
        FileName = "..\Box.stl";
        ScaleXYZ = {0.001,0.001,0.001};
    };
    AnySurfSTL Cylinder = {
        FileName = "..\Box.stl";
    };
    */
    AnyDrawSurf DrwSurf = {
        FileName = "..\Box.anysurf";
        ScaleXYZ = {0.001,0.001,0.001};
        RGB = {82/256,85/256,111/256}; //Blue
    };
    AnyDrawSeg DrwSeg = {};
}; //Box

```

Table 7.9 – Anyfiles of Upper Extremity Segments & Box

7.5.2 Initial Conditions

The initial conditions have already been discussed in the previous chapter, so all the associated “anyfiles” are presented in the following table (Table 7.10).

InitialConditions.any	RotMatrices.any File
<pre>//-----ROTATION X-AXIS----- //-----Segment 7----- AnyVar alpha7 = 0; //-----Segment 7L----- AnyVar alpha7L = 0; //-----Segment 8----- AnyVar alpha8 = 0; //-----Segment 8L----- AnyVar alpha8L = 0; //-----ROTATION Y-AXIS----- //-----Segment 7----- AnyVar beta7 = 0; //-----Segment 7L----- AnyVar beta7L = 0; //-----Segment 8----- AnyVar beta8 = 0; //-----Segment 8L----- AnyVar beta8L = 0; //-----ROTATION Z-AXIS----- //-----Segment 7----- AnyVar gamma7 = 0.523599; //-----Segment 7L----- AnyVar gamma7L = 0.523599; //-----Segment 8----- AnyVar gamma8 = -0.872665; //-----Segment 8L----- AnyVar gamma8L = -0.872665;</pre>	<pre>//-----SEGMENT7 ROTATION VECTOR----- AnyMat33 R7 = { {cos(beta7) * cos(gamma7) , cos(gamma7) * sin(beta7) * sin(alpha7) + sin(gamma7) * cos(alpha7) , sin(gamma7) * sin(alpha7) - cos(gamma7) * sin(beta7) * cos(alpha7)} , {-sin(gamma7) * cos(beta7) , cos(alpha7) * cos(gamma7) - sin(alpha7) * sin(beta7) * sin(gamma7) , sin(gamma7) * sin(beta7) * cos(alpha7) + cos(gamma7) * sin(alpha7)} , { sin(beta7) , -cos(beta7) * sin(alpha7) , cos(alpha7) * cos(beta7)}}; //-----SEGMENT7L ROTATION VECTOR----- AnyMat33 R7L = { {cos(beta7L) * cos(gamma7L) , cos(gamma7L) * sin(beta7L) * sin(alpha7L) + sin(gamma7L) * cos(alpha7L) , sin(gamma7L) * sin(alpha7L) - cos(gamma7L) * sin(beta7L) * cos(alpha7L)} , {-sin(gamma7L) * cos(beta7L) , cos(alpha7L) * cos(gamma7L) - sin(alpha7L) * sin(beta7L) * sin(gamma7L) , sin(gamma7L) * sin(beta7L) * cos(alpha7L) + cos(gamma7L) * sin(alpha7L)} , { sin(beta7L) , -cos(beta7L) * sin(alpha7L) , cos(alpha7L) * cos(beta7L)}}; //-----SEGMENT8 ROTATION VECTOR----- AnyMat33 R8 = { {cos(beta8) * cos(gamma8) , cos(gamma8) * sin(beta8) * sin(alpha8) + sin(gamma8) * cos(alpha8) , sin(gamma8) * sin(alpha8) - cos(gamma8) * sin(beta8) * cos(alpha8)} , {-sin(gamma8) * cos(beta8) , cos(alpha8) * cos(gamma8) - sin(alpha8) * sin(beta8) * sin(gamma8) , sin(gamma8) * sin(beta8) * cos(alpha8) + cos(gamma8) * sin(alpha8)} , { sin(beta8) , -cos(beta8) * sin(alpha8) , cos(alpha8) * cos(beta8)}}; //-----SEGMENT8L ROTATION VECTOR----- AnyMat33 R8L = { {cos(beta8L) * cos(gamma8L) , cos(gamma8L) * sin(beta8L) * sin(alpha8L) + sin(gamma8L) * cos(alpha8L) , sin(gamma8L) * sin(alpha8L) - cos(gamma8L) * sin(beta8L) * cos(alpha8L)} , {-sin(gamma8L) * cos(beta8L) , cos(alpha8L) * cos(gamma8L) - sin(alpha8L) * sin(beta8L) * sin(gamma8L) , sin(gamma8L) * sin(beta8L) * cos(alpha8L) + cos(gamma8L) * sin(alpha8L)} , { sin(beta8L) , -cos(beta8L) * sin(alpha8L) , cos(alpha8L) * cos(beta8L)}};</pre>

Table 7.10 – Initial Conditions Anyfiles of Upper Extremities

7.5.3 Joints

The joints that will be used in the upper extremity are shown in Figure 7.29 below, while the corresponding “anyfile” is depicted in Table 7.11.

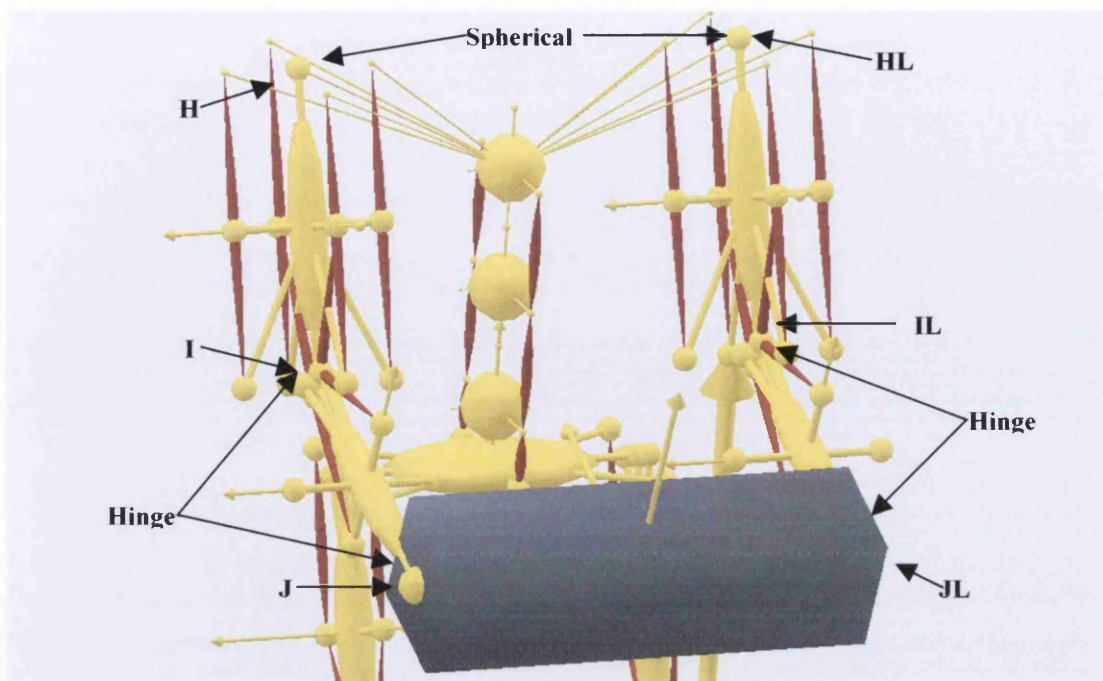


Figure 7.29 – Joints of Upper Extremity & Box

Joints.any File

```
AnySphericalJoint H = {  
  AnyRefNode &NodeS6_4 = ...Segs.Trunk.Seg6.NodeSeg6_4;  
  AnyRefNode &NodeS7_1 = ...Segs.UpperExtremities.Seg7.NodeSeg7_1;  
}; //JointH  
AnySphericalJoint HL = {  
  AnyRefNode &NodeS6_5 = ...Segs.Trunk.Seg6.NodeSeg6_5;  
  AnyRefNode &NodeS7L_1 = ...Segs.UpperExtremities.Seg7L.NodeSeg7L_1;  
}; //JointHL  
AnyRevoluteJoint I = {  
  Axis = z;  
  AnyRefNode &NodeS7_2 = ...Segs.UpperExtremities.Seg7.NodeSeg7_2;  
  AnyRefNode &NodeS8_1 = ...Segs.UpperExtremities.Seg8.NodeSeg8_1;  
}; //JointI  
AnyRevoluteJoint IL = {  
  Axis = z;  
  AnyRefNode &NodeS7L_2 = ...Segs.UpperExtremities.Seg7L.NodeSeg7L_2;  
  AnyRefNode &NodeS8L_1 = ...Segs.UpperExtremities.Seg8L.NodeSeg8L_1;  
}; //JointIL
```

Joints.any File

```
AnyRevoluteJoint J = {  
  Axis = y;  
  AnyRefNode &NodeS8_2 = ...Segs.UpperExtremities.Seg8.NodeSeg8_2 ;  
  AnyRefNode &NodeBox_1 = ...Segs.Box.Box.NodeBox_1;  
}; //JointJ  
AnyRevoluteJoint JL = {  
  Axis = y;  
  AnyRefNode &NodeS8L_2 = ...Segs.UpperExtremities.Seg8L.NodeSeg8L_2;  
  AnyRefNode &NodeBox_2 = ...Segs.Box.Box.NodeBox_2;  
}; //JointJL
```

Table 7.11 – Joints.any Files of the Upper Extremity & Box

7.5.4 Muscles

The muscles that have been used in the upper extremity are clearly displayed in Figure 7.30. One more time, the minimum number of muscles for the implementation of motion is used. The corresponding “anyfile” is shown below in Figure 7.31.

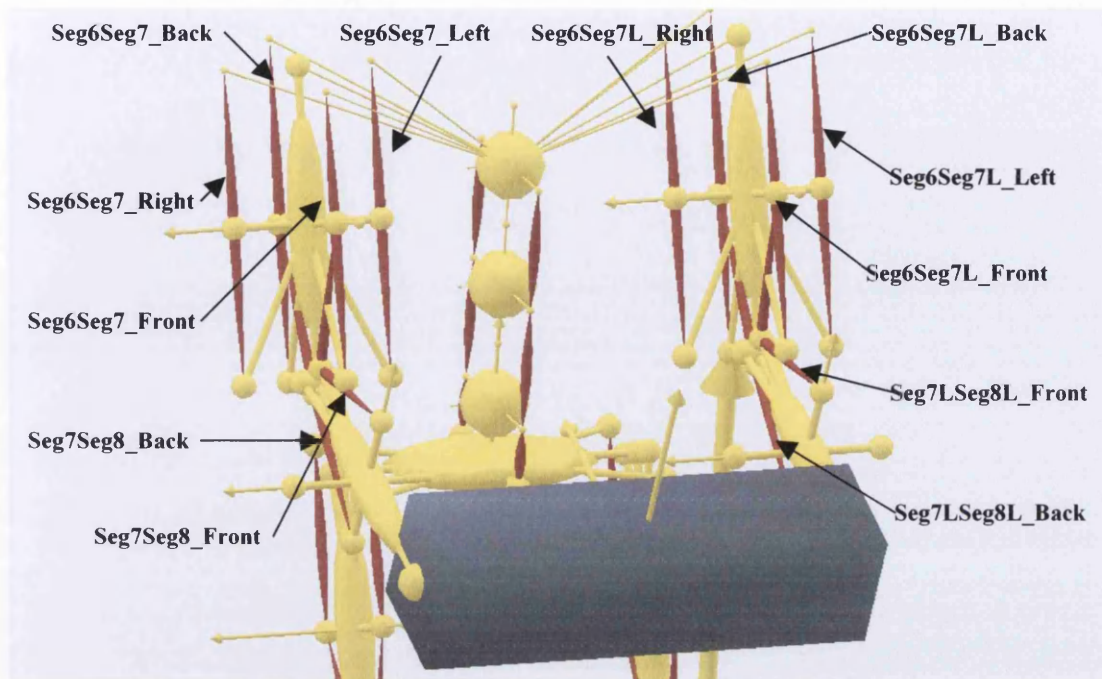


Figure 7.30 – Muscles of the Upper Extremity

Muscles.any File

```
//-----Simple muscle model with constant strength 300N-----  
AnyMuscleModel MusMd11 = {  
    FO = 10;  
}; //Muscle Strength  
  
//-----SEG6-SEG7 BACK-----  
AnyViaPointMuscle Seg6Seg7_Back = {  
    AnyMuscleModel &MusMd11 = .MusMd11;  
    AnyRefNode &NodeS6_7 = ...Segs.Trunk.Seg6.NodeSeg6_7;  
    AnyRefNode &NodeS7_4 = ...Segs.UpperExtremities.Seg7.NodeSeg7_4;  
    AnyDrawMuscle DrwMus = {};  
}; //Seg6/Seg7 Back  
  
//-----SEG6-SEG7 FRONT-----  
AnyViaPointMuscle Seg6Seg7_Front = {  
    AnyMuscleModel &MusMd11 = .MusMd11;  
    AnyRefNode &NodeS6_6 = ...Segs.Trunk.Seg6.NodeSeg6_6;  
    AnyRefNode &NodeS7_3 = ...Segs.UpperExtremities.Seg7.NodeSeg7_3;  
    AnyDrawMuscle DrwMus = {};  
}; //Seg6/Seg7 Front  
  
//-----SEG6-SEG7L BACK-----  
AnyViaPointMuscle Seg6Seg7L_Back = {  
    AnyMuscleModel &MusMd11 = .MusMd11;  
    AnyRefNode &NodeS6_9 = ...Segs.Trunk.Seg6.NodeSeg6_9;  
    AnyRefNode &NodeS7L_4 = ...Segs.UpperExtremities.Seg7L.NodeSeg7L_4;  
    AnyDrawMuscle DrwMus = {};  
}; //Seg6/Seg7L Back
```



```

-----SEG6-SEG7L FRONT-----
AnyViaPointMuscle Seg6Seg7L_Front = {
  AnyMuscleModel &MusMdl1 = .MusMdl1;
  AnyRefNode &NodeS6_8 = ...Segs.Trunk.Seg6.NodeSeg6_8;
  AnyRefNode &NodeS7L_3 = ...Segs.UpperExtremities.Seg7L.NodeSeg7L_3;
  AnyDrawMuscle DrwMus = {};
}; //Seg6/Seg7L Front
//-----
-----SEG6-SEG7 RIGHT-----
AnyViaPointMuscle Seg6Seg7_Right = {
  AnyMuscleModel &MusMdl1 = .MusMdl1;
  AnyRefNode &NodeS6_10 = ...Segs.Trunk.Seg6.NodeSeg6_10;
  AnyRefNode &NodeS7_7 = ...Segs.UpperExtremities.Seg7.NodeSeg7_7;
  AnyDrawMuscle DrwMus = {};
}; //Seg6/Seg7 Right
//-----
-----SEG6-SEG7 LEFT-----
AnyViaPointMuscle Seg6Seg7_Left = {
  AnyMuscleModel &MusMdl1 = .MusMdl1;
  AnyRefNode &NodeS6_11 = ...Segs.Trunk.Seg6.NodeSeg6_11;
  AnyRefNode &NodeS7_8 = ...Segs.UpperExtremities.Seg7.NodeSeg7_8;
  AnyDrawMuscle DrwMus = {};
}; //Seg6/Seg7 Left
//-----
-----SEG6-SEG7L LEFT-----
AnyViaPointMuscle Seg6Seg7L_Left = {
  AnyMuscleModel &MusMdl1 = .MusMdl1;
  AnyRefNode &NodeS6_12 = ...Segs.Trunk.Seg6.NodeSeg6_12;
  AnyRefNode &NodeS7L_8 = ...Segs.UpperExtremities.Seg7L.NodeSeg7L_8;
  AnyDrawMuscle DrwMus = {};
}; //Seg6/Seg7L Left
//-----
-----SEG6-SEG7L RIGHT-----
AnyViaPointMuscle Seg6Seg7L_Right = {
  AnyMuscleModel &MusMdl1 = .MusMdl1;
  AnyRefNode &NodeS6_13 = ...Segs.Trunk.Seg6.NodeSeg6_13;
  AnyRefNode &NodeS7L_7 = ...Segs.UpperExtremities.Seg7L.NodeSeg7L_7;
  AnyDrawMuscle DrwMus = {};
}; //Seg6/Seg7L Right
//-----
-----SEG7-SEG8 BACK -----
AnyViaPointMuscle Seg7Seg8_Back = {
  AnyMuscleModel &MusMdl1 = .MusMdl1;
  AnyRefNode &NodeS7_6 = ...Segs.UpperExtremities.Seg7.NodeSeg7_6;
  AnyRefNode &NodeS8_8 = ...Segs.UpperExtremities.Seg8.NodeSeg8_8;
  AnyRefNode &NodeS8_4 = ...Segs.UpperExtremities.Seg8.NodeSeg8_4;
  AnyDrawMuscle DrwMus = {};
}; //Seg7-Seg8 Back
//-----
-----SEG7L-SEG8L BACK -----
AnyViaPointMuscle Seg7Lseg8L_Back = {
  AnyMuscleModel &MusMdl1 = .MusMdl1;
  AnyRefNode &NodeS7L_6 = ...Segs.UpperExtremities.Seg7L.NodeSeg7L_6;
  AnyRefNode &NodeS8L_8 = ...Segs.UpperExtremities.Seg8L.NodeSeg8L_8;
  AnyRefNode &NodeS8L_4 = ...Segs.UpperExtremities.Seg8L.NodeSeg8L_4;
  AnyDrawMuscle DrwMus = {};
}; //Seg7Left-Seg8Left Front
//-----
-----SEG7-SEG8 FRONT -----
AnyViaPointMuscle Seg7Seg8_Front = {
  AnyMuscleModel &MusMdl1 = .MusMdl1;
  AnyRefNode &NodeS7_5 = ...Segs.UpperExtremities.Seg7.NodeSeg7_5;
  AnyRefNode &NodeS8_7 = ...Segs.UpperExtremities.Seg8.NodeSeg8_7;
  AnyRefNode &NodeS8_3 = ...Segs.UpperExtremities.Seg8.NodeSeg8_3;
  AnyDrawMuscle DrwMus = {};
}; //Seg7-Seg8 Front
//-----
-----SEG7L-SEG8L FRONT -----
AnyViaPointMuscle Seg7Lseg8L_Front = {
  AnyMuscleModel &MusMdl1 = .MusMdl1;
  AnyRefNode &NodeS7L_5 = ...Segs.UpperExtremities.Seg7L.NodeSeg7L_5;
  AnyRefNode &NodeS8L_7 = ...Segs.UpperExtremities.Seg8L.NodeSeg8L_7;
  AnyRefNode &NodeS8L_3 = ...Segs.UpperExtremities.Seg8L.NodeSeg8L_3;
  AnyDrawMuscle DrwMus = {};
}; //Seg7-Seg8 Front

```

Figure 7.31 – Muscles.any File of the Upper Extremity

7.5.5 Kinematic Measures & Drivers

There are two ways to drive the whole model with the addition of the upper extremity and the box. The first way is to keep all the drivers of the previous investigations and add four additional drivers, one for each of the joints of the upper extremity. The second way is to keep again all the drivers of the previous investigations and use only one driver, a driver that will drive the box, and the upper extremities will follow its motion. In this particular case the second way will be followed. One reason of adapting this way is that it is more realistic because it mimics the human organism that follows the shortest path for the implementation of a motion for energetic purpose. Another reason for example, is the difficulty to define the exact kinematic data of the driver of every joint of the upper extremity, when someone wants to turn a steering wheel (closed chain). In this case only one driver could be used which could move the wheel and make the upper extremity follow its motion. One more reason is that if accurate kinematic data is not necessary then it would be easier to use only one driver to implement the motion instead of using more.

The first step is to decide the exact position of the upper extremities and the box. In order to do this the “kinematic measures” which will measure *Cartesian* angles (*AnyKinRotational/RotVector*) and *Cardan* angles (*AnyKinRotational/RotAxesAngles*) will be defined (Figure 7.32).

KinematicMeasures.any File

```
//-----CARTESIAN ROTATION VECTORS-----  
//-----SEG7 KINEMATIC ROTATIONAL-----  
AnyKinRotational Seg7Rot1 = {  
  AnyFixedRefFrame &Ground = ..GlobalRef;  
  AnyRefFrame &Seg7 = ..Segs.UpperExtremities.Seg7;  
  Type = RotVector;  
};  
//-----SEG7L KINEMATIC ROTATIONAL-----  
AnyKinRotational Seg7LeftRot1 = {  
  AnyFixedRefFrame &Ground = ..GlobalRef;  
  AnyRefFrame &Seg7L = ..Segs.UpperExtremities.Seg7L;  
  Type = RotVector;  
};  
//-----SEG8 KINEMATIC ROTATIONAL-----  
AnyKinRotational Seg8Rot1 = {  
  AnyFixedRefFrame &Ground = ..GlobalRef;  
  AnyRefFrame &Seg8 = ..Segs.UpperExtremities.Seg8;  
  Type = RotVector;  
};  
//-----SEG8L KINEMATIC ROTATIONAL-----  
AnyKinRotational Seg8LeftRot1 = {  
  AnyFixedRefFrame &Ground = ..GlobalRef;  
  AnyRefFrame &Seg8L = ..Segs.UpperExtremities.Seg8L;  
  Type = RotVector;  
};  
//-----CARDAN ANGLES-----  
//-----SEG7 KINEMATIC ROTATIONAL-----  
AnyKinRotational Seg7Rot2 = {  
  AnyRefFrame &Seg7 = ..Segs.UpperExtremities.Seg7;  
  AnyFixedRefFrame &Ground = ..GlobalRef;  
  Axis1=x;  
  Axis2=y;  
  Axis3=z;  
  Type = RotAxesAngles;  
};
```

```

//-----SEG7L KINEMATIC ROTATIONAL-----
AnyKinRotational Seg7LeftRot2 = {
  AnyRefFrame &Seg7L = ..Segs.UpperExtremities.Seg7L;
  AnyFixedRefFrame &Ground = ..GlobalRef;
  Axis1=x;
  Axis2=y;
  Axis3=z;
  Type = RotAxesAngles;
};

//-----SEG8 KINEMATIC ROTATIONAL-----
AnyKinRotational Seg8Rot2 = {
  AnyRefFrame &Seg8 = ..Segs.UpperExtremities.Seg8;
  AnyFixedRefFrame &Ground = ..GlobalRef;
  Axis1=x;
  Axis2=y;
  Axis3=z;
  Type = RotAxesAngles;
};

//-----SEG8L KINEMATIC ROTATIONAL-----
AnyKinRotational Seg8LeftRot2 = {
  AnyRefFrame &Seg8L = ..Segs.UpperExtremities.Seg8L;
  AnyFixedRefFrame &Ground = ..GlobalRef;
  Axis1=x;
  Axis2=y;
  Axis3=z;
  Type = RotAxesAngles;
};

```

Figure 7.32 – Kinematic Measures of Upper Extremities

The next step is to create one more “anyfile”, the Drivers2.any file. In this file the heel, pelvis, trunk drivers are added, transferred from the Drivers1.any file. It has to be mentioned that the pelvis (*Seg3*) driver has to contain the x-translational coordinate. In the new driver “anyfile”, four more drivers are added which are responsible for the motion of the upper extremities. The rotation angles that have been chosen are -30 degrees for *Seg7* and *Seg7L*, and 50 degrees for *Seg8* and *Seg8L*. These drivers are related to the *Cartesian* “kinematic measures”. The Driver2.any file is shown below (Figure 7.33).

Drivers2.any File

```

//-----RIGHT HEEL DRIVER-----
AnyKinEqSimpleDriver RightHeelDriver = {
  AnyKinMeasureOrg &ref = ..KinematicMeasures.RightHeelPos;
  DriverPos = {..Position.RF_X,..Position.RF_Y,..Position.RF_Z,
    ..Position.RFrot_Y,..Position.RFrot_Z};
  DriverVel = {..Position.RF_XV,..Position.RF_YV,..Position.RF_ZV,
    ..Position.RFrot_YV,..Position.RFrot_ZV};
  Reaction.Type = {0,0,0,0,0};
};

//-----RIGHT HEEL LIN/ROT REACTION-----
AnyReacForce RightHeelToGroundReaction = {
  AnyKinLinear &ref1 = ..KinematicMeasures.RightHeelLin;
  AnyKinRotational &ref2 = ..KinematicMeasures.RightHeelRot;
};

```

```

//-----LEFT HEEL DRIVER-----
AnyKinEqSimpleDriver LeftHeelDriver = {
  AnyKinMeasureOrg &ref = ..KinematicMeasures.LeftHeelLPos;
  DriverPos = {..Position.LF_X,..Position.LF_Y,..Position.LF_Z,
    ..Position.LFrot_Y,..Position.LFrot_Z};
  DriverVel = {..Position.LF_XV,..Position.LF_YV,..Position.LF_ZV,
    ..Position.LFrot_YV,..Position.LFrot_ZV};
  Reaction.Type = {0,0,0,0,0};
};
//-----LEFT HEEL LIN/ROT REACTION-----
AnyReacForce LeftHeelToGroundReactionL = {
  AnyKinLinear &ref1 = ..KinematicMeasures.LeftHeelLLin;
  AnyKinRotational &ref2 = ..KinematicMeasures.LeftHeelLRot;
};
//-----SEG3 DRIVER-----
AnyKinEqSimpleDriver Seg3Motion= {
  AnyKinMeasureOrg &ref = ..KinematicMeasures.Seg3Pos;
  DriverPos = {..Position.P_X,..Position.P_Y,..Position.P_Z,
    ..Position.PR_X,..Position.PR_Y,..Position.PR_Z};
  DriverVel = {..Position.P_XV,..Position.P_YV,..Position.P_ZV,
    ..Position.PR_XV,..Position.PR_YV,..Position.PR_ZV};
  Reaction.Type = {0,0,0,0,0,0};
};
//-----TRUNK DRIVER-----
AnyKinEqSimpleDriver Trunk = {
  AnyKinMeasureLinComb &ref = ..KinematicMeasures.TrunkMeasure;
  DriverPos = {-15*pi/180, -15*pi/180, -15*pi/180};
  DriverVel = {0.25,0.25,0.25};
  Reaction.Type = {0,0,0};
};
//-----SEG7 DRIVER-----
AnyKinEqSimpleDriver Seg7Driver = {
  AnyKinRotational &Jnt = ..KinematicMeasures.Seg7Rot1;
  MeasureOrganizer = {0,1,2};
  DriverPos = {0,0,-30*pi/180};
  DriverVel = {0,0,0};
  DriverAcc = {0,0,0};
  Reaction.Type = {1,1,1};
};
//-----SEG7Left DRIVER-----
AnyKinEqSimpleDriver Seg7LeftDriver = {
  AnyKinRotational &Jnt = ..KinematicMeasures.Seg7LeftRot1;
  MeasureOrganizer = {0,1,2};
  DriverPos = {0,0,-30*pi/180};
  DriverVel = {0,0,0};
  DriverAcc = {0,0,0};
  Reaction.Type = {1,1,1};
};
//-----SEG8 DRIVER-----
AnyKinEqSimpleDriver Seg8Driver = {
  AnyKinRotational &Jnt = ..KinematicMeasures.Seg8Rot1;
  MeasureOrganizer = {2};
  DriverPos = {50*pi/180};
  DriverVel = {0};
  DriverAcc = {0};
  Reaction.Type = {1};
};
//-----SEG8Left DRIVER-----
AnyKinEqSimpleDriver Seg8LeftDriver = {
  AnyKinRotational &Jnt = ..KinematicMeasures.Seg8LeftRot1;
  MeasureOrganizer = {2};
  DriverPos = {50*pi/180};
  DriverVel = {0};
  DriverAcc = {0};
  Reaction.Type = {1};
};

```

Figure 7.33 – Drivers2.any File of the Simplified Human Model

Since the “kinematic measures” and drivers of the upper extremities are defined, it is time to create the “kinematic measures” and the corresponding driver that will drive the box, and to update the CoM “kinematic measure”, by adding the new segments (Table 7.12). The box driver is added to the Drivers1.any file.

KinematicMeasures.any File

```
//-----CENTRE OF MASS MEASURE-----
AnyKinCoM CoMeasure = {
  AnySeg &seg0 = ..Segs.LowerExtremities.Seg0;
  AnySeg &seg0L = ..Segs.LowerExtremities.Seg0L;
  AnySeg &seg1 = ..Segs.LowerExtremities.Seg1;
  AnySeg &seg1L = ..Segs.LowerExtremities.Seg1L;
  AnySeg &seg2 = ..Segs.LowerExtremities.Seg2;
  AnySeg &seg2L = ..Segs.LowerExtremities.Seg2L;
  AnySeg &seg3 = ..Segs.LowerExtremities.Seg3;
  AnySeg &seg4 = ..Segs.Trunk.Seg4;
  AnySeg &seg5 = ..Segs.Trunk.Seg5;
  AnySeg &seg6 = ..Segs.Trunk.Seg6;
  AnySeg &seg7 = ..Segs.UpperExtremities.Seg7;
  AnySeg &seg7L = ..Segs.UpperExtremities.Seg7L;
  AnySeg &seg8 = ..Segs.UpperExtremities.Seg8;
  AnySeg &seg8L = ..Segs.UpperExtremities.Seg8L;
  AnySeg &Box = ..Segs.Box.Box;
}; //CentreOfMassMeasure

//-----BOX KINEMATIC LINEAR-----
AnyKinLinear BoxLin = {
  AnyFixedRefFrame &Ground = ..GlobalRef;
  AnyRefNode &Nodes = ..Segs.Box.Box.NodeBox_0;
};

//-----BOX KINEMATIC ROTATIONAL-----
AnyKinRotational BoxRot = {
  AnyFixedRefFrame &Ground = ..GlobalRef;
  AnyRefNode &Nodes = ..Segs.Box.Box.NodeBox_0;
  Type = RotVector;
};

//-----BOX ORGANIZER-----
AnyKinMeasureOrg BoxPos = {
  AnyKinLinear &ref1 = .BoxLin;
  AnyKinRotational &ref2 = .BoxRot;
  MeasureOrganizer = {0,1,2,3};
};
```

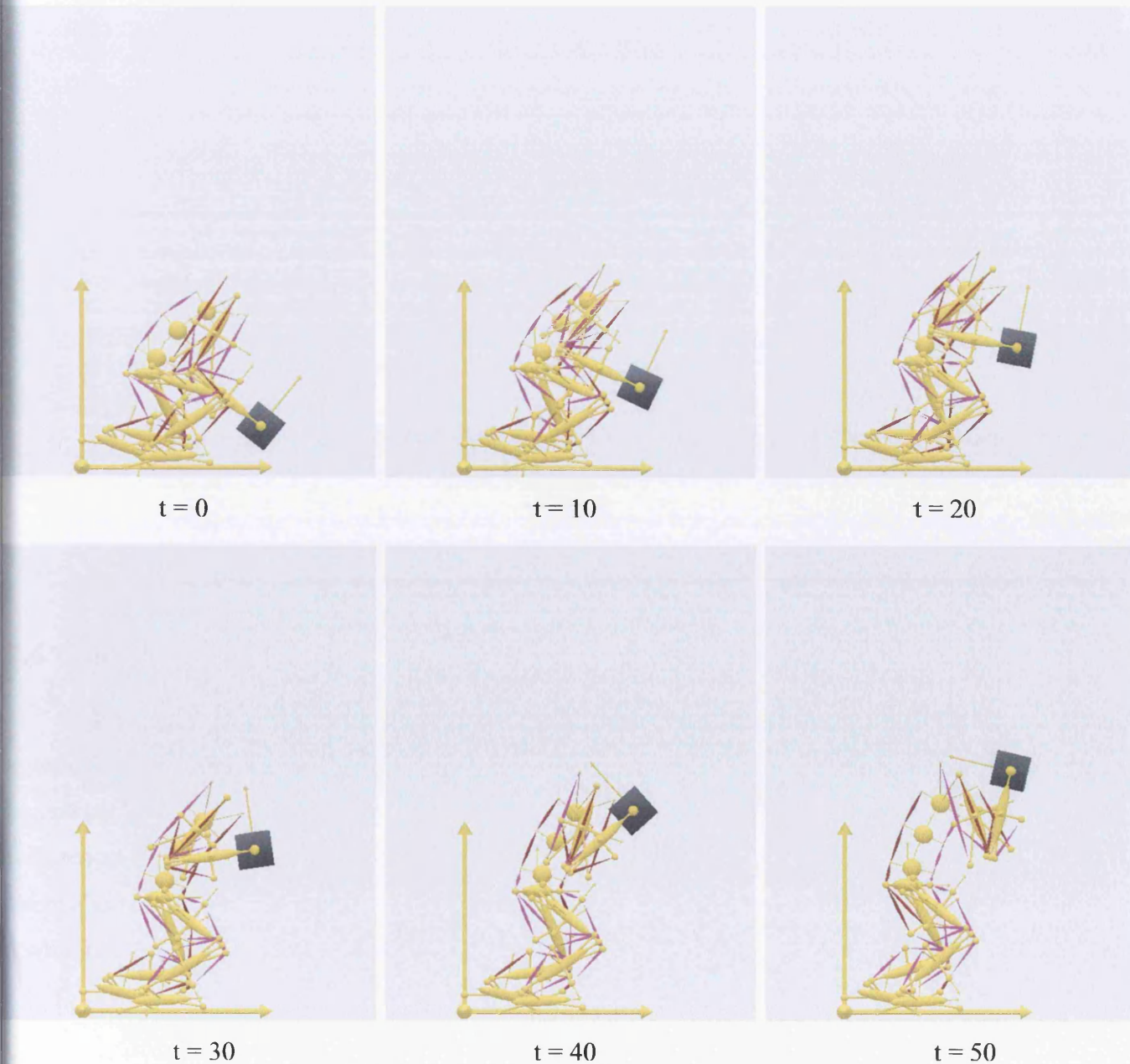
Drivers1.any File

```
//-----BOX DRIVER-----
AnyKinEqSimpleDriver BoxMotion= {
  AnyKinMeasureOrg &ref = ..KinematicMeasures.BoxPos;
  DriverPos = {0.403673, 0.084523, 0.150000,0};
  DriverVel = {-0.05,0.9,0,0};
  Reaction.Type = {1,1,1,1};
};
```

Table 7.12 – CoM/Box Kinematic Measures & Box Driver

The next step, is to deactivate the Drivers1.any file and to activate the Drivers2.any file. The model is reloaded and at the time “**study operation: Initial conditions are fully updated**” the position of *CoM*, the position *BoxPos* of the box driver and the *Cardan* angles of the upper extremities have to be recorded. But the addition of the new segments caused reverse rotation of the right lower extremity and the only way to make it again work, was to increase the y-coordinate of the pelvis (*Seg3*) from $0.2m$ to $0.21004m$. After this change, the model works again and the above recordings are being taken. The position of *CoM* is $0.244655m$, the position of the box is $\{0.403673, 0.084523, 0.150000, 0.000000\}$ with units m, m, m, rad respectively and the *Cardan* angles *gamma7* and *gamma7L* are 0.523599 rad, while *gamma8* and *gamma8L* are -0.872665 rad. Then, the associated changes have to be implemented.

Finally, it is time to activate Driver1.any again and to run the model with the velocities that have been chosen in the box driver as shown in Table 7.12. The resultant motion is represented in Figure 7.34 below, which is successful and reflects one more time the over-activation of the muscles because of the low value of the muscle force $F_0=10N$.



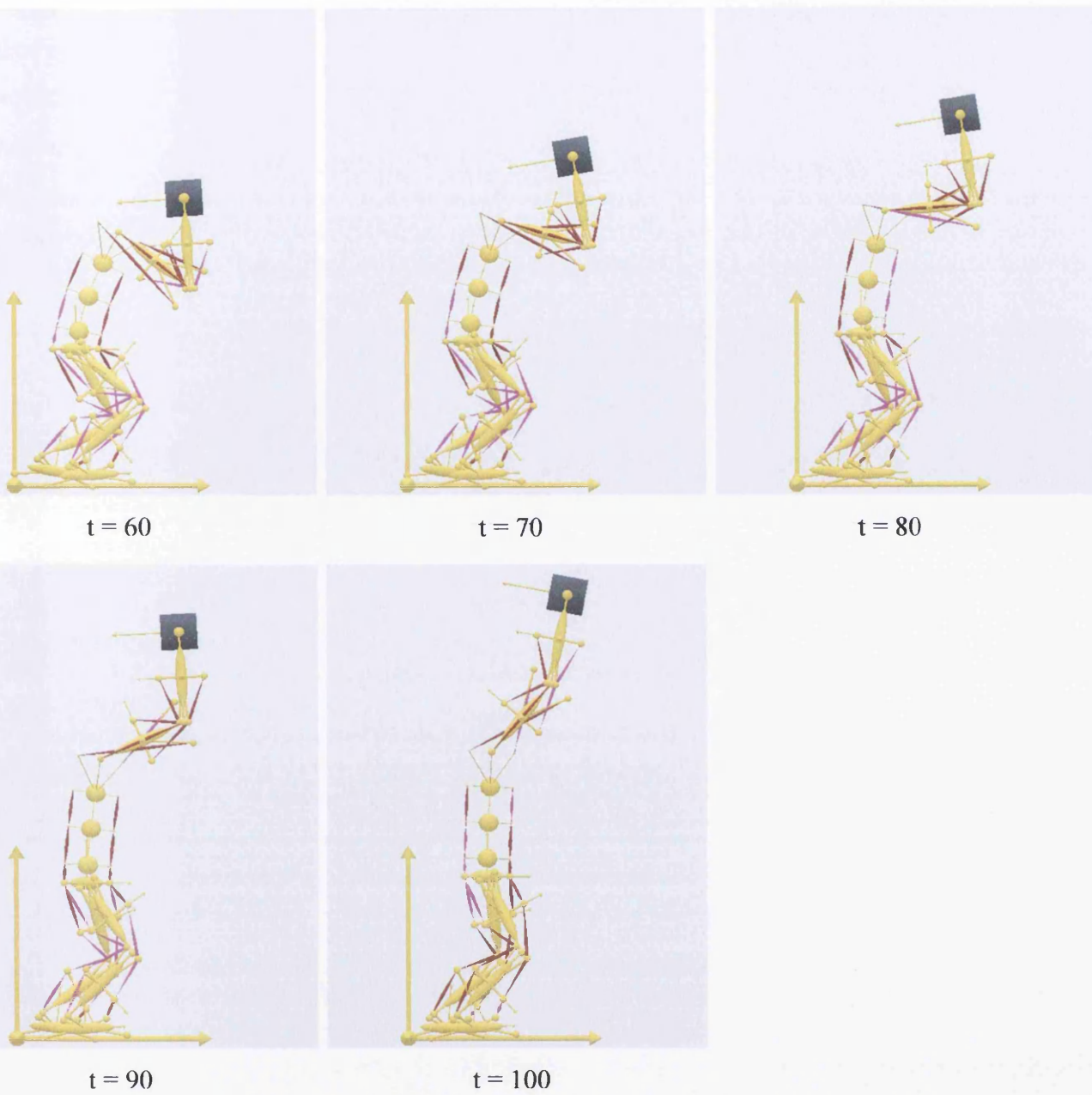


Figure 7.34 – Box Driven Motion of the Simplified Human Model

7.6 Conclusions

The present chapter demonstrated the way of implementing a simple model, which can be an approach to a real human model. The remarkable aspect of this human like model is that it is developed by just using the knowledge gained in the previous chapter adding just a few more features. These features are the spine and the center of mass drivers, and the reaction forces at the feet. From a validation point of view, the human like model proved to be very reliable in case of implementing a wide range of postures, as it was successful in a representative extreme one. A very important modelling

feature that enabled this implementation, is that only one driver is needed in order to position the pelvis instead of assigning one driver in each joint of the lower extremity. And that is because a driver in each joint would cause limitations in choosing a specific range of postures as it makes it impossible to resolve the positions by stepping from one joint to the next until the free ends of the mechanism are reached. Finally, the current study intends to be the basic “platform” for the more complex model that follows, the musculoskeletal model of the human body.

CHAPTER 8

DEVELOPMENT OF A PARAMETERISED HUMAN MODEL

8.1 Introduction

Following the work in previous chapters, here a fully parameterized model of the human body will be developed, dependent on the global parameters *height* and *mass*. The reason for this is that, both these parameters can represent a wide range of human bodies and they are very important factors that affect the analysis of the human movement as well. This chapter brings together the literature review and what was learned from the modelling as shown in the previous simple models. The parameterized musculoskeletal model is based on the model of the *Anybody Repository2* regarding the nodes in the segments, the joints, the muscles, the drivers of the shoulder complex and the bone geometries of scapula, thorax, radius and clavicle. Everything else is based on the research described throughout this project.

The parameterized model is using the same “design philosophy” with the simplified model of Chapter 7 regarding the organization, the “kinematic solutions” and the implementation of motion. The rest of the bone geometries are coming from the University of Brussels *Virtual Human* project, and they can be found in the *BEL* (European Biomechanics Laboratory) *Repository*⁵⁶. These geometries can be visualized and converted to *STL* format through the use of a freely downloadable software, the *DataManager*⁵⁷. The geometrical properties of the different bones of the body are derived from the research, presented in Chapter 3, of Drillis/Contini²⁴ (Figure 8.1), Dempster¹⁴ (Table 8.1) and Hanavan²⁵. The reason that Dempster’s data have been chosen, is because the segment endpoints are defined by their joint centers of rotation.

The final result is the development of three models of different height which are contained in the same windows folder (*Combined*) and share the same “CPU”/Main.any file. These models can be part of the same study, i.e. work all three (or two) together, or work each one individually. Because of the large amount of files needed in this investigation, without the need to be all included, the *Repository2* and the *Combined* study will be found on the supplementary *CD* of this thesis.

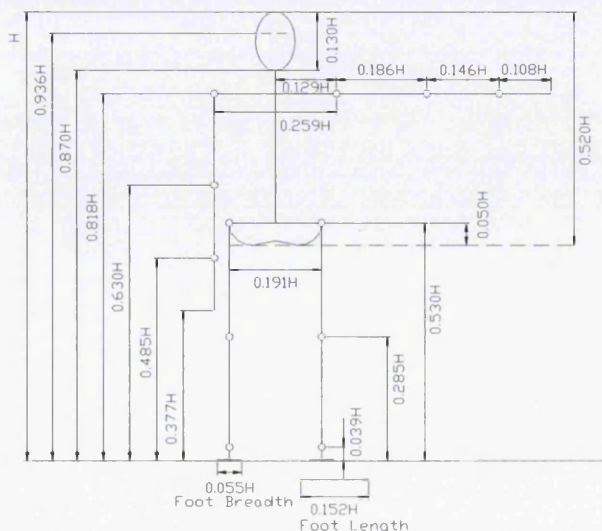


Figure 8.1 – Body Segment Lengths Expressed as a Fraction of Body Height *H* [18]

Segment	Endpoints (proximal to distal)	Segmental mass/total mass (P)	Center of mass/ Segment length		Radius of gyration/ Segment length			Density (gr/cm ³)
			(R _{proximal})	(R _{distal})	(K _{cg})	(K _{proximal})	(K _{distal})	
Hand	Wrist center to knuckle II of third finger	0.006	0.506	0.494	0.298	0.587	0.577	1.16
Forearm	Elbow to wrist center	0.016	0.430	0.570	0.303	0.526	0.647	1.13
Upper arm	Glenohumeral joint to elbow center	0.028	0.436	0.564	0.322	0.542	0.645	1.07
Forearm and hand	Elbow to wrist center	0.022	0.682	0.318	0.468	0.827	0.565	1.14
Upper extremity	Glenohumeral joint to wrist center	0.050	0.530	0.470	0.368	0.645	0.596	1.11
Foot	Ankle to ball of foot	0.0145	0.500	0.500	0.475	0.690	0.690	1.10
Leg	Knee to ankle center	0.0465	0.433	0.567	0.302	0.528	0.643	1.09
Thigh	Hip to knee center	0.100	0.433	0.567	0.323	0.540	0.653	1.05
Lower extremity	Hip to ankle center	0.161	0.447	0.553	0.326	0.560	0.650	1.06
Head	C7-T1 to ear canal	0.081	1.000	0.000	0.495	1.116	0.495	1.11
Shoulder	Sternoclavicular joint to glenohumeral joint center	0.0158	0.712	0.288	-	-	-	1.04
Thorax	C7-T1 to T12-L1	0.216	0.820	0.180	-	-	-	0.92
Abdomen	T12-L1 to L4-L5	0.139	0.440	0.560	-	-	-	-
Pelvis	L4-L5 to trochanter	0.142	0.105	0.895	-	-	-	-
Thorax and abdomen	C7-T1 to L4-L5	0.355	0.630	0.370	-	-	-	-
Abdomen and pelvis	T12-L1 to greater trochanter	0.281	0.270	0.730	-	-	-	1.01
Trunk	Greater trochanter to glenohumeral joint	0.497	0.495	0.505	0.406	0.640	0.648	1.03
Head, arms and trunk	Greater trochanter to glenohumeral joint	0.678	0.626	0.374	0.496	0.798	0.621	-
Head, arms and trunk	Greater trochanter to mid-rib	0.678	1.142	-0.142	0.903	1.456	0.914	-

Table 8.1 - Dempster's Body Segment Parameters [10]

8.2 Lower Extremity

8.2.1 Segments

The lower extremity is constituted by the feet, the upper and lower legs and the pelvis (Figure 8.2). The feet are considered as rectangular prisms, the legs as circular cylinders while the pelvis as an elliptical cylinder. The mass and the density of the various segments are defined by Dempster's data¹⁴.

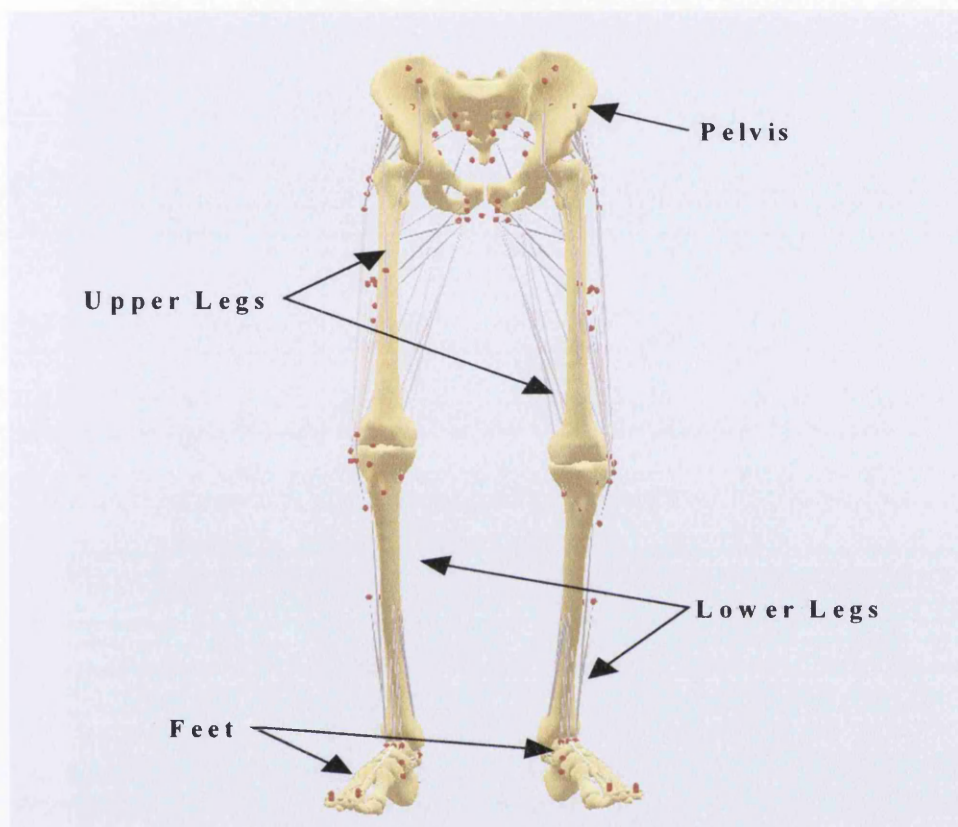
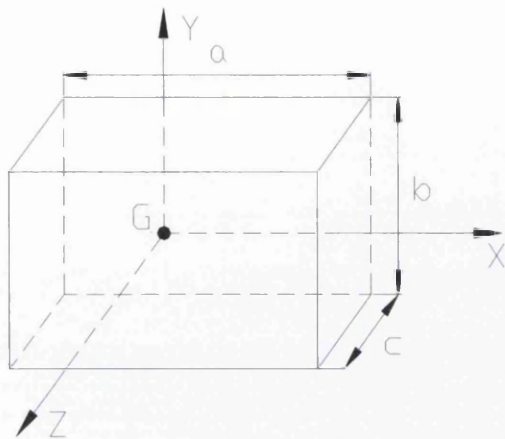


Figure 8.2 – Segments of the Lower Extremity

In the investigation that will follow, the way of parameterization of each of the segments will be presented. Due to the large number of nodes that each segment contains, only the part of the corresponding “anyfile”, that has to do with the geometrical properties of it, will be presented, and also just a few of the nodes as well, for illustrative purpose only. The full “anyfiles” can be found on the supplementary *CD*. A very important aspect that has to be mentioned, is that the bones are attached at the joints instead of at the center of mass of the segments. And that is because the bone geometries, are such complex, that it is difficult to place them in the right position during scaling if these are attached at the center of mass of the segments. It has to also be noted that M denotes the whole mass the of human body, while H is the overall height. Both of these parameters can be defined at the CPU/Main.any file of the whole model.

- **Feet**

Rectangular prism



Centre of mass (G): $G = \frac{b}{2}$

Mass: $m = 0.0145M(\text{kg})$

Density: $d = 1100\text{kg} / \text{m}^3$

Moment of Inertia (G)

$a = 0.152H(\text{m}), b = 0.039H(\text{m}), c = 0.055H(\text{m})$

$I_{x_G} = \frac{1}{12} \cdot m \cdot (b^2 + c^2) \text{kgm}^2$

$I_{y_G} = \frac{1}{12} \cdot m \cdot (a^2 + c^2) \text{kgm}^2$

$I_{z_G} = \frac{1}{12} \cdot m \cdot (a^2 + b^2) \text{kgm}^2$

Figure 8.3 – Geometrical Properties of Feet

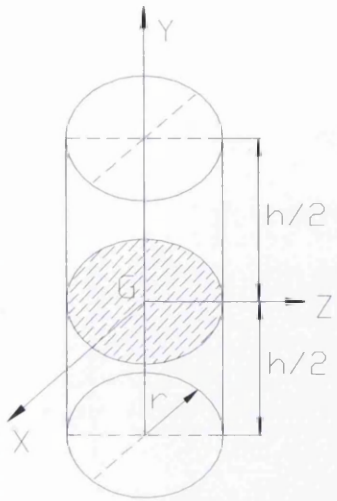
```
//-----RIGHT FOOT-----
AnySeg RightFoot = {
  #include "1_90/DrawSettings/Nodes.any"
  #include "1_90/DrawSettings/SegmentAxes.any"
  Mass = 0.0145*....BP.M;
  AnyVar d = 1100; // density
  AnyVar a = 0.152*....BP.H;
  AnyVar b = 0.039*....BP.H;
  AnyVar c = 0.055*....BP.H;
  AnyVar Ixx = (1/12)*Mass*(b^2+c^2);
  AnyVar Iyy = (1/12)*Mass*(a^2+c^2);
  AnyVar Izz = (1/12)*Mass*(a^2+b^2);
  Jii = {Ixx, Iyy, Izz};
  AnyVar Sign = 1;
  AnyRefNode AnkleJoint = {
    sRel = {-0.06, 0.06, .Sign * 0} * .a/0.24;
  }
  AnyDrawSurf DrwSurf = {
    FileName = "1_90/Bones/Foot.anysurf";
    ScaleXYZ = {0.001, 0.001, 0.001*..Sign} * ..a;
    RGB = Main.Model3.HumanModel.DrawSettings.Colors.Segments;
  };
};
AnyRefNode HeelJoint = {
  sRel = {-0.115, 0, 0 * .Sign} * .a/0.24;
};
```

Figure 8.4 –RightFoot.any File of the Parameterized Model

In the above figure *Sign* is positive for the right foot and leg, and negative for the left. The number *0.24* which divides the *a* parameter, is the length of the foot of the model in the *Repository2* of the Anybody technology. In other words the position of the various nodes is based on the Anybody human model contained in the *Repository2*.

• Lower Legs

Circular Cylinder



Centre of mass (G): $G = \frac{h}{2}$

Volume: $V = \pi \cdot r^2 \cdot h$

Mass: $m = 0.0465M$

Density: $d = \frac{m}{V} = 1090$

Moment of Inertia (G)

$$h = 0.285H - 0.039H, r = \left(\frac{m}{d \cdot \pi \cdot h} \right)^{\frac{1}{2}}$$

$$I_{x_G} = \frac{1}{12} \cdot m \cdot (3r^2 + h^2)$$

$$I_{y_G} = \frac{1}{2} \cdot m \cdot r^2$$

$$I_{z_G} = I_{x_G}$$

Figure 8.5 – Geometrical Properties of Lower Legs

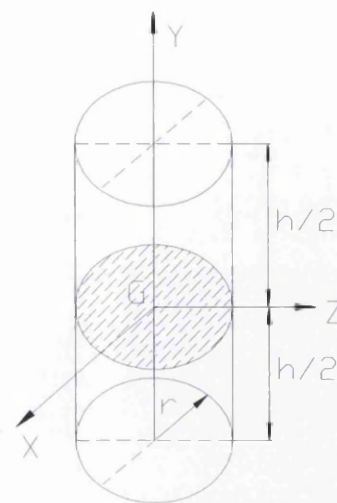
```
//-----RIGHT LOWER LEG-----
AnySeg RightLowerLeg = {
  #include "1_90/DrawSettings/Nodes.any"
  #include "1_90/DrawSettings/SegmentAxes.any"
  Axes = ....InitialConditions.LowerLeg;
  Mass = 0.0465 * ....BP.M;
  AnyVar d = 1090; // density
  AnyVar p = 3.1416; //pi
  AnyVar h = (0.285 * ....BP.H) - (0.039 * ....BP.H);
  AnyVar radius = (Mass/(d*p*h))^0.5; // radius
  AnyVar Ixx = (1/12)*Mass*((3*radius^2)+(h^2));
  AnyVar Iyy = (1/2)*Mass*radius^2;
  AnyVar Izz = Ixx;
  Jii = {Ixx, Iyy, Izz};
  AnyVar Sign = 1;
  AnyRefNode Node0 = {
    sRel = {0,0,0};
  };
  AnyRefNode AnkleJoint = {
    sRel = {0,-0.2608,0*.Sign} * .h/0.46;
  };
  AnyDrawSurf DrwSurf = {
    FileName = "1_90/Bones/Tibia.any surf";
    ScaleXYZ = {0.001, 0.001, 0.001*..Sign} * ..h;
    RGB = Main.Model3.HumanModel.DrawSettings.Colors.Segments;
  };
};
AnyRefNode KneeJoint = {
  sRel = {0, 0.1992,0*.Sign} * .h/0.46;
};
```

Figure 8.6 – RightLowerLeg.any File of the Parameterized Model

In the above figure the number 0.46 is the length of the lower leg in the *Repository2* model, while because of the absence of data about the radius of the cylinder the calculation is possible with the use of the volume and the mass.

• Upper Legs

Circular Cylinder



Centre of mass (G): $G = \frac{h}{2}$

Volume: $V = \pi \cdot r^2 \cdot h$

Mass: $m = 0.1M$

Density: $d = \frac{m}{V} = 1050$

Moment of Inertia (G)

$$h = 0.530H - 0.285H, r = \left(\frac{m}{d \cdot \pi \cdot h} \right)^{\frac{1}{2}}$$

$$I_{x_G} = \frac{1}{12} \cdot m \cdot (3r^2 + h^2)$$

$$I_{y_G} = \frac{1}{2} \cdot m \cdot r^2$$

$$I_{z_G} = I_{x_G}$$

Figure 8.7 – Geometrical Properties of Upper Legs

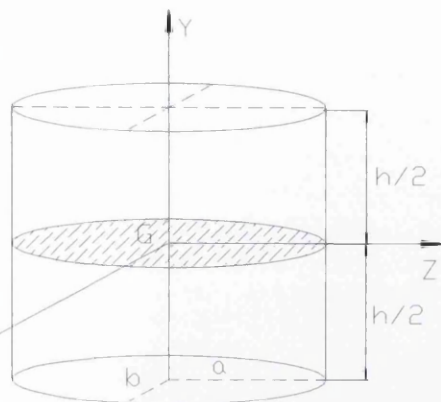
```
//-----RIGHT UPPER LEG-----
AnySeg RightUpperLeg = {
  #include "1_90/DrawSettings/Nodes.any"
  #include "1_90/DrawSettings/SegmentAxes.any"
  Axes = ...InitialConditions.UpperLeg;
  Mass = 0.1*...BP.M;
  AnyVar d = 1050; // density
  AnyVar p = 3.1416; //pi
  AnyVar h = (0.530*...BP.H) - (0.285*...BP.H);
  AnyVar radius = (Mass/(d*p*h))^0.5; // radius
  AnyVar Ixx = (1/12)*Mass*((3*radius^2)+(h^2));
  AnyVar Iyy = (1/2)*Mass*radius^2;
  AnyVar Izz = Ixx;
  Jii = {Ixx, Iyy, Izz};
  AnyVar Sign = 1;
  AnyRefNode Node0 = {
    sRel = {0,0,0};
  };
  AnyRefNode KneeJoint = {
    sRel = {-0.025, -0.258, -0.028*.Sign} * .h/0.455;
  };
  AnyDrawSurf DrwSurf = {
    FileName = "1_90/Bones/Femur.anysurf";
    ScaleXYZ = {0.001, 0.001, 0.001*..Sign} * ..h;
    RGB = Main.Model3.HumanModel.DrawSettings.Colors.Segments;
  };
};
AnyRefNode HipJoint = {
  sRel = {0, 0.197, -0.038*.Sign} * .h/0.455;
};
```

Figure 8.8 – RightUpperLeg.any File of the Parameterised model

In the above figure the number $0.455m$ is the length of the upper leg in the *Repository2* model.

- Pelvis

Elliptical Cylinder



Centre of mass (G): $G = \frac{h}{2}$

Volume: $V = \pi \cdot a \cdot b \cdot h$

Mass: $m = 0.142M$

Density: $d = \frac{m}{V} = 1010$

Moment of Inertia (G)

$h = 0.11H, a = 0.259H/2, b = \frac{m}{d \cdot \pi \cdot a \cdot h}$

$I_{x_G} = \frac{1}{4} \cdot m \cdot a^2 + \frac{1}{12} \cdot m \cdot h^2$

$I_{y_G} = \frac{1}{4} \cdot m \cdot (a^2 + b^2)$

$I_{z_G} = \frac{1}{4} \cdot m \cdot b^2 + \frac{1}{12} \cdot m \cdot h^2$

Figure 8.9 – Geometrical Properties of the Pelvis

Because of the absence of the definition of the exact height h of the pelvis in the Drillis/Contini model, the length of the pelvis above the hip joints is calculated using the pelvis of the *Repository2* model. As shown below in Figure 8.10, the length of the pelvis in the *Repository2* model is 115mm , while the overall associated body height is 1900mm . That all means that this length is $0.06 \cdot \text{overall height}$ and with the addition of the Drillis/Contini length below the hip joint which is $0.05H$, an approximate value for the pelvis length is $0.11H$.

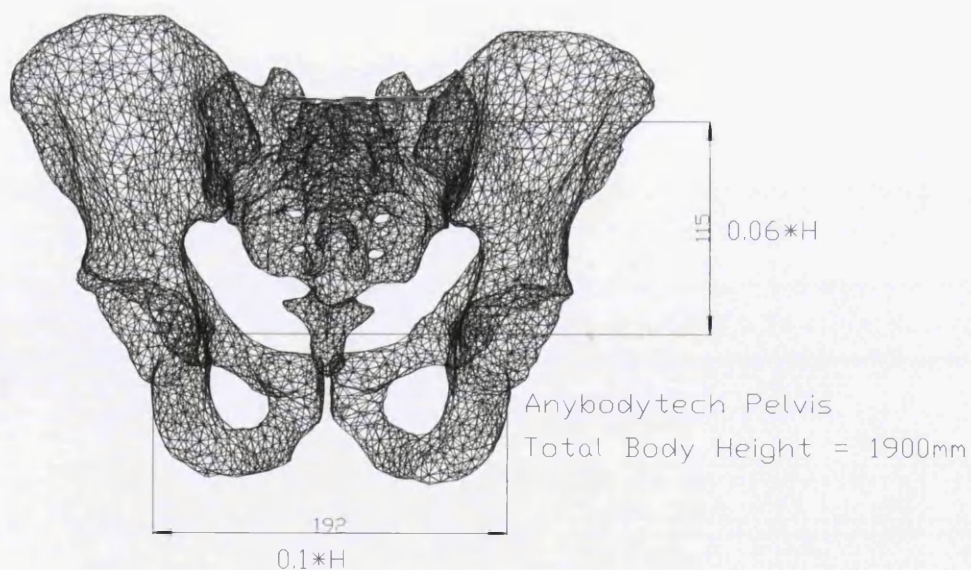


Figure 8.10 – Key Dimensions of the Pelvis of the Repository2 Model

```

AnySeg Pelvis = {

#include "l_90/DrawSettings/Nodes.any"
#include "l_90/DrawSettings/SegmentAxes.any"
Mass = 0.142*...BP.M;
AnyVar d = 1010; // density
AnyVar p = 3.1416;
AnyVar h = 0.11*...BP.H;
AnyVar a = (0.259*...BP.H)/2;
AnyVar b = Mass/(d*p*a*h);
AnyVar Ixx = ((1/4)*Mass*a^2)+((1/12)*Mass*h^2);
AnyVar Iyy = (1/4)*Mass*(a^2+b^2);
AnyVar Izz = ((1/4)*Mass*b^2)+((1/12)*Mass*h^2);
Jii = {Ixx, Iyy, Izz};
sCoM = {0, (RightHipJoint.sRel[1])/2, 0};

AnyRefNode RightHipJoint = {
  sRel = {0, -0.115, 0.096} * (((0.101 * Main.Model13.HumanModel.BP.H)/2)/0.096);
};

AnyRefNode LeftHipJoint = {
  sRel = {0, -0.115, -0.096} * (((0.101 * Main.Model13.HumanModel.BP.H)/2)/0.096);
};

AnyRefNode PelvisSacrumJoint = {
  sRel = {-0.003, -0.01, 0} * (((0.101 * Main.Model13.HumanModel.BP.H)/2)/0.096);
};

AnyRefNode Node0 = {
  sRel = {0,0,0};
};

AnyDrawSurf DrwSurf = {
  FileName = "l_90/Bones/Pelvis.anysurf";
  ScaleXYZ = {0.001, 0.001, 0.001} * (0.101 * Main.Model13.HumanModel.BP.H);
  RGB = Main.Model13.HumanModel.DrawSettings.Colors.Segments;
};

```

Figure 8.11 – Pelvis.any File of the Parameterised model

8.2.2 Joints

The joints that will be used in the lower extremity are shown in Figure 8.12 below, while the corresponding “anyfile” of the right leg is shown in Figure 8.13.

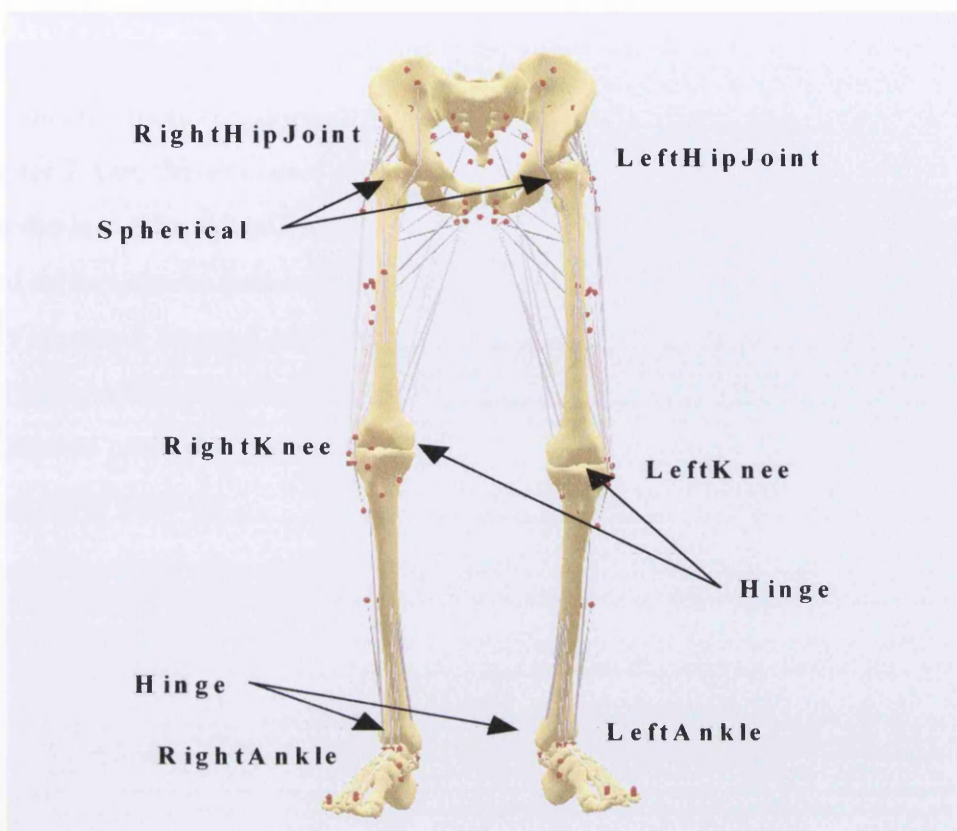


Figure 8.12 – Joints of the Lower Extremity

Joints.any File

```
//-----RIGHT ANKLE JOINT-----  
AnyRevoluteJoint RightAnkle = {  
  Axis = z;  
  AnyRefNode &RFoot = ....Segs.Legs.RightLeg.RightFoot.AnkleJoint;  
  AnyRefNode &RShank = ....Segs.Legs.RightLeg.RightLowerLeg.AnkleJoint;  
}; //Right Ankle Joint  
  
//-----RIGHT KNEE JOINT-----  
AnyRevoluteJoint RightKnee = {  
  Axis = z;  
  AnyRefNode &RShank = ....Segs.Legs.RightLeg.RightLowerLeg.KneeJoint;  
  AnyRefNode &RThigh = ....Segs.Legs.RightLeg.RightUpperLeg.KneeJoint;  
}; //Right Knee Joint  
  
//-----RIGHT HIP JOINT-----  
AnySphericalJoint RightHipJoint = {  
  AnyRefNode &RThigh=....Segs.Legs.RightLeg.RightUpperLeg.HipJoint;  
  AnyRefNode &Pelvis=....Segs.Trunk.Pelvis.RightHipJoint;  
}; //Right Hip Joint
```

Figure 8.13 – Joints.any File of the Lower Extremity

8.2.3 Validation

Figure 8.14 below, presents some time steps of the motion of the parameterized lower extremity model in three versions. The first version is for a human body with a height of $1.5m$ and a weight of $40kg$, the second is of $1.7m$ height and $65kg$ weight, while the third is of $1.9m$ height and $75kg$ weight. All the models are included in the same study, which means that they run all together. They are muscle driven as well. The way that the motion is implemented is exactly the same with the model presented in the previous Chapter 7. One driver is used for each foot, one driver for the pelvis and the proper initial conditions for the legs. More details about the position parameters of each version and the “kinematic measures” and drivers can be found on the supplementary *CD*.

As it is clearly shown in Figure 8.14 below, the validation test ended successfully. All the three models accomplished successfully the complex motion that they were assigned to perform, which means that the models may respond positively even in the most extreme postures. It can also be seen that there is no over-activation at the muscles.

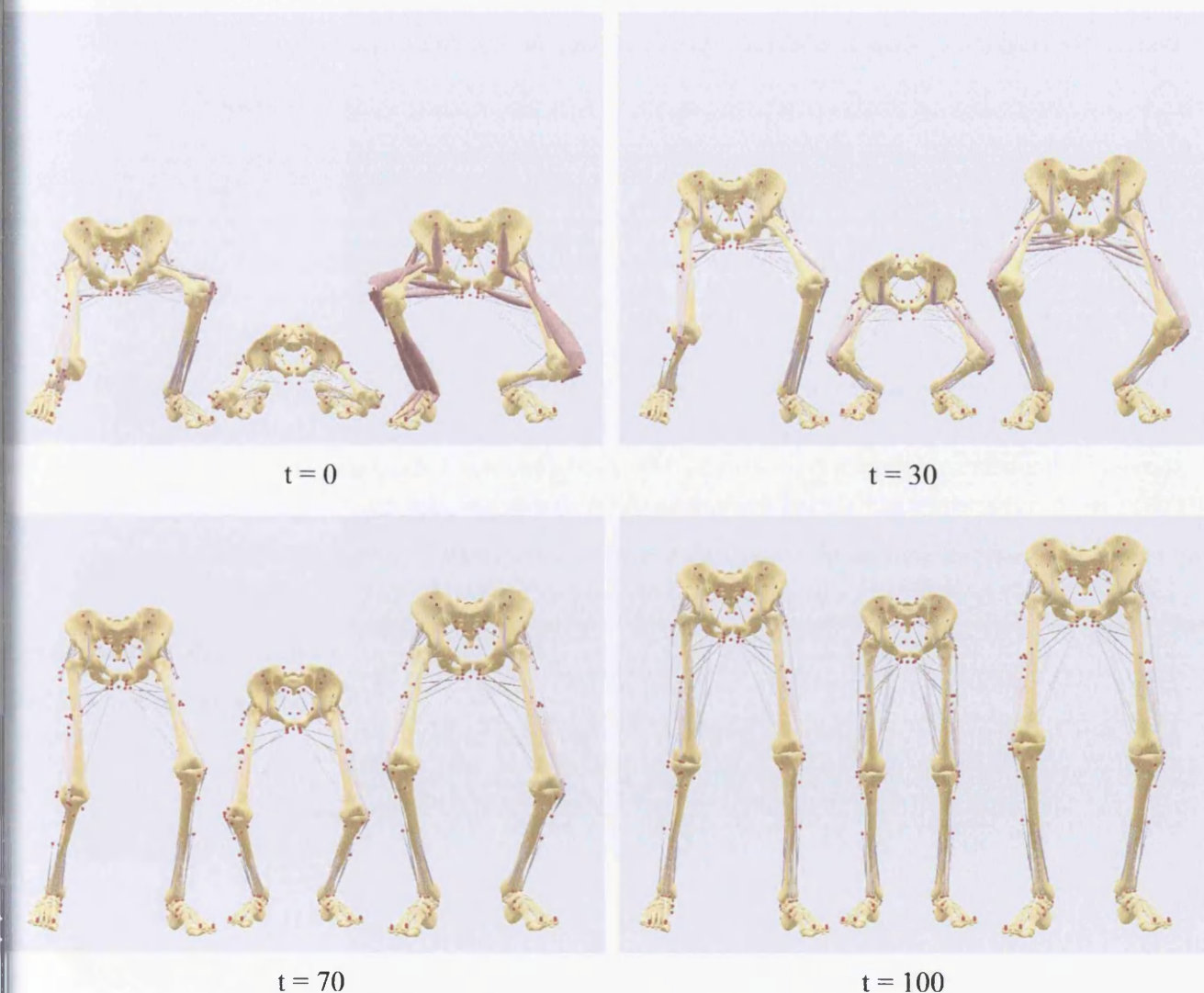


Figure 8.14 – Motion of the Lower Extremity in Three Versions

8.3 Lumbar, Thorax & Head

8.3.1 Segments

The lumbar segment is constituted by the L5L4 segment and the L3L2 segment while L1 is assumed to be part of the thorax. The thorax and head are assumed to be as one segment, which is connected to the L3L2 segment (Figure 8.15).

The L5L4, L3L2 and the thorax segments are considered elliptical cylinders, while the head is a sphere. The mass and the density of the various segments are defined using the Dempster's data¹⁴.

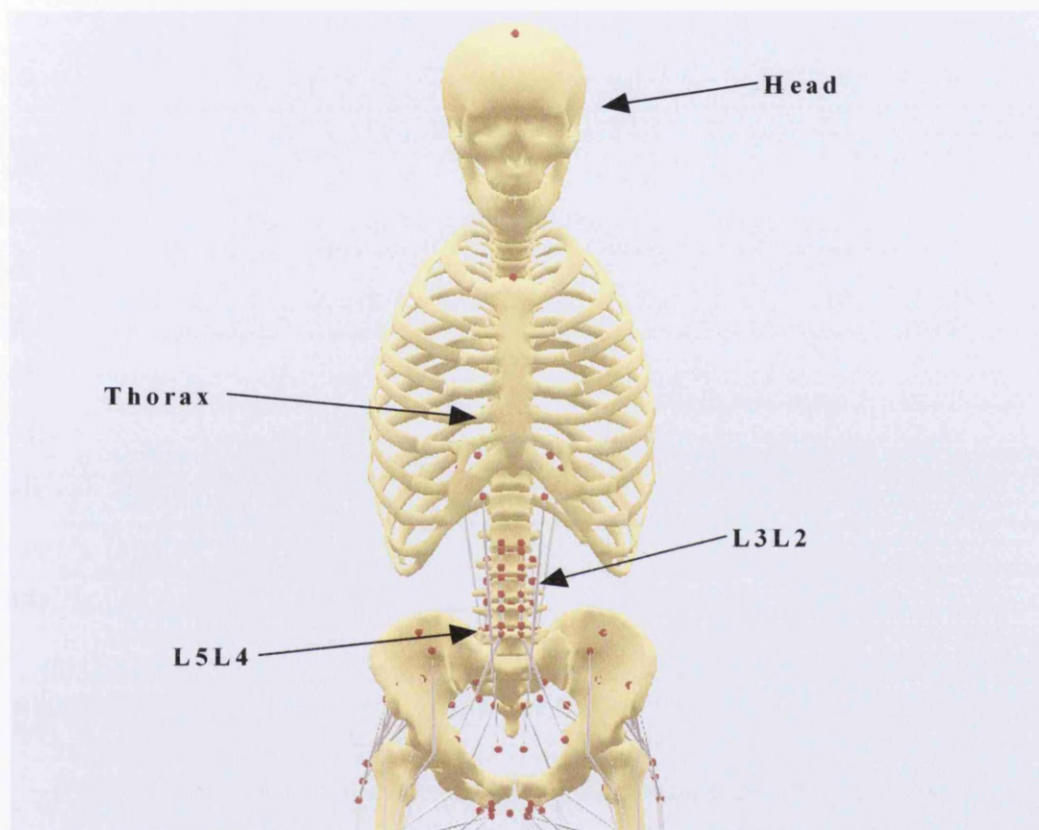


Figure 8.15 – L5L4, L3L2, Thorax & Head Segments

Because of the absence of the definition of the exact length of the lumbar and the thorax in the Drillis/Contini model, the length of these segments is calculated using *Repository2* model (Table 8.2).

Repository2 Model		
Segment	Length	Fraction of H (1.9m)
L5L4	0.0666m	$0.035H$
L3L2	0.0654m	$0.034H$
HAT	0.69m	$0.36H$

Table 8.2 – Repository2 Model's Length of L5L4/L3L2/HAT Segments

As it is shown in the Table 8.2, the total length of the segments, as a fraction of the overall height H , of the *Repository2* model, is $0.43H$. On the other hand in the Drillis/Contini model (Figure 8.1) the distance from the bottom of the pelvis to the top of the head is $0.52H$, while the length of the pelvis is $0.11H$. That means that the total length of L5L4, L3L2 and HAT segments in the Drillis/Contini model is $0.41H$. If now, the data of the Table 8.2 is adapted to the Drillis/Contini model, then the revised lengths of the segments of the *Repository2* model are presented in Table 8.3.

Drillis/Contini Model	
Segment	Fraction of H
L5L4	$0.033H$
L3L2	$0.032H$
HAT	$0.34H$

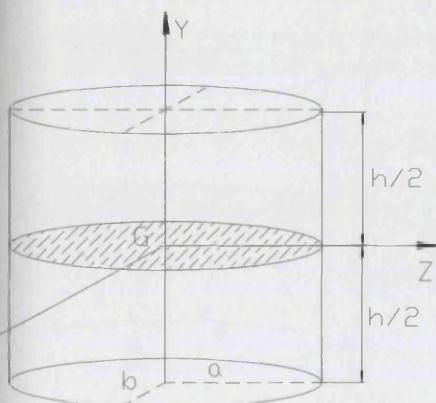
Table 8.3 – Rough Estimation of L5L4/L3L2/HAT Segments in the Drillis/Contini model

Finally, the length of the thorax in the Drillis/Contini model is obtained by the subtraction of the lengths of the head ($0.130H$) and the neck ($0.052H$) from the overall length (HAT) which is $0.34H$ (Table 8.3). As a result the length of the thorax is $0.158H$.

In the investigation that will follow, the way of parameterization of each of the segments will be presented. Due to the large number of nodes that each segment contains, only the part of the corresponding “anyfile” that has to do with the geometrical properties of it, will be presented, and also just a few of the nodes, for illustrative purpose only. The full “anyfiles” can be found on the supplementary CD.

• L5L4

Elliptical Cylinder



Centre of mass (G): $G = \frac{h}{2}$

Volume: $V = \pi \cdot a \cdot b \cdot h$

Mass: $m = 0.139M / 2$

Density: $d = \frac{m}{V} = 1010$

Moment of Inertia (G)

$$h = 0.033H, a = 0.259H / 2, b = \frac{m}{d \cdot \pi \cdot a \cdot h}$$

$$I_{x_G} = \frac{1}{4} \cdot m \cdot a^2 + \frac{1}{12} \cdot m \cdot h^2$$

$$I_{y_G} = \frac{1}{4} \cdot m \cdot (a^2 + b^2)$$

$$I_{z_G} = \frac{1}{4} \cdot m \cdot b^2 + \frac{1}{12} \cdot m \cdot h^2$$

Figure 8.16 – Geometrical Properties of L5L4

```
L5L4Seg.any File
```

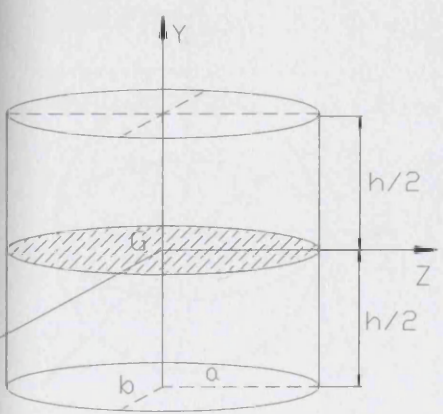
```
//-----L5/L4 Segment-----
AnySeg L5L4Seg = {
  #include "1_90/DrawSettings/Nodes.any"
  #include "1_90/DrawSettings/SegmentAxes.any"
  Mass = (0.139*...BP.M)/2;
  AnyVar d = 1010;          // density
  AnyVar p = 3.1416;
  AnyVar h = 0.033*...BP.H;
  AnyVar a = (0.259*...BP.H)/2;
  AnyVar b = Mass/(d*p*a*h);
  AnyVar Ixx = ((1/4)*Mass*a^2)+((1/12)*Mass*h^2);
  AnyVar Iyy = (1/4)*Mass*(a^2+b^2);
  AnyVar Izz = ((1/4)*Mass*b^2)+((1/12)*Mass*h^2);
  Jii = {Ixx, Iyy, Izz};
  AnyRefNode L5SacrumJoint = {
    sRel = {-0.0072, -0.0333, 0} * .h/0.0666;
  };
  AnyRefNode L4L3Joint = {
    sRel = {0.0071, 0.0333, 0} * .h/0.0666;
  };
  AnyDrawSurf DrwSurf = {
    FileName = "1_90/Bones/L5L4.anysurf";
    ScaleXYZ = {0.001, 0.001, 0.001} * .h;
    RGB = Main.Model3.HumanModel.DrawSettings.Colors.Segments;
  };
};
```

Figure 8.17 – L5L4Seg.any File of the Parameterized Model

In the above figure, the number *0.0666m* is the length of the L5L4 segment in the *Repository2* model.

• **L3L2**

Elliptical Cylinder



Centre of mass (G): $G = \frac{h}{2}$

Volume: $V = \pi \cdot a \cdot b \cdot h$

Mass: $m = 0.139M / 2$

Density: $d = \frac{m}{V} = 1010$

Moment of Inertia (G)

$h = 0.032H, a = 0.259H/2, b = \frac{m}{d \cdot \pi \cdot a \cdot h}$

$I_{xG} = \frac{1}{4} \cdot m \cdot a^2 + \frac{1}{12} \cdot m \cdot h^2$

$I_{yG} = \frac{1}{4} \cdot m \cdot (a^2 + b^2)$

$I_{zG} = \frac{1}{4} \cdot m \cdot b^2 + \frac{1}{12} \cdot m \cdot h^2$

Figure 8.18 – Geometrical Properties of L3L2

```
L3L2Seg.any File
```

```

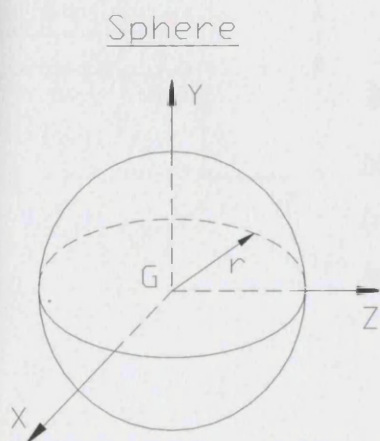
//-----L3/L2 Segment-----
AnySeg L3L2Seg = {
  #include "1_90/DrawSettings/Nodes.any"
  #include "1_90/DrawSettings/SegmentAxes.any"
  Mass = (0.139*...BP.M)/2;
  AnyVar d = 1010;           // density
  AnyVar p = 3.1416;
  AnyVar h = 0.032*...BP.H;
  AnyVar a = (0.259*...BP.H)/2;
  AnyVar b = Mass/(d*p*a*h);
  AnyVar Ixx = ((1/4)*Mass*a^2)+((1/12)*Mass*h^2);
  AnyVar Iyy = (1/4)*Mass*(a^2+b^2);
  AnyVar Izz = ((1/4)*Mass*b^2)+((1/12)*Mass*h^2);
  Jii = {Ixx, Iyy, Izz};
  AnyRefNode L3L4Joint = {
    sRel = {0.0085, -0.0327, 0} * .h/0.0654;
  };
  AnyRefNode L2HATJoint = {
    sRel = {-0.0085, 0.0327, 0} * .h/0.0654;
  };
  AnyDrawSurf DrwSurf = {
    FileName = "1_90/Bones/L3L2.anysurf";
    ScaleXYZ = {0.001, 0.001, 0.001} * .h;
    RGB = Main.Model3.HumanModel.DrawSettings.Colors.Segments;
  };
};

```

Figure 8.19 – L3L2Seg.any File of the Parameterized Model

In the above figure, the number $0.0654m$ is the length of the L3L2 segment in the *Repository2* model.

• **Head & Thorax (HAT)**



Centre of mass (G): $G = r$

Volume: $V = \frac{4}{3} \cdot \pi \cdot r^3$

Mass: $m = 0.081M$

Density: $d = \frac{m}{V} = 1110$

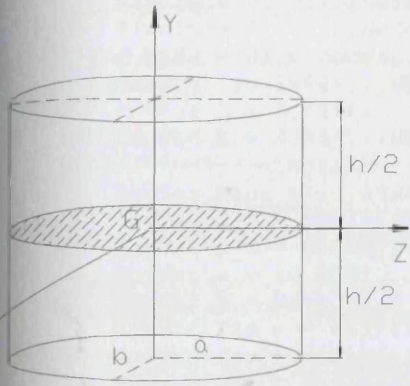
Moment of Inertia (G)

$$r = \left(\frac{3 \cdot m}{d \cdot 4 \cdot \pi} \right)^{\frac{1}{3}}$$

$$I_{x_G} = I_{y_G} = I_{z_G} = \frac{2}{5} \cdot m \cdot r^2$$

Figure 8.20 – Geometrical Properties of Head

Elliptical Cylinder



Centre of mass (G): $G = \frac{h}{2}$

Volume: $V = \pi \cdot a \cdot b \cdot h$

Mass: $m = 0.216M - 0.0266M$
 $= 0.1894M$

Density: $d = \frac{m}{V} = 920$

Moment of Inertia (G)

$h = 0.158H, a = 0.259H/2, b = \frac{m}{d \cdot \pi \cdot a \cdot h}$

$I_{x_G} = \frac{1}{4} \cdot m \cdot a^2 + \frac{1}{12} \cdot m \cdot h^2$

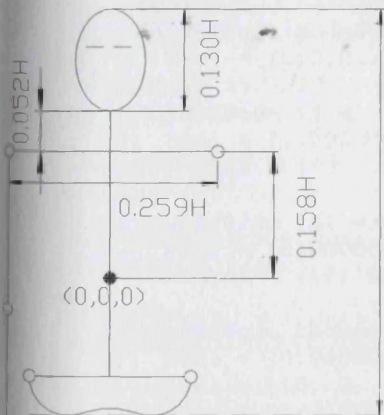
$I_{y_G} = \frac{1}{4} \cdot m \cdot (a^2 + b^2)$

$I_{z_G} = \frac{1}{4} \cdot m \cdot b^2 + \frac{1}{12} \cdot m \cdot h^2$

*Scapula & Clavicula are 2kg in the 75kg Repository2 model which is 0.0266M

Figure 8.21 – Geometrical Properties of Thorax

Head & Thorax



Centre of Mass

$HAT(G) = (\bar{X}, \bar{Y}, \bar{Z}) = (0, 0.137H, 0)$

$\bar{X} = 0, \bar{Z} = 0$

$\bar{Y} = \frac{m_T \cdot 0.079H + m_H \cdot 0.275H}{m_T + m_H}$

$m_T = 0.1894M$

$m_H = 0.081M$

$\Rightarrow \bar{Y} = \frac{0.015 \cdot H \cdot M + 0.022 \cdot H \cdot M}{0.27M} = 0.137H$

Moment of Inertia

$I_{xx_G} = [I_{xx_T} + m_T \cdot (0.137H - 0.079H)^2] + [I_{xx_H} + m_H \cdot (0.275H - 0.137H)^2]$

$I_{yy_G} = [I_{yy_T} + I_{yy_H}]$

$I_{zz_G} = [I_{zz_T} + m_T \cdot (0.137H - 0.079H)^2] + [I_{zz_H} + m_H \cdot (0.275H - 0.137H)^2]$

Figure 8.22 – CoM & Mol of HAT

In the Figure 8.23 below, the corresponding “anyfile” of HAT is presented, where the number 0.69m is the length of the HAT segment in the Repository2 model.

```

//-----HAT Segment-----
AnySeg HATSeg = {
  #include "1_90/DrawSettings/Nodes.any"
  #include "1_90/DrawSettings/SegmentAxes.any"
  // Axes = Main.Model3.HumanModel.InitialConditions.HATSeg;
  Mass = (0.1894*...BP.M + 0.081*...BP.M);
  AnyVar p = 3.1416;
  AnyVar h = 0.34*...BP.H;
  //-----Thorax-----
  AnyVar Mass_t=(0.1894*...BP.M);
  AnyVar d_t = 920; // density
  AnyVar h_t = 0.158*...BP.H; //t for thorax
  AnyVar a = (0.259*...BP.H)/2;
  AnyVar b = Mass_t/(d_t*p*a*h_t);
  AnyVar Ixx_t = ((1/4)*Mass_t*a^2)+((1/12)*Mass_t*h_t^2); //t for thorax
  AnyVar Iyy_t = (1/4)*Mass_t*(a^2+b^2);
  AnyVar Izz_t = ((1/4)*Mass_t*b^2)+((1/12)*Mass_t*h_t^2);
  //-----Head-----
  AnyVar Mass_h = 0.081*...BP.M;
  //AnyVar h_h = 0.130*...BP.H; //h for head
  AnyVar d_h = 1110; // density
  AnyVar radius = ((3*Mass_h)/(d_h*4*p))^(1/3); // radius
  AnyVar Ixx_h = (2/5)*Mass_h*radius^2;
  AnyVar Iyy_h = Ixx_h;
  AnyVar Izz_h = Ixx_h;
  //-----
  AnyVar Ixx = (Ixx_t + Mass_t*(0.137*...BP.H - 0.079*...BP.H)^2) + (Ixx_h +
    Mass_h*(0.275*...BP.H - 0.137*...BP.H)^2);
  AnyVar Iyy = (Iyy_t + Iyy_h);
  AnyVar Izz = (Izz_t + Mass_t*(0.137*...BP.H - 0.079*...BP.H)^2) + (Izz_h +
    Mass_h*(0.275*...BP.H - 0.137*...BP.H)^2);
  Jii = {Ixx, Iyy, Izz};
  sCoM = {-0.04*h/0.69, (HATL2Joint.sRel[1]+0.137*...BP.H), 0};
  AnyRefNode Node0 = {
    sRel = {0,0,0};
  };
  AnyRefNode ij = {
    sRel = {0.000000, 0.000000 , 0.000000};
    // ARel = {{0, 0, -1}, {0, 1, 0}, {1, 0, 0}};
  };
  AnyRefNode iji = {
    sRel = {0.000000, 0.000000 , 0.000000};
    // ARel = {{0, 0, -1}, {0, 1, 0}, {1, 0, 0}};
  };
  AnyRefNode thorax_bone = {
    sRel = {0.000000, 0.000000 , 0.000000};
    ARel = {{0, 0, -1}, {0, 1, 0}, {1, 0, 0}};
    AnyDrawSurf DrwSurf = {
      FileName = "1_90/Bones/thorax_bones";
      ScaleXYZ = {1, 1, 1}*..h/0.69;
      RGB = Main.Model3.HumanModel.DrawSettings.Colors.Segments;
    };
  };
  AnyRefNode skull_bone = {
    sRel = {0.000000, 0.000000 , 0.000000};
    ARel = {{0, 0, -1}, {0, 1, 0}, {1, 0, 0}};
    AnyDrawSurf DrwSurf = {
      FileName = "1_90/Bones/skull";
      ScaleXYZ = {1, 1, 1}*..h/0.69;
      RGB = Main.Model3.HumanModel.DrawSettings.Colors.Segments;
    };
  };
  AnyRefNode TopOfHead = {
    sRel = {0, 0.32 , 0} * .h/0.69;
  };
  AnyRefNode HATL2Joint = {
    sRel = {-0.02, -0.37, 0} * .h/0.69;
  };
};

```

Figure 8.23 – HATSeg.any File of the Parameterised Model

8.3.2 Joints

The joints that will be used to connect the L5L4, L4L3 and HAT to each other and to the pelvis are shown in Figure 8.24 below, while the corresponding “anyfile” is depicted in Figure 8.25. It has to be noted that the sacrum does not have any geometrical properties because it is assumed to be part of the pelvis.

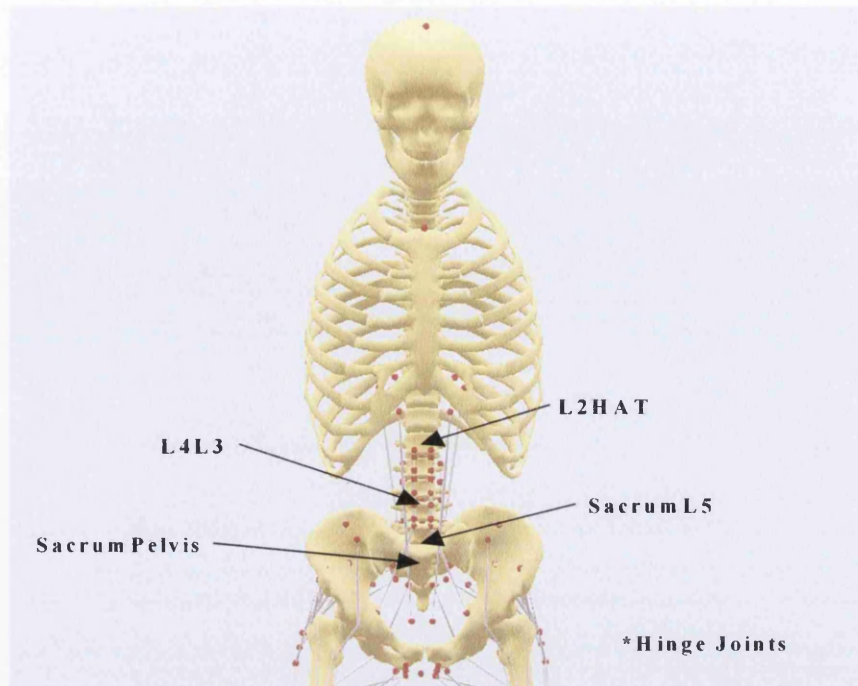


Figure 8.24 – Joints of Sacrum, L5L4, L4L3 & HAT Segments

Joints.any File

```
//-----PELVIS-SACRUM JOINT-----
AnyRevoluteJoint SacrumPelvis = {
  Axis = z;
  AnyRefNode &Pelvis = ...Segs.Trunk.Pelvis.PelvisSacrumJoint;
  AnyRefNode &Sacrum = ...Segs.Trunk.Sacrum.SacrumPelvisJoint;
};
//-----SACRUM-L5 JOINT-----
AnyRevoluteJoint SacrumL5 = {
  Axis = z;
  AnyRefNode &Sacrum = ...Segs.Trunk.Sacrum.SacrumL5Joint;
  AnyRefNode &L5L4 = ...Segs.Trunk.L5L4Seg.L5SacrumJoint;
};
//-----L4-L3 JOINT-----
AnyRevoluteJoint L4L3 = {
  Axis = z;
  AnyRefNode &L5L4 = ...Segs.Trunk.L5L4Seg.L4L3Joint;
  AnyRefNode &L3L2 = ...Segs.Trunk.L3L2Seg.L3L4Joint;
};
//-----L3-HAT JOINT-----
AnyRevoluteJoint L2HAT = {
  Axis = z;
  AnyRefNode &L3L2 = ...Segs.Trunk.L3L2Seg.L2HATJoint;
  AnyRefNode &HAT = ...Segs.Trunk.HATSeg.HATL2Joint;
};
```

Figure 8.25 – Joints.any File of Sacrum, L5L4, L3L2 & HAT Segments

8.3.3 Validation

Figure 8.26 below, presents some time steps of the motion of the parameterized lower extremity models presented in Section 8.2 extended with the new parameterized segments discussed in this section. All the models are included in the same study again, and they are muscle driven. The way that the motion is implemented is exactly the same as it was with the model presented in Chapter 7, regarding the use of the *AnyKinMeasureLinComb* facility for the joints of Figure 8.24. The only difference now is, that the coefficient matrix $Cqef$, because of the four joints, has four columns. More details can be found in the “kinematic measures” and drivers “anyfiles” on the supplementary CD.

As it is shown in Figure 8.26 below, all the three models worked properly with the addition of the lumbar, thorax and head. It has to be noted that the vivid red colour in the muscles of the model which is of $1.9m$ height, at time step $t=0$, denotes that these muscles are over activated. That means that there is heavy loading exceeding the muscle capacity. To avoid this, the movement should be changed or the muscle capacity should be increased.

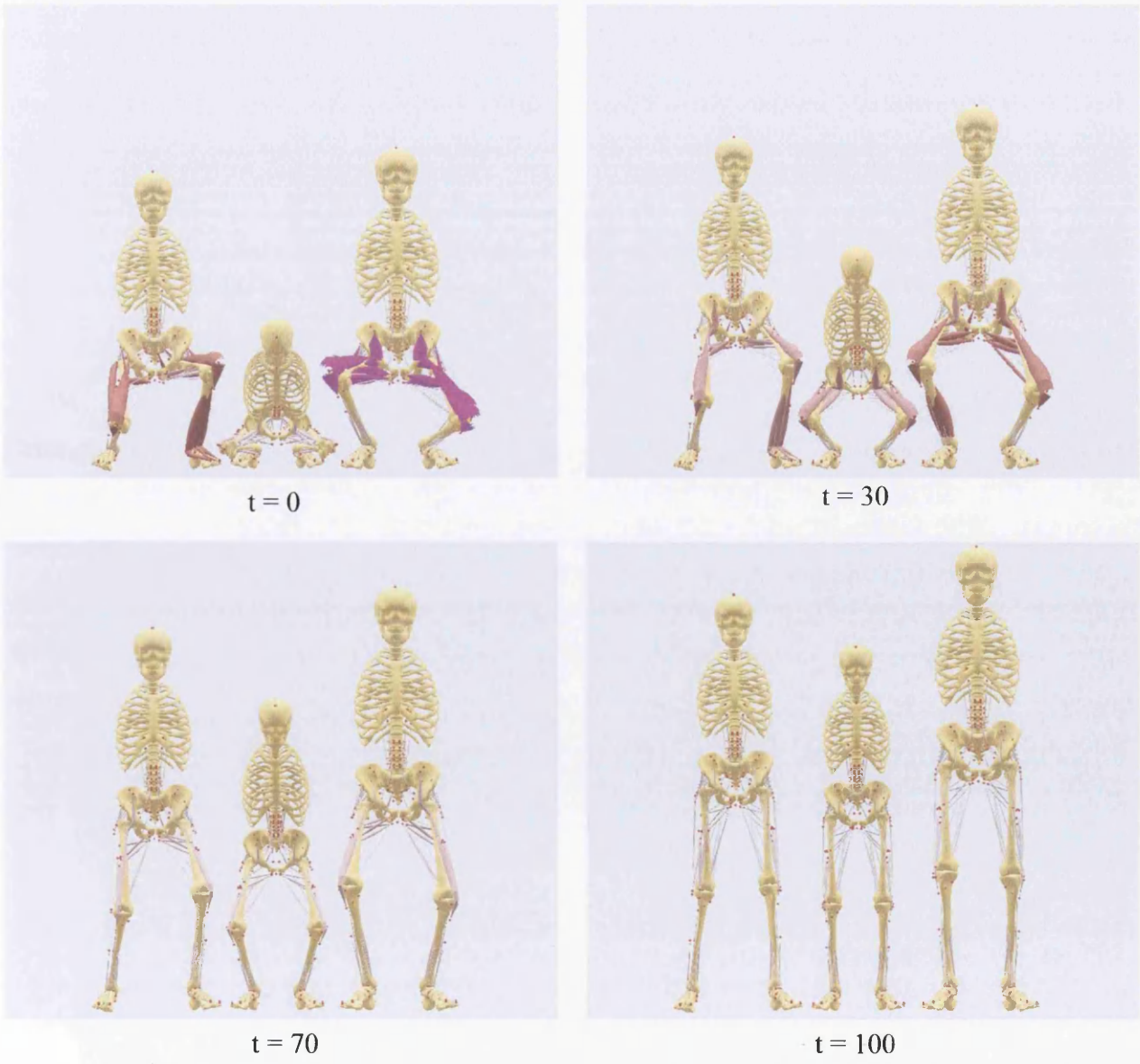


Figure 8.26 – Motion of the Lower Extremity with L5L4, L3L2 & HAT Segments

8.4 Upper Limb

8.4.1 Segments

The upper limb is constituted by the clavicle, the scapula, the humerus, the ulna, the radius and the hand (Figure 8.27). All these segments are assumed to be circular cylinders. The mass and the density of these segments are defined employing Dempster's data¹⁴.

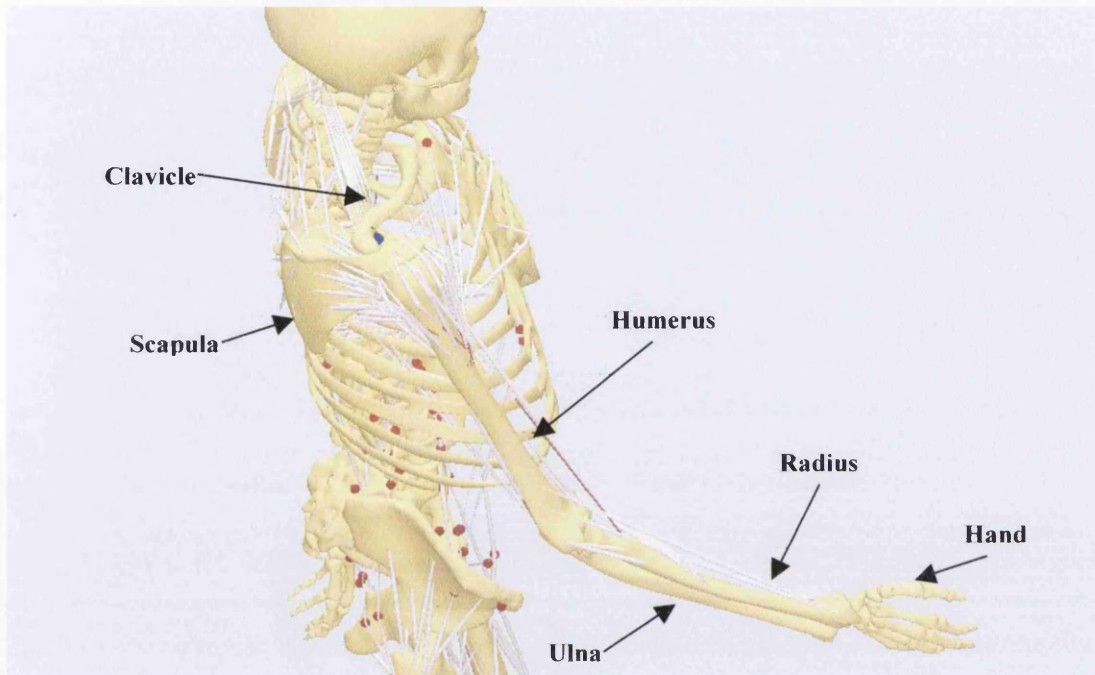
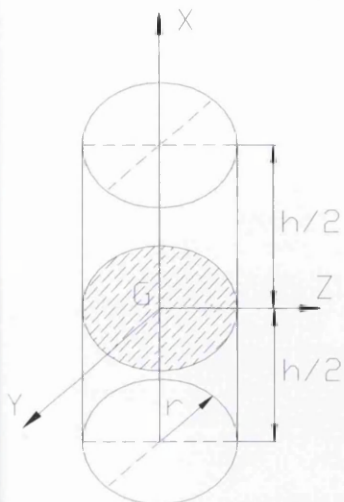


Figure 8.27 – Segments of the Upper Limb

Because of the absence of the definition of the exact dimensions of the scapula and the clavicle in the Drillis/Contini model, the dimensions of these segments are calculated using *Repository2* model. In the investigation that will follow, the way of parameterization of each of the segments will be presented. Due to the large number of nodes that each segment contains, only the part of the corresponding “anyfile” that has to do with the geometrical properties of it, will be presented, and also just a few of the nodes, for illustrative purpose only. The full “anyfiles” can be found on the supplementary *CD*.

- **Clavicle**

Circular Cylinder



Centre of mass (G): $G = \frac{h}{2}$

Volume: $V = \pi \cdot r^2 \cdot h$

Mass: $m = 0.0066M$

Density: $d = \frac{m}{V} = 920$

Moment of Inertia (G)

$$h = \left((ac.sRel[0] - scj.srel[0])^2 \right)^{0.5}, r = \left(\frac{m}{d \cdot \pi \cdot h} \right)^{\frac{1}{2}}$$

$$I_{y_G} = \frac{1}{12} \cdot m \cdot (3r^2 + h^2)$$

$$I_{x_G} = \frac{1}{2} \cdot m \cdot r^2$$

$$I_{z_G} = I_{y_G}$$

Figure 8.28 – Geometrical Properties of Clavicle

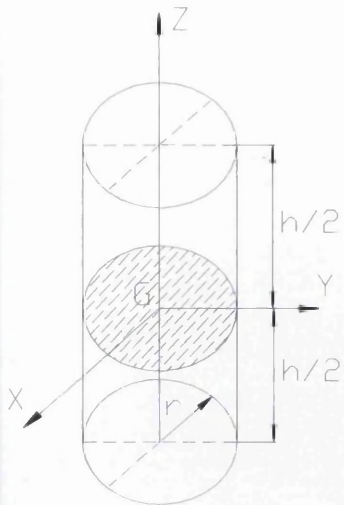
In the above figure, *ac* is the *acromion* and *scj* is a joint location which will be explained later in this chapter. In the Figure 8.29, the number *0.69m* is the length of the HAT segment in the *Repository2* model.

```
//-----RIGHT CLAVICULA-----
AnySeg RightClavicula = {
  #include "1_90/RightUpperLimbs\Clavicula-Nodes.any"
  Mass = 0.0066 * .....BP.M;
  // Axes = .....InitialConditions.R_Clavicula;
  AnyVar d = 920; // density
  AnyVar p = 3.1416; //pi
  AnyVar h = ((ac.sRel[0]-SCJoint.sRel[0])^2)^0.5;
  AnyVar radius = (Mass/(d*p*h))^0.5; // radius
  AnyVar Ixx = (1/2)*Mass*radius^2;
  AnyVar Iyy = (1/12)*Mass*((3*radius^2)+(h^2));
  AnyVar Izz = Iyy;
  Jii = {Ixx, Iyy, Izz};
  sCoM = {h/2,0,0};
  AnyRefNode Node0 = {
    sRel = {0,0,0};
  };
  AnyRefNode ac = {
    sRel = {0.168874, 0, 0 * ...Sign} *
    Main.Model13.HumanModel.Segs.Trunk.HATSeg.h/0.69;
  };
  AnyRefNode ACJoint = {
    sRel = {0.155254, 0.002989, -0.006589 * ...Sign} *
    Main.Model13.HumanModel.Segs.Trunk.HATSeg.h/0.69;
  };
  AnyRefNode SCJoint = {
    sRel = {0, 0, 0 * ...Sign} *
    Main.Model13.HumanModel.Segs.Trunk.HATSeg.h/0.69;
  };
};
```

Figure 8.29 – RightClavicula.any File of the Parameterised Model

• **Scapula**

Circular Cylinder



Centre of mass (G): $G = \frac{h}{2}$

Volume: $V = \pi \cdot r^2 \cdot h$

Mass: $m = 0.0066M$

Density: $d = \frac{m}{V} = 920$

Moment of Inertia (G)

$h = 0.00526H, r = \left(\frac{m}{d \cdot \pi \cdot h}\right)^{\frac{1}{2}}$

$I_{y_G} = \frac{1}{12} \cdot m \cdot (3r^2 + h^2)$

$I_{z_G} = \frac{1}{2} \cdot m \cdot r^2$

$I_{x_G} = I_{y_G}$

Figure 8.30 – Geometrical Properties of Scapula

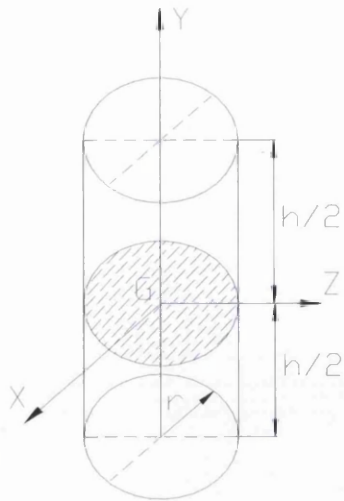
Because of the absence of the definition of the exact height h of the scapula in the Drillis/Contini model, the length of this segment is calculated using the scapula ($0.01m/1.9m=0.00526M$) of the *Repository2* model. In the Figure 8.31, the number $0.69m$ is the length of the HAT segment in the *Repository2* model.

```
//-----RIGHT SCAPULA-----
AnySeg RightScapula = {
  #include "1_90/RightUpperLimbs\Scapula-Nodes.any"
  #include "1_90/RightUpperLimbs\Scapula-Nodes-Forearm.any"
  // r = .....InitialConditions.R_Scapula_r;
  Mass = 0.0066 * Main.Model3.HumanModel.BP.M;
  Axes = .....InitialConditions.R_Scapula;
  AnyVar d = 920; // density
  AnyVar p = 3.1416; //pi
  AnyVar h = 0.00526* Main.Model3.HumanModel.BP.H;
  AnyVar radius = (Mass/(d*p*h))^0.5; // radius
  AnyVar Ixx = Iyy;
  AnyVar Iyy = (1/12)*Mass*((3*radius^2)+(h^2));
  AnyVar Izz = (1/2)*Mass*radius^2;
  Jii = {Ixx, Iyy, Izz};
  sCoM = {-0.06, -0.04, 0.02 * ..Sign} *
    Main.Model3.HumanModel.Segs.Trunk.HATSeg.h/0.69;
  AnyRefNode Node0 = {
    sRel = {0,0,0};
  };
  AnyRefNode ACJoint = {
    sRel = {-0.000929, 0.001401, -0.015331 * ...Sign} *
      Main.Model3.HumanModel.Segs.Trunk.HATSeg.h/0.69;
  };
  AnyRefNode gh = {
    sRel = {0.015898, -0.050863, -0.024381 * ...Sign} *
      Main.Model3.HumanModel.Segs.Trunk.HATSeg.h/0.69;
  };
};
```

Figure 8.31 – RightScapula.any File of the Parameterised Model

• **Humerus**

Circular Cylinder



Centre of mass (G): $G = \frac{h}{2}$

Volume: $V = \pi \cdot r^2 \cdot h$

Mass: $m = 0.028M$

Density: $d = \frac{m}{V} = 1070$

Moment of Inertia (G)

$h = 0.818H - 0.630H, r = \left(\frac{m}{d \cdot \pi \cdot h} \right)^{\frac{1}{2}}$

$I_{x_G} = \frac{1}{12} \cdot m \cdot (3r^2 + h^2)$

$I_{y_G} = \frac{1}{2} \cdot m \cdot r^2$

$I_{z_G} = I_{x_G}$

Figure 8.32 – Geometrical Properties of Humerus

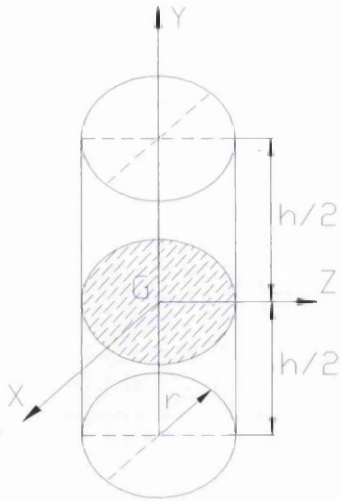
```
//-----RIGHT HUMERUS-----
AnySeg RightHumerus = {
  #include "1_90/RightUpperLimbs\Humerus-Nodes.any"
  #include "1_90/RightUpperLimbs\Humerus-Nodes-Forearm.any"
  Mass = 0.028* Main.Model3.HumanModel.BP.M;
  // Axes = .....InitialConditions.R_Humerus;
  AnyVar d = 1070; // density
  AnyVar p = 3.1416;
  AnyVar h = (0.818* Main.Model3.HumanModel.BP.H) - (0.630*
    Main.Model3.HumanModel.BP.H);
  AnyVar radius = (Mass/(d*p*h))^0.5; // radius
  AnyVar Ixx = (1/12)*Mass*((3*radius^2)+(h^2));
  AnyVar Iyy = (1/2)*Mass*radius^2;
  AnyVar Izz = Ixx;
  Jii = {Ixx, Iyy, Izz};
  sCoM = {0, -0.18 * h , 0};
  AnyRefNode gh = {
    sRel = { 0, 0 , 0 * ...Sign} * .h/0.31121;
  };
  AnyRefNode em = {
    sRel = { -0.03051, -0.311210, 0 * ...Sign} * .h/0.31121;
  };
  AnyRefNode el = {
    sRel = { 0.03051, -0.306605, 0 * ...Sign} * .h/0.31121;
  };
  AnyRefNode fe = {
    sRel = {(0.013072-.01), -0.331048 , -0.007917 * ...Sign} * .h/0.31121;
  };
};
```

Figure 8.33 – RightHumerus.any File of the Parameterised Model

In the Figure 8.33 above, the number $0.31121m$ is the length of the humerus segment in the *Repository2* model.

- **Ulna**

Circular Cylinder



Centre of mass (G): $G = \frac{h}{2}$

Volume: $V = \pi \cdot r^2 \cdot h$

Mass: $m = (0.016M)/2$

Density: $d = \frac{m}{V} = 1130$

Moment of Inertia (G)

$$h = 0.630H - 0.485H, r = \left(\frac{m}{d \cdot \pi \cdot h} \right)^{\frac{1}{2}}$$

$$I_{x_G} = \frac{1}{12} \cdot m \cdot (3r^2 + h^2)$$

$$I_{y_G} = \frac{1}{2} \cdot m \cdot r^2$$

$$I_{z_G} = I_{x_G}$$

Figure 8.34 – Geometrical Properties of Ulna

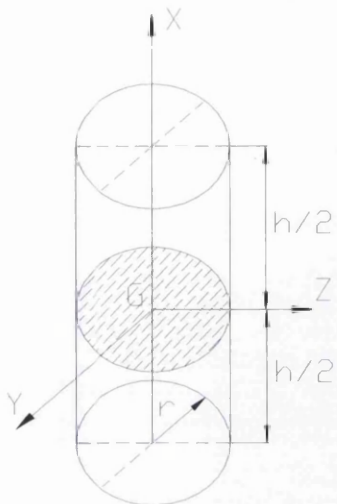
```
//-----RIGHT ULNA-----
AnySeg RightUlna = {
  #include "1_90/RightUpperLimbs/Ulna-Nodes.any"
  Mass = (0.016* Main.Model3.HumanModel.BP.M)/2;
  // Axes = Main.Model3.HumanModel.InitialConditions.R_Ulna;
  AnyVar d = 1130; // density
  AnyVar p = 3.1416;
  AnyVar h = 1.03*(0.630* Main.Model3.HumanModel.BP.H) -
    (0.485* Main.Model3.HumanModel.BP.H);
  AnyVar radius = (Mass/(d*p*h))^0.5; // radius
  AnyVar Ixx = (1/12)*Mass*((3*radius^2)+(h^2));
  AnyVar Iyy = (1/2)*Mass*radius^2;
  AnyVar Izz = Ixx;
  Jii = {Ixx, Iyy, Izz};
  sCoM = {0, 0.43 *-1 * h, 0 * ..Sign};
  AnyRefNode Node0 = {
    sRel = {0,0,0};
  };
  AnyRefNode NodeHand = {
    sRel = {-0.05,-0.36,-0.025 *...Sign} * .h/0.265;
  };
  AnyRefNode fe = {
    sRel = {0, 0, 0 * ...Sign} * .h/0.265;
  };
  AnyRefNode ol = {
    sRel = {-1*0.019665, -0.005124 *-1 , 0.033 * ...Sign} * .h/0.265;
  };
  AnyRefNode us = {
    sRel = {-1*0.024878, 0.270451 *-1 , 0 * ...Sign} * .h/0.265;
  };
};
```

Figure 8.35 – RightUlna.any File of the Parameterised Model

In the Figure 8.35 above, the number *0.265m* is the length of the ulna segment in the *Repository2* model.

- **Radius**

Circular Cylinder



Centre of mass (G): $G = \frac{h}{2}$

Volume: $V = \pi \cdot r^2 \cdot h$

Mass: $m = (0.016M)/2$

Density: $d = \frac{m}{V} = 1130$

Moment of Inertia (G)

$h = 0.630H - 0.485H, r = \left(\frac{m}{d \cdot \pi \cdot h} \right)^{\frac{1}{2}}$

$I_{y_G} = \frac{1}{12} \cdot m \cdot (3r^2 + h^2)$

$I_{x_G} = \frac{1}{2} \cdot m \cdot r^2$

$I_{z_G} = I_{y_G}$

Figure 8.36 – Geometrical Properties of Radius

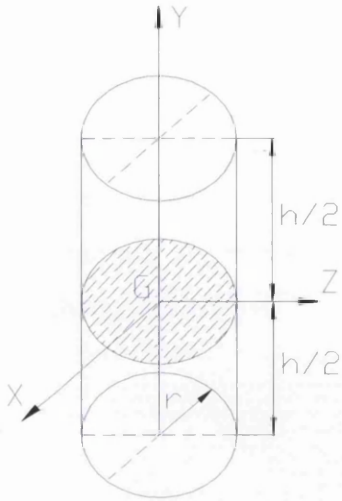
```
//-----RIGHT RADIUS-----
AnySeg RightRadius = {
  #include "1_90/RightUpperLimbs\Radius-Nodes.any"
  Mass = (0.016* Main.Model3.HumanModel.BP.M)/2;
  //Axes = Main.Model3.HumanModel.InitialConditions.R_Radius;
  AnyVar d = 1130; // density
  AnyVar p = 3.1416;
  AnyVar h = (0.630* Main.Model3.HumanModel.BP.H) - (0.485*
    Main.Model3.HumanModel.BP.H);
  AnyVar radius = (Mass/(d*p*h))^0.5; // radius
  AnyVar Iyy = (1/12)*Mass*((3*radius^2)+(h^2));
  AnyVar Ixx = (1/2)*Mass*radius^2;
  AnyVar Izz = Iyy;
  Jii = {Ixx, Iyy, Izz};
  sCoM = {-0.43* h *-1, 0 * -1, 0 *..Sign};
  AnyRefNode Node0 = {
    sRel = {0,0,0};
  };
  AnyRefNode PointPS = {
    sRel = {0 * -1, 0* -1, 0 * ...Sign} * .h/0.236653;
  };
  AnyRefNode rs = {
    sRel = {-1*-0.236653,-1* -0.041349, 0 * ...Sign} * .h/0.236653;
  };
  AnyRefNode wj = {
    sRel = {-1*-0.232338, -1*-0.017963, 0.000742 * ...Sign} * .h/0.236653;
  };
};
```

Figure 8.37 – RightRadius.any File of the Parameterised Model

In the Figure 8.37 above, the number *0.236653m* is the length of the radius segment in the *Repository2* model.

- **Hand**

Circular Cylinder



Centre of mass (G): $G = \frac{h}{2}$

Volume: $V = \pi \cdot r^2 \cdot h$

Mass: $m = 0.006M$

Density: $d = \frac{m}{V} = 1160$

Moment of Inertia (G)

$$h = 0.485H - 0.377H, r = \left(\frac{m}{d \cdot \pi \cdot h} \right)^{\frac{1}{2}}$$

$$I_{x_G} = \frac{1}{12} \cdot m \cdot (3r^2 + h^2)$$

$$I_{y_G} = \frac{1}{2} \cdot m \cdot r^2$$

$$I_{z_G} = I_{x_G}$$

Figure 8.38 – Geometrical Properties of Hand

```
//-----RIGHT HAND-----
AnySeg RightHand = {
  Mass = 0.006* Main.Model3.HumanModel.BP.M;
  // Axes = Main.Model3.HumanModel.InitialConditions.R_Hand;
  AnyVar d = 1160; // density
  AnyVar p = 3.1416;
  AnyVar h = ((0.485 * Main.Model3.HumanModel.BP.H) - (0.377 *
    Main.Model3.HumanModel.BP.H))/2;
  AnyVar radius = (Mass/(d*p*h))^0.5; // radius
  AnyVar Ixx = (1/12)*Mass*((3*radius^2)+(h^2));
  AnyVar Iyy = (1/2)*Mass*radius^2;
  AnyVar Izz = Ixx;
  Jii = {Ixx, Iyy, Izz};
  sCoM = {-0.06, 0, 0} * h/0.1;
  AnyRefNode wj={
    sRel={0,0,0*...Sign} * .h/0.1;
    //Arel = Main.Model3.HumanModel.InitialConditions.R_Hand;
    Arel = {{1,0,0},
            0,...Sign*1,0},
            0,0,...Sign}
  };
};
AnyRefNode Node0 = {
  sRel = {0,0,0};
};
AnyRefNode Hand={ sRel= {-0.1, 0, 0*...Sign} * .h/0.1;};
AnyRefNode PalmJoint={ sRel= {-0.08, -0.02, 0} * .h/0.1;};
AnyRefNode Thumb3joint={ sRel= {0, 0, 0.05} * .h/0.1;};
```

Figure 8.39 – RightHand.any File of the Parameterised Model

8.4.2 Joints

The joints that will be used to connect the segments of the upper limb to each other and to the thorax are shown in Figure 8.40 below, while the corresponding “anyfile” is shown in Figure 8.41. It has to be noted that the joints that connect the radius with the ulna and the wrist, are not the proper joints which are used by *Repository2* model. In the *Repository2* model, the radius is connected to the ulna by a combination of joints with one degree of freedom, while a hinge joint connects the radius and the hand. In the current model, because of the failure to place the radius in the desired position and the kinematic indeterminacy of the system, both of the joints that connect the ulna and hand to the radius are spherical with their own drivers.

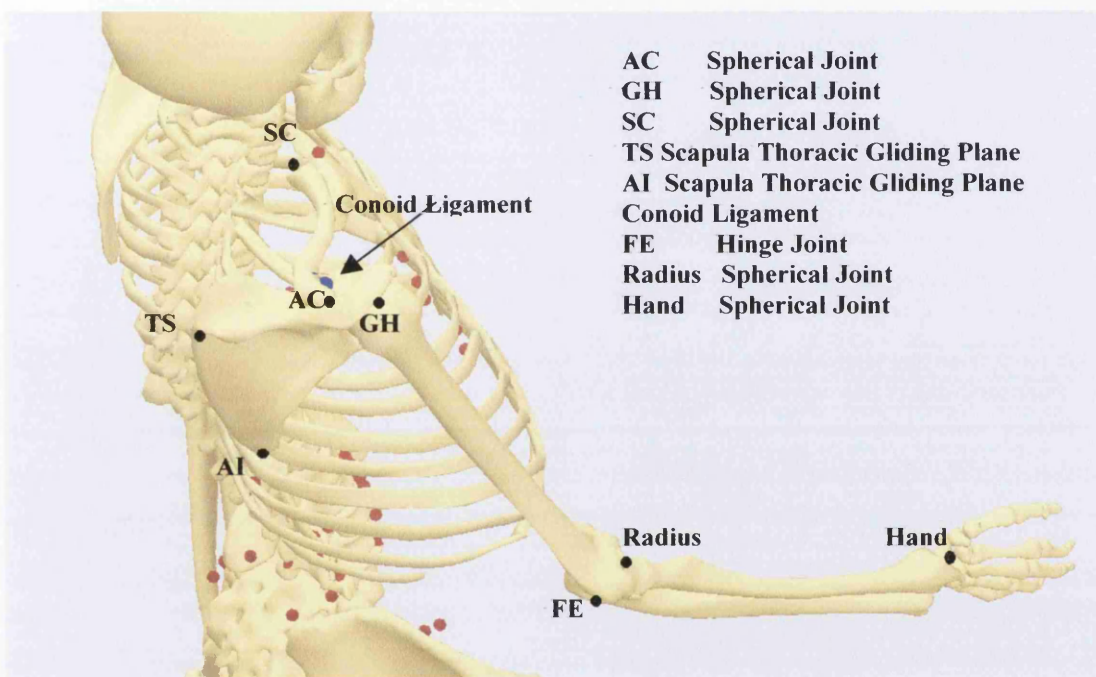


Figure 8.40 – Joints of the Upper Limb

Joints.any File

```
//-----SC JOINT-----  
AnyKinRotational R_SCRot = {  
  AnyRefNode &Thorax = ..\..\Segs.Arms.RightArm.RightShoulder.RightShoulderRef.R_SCJoint;  
  AnyRefNode &R_Clavicula = ..\..\Segs.Arms.RightArm.RightShoulder.RightClavicula.SCJoint;  
  Type = RotAxesAngles;  
  Axis1 = y;  
  Axis2 = z;  
  Axis3 = x;  
};  
AnyKinLinear R_SCLin = {  
  AnyRefNode &Thorax = ..\..\Segs.Arms.RightArm.RightShoulder.RightShoulderRef.R_SCJoint;  
  AnyRefNode &R_Clavicula = ..\..\Segs.Arms.RightArm.RightShoulder.RightClavicula.SCJoint;  
};  
AnyKinEq R_SCLinCon = {  
  AnyKinMeasure &SCLinCon = .R_SCLin;  
};
```

```

Chapter 8
//-----Scapula thorax gliding plane-----
AnyKinPLine R_ScapulaAThoraxNode = {
  AnyRefNode &R_Scapula = ..\..\Segs.Arms.RightArm.RightShoulder.RightScapula.ai;
  AnyRefNode &Thorax= ..\..\Segs.Arms.RightArm.RightShoulder.RightShoulderRef.R_SphereNode;
};
AnyKinPLine R_ScapulaTSThoraxNode = {
  AnyRefNode &Thorax= ..\..\Segs.Arms.RightArm.RightShoulder.RightShoulderRef.R_SphereNode;
  AnyRefNode &R_Scapula = ..\..\Segs.Arms.RightArm.RightShoulder.RightScapula.ts;
};
//-----AC JOINT-----
AnyKinRotational R_ACRot = {
  AnyRefNode &R_Clavicula = ..\..\Segs.Arms.RightArm.RightShoulder.RightClavicula.ACJoint;
  AnyRefNode &R_Scapula = ..\..\Segs.Arms.RightArm.RightShoulder.RightScapula.ACJoint;
  Type = RotAxesAngles;
  Axis1 = y;
  Axis2 = z;
  Axis3 = x;
};
AnyKinLinear R_ACLin = {
  AnyRefNode &R_Clavicula = ..\..\Segs.Arms.RightArm.RightShoulder.RightClavicula.ACJoint;
  AnyRefNode &R_Scapula = ..\..\Segs.Arms.RightArm.RightShoulder.RightScapula.ACJoint;
};
AnyKinEq R_ACLinCon = {
  AnyKinMeasure &ACLinCon = .R_ACLin;
};
AnyKinPLine R_ConoidLigament = {
  AnyRefNode &R_Scapula = ..\..\Segs.Arms.RightArm.RightShoulder.RightScapula.conoid;
  AnyRefNode &R_Clavicula = ..\..\Segs.Arms.RightArm.RightShoulder.RightClavicula.conoid;
};
//-----Definition of GH joint-----
AnyKinRotational R_GHRot = {
  AnyRefNode &R_Scapula_gh = ..\..\Segs.Arms.RightArm.RightShoulder.RightScapula.gh;
  AnyRefNode &R_Humerus_gh = ..\..\Segs.Arms.RightArm.RightUpperLimb.RightHumerus.gh;
  Type = RotAxesAngles;
  Axis1 = y;
  Axis2 = z;
  Axis3 = y;
};
AnyKinLinear R_GHLin = {
  AnyRefNode &R_Scapula_gh = ..\..\Segs.Arms.RightArm.RightShoulder.RightScapula.gh;
  AnyRefNode &R_Humerus_gh = ..\..\Segs.Arms.RightArm.RightUpperLimb.RightHumerus.gh;
};
AnyKinEq R_GHLinCon = {
  AnyKinMeasure &R_GHLinCon = .R_GHLin;
};
//-----Definition of elbow flexion extension joint-----
AnyRevoluteJoint FE = {
  Axis = x;
  AnyRefNode &R_Humerus_fe = ..\..\Segs.Arms.RightArm.RightUpperLimb.RightHumerus.fe;
  AnyRefNode &R_Ulna_fe = ..\..\Segs.Arms.RightArm.RightLowerLimb.RightUlna.fe;
}; //End FE
//-----Radius-Ulna Joint-----
AnySphericalJoint R_PSRotProximal = {
  AnyRefNode &R_RadiusPs = ..\..\Segs.Arms.RightArm.RightLowerLimb.RightRadius.PointPS;
  AnyRefNode &R_UlnaPs = ..\..\Segs.Arms.RightArm.RightLowerLimb.RightUlna.ps;
};
AnyKinRotational R_RadiusRot = {
  AnyRefNode &Node = Main.Model3.HumanModel.Segs.Arms.RightArm.RightLowerLimb.RightUlna.Node0;
  AnyRefNode &Nodes Main.Model3.HumanModel.Segs.Arms.RightArm.RightLowerLimb.RightRadius.Node0;
  Type = RotVector;
};
//-----Hand Joint-----
AnySphericalJoint R_WristJoint = {
  AnyRefNode &R_Radius_wj = ..\..\Segs.Arms.RightArm.RightLowerLimb.RightRadius.wj;
  AnyRefNode &R_Hand_wj = ..\..\Segs.Arms.RightArm.RightHand.RightHand.wj;
};

```

Figure 8.41 – Joints.any File of the Right Upper Limb

8.4.3 Validation

Figure 8.42 below, presents some time steps of the motion of the full parameterized human model. It can be seen that the model proved to be reliable in the implementation of different postures. All the models are included in the same study again, but they are not muscle driven because the study reported an error perhaps due to the high computational demand. On the other hand, the muscle driven simulation worked without problem when only two models were included in the same study (Figures 8.43-8.46). It has to be noted that there are time steps at the simulations with some muscles being over-activated. That means that there is heavy loading exceeding the muscle capacity. To avoid this, the movement should be changed or the muscle capacity should be increased.

It has to be mentioned that at this time it was not possible the lifting of a box with only one driver on the box, as it was presented in the previous chapter. This happened because the model experienced constraint violations at the *conoid ligament*. Another limitation of this model is, that a driver has been used at the joint of radius and ulna, which does not allow the radius to move and have a realistic motion. Details about the “kinematic measures” and the drivers “anyfiles” can be found on the supplementary *CD*.

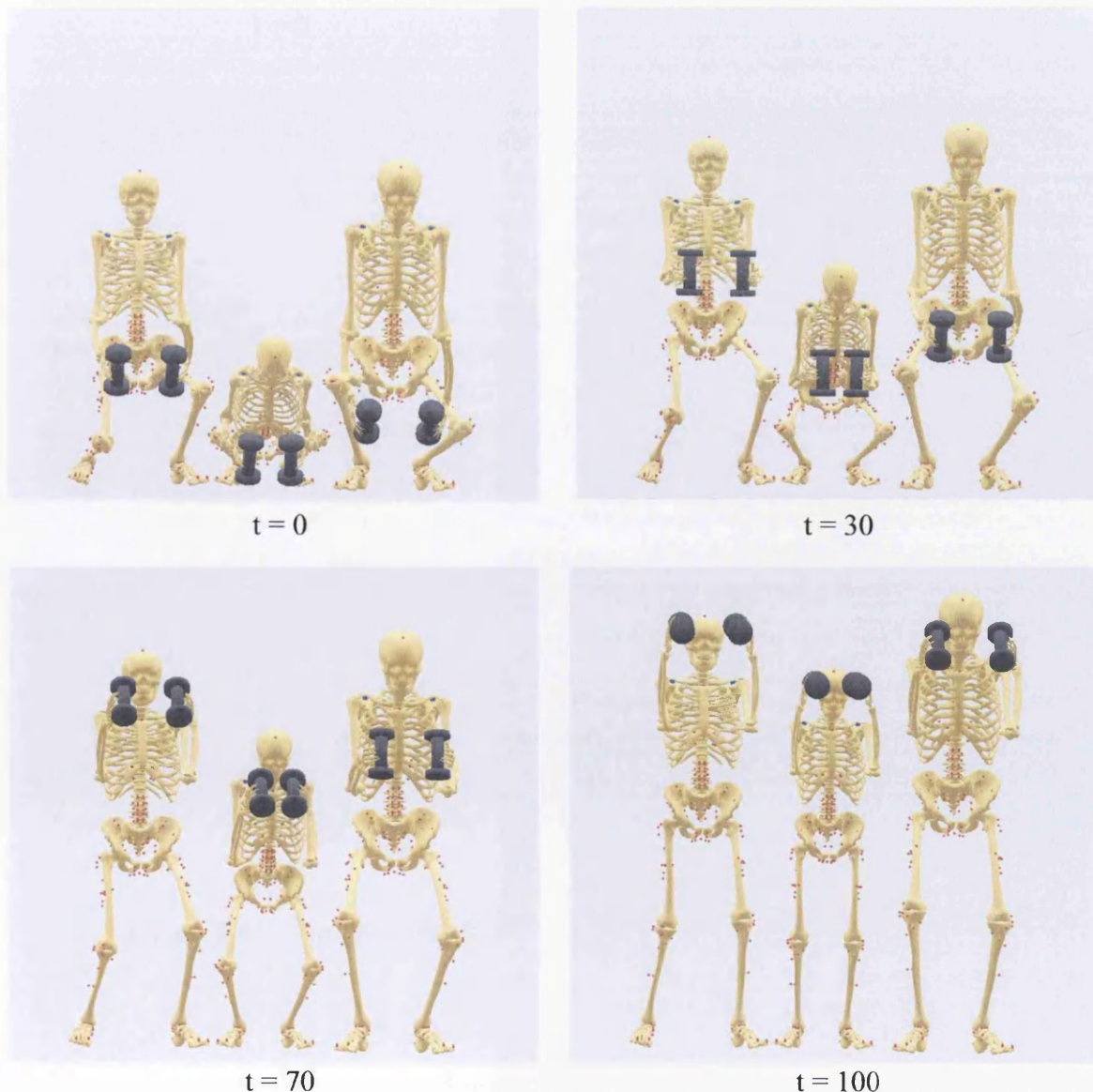


Figure 8.42 – Motion of the Human Parameterized Models



t = 0



t = 30



t = 70



t = 100

Figure 8.43 – Anterior View of the Motion of the 1.7m & 1.5m Human Models



t = 0



t = 30



t = 70



t = 100

Figure 8.44 – Posterior View of the Motion of the 1.7m & 1.5m Human Models



t = 0



t = 30



t = 70



t = 100

Figure 8.45 – Anterior View of the Motion of the 1.9m & 1.5m Human Models



t = 0



t = 30



t = 70



t = 100

Figure 8.46 – Posterior View of the Motion of the 1.9m & 1.5m Human Models

8.5 Application

In this section a simple investigation of an application of a human musculoskeletal model will be discussed. For this purpose, a musculoskeletal model of $1.7m$ height and $65kg$ weight has been chosen, with the legs twisted about y -axis 20 degrees and the pelvis being $0.6m$ above the ground and moving with a velocity of $0.35m/sec$. It has to be mentioned that the upper limbs do not move and the center of mass driver is not activated.

There are two tasks that have been assigned to the specific model to perform. The first task (Case 1) is the upward movement with the two weights close to the trunk (Figure 8.47) while the second task (Case 2) is the same movement with the weights further away from the trunk (Figure 8.48).

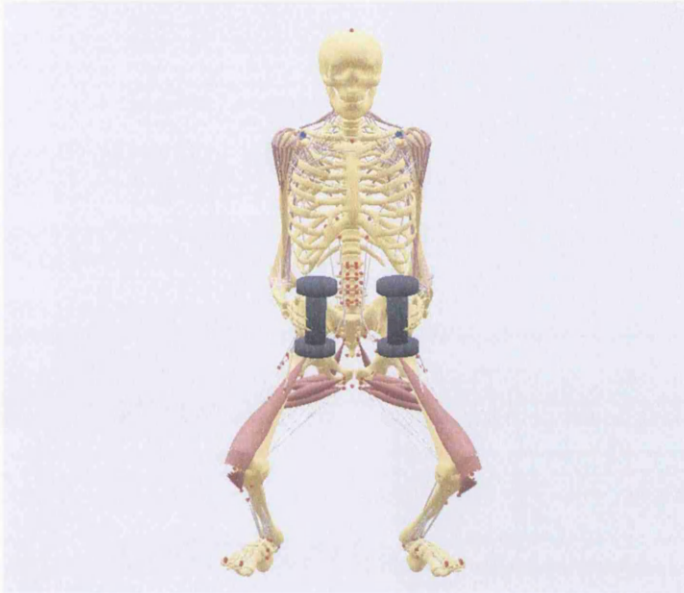


Figure 8.47 – Anterior View of Case 1 Application of the Human Musculoskeletal Model



Figure 8.48 – Anterior View of Case 2 Application of the Human Musculoskeletal Model

The investigation will focus, in both cases, on the load of the *sacrum/lumbar L5* joint and the muscle force of the right *rectus femoris* muscle.

Figures 8.49 and 8.50 below show some time steps of the motion of Case 1 and Case 2 respectively, while Figure 8.51 shows the corresponding reaction forces on *the sacrum/lumbar L5* joint. Furthermore, Figure 8.52 indicates the corresponding muscle force of the right *rectus femoris* muscle.

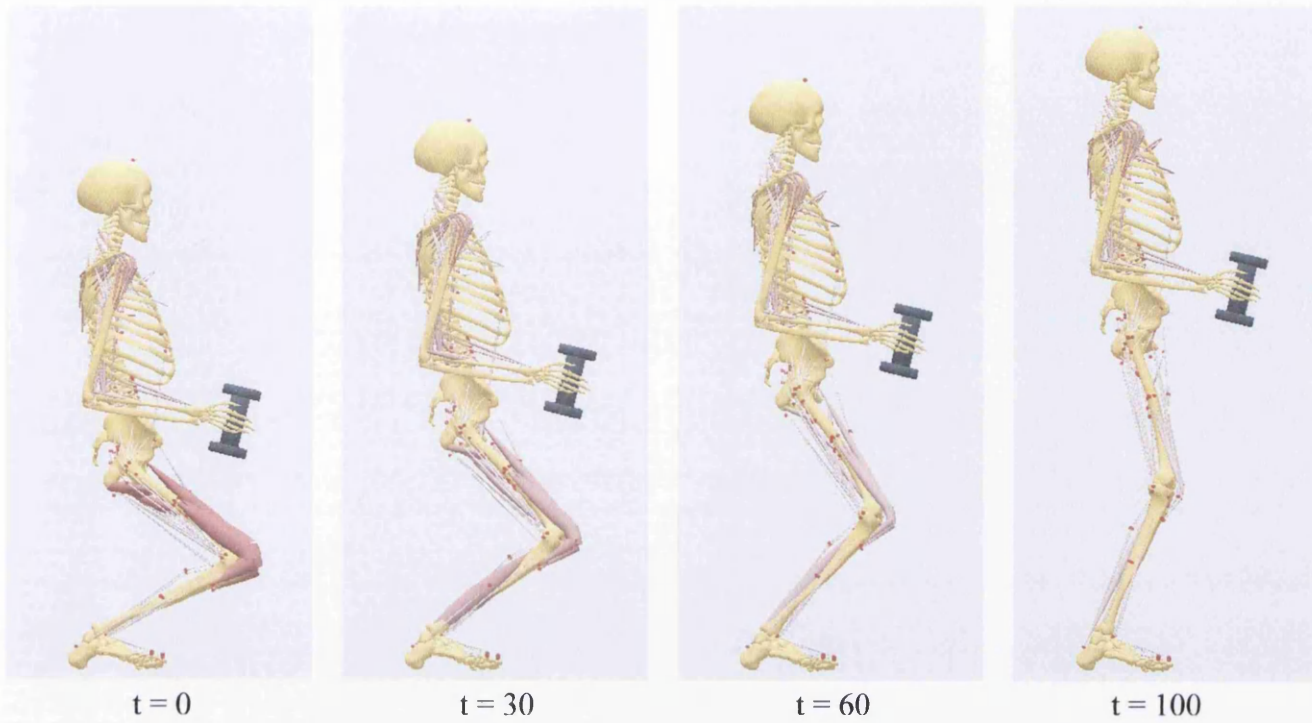


Figure 8.49 – Right Lateral View of Case 1

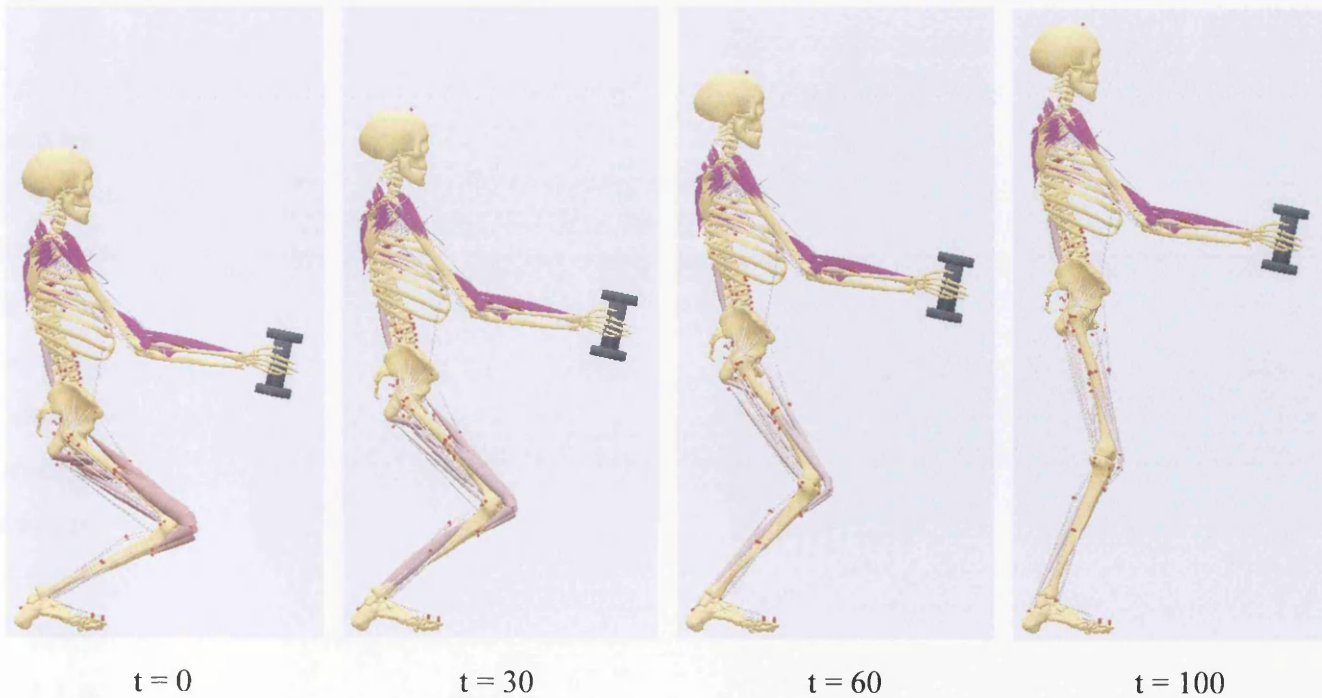


Figure 8.50 – Right Lateral View of Case 2

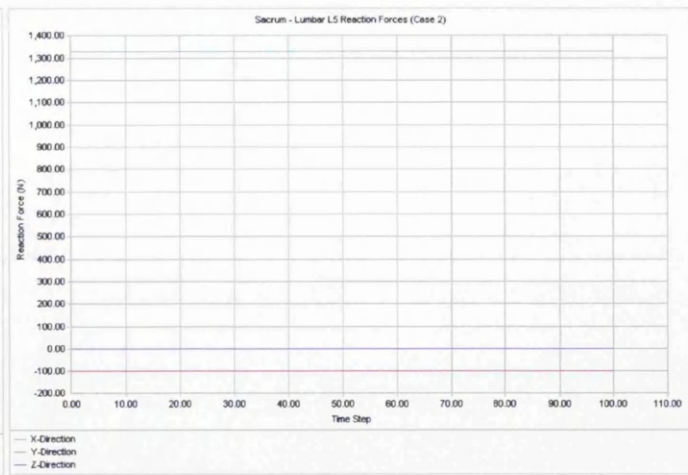
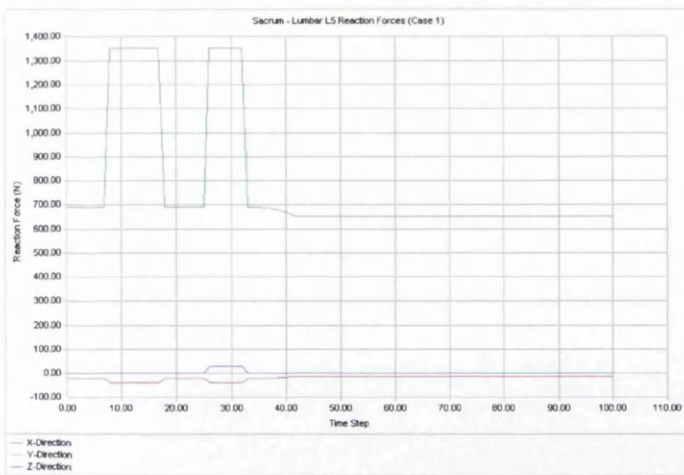


Figure 8.51 – Cases 1 & 2 Reaction Forces of Sacrum/Lumbar L5 Joint

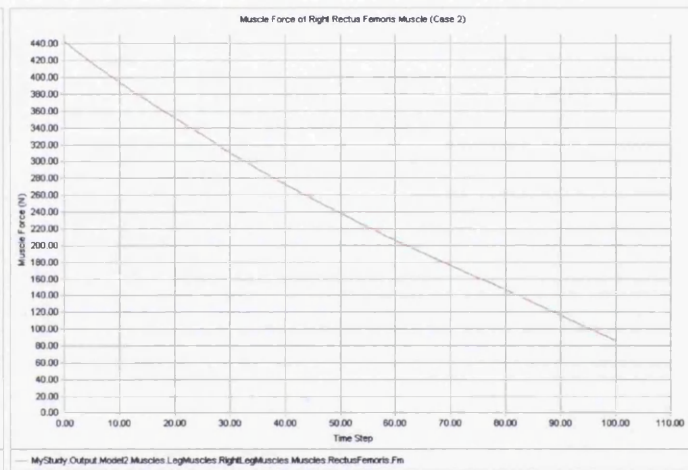
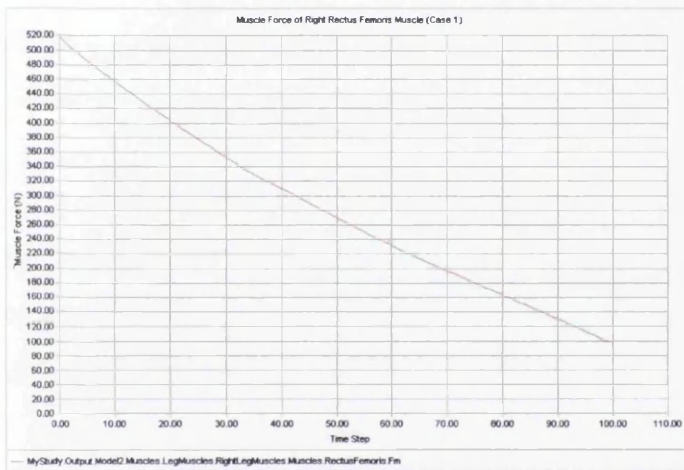


Figure 8.52 – Cases 1 & 2 Muscle Force of Right Rectus Femoris Muscle

It can be seen from Figure 8.50 that during the upward movement of Case 2 the muscles of the upper limb are over-activated. That means that there is heavy loading exceeding the muscle capacity. To avoid this, the movement should be changed or the muscle capacity should be increased.

Figure 8.51 indicates that the joint load at *y-direction* is reduced when the weights are close to the trunk and increased when they are further away from the trunk. On the other hand, Figure 8.52 shows that the muscle force of the *rectus femoris* muscle is increased when the weights are close to the trunk and reduced for Case 2. So, what is the best way to lift the weights taking into account the particular joint load and muscle force?

As it is indicated in Figure 8.51, the reaction force of Case 2 in *y-direction* is too high in comparison with the corresponding reaction force of Case 1. Besides this, Figure 8.52 shows that the muscle forces of Case 1 and Case 2 are close to each other. As a result the recommended lifting technique is the one with the weights close to the trunk.

8.6 Conclusions

The present chapter demonstrated the way of implementing a parameterized human musculoskeletal model representing the various segments of the body as simple geometrical models. The parameterization is based on the two global parameters, the *height* and the *mass*, while the design “philosophy” that was followed in developing this complex model was the result of the modelling experience of the prior simple models.

The geometrical models that were chosen and constitute the mechanical model, which are the rectangular prism, the circular cylinder, the elliptical cylinder and the sphere, represent a good approximation of the various parts of a real body without long and complex calculations.

The types of drivers that were used, and especially their position in the model, permitted a wide range of motions that make the model reliable for investigating the human movement under various postures.

Problems occurred with muscle wrapping, but did not cause any problem to the motion. Because of scaling, sometimes the muscles that wrap over bones were located below that wrapping surface. Further problems occurred when using the *ulna-radius* joint, which was constrained by a driver because *ulna* and *radius* could not be positioned at a desired orientation using the rotating matrices. The attempt to lift a box with both hands and with only one driver allocated on the box did not work at all. In this case this lifting technique failed because of a problem occurred at the *conoid ligament* due to constraints violation. In general, beside the malfunction of the *conoid ligament* and the *ulna-radius* joint, the performance of the parameterized model was satisfactory during the validation test.

CHAPTER 9 CONCLUDING REMARKS

9.1 Summary

In this thesis, the final target was the development of a parameterized musculoskeletal model of the human body dependent on the two global parameters, *mass* and *height*, using a simulation software named Anybody Technology. The necessary steps in achieving this target were expanded in a logical manner.

Research started with an analysis on the musculoskeletal system describing the skeletal, the articular and the muscular system. It continued with a review of anthropometry, a science that provides the proper background for the model's parameterization, a discussion on the reasons that prompt scientists to simulate human movement and the research method that is followed in achieving this simulation. The investigation went on to an outline of the most popular musculoskeletal software packages and some of their application areas and continued with an analysis of the basic features of the simulation system, Anybody Technology, which is used in the present research. The next step was an introduction to the Anybody modelling system developing a triple inverted pendulum in three phases, and the development of an advanced "human like" model which intended to be the basis for the development of the final parameterized model. In this stage basic features that constitute a multibody mechanism became familiar. These features are the *global reference frame*, the *segments*, the *joints*, the *drivers*, the *muscles*, the *real geometries* and some advanced like the *kinematic measures* and the *initial conditions*. Finally, the knowledge gained from the analysis of the musculoskeletal system, the science of Anthropometry and the two models developed using Anybody Technology, made the development of a parameterized musculoskeletal human body real. This model, modifying the two global parameters *mass* and *height*, may represent a wide range of the human population, and provide valuable information for the investigation of the human movement, like the force and the activation of the muscles and the reaction forces at the joints.

9.2 Conclusions

Apart from the work, this project has demonstrated that it is possible for a novel user of Anybody Technology to develop a complex model such the parameterized musculoskeletal model of the human body presented in the previous chapter. This way possible because prior to the complicated model, two simple models had been developed that constituted the basic platform for more challenging models. Simple models tend to give better insight into the underlying mechanisms of complex systems, provide

confidence for a next more advanced step, and are time and cost effective. The latter is maybe the most important thing of this research.

Of course, although the final model is not perfect, it has demonstrated a few strong points. One of these points is the simple structure that allows the user to access files and folders quickly. In the same way, the strategy of using only one file, the *Main.any* file, to control the whole model, proved to be very useful for making easy and quick changes to the model, and understand how it works. Another strong point is that the model has been constructed in such a way, that a wide range of motions, which can be extreme as well, can be implemented.

Weak points of the parameterized model are some of the joints of the upper extremity, which are not of the proper type. These are the *ulna-radius* joint and the *radius-hand* joint, which caused problems to the model at their original form. Another weak point is the *conoid ligament* which experienced constraints violation during the attempt of lifting a box with only one driver, a box driver. Finally, the parameterized model reported, sometimes more and sometimes less, warnings during the simulation. These warnings have to do with some of the muscles that wrap over the bones, which sometimes, because of the scaling, were located below the wrapping surface.

9.3 Further Work

The research that is undertaken during this project could be seen as a “pilot” for future advanced musculoskeletal models of humans or animals too. However, it seems that some further work has to be done in order to improve the current parameterized model. The way that the joints and drivers of the complex upper extremity interact with each other, needs further investigation. The correct wrapping of the muscles over the bones during scaling, has also to be investigated. And finally, as soon as these problems are solved, the model could move to the more complicated *Repository* that Anybody Technology provides, the *Repository version 4*.

APPENDIX A SIMPLEX ALGORITHM METHOD⁵⁸

In the study that follows, the *simplex method* will be analysed in simple steps with the use of an example. The process is as follows:

1. The Initial Problem

The initial problem is described as follows:

$$\square \text{ Maximise } Z = 10x_1 + 8x_2$$

Subject to the constraints

$$\square 279x_1 + 119x_2 \leq 700$$

$$\square 0.18x_1 + 2.8x_2 \leq 3.5$$

$$\square x_1 \geq 0, \quad x_2 \geq 0$$

2. Slack Variables

Introduce slack variables, which represent unused capacity in the constraints:

$$\square 279x_1 + 119x_2 + s = 700$$

$$\square 0.18x_1 + 2.8x_2 + t = 3.5$$

$$\square x_1 \geq 0, \quad x_2 \geq 0, \quad s \geq 0, \quad t \geq 0$$

Rewrite the objective to get the (ideal) objective function (i.e. all excess capacity used):

$$10x_1 + 8x_2 + 0s + 0t = Z$$

$$Z - 10x_1 - 8x_2 - 0s - 0t = 0$$

3. Simplex Tableaux

This application of the simplex algorithm uses tables, called tableaux, to represent calculations and intermediate steps to completing a problem. The first step is to form the initial tableau. There are four variables: x_1, x_2, s, t , and two equations. Therefore, two variables have to be set to zero in order these equations to be solved.

The variables set to zero are called *non-basic variables*, while the variables not set to zero are called *basic variables*.

For the initial tableau, the basic variables are always the slack variables. The initial tableau is set up as follows:

+Tableau I					
Basic variable	x_1	x_2	s	t	Value
s	279	119	1	0	700
t	0.18	2.8	0	1	3.5
Z	-10	-8	0	0	0

- The first row represents the first constraint.
- The second row represents the second constraint.
- The third row represents the objective function.

It has to be noticed that the basic variables appear only in one row, and only with a coefficient of one, and in their column, all other entries read zero.

This tableau represents the solution $x_1 = 0$, $x_2 = 0$, $s = 700$, $t = 3.5$, $Z = 0$. These values are found when the first and the last columns are read off:

- $s = 700$
- $t = 3.5$
- $Z = 0$

It is known that x_1 and x_2 are zero because they are non-basic variables.

4. The Optimality Condition

A tableau represents an optimal solution if:

- The objective row contains zero entries in the columns of basic variables.
- The objective row contains no negative entries in the columns of non-basic variables.

Tableau I does not satisfy this condition, so the solution is not optimal. Therefore, this tableau has to be modified and give a second one corresponding to another vertex in the linear programming space.

5. Choosing a Pivot

What is needed now, is the finding of a non-basic variable to become a basic variable and also the basic variable to be replaced with this non-basic variable. For finding the new basic variable, the column with the most negative entry must be selected in the objective row. In tableau I, this is the " x_1 " column with -10 . This is called the *pivotal column*.

In order to find the basic variable being replaced, θ values for each row must be calculated (except the objective row). θ is given by the entry in the "value" column divided by the entry in the pivotal column.

For the first row: $\theta = \frac{700}{279} = 2.51$

For the second: $\theta = \frac{3.5}{0.18} = 19.44$

The *pivotal row* is then taken to be the smallest of these θ values. The *pivot* is found at the intersection of the pivotal row and the pivotal column. In this case, the pivot is the number 279:

+Tableau I with pivot					
Basic variable	x_1	x_2	s	t	Value
s	279	119	1	0	700
t	0.18	2.8	0	1	3.5
Z	-10	-8	0	0	0

6. Creating a New Tableau

The first step to tableau II is the division of the pivotal row by the pivot so that the pivot will be 1:

Basic variable	x_1	x_2	s	t	Value
s	$\frac{279}{279} = 1$	$\frac{119}{279} = 0.43$	$\frac{1}{279} = 0.0036$	0	$\frac{700}{279} = 2.51$
t	0.18	2.8	0	1	3.5
Z	-10	-8	0	0	0

The next step is the addition or subtraction of multiples of this new row from the other rows in the tables, so that they become zero, as they should for a basic variable.

For the second row (the t row), subtract $0.18 \times$ row1 (the s row):

Basic variable	x_1	x_2	s	t	Value
s	1	0.43	0.0036	0	2.51
t	0	2.72	-0.00065	1	3.05
Z	-10	-8	0	0	0

For the third row (the Z row) add $10 \times$ row1:

Basic variable	x_1	x_2	s	t	Value
s	1	0.43	0.0036	0	2.51
t	0	2.72	-0.00065	1	3.05
Z	0	-3.7	0.036	0	25.1

The final step is the replacement of the old basic variable with the new:

+Tableau II					
Basic variable	x_1	x_2	s	t	Value
x_1	1	0.43	0.0036	0	2.51
t	0	2.72	-0.00065	1	3.05
Z	0	-3.7	0.036	0	25.1

This tableau represents the solution where $t = 3.05$, $x_1 = 2.51$, $x_2 = 0$, $s = 0$, $Z = 25.1$. However, there is still a negative value in the objective row, so the solution is not optimal either. Therefore a third tableau is needed.

7. Repeat the Pivoting Process

The pivotal column is the one with -3.7 in the objective row, which is " x_2 ".

The pivotal row is the one with the smallest θ value:

$$\text{For the first row: } \theta = \frac{2.51}{0.43} = 5.84$$

$$\text{For the second row: } \theta = \frac{3.05}{2.72} = 1.12$$

Therefore, the pivot is the intersection of the t row and the x_2 column, which is 2.72.

The next step is the division of the pivotal row by the pivot:

Basic variable	x_1	x_2	s	t	Value
x_1	1	0.43	0.0036	0	2.51
t	0	1	-0.00024	0.37	1.12
Z	0	-3.7	0.036	0	25.1

The next step is the addition or subtraction of this row with the remaining rows:

Row1: subtract $0.43 \times$ row2

Row3: add $3.7 \times$ row2

The next step is the replacement of the basic variable:

+Tableau III					
Basic variable	x_1	x_2	s	t	Value
x_1	1	0	0.0037	-0.16	2.03
x_2	0	1	-0.00024	0.37	1.12
Z	0	0	0.035	1.37	29.24

This tableau represents the solution where $x_1 = 2.03$, $x_2 = 1.12$, $s = 0$, $t = 0$, $Z = 29.24$. It is an optimal solution, as there exist no negative basic variables and all non-basic variables are zero.

REFERENCES

- [1] McArdle, W.D., F.I. Katch, and V.L. Katch. Exercise Physiology: Energy, Nutrition, and Human Performance. Philadelphia: Lea & Febiger 1996
- [2] Forensic Anthropology
<http://www-personal.une.edu.au/~pbrown3/skeleton.pdf>
- [3] Musculoskeletal System
<http://www.sirinet.net/~jgjjohnso/>
- [4] Watkins, James. Structure and Function of the Musculoskeletal System. Human Kinetics, 1999.
Zatsiorsky, V.M. Kinetics of Human Motion. Human Kinetics, Champaign, IL, 2002.
- [5] Martini, F., Timmons, M., and McKinley, M. Human Anatomy. 3rd Edition. Prentice Hall Publishers, 2000.
- [6] Muscle Groups
http://training.seer.cancer.gov/module_anatomy/unit4_4_muscle_grp.html
- [7] Loyola University Medical Education Network – Master Muscle list.
<http://www.meddean.luc.edu/lumen/MedEd/GrossAnatomy/dissector/mml/abdomen.htm>
- [8] Bach, T.M. Optimising Mass and Mass Distribution in Lower Limb Prostheses. Prosthetics & Orthotics Australia 1995; 10:29-34.
- [9] Tsai, C.S. and Mansour, J.M. Swing Phase Simulation and Design of Above Knee Orthoses. Journal of Biomechanical Engineering 1986; 108:65-72.
- [10] Robertson, D.G.E., Caldwell, G.E., Hamill, J., Kamen, G., and Whittlesey, S.N. Research Methods in Biomechanics. Human Kinetics, 2004.
- [11] Harles, E. The Static Moments of the Component Masses of the Human Body. *Treatises of the Mathematics-Physics Class, Royal Bavarian Academy of Sciences* 1860; 8:69-96, 257-94. Trans. FTD Technical Report 61-295. Wright-Patterson Air Force Base, OH, 1962.
- [12] Braune, W., and Fischer, O. Uber den Scherpunkt des menschlichen Körpers, mit Rücksicht auf die Ausrüstung des deutschen Infanteristen [The center of gravity of the human body as related to the equipment of the German Infantry]. *Abhandlungen der mathematisch-physischen Klasse der Königlich-Sächsischen Gesellschaft der Wissenschaften* 1889;26:561-672.
- [13] Fischer, O. *Theoretische Grundlagen für eine Mechanik der Lebenden Körper mit Speziellen Anwendungen auf den Menschen, sowie auf einige Bewegungs-voränge an Maschine* [Theoretical fundamentals for a mechanics of living bodies, with special applications to man, as well as to some processes of motion in machines]. Leipzig: B.G. Tuebner, 1906.
- [14] Dempster, W.T. Space Requirements of the Seated Operator: Geometrical, and Mechanical Aspects of the Body with Special Reference to the Limbs. WADC Technical Report 55-159. Wright-Patterson Air Force Base, OH, 1955.

- [15] Reuleaux, F. *The Kinematics of Machinery: Outlines of a Theory of Machines*. London 1876; Macmillan
- [16] Miller, D.I., and Nelson, R.C. *Biomechanics of Sport*. Lea & Febiger, Philadelphia, 1973.
- [17] Plagenhoef, S. *Patterns of Human Motion: A Cinematographic Analysis*. Englewood Cliffs, Prentice Hall, NJ, 1971.
- [18] Winter, D.A. *Biomechanics and Motor Control of Human Movement*. 3rd Edition. John Wiley & Sons, Inc., Hoboken, New Jersey, 2005.
- [19] Barter, J.T. *Estimation of the Mass of the Body Segments*. WADC Technical Report 57-260. Wright-Patterson Air Force Base, OH, 1957.
- [20] Clauser, C.E., McConville, J.T., and Young, J.W. *Weight, Volume and Center of Mass of Segments of the Human Body*. AMRL Technical Report 60-70. Wright-Patterson Air Force Base, OH, 1969.
- [21] Chandler, R.F., Clauser, C.E., McConville, J.T., Reynolds, H.M., and Young, J.W. *Investigation of Inertial Properties of the Human Body*. AMRL Technical Report 74-137. Wright-Patterson Air Force Base, OH, 1975.
- [22] Vaughan, C.L., Davis, B.L., and O'Connor, J.C. *Dynamics of Human Gait*. Human Kinetics, Champaign, IL, 1992.
- [23] Hinrichs, R.N. *Regression Equations to Predict Segmental Moments of Inertia from Anthropometric Measurements: An Extension of the Data of Chandler et al (1975)*. *Journal of Biomechanics* 1985;18:621-4
- [24] Drillis, R., and Contini, R. *Body Segment Parameters*. Rep. 1163-03. Office of Vocational Rehabilitation, Department of Health, Education, and Welfare, New York, 1966.
- [25] Hanavan, E.P. *A Mathematical Model of the Human Body*. AMRL Technical Report 64-102. Wright-Patterson Air Force Base, OH, 1964.
- [26] Hatze, H. *A Mathematical Model for the Computational Determination of Parameter Values of Anthropometric Segments*. *Journal of Biomechanics* 1980; 13:833-43.
- [27] Yeadon, M.R. *The Simulation of Aerial Movement: II. A Mathematical Inertia Model of the Human Body*. *Journal of Biomechanics* 1990; 23:67-74.
- [28] Zatsiorsky, V.M., and Seluyanov, V.N. *The Mass and Inertia Characteristics of the Main Segments of the Human Body*. *Biomechanics VIII-B*, ed. H. Matsui and K. Kobayashi, 1152-9. Human Kinetics, Champaign, IL, 1983.
- [29] Zatsiorsky, V.M., and Seluyanov, V.N. *Estimation of the Mass and Inertia Characteristics of the Human Body by Means of the Best Predictive Regression Equations*. *Biomechanics IX-B*, ed. D.A. Winter et al., 233-9. Human Kinetics, Champaign, IL, 1985.
- [30] Zatsiorsky, V.M. *Kinetics of Human Motion*. Human Kinetics, Champaign, IL, 2002.
- [31] Jensen, R.K. *Model for Body Segment Parameters*. In *Biomechanics V-B*, ed. P.V. Komi, 380-6. University Park Press, Baltimore, 1976.

- [32] Jensen, R.K. Estimation of the Biomechanical Properties of Three Body Types Using a Photogrammetric Method. *Journal of Biomechanics* 1978; 11:349-58.
- [33] Mungiole, M., and Martin, P.E. Estimating Segmental Inertia Properties: Comparison of Magnetic Resonance Imaging with Existing Methods. *Journal of Biomechanics* 1990; 23:1039-46.
- [34] Durkin, J.L., and Dowling, J. Analysis of Body Segment Parameter Differences Between Four Human Populations and the Estimation of Errors of Four Popular Mathematical Models. *Journal of Biomechanical Engineering* 2003; 125:515-22.
- [35] Durkin, J.L., Dowling, J., and Andrews, D.M. The Measurement of Body Segment Inertial Parameters Using Dual Energy X-Ray Absorptiometry. *Journal of Biomechanics* 2002; 35:1575-80.
- [36] Jensen, R.K. Body Segment Mass, Radius and radius of Gyration Proportions of Children. *Journal of Biomechanics* 1986; 19:359-68.
- [37] Jensen, R.K. Changes in Segment Inertial Proportions Between 4 and 20 years. *Journal of biomechanics* 1989; 22:529-36.
- [38] Jensen, R.K., Treitz, T., and Doucet, S. Prediction of Human Segment Inertias During Pregnancy. *Journal of Applied Biomechanics* 1996; 12:15-30.
- [39] Schneider, K., and Zernicke, R.F. Mass, Center of Mass and Moment of Inertia Estimates for Infant Limb Segments. *Journal of Biomechanics* 1992; 25:145-8.
- [40] Pavol, M.J., Owings, T.M., and Grabiner, M.D. Body Segment Parameter Estimation for the General Population of Older Adults. *Journal of Biomechanics* 2002; 35:707-12.
- [41] Hatze, H. A New Method for the Simultaneous Measurement of the Moment of Inertia, the Damping Coefficient and the Location of the Center of Mass of a Body Segment In Situ. *European Journal of Applied Physiology* 1975; 34:217-26.
- [42] Drillis, R., Contini, R., and Bluestein, M. Body Segment Parameters: A Survey of Measurement Techniques. *Artificial Limbs* 1964; 8:44-66.
- [43] Anybody Technology, (2005), The Anybody Modelling System. Version 1.3.0, Anybody Technology ApS, Denmark.
<http://www.anybodytech.com/26.0.html>
- [44] Musculographics, Inc, SIMM Software Suite. Version 4.1.1, c/o Motion Analysis Corporation, Ca, USA, 2005.
<http://www.musculographics.com/products/products.html>
- [45] Biomechanics Research Group, Inc, LifeMOD Biomechanics Modeller. Version 2005.1.0, Biomechanics Research Group, Inc., Ca, USA, 2005.
<http://www.lifemodeler.com/>
- [46] MSC Software Corporation, ADAMS. Version 2005 r2, MSC Software Corporation, Ca, USA, 2005.
<http://www.mssoftware.com/products/adams.cfm>

[47] The Mathworks, Inc, SimMechanics. Version 2.3, The Mathworks, Inc, Ma, USA, 2005.

<http://www.mathworks.com/products/simmechanics/>

[48] TNO Automotive, MADYMO. Version 6.2.1, TNO MADYMO BV, Delft, The Netherlands, 2005.

<http://www.automotive.tno.nl/smartsite.dws?id=1530>

[49] Anybody Technology, (2005), The Anybody Modelling System: Examples. Version 1.3.0, Anybody Technology ApS, Denmark.

<http://www.anybodytech.com/27.0.html>

[50] Biomechanics Research Group, Inc, LifeMOD Biomechanics Modeller: Functions. Version 2005.1.0, Biomechanics Research Group, Inc., Ca, USA, 2005.

<http://www.lifemodeler.com/>

[51] Anybody Technology, (2005), The Anybody Modelling System: Tutorials. Version 1.3.0, Anybody Technology ApS, Denmark.

<http://www.anybodytech.com/44.0.html>

[52] Anybody Technology, (2005), The Anybody Modelling System: Reference Manual. Version 1.3.0, Anybody Technology ApS, Denmark.

<http://www.anybodytech.com/fileadmin/doc/AnyBodyRefManual.pdf>

[53] Forster, E. (2004): Predicting muscle forces in the human lower limb during locomotion. The Medical Faculty, University of Ulm, Germany.

http://vts.uni-ulm.de/query/longview.meta.asp?document_id=3722

[54] Rasmussen, J., Damsgaard, M., and Voigt, M. Muscle recruitment by the min/max criterion – a comparative numerical study. *Journal of Biomechanics* 2001; 34(3): 409-415.

[55] Damsgaard, M., Christensen, S.T., and Rasmussen, J. An efficient numerical algorithm for solving muscle recruitment problem in inverse dynamics simulations. *International Society of Biomechanics, XVIIth Congress, July 8-13, 2001. Zurich, Switzerland.*

[56] The BEL Repository

http://www.tecno.ior.it/VRLAB/researchers/repository/BEL_repository.html

[57] The Multimod Data Manager Home Page

http://www.tecno.ior.it/multimod/DataManager/dm_home.html

[58] The Simplex Algorithm Method

http://en.wikipedia.org/wiki/Simplex_algorithm_method

BIBLIOGRAPHY

- Winter, D.A. Biomechanics and Motor Control of Human Movement. 3rd Edition. John Wiley & Sons, Inc., Hoboken, New Jersey, 2005.
- Robertson, D.G.E., Caldwell, G.E., Hamill, J., Kamen, G., and Whittlesey, S.N. Research Methods in Biomechanics. Human Kinetics, 2004.
- Martini, F., Timmons, M., and McKinley, M. Human Anatomy. 3rd Edition. Prentice Hall Publishers, 2000.
- Watkins, James. Structure and Function of the Musculoskeletal System. Human Kinetics, 1999.
- Zatsiorsky, V.M. Kinetics of Human Motion. Human Kinetics, Champaign, IL, 2002.
- Chaffin, D.B. and Andersson, G.B.J. Occupational Biomechanics. Second Edition. John Wiley & Sons, 1991.
- Frey, D. Autocad 2000: Visual JumpStart. Sybex, Ca, 2000.
- McGill, D.J. Engineering Mechanics: Dynamics. Brooks/Cole, California, 1984.
- Greenwood, D.T. Principles of dynamics. Prentice-Hall, New Jersey, 1988.