Department of Health Science and Technology
Fredrik Bajers Vej 7
9220 Aalborg Øst
Denmark

Title:
Implementation of Model for Estimation of Joint Torques for Designing the Control for Hip Orthosis

Project Period:
4th Semester - Master, Spring 2011

Project Group:
1086g, Medical Informatics

Participants:
Kasper Esben Kannik

Supervisor at Aalborg University:
Dejan B. Popovic

Copies: 3

Attachments: CD

Finished: June 1st 2011

Preface

This report is a master thesis written in the spring 2011 by group 11gr1086g in Medical Informatics at the Department of Health Science and Technology, Aalborg University. The project period lasted from the Febuary 1st to June 1st 2011.
The target group of the project is students and supervisors at Aalborg University and others who might be interested.
The project is based on a problem oriented method and consists of this project report which is divided into two parts:

– Problem Analysis.
– Problem Solving.

In the end of the report appendices are represented. The appendices are composed by the group and contain further explanations about selected topics.

The references in the report are indicated both before and after periods. If the reference is before a period it refers only to the sentence. If the reference is after a period, it refers to the section or until the latest reference.
Figures without source reference is made by the group.

This report is prepared by:

Kasper Esben Kannik

Contents

Introduction

Individuals with gait disabilities often need assitance with postural control and body weight while standing on the paretic leg and with the swing of the paretic leg [1]. A new approach to restoring gait of the physically disabled uses a motor together with a spring as the control mechanism for a hip orthosis controlling the leg, where the motor produces joint flexion and loads the spring with energy that it uses to produce joint extension. The approach integrates the hip orthosis with a rehabilitation device called Walkaround [1], which provides postural and body weight assistance.

The hip orthosis is a module of a self-fitting modular orthosis, that has been developed by professor Dejan B. Popović [2]. In order to design the control for the hip orthosis, a model must be developed to analyze the effects of the spring during walking and determine the necessary characteristics of the motor drive. A biomechanical model[3] used for designing the control for walking by means of functional electrical stimulation is adopted to construct the model to be used as a part of a tool for the analysis. The purpose of the model is to estimate the joint torques that are necessary to produce a given gait pattern for a given individual and is implemented in a computer program.

In order to understand the problem it is necessary to understand the elements of it. The elements of the problem are the Walkaround, the hip orthosis and the biomechanical model. Each element is adressed in the following problem analysis leading up to a description of the computer program development process.

Chapter 1

Problem Analysis

1.1 The Walkaround

The Walkaround is a walker with wheels that controls the orientation and position of the trunk with respect to the vertical. It allows walking without the use of hands for support and imposes no constraints on leg movements. The control is provided through a lumbar belt and an adjustable suspension system with springs that connects the belt with the walker. The Walkaround has telescopic bars allowing it to fit the height of the individual user. A sketch of the Walkaround is shown in figure 1.1.[1]

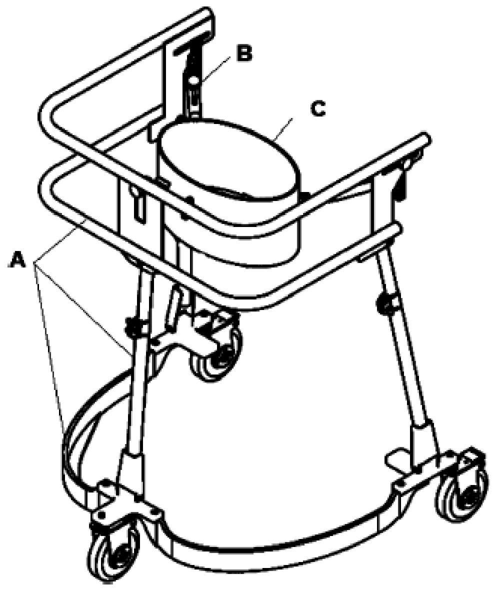

Figure 1.1: Sketch of the Walkaround. A: Telescopic bars that allow fitting to individual height. B: Adjustable suspension system for providing orientation and stifness control to pelvis and trunk. C: Lumbar belt.[1]

1.2 The Self-Fitting Modular Orthosis

The self-fitting modular orthosis (SFMO) is a light metal brace designed to control the hip, knee and ankle joints. The brace comprises three joint modules that can be used independently to control one joint or together to control multiple joints. The SFMO modules have telescopic features, which allows it to fit itself to the human user by self-centering of mechanical to human joints.

The control must be designed to assist a walking disabled individual in obtain a normal gait pattern. The SFMO is an active orthosis, which can be instrumented with an externally powered unit called a cybernetic actuator that provides externally controlled flexion, extension and stiffness. The cybernetic actuator is a motor driving a ballscrew mechanism which turns rotational movement of the motor into linear movement of the slider. The SFMO is shown in figures 1.2 and 1.3.

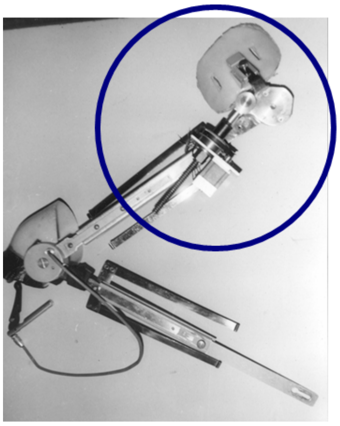

Figure 1.2: Self-fitting modular orthosis developed by Dejan B. Popović. The encircled part is the hip module. [4]

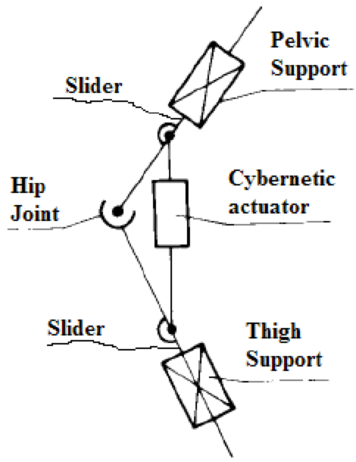

Figure 1.3: Kinematic scheme of a single joint module. The supports of the hip module are attached to the pelvis and the thigh. A support connects the brace with a slider, which is controlled by the cybernetic actuator. Modified from [2]

1.3 The Joined System

The hip module of the SFMO is integrated with the Walkaround by attaching it to the lumbar belt as shown in figure 1.4.

Figure 1.4: The Walkaround with the hip module of the SFMO integrated. The blue ellipse encapsulates the hip orthosis.[4]

The control mechanism for the hip orthosis is illustrated in figure 1.5. It consists of a motor that produces flexion of the hip joint, while it loads an opposing spring with energy, that is used to extend the joint.

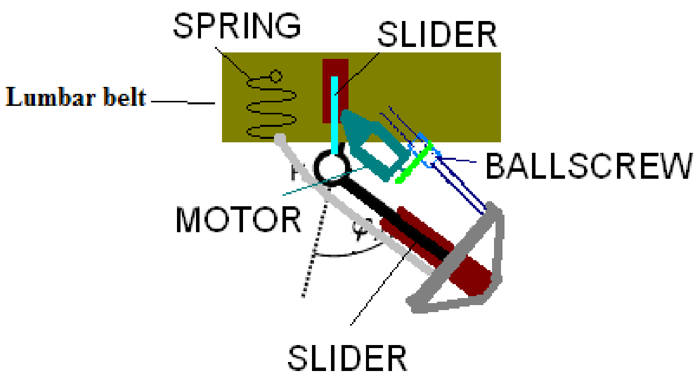

Figure 1.5: Control mechanism for the hip orthosis. The upper support of the SFMO is attached to the lumbar belt of the Walkaround and the lower support is attached to the thigh segment. The cybernetic actuator provides flexion of the hip joint and energy is stored in the spring, which is used for extension.[4]

1.4 Model Construction

The purpose of the control mechanism is to generate the joint moments that are necessary to produce a desired gait pattern for a given individual. The joint torques are normally produced by the muscles contracting and external forces. Measuring muscle forces or joint moments directly is technically complex and requires invasive procedures. These methods do not have application in disabled humans. However, the joint reaction forces and muscle moments can be estimated indirectly by modelling the human body as a system of rigid bodies connected by joints. It is called link-segment modelling and requires a full kinematic description, accurate anthropometric measures and the external forces.[5]

A biomechanical model[6, 3] has been developed for optimal control of walking using functional electrical stimulation. The model is used for estimating the optimal activation of the muscles that produces the necessary joint moments and minimizes muscle activations. The biomechanical model reduces the whole body to a planar model of one leg in the saggital plane where the influence of the rest of the body is represented by interface forces and torques acting at the hip. The leg is modelled as a three-segment linkage of rigid bodies connected by pin joints allowing rotation. The three rigid bodies are the thigh, shank and foot segments and the pin joints between them are the hip, knee and ankle joints. The model is shown in figure 1.6.

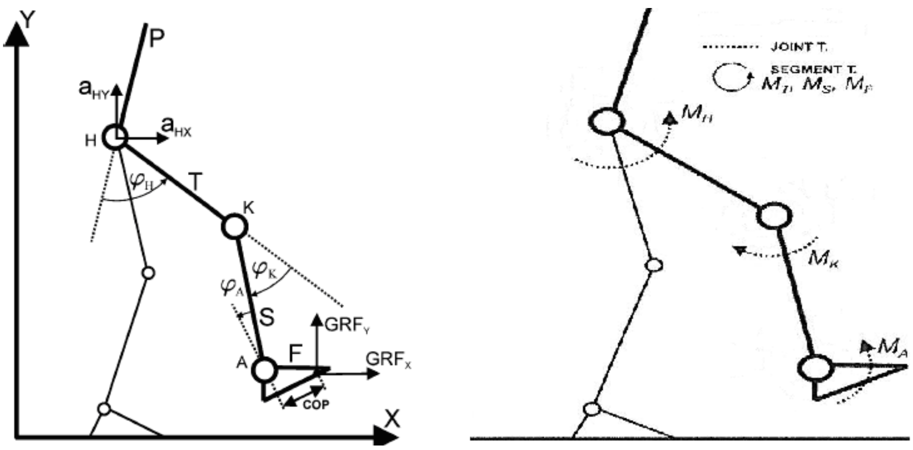

Figure 1.6: The reduced biomechanical model showing a) segment angles: $\varphi_T, \varphi_S, \varphi_F$, joint angles: $\varphi_H, \varphi_K, \varphi_A$, components of hip acceleration: a_{HX} and a_{HY}, components of GRF: GRF_x and GRF_y, COP, pelvis, thigh, shank and foot segments: P, T, S, F and hip, knee and ankle joints: H, K and A. b) the net joint moments at the hip, knee and ankle: $M_{H,K,A}$. [7]

Each leg segment is described by four properties, which are the mass, length, distance to the center of mass from the proximal joint and the moment of inertia about the central axis perpendicular to the saggital plane. The interaction with the ground is modelled by the ground

reaction force (GRF) acting at the sole of the foot along the center of pressure (COP).
Basic theorems of mechanics are used to derive the equations of motion, which are given in appendix.

1.5 Summary

A system is developed to try a new approach to restoring gait, where the hip joint is controlled by an orthosis connected to a walker with wheels called a Walkaround. The Walkaround assists with postural control and weight. The orthosis has a control mechanism comprised by a motor and a spring, where the motor flexes the hip joint and the spring extends the joint. A model has been constructed to simulate the joint torques that are needed to generate a given gait pattern based on the moments of inertia, mass, center of mass and length of the thigh, shank and foot for a given individual. The gait pattern is represented by the hip acceleration, the angular displacements, velocities and accelerations of leg segment, the ground reaction force and center of pressure.

1.6 Problem Statement

How can a program be designed and implemented that applies a biomechanical model to estimate the joint torques that are needed to produce a given gait pattern using the segment parameters of a given individual to be used as a part of an analysis tool for designing the control for the hip orthosis?

Chapter 2

System Development

2.1 Methods

The system is developed using object oriented programming in Matlab [8]. Unified modelling language (UML) [9] is used in the development process to describe and document the functionality of the program. The starting point is to model the system functionality using use case modelling. Use cases models are used to capture the functional aspects of the system to ensure that the requirements are met. The use cases are documented with thorough use case specifications and the work flows are modelled with activity diagrams. The use cases are used as input for the subsequent development phases, which are the design of classes and implementation. The design of classes is modelled using class diagrams and explored using sequence diagrams before implementation.

Chapter 3

Requirements

3.1 System Requirements

In this section it will be assessed what functionality is relevant to the system user and what processes are required to obtain this functionality. This is modelled using a use case diagram providing a visual depiction of the different scenarios of interactions between the actors and the use cases. The use cases are documented with use case specifications and the work flows that the use cases represent are modelled using activity diagrams.

3.1.1 Required Functionality

The actors and use cases identifyed are shown in the use case diagram in figure 3.1. The identified use cases are **Load Model Inputs**, **Present Gait Pattern** and **Estimate Joint Torques**. One actor has been identifyed, which is the **System User**, who operates the system. The actor has association relationships with use cases denoted as solid lines. The solid lines going from the **System User** with arrowheads pointing to **Load Model Inputs** and **Present Gait Pattern** use cases indicate that the **System User** initiates these interactions with the system.

A number of base use cases have include relationships with inclusion use cases denoted as dashed lines with open arrowheads pointing to the inclusion use case and are labelled with the «include» stereotype. An include relationship states that the inclusion use case needs to be executed before the base use case. Thus **Load Model Inputs** must be executed before **Present Gait Pattern** and **Estimate Joint Torques** can be executed.

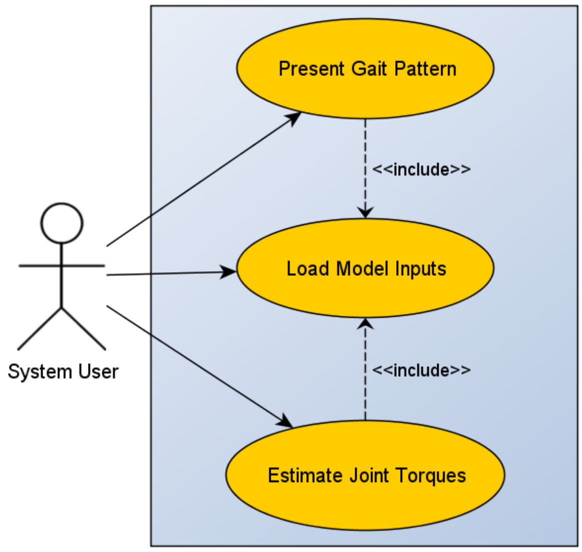

Figure 3.1: Use case diagram showing actors and use cases and the relationships between these.

3.1.2 Use case: Present Gait Pattern

The execution of **Present Gait Pattern** plots the gait pattern that has been chosen by the user for the simulation. The specification of **Present Gait Pattern** is shown in table 3.1.

Present Gait Pattern
Actors: System User
Pre-conditions: A gait pattern has been loaded.
Basic Flow: 1. User requests to view the loaded gait pattern. 2. The gait pattern is displayed to the user.
Post-conditions: None.
Use Case Relationships: Includes **Load Model Inputs**.

Table 3.1: *Specification of* ***Present Gait Pattern***

3.1.3 Use case: Load Model Inputs

Load Model Inputs allows the **System User** to load the set of segment parameters and gait pattern to be used as inputs for the model. The segment parameters and gait patterns are stored in different files, so use case must be executed twice for all inputs to be loaded. The specification of **Load Model Inputs** is shown in table 3.2 and its work flows are shown in figure 3.2.

Load Model Inputs
Actors: System User
Pre-conditions: None
Basic Flow: 1. User browses to directory. 2. User selects file. 3. The model inputs are set the from file.
Post-conditions: All model inputs have been set.
Use Case Relationships: Included by **Present Gait Pattern**, **Estimate Joint Torques** and **Present Results**.

Table 3.2: Specification of ***Load Model Inputs***

3.1.4 Use case: Estimate Joint Torques

Estimate Joint Torques uses the loaded model inputs to estimate the joint torques necessary for the subject represented by the segment parameters to produce the chosen gait pattern and presents them to the user. The specification of **Estimate Joint Torques** is shown in table 3.3 and its work flows are shown in figure 3.2.

Estimate Joint Torques
Actors: System User
Pre-conditions: The model inputs have been loaded.
Basic Flow: 1. Estimate joint torques. 2. Display joint torques
Post-conditions: Joint torques have been estimated.
Use Case Relationships: Includes **Load Model Inputs**.

Table 3.3: *Specification of* ***Estimate Joint Torques***

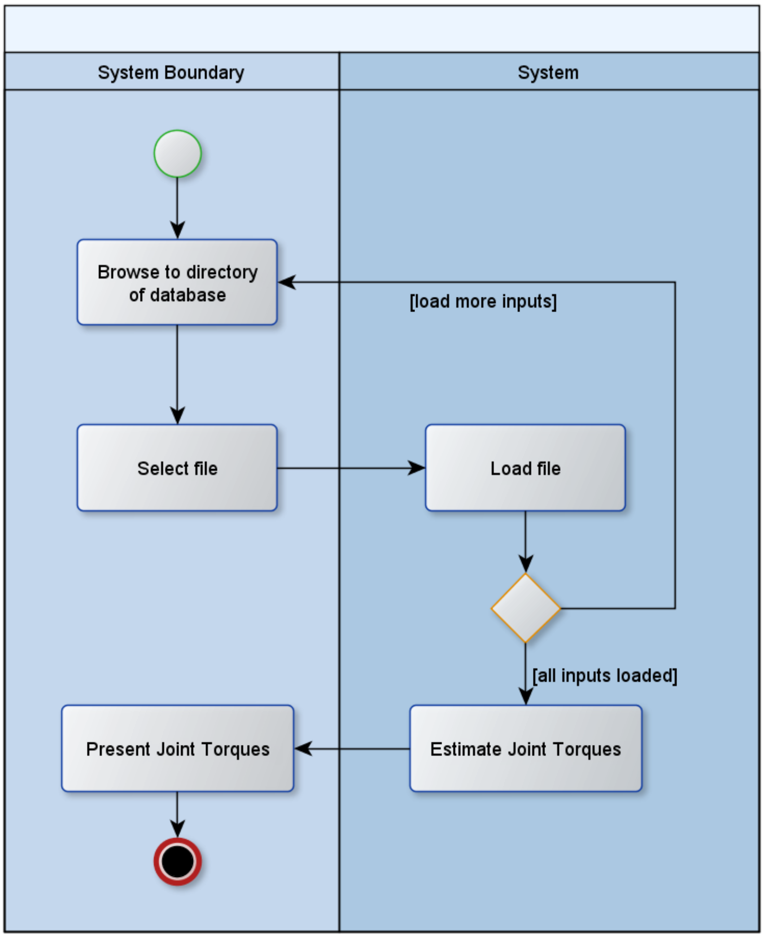

Figure 3.2: Work flows of **Load Model Inputs** and **Estimate Joint Torques**.

Chapter 4

Analysis

4.1 Analysis Classes

In this section analysis classes are identified, which represent candidates for potential classes. The purpose of creating analysis classes is to describe how the use cases can be realized. The analysis classes are identified by breaking up the use cases into components that form the basis for the classes. This is done using name and verb analysis. The realization of each use case is documented by a flow of events description and a communication diagram, that illustrates how the objects are linked together and how they interact in space. The analysis classes are organized into stereotypes represented by the symbols shown in figure 4.1

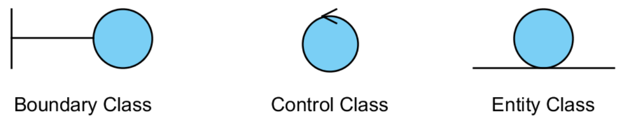

Figure 4.1: The symbols representing the boundary, control and entity stereotypes of the analysis classes.

The stereotypes characterize the behavior of a class:

- Boundary classes model the interaction between the system and the actors. They represent the functionalities necessary for the actors to interact with the system.
- Entity classes model information that is persistent or has a long life time in the system. These classes can exist beyond the execution of one use case or session.
- Control classes are used to represent coordination, sequences of operations and control of other objects. They are often used to encapsulate control related to a specific use case.

4.1.1 Realization of Load Model Inputs

The flow of events for the realization of **Load Model Inputs** is:

1. A **System User** has started the program and activates **DialogBox** object. The system user browses the directory and selects a file.
2. The **DialogBox** object asks a **Load** object to perform the loading of the file.
3. The **Load** object asks a **Database** object to open the specified file.
4. The **Load** object asks a **Segment Parameters** object to set the segment parameters with the values in the file.
5. The process is repeated for a **GaitPattern** object.

The communication diagram is shown in figure 4.2.

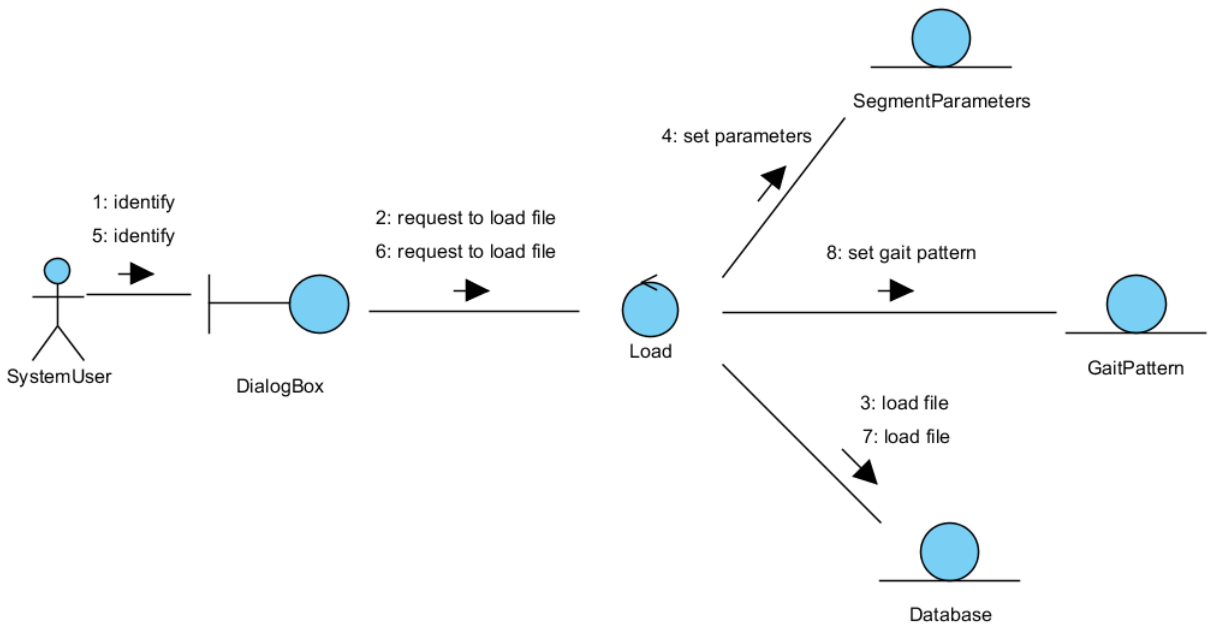

Figure 4.2: Communication diagram showing the analysis classes for the realization of **Load Model Inputs**.

4.1.2 Realization of Present Gait Pattern Use Case

The flow of events for the realization of **Present Gait Pattern** is:

1. A **System User** asks a **UserInterface** object to present the loaded gait pattern.
2. The **UserInterface** object asks a **Presentation** object to perform the presentation of the gait pattern.
3. The **Presentation** object requests the gait pattern from a **GaitPattern** object.
4. The **Presentation** object asks the **UserInterface** object to plot the gait pattern.

The communication diagram is shown in figure 4.3.

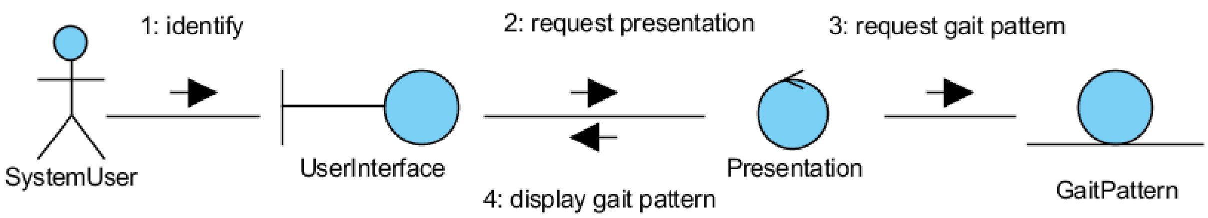

Figure 4.3: Communication diagram showing the analysis classes for the realization of **Present Gait Pattern**.

4.1.3 Realization of Estimate Joint Torques Use Case

The flow of events for the realization of **Estimate Joint Torques** is:

1. A **Load** object has finished loading model inputs and requests an estimation from an **Estimation** object.
2. The **Estimation** object asks a **Model** object to estimate joint torques for the loaded input.
3. The **Model** object requests the loaded segment parameters from a **Segment Parameters** object.
4. and then requests the loaded gait pattern from a **GaitPattern** object and estimates the joint torques.
5. The **Estimation** object asks a **UserInterface** object to display the joint torques.

The communication diagram is shown in figure 4.4.

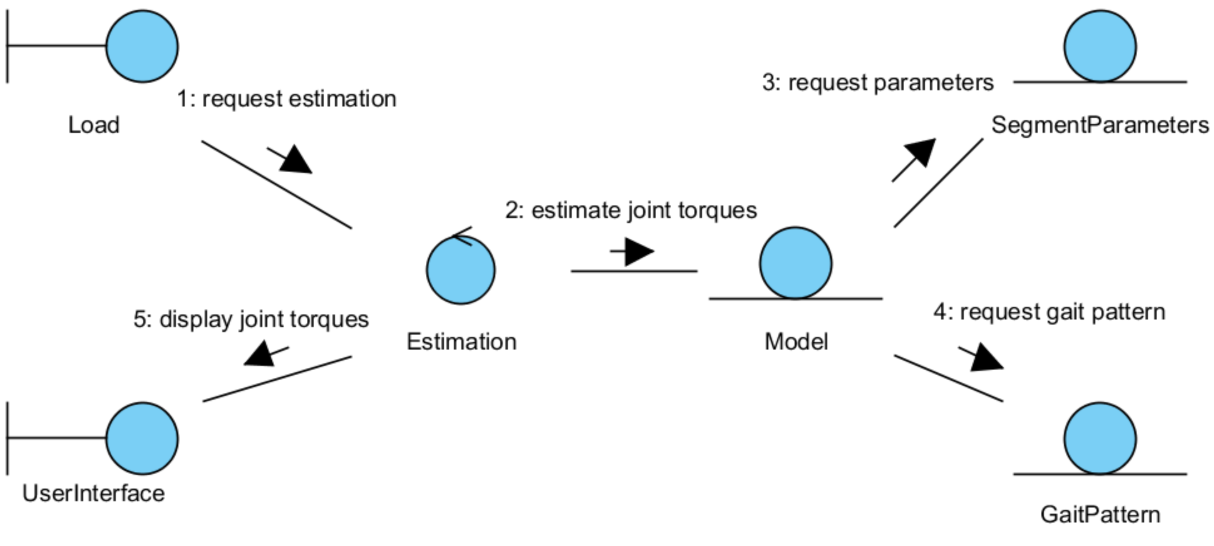

Figure 4.4: Communication diagram showing the analysis classes for the realization of **Estimate Joint Torques**.

Chapter 5

Design

5.1 Design

The identified analysis classes are transformed into design classes using the analysis classes as candidates. The classes are expanded to methods and associations and a more technical description is made of the activities and interactions of the classes using class diagrams and sequence diagrams. The purpose of this process is to create classes ready for implementation. The analysis classes are transformed into design classes as shown in figure 5.1.

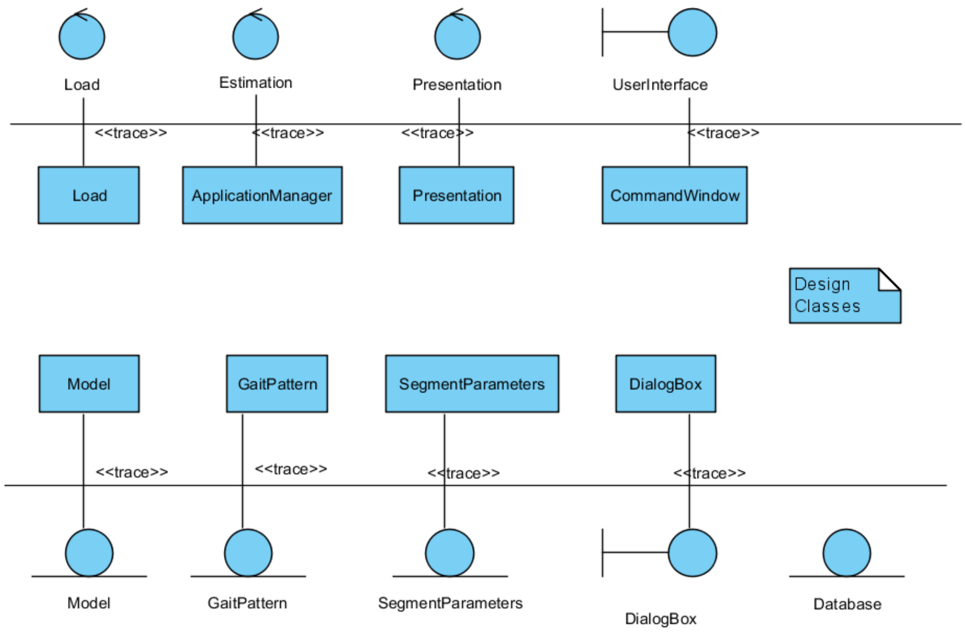

Figure 5.1: Transformation from analysis classes, shown in the top and bottom, into design classes, shown in the middle.

5.1.1 Class Collaborations

The class diagrams shown in the figures 5.2, 5.3 and 5.4 show the static collaboration between the objects for each of the three use cases. The dynamic collaborations are shown in the figures 5.5, 5.7 and 5.7.

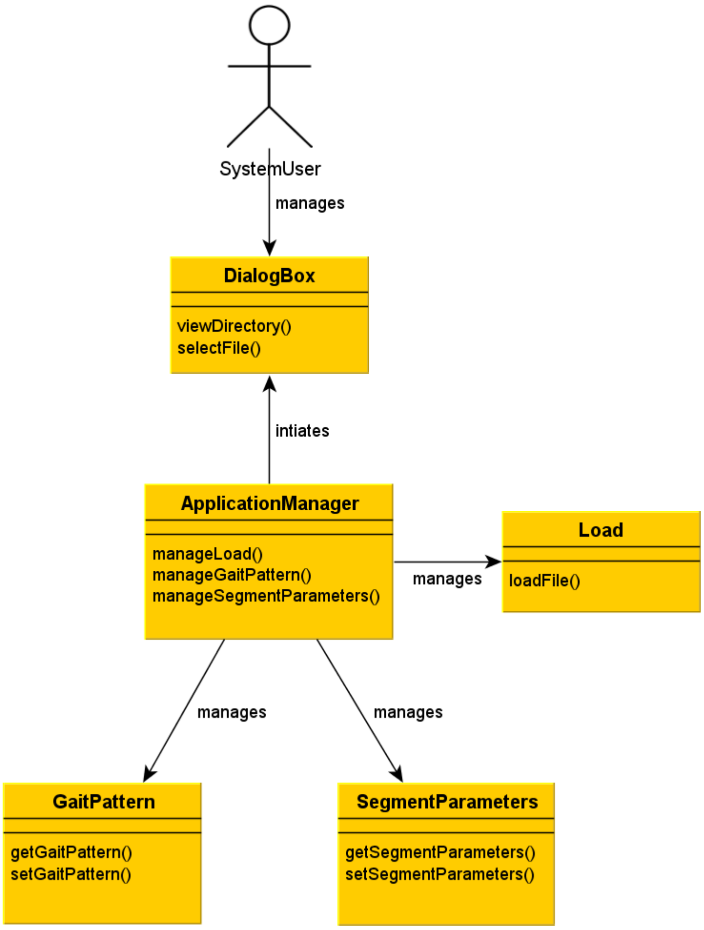

Figure 5.2: Class diagram for **Load Model Inputs**.

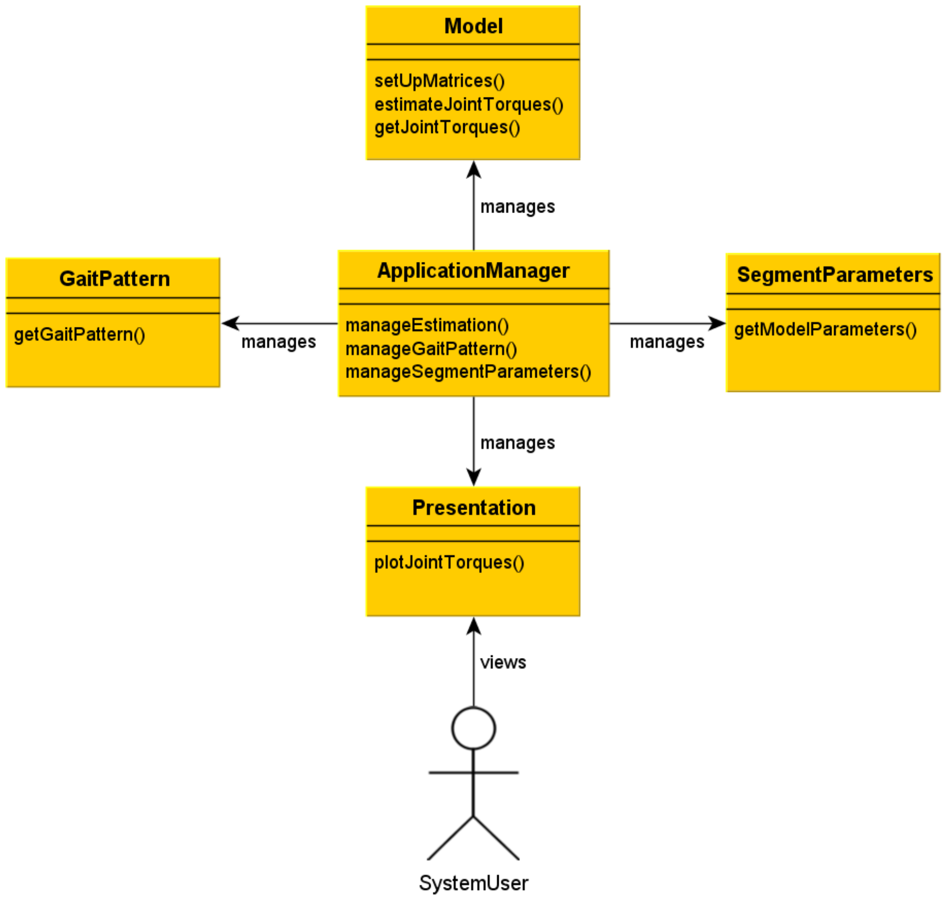

Figure 5.3: Class diagram for **Estimate Joint Torques**.

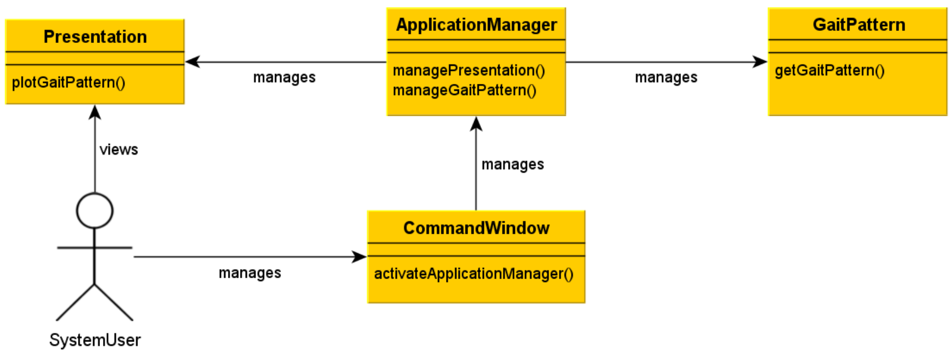

Figure 5.4: Class diagram for **Present Gait Pattern**.

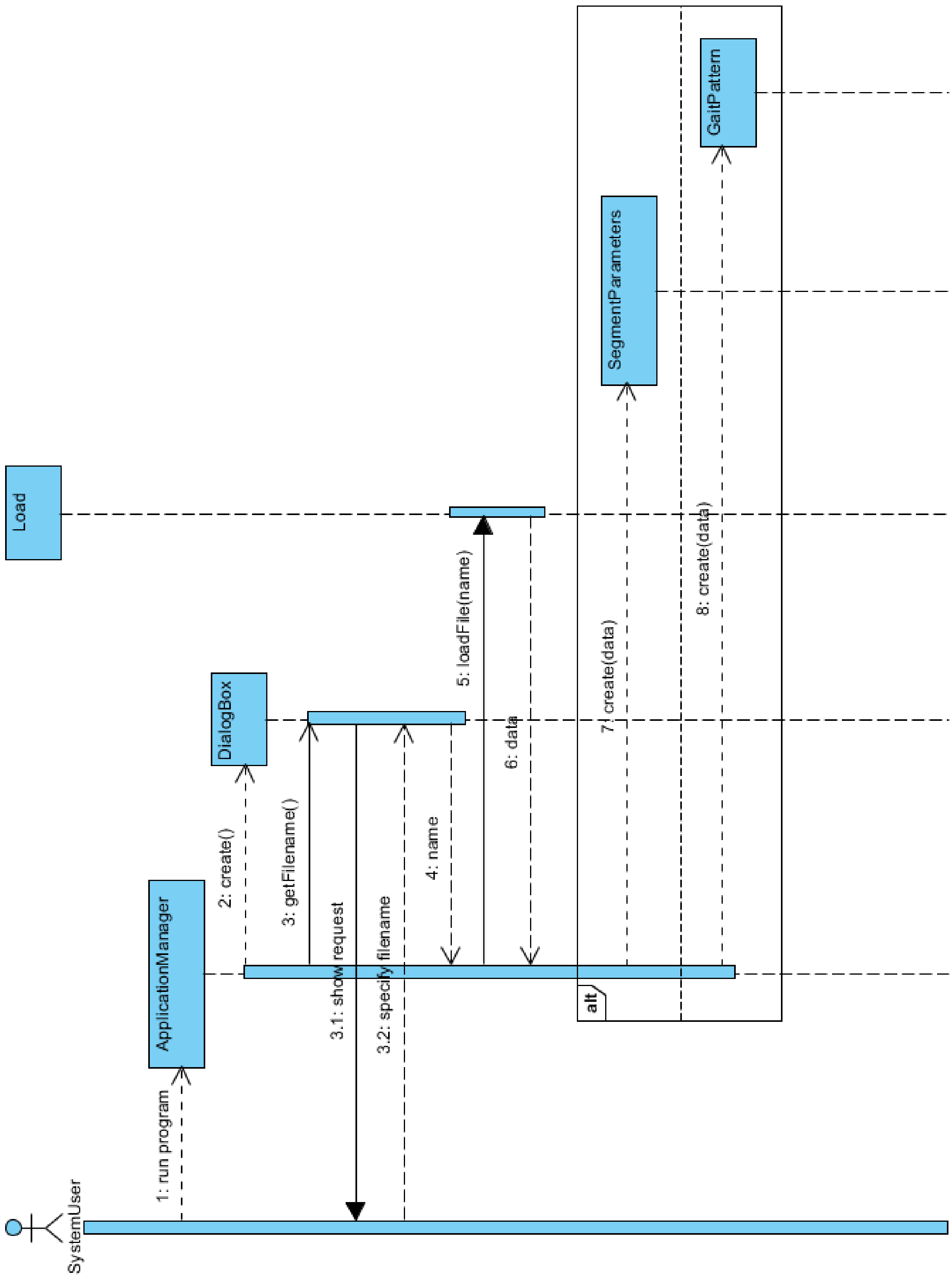

Figure 5.5: Sequence diagram for execution of **Load Model Inputs**.

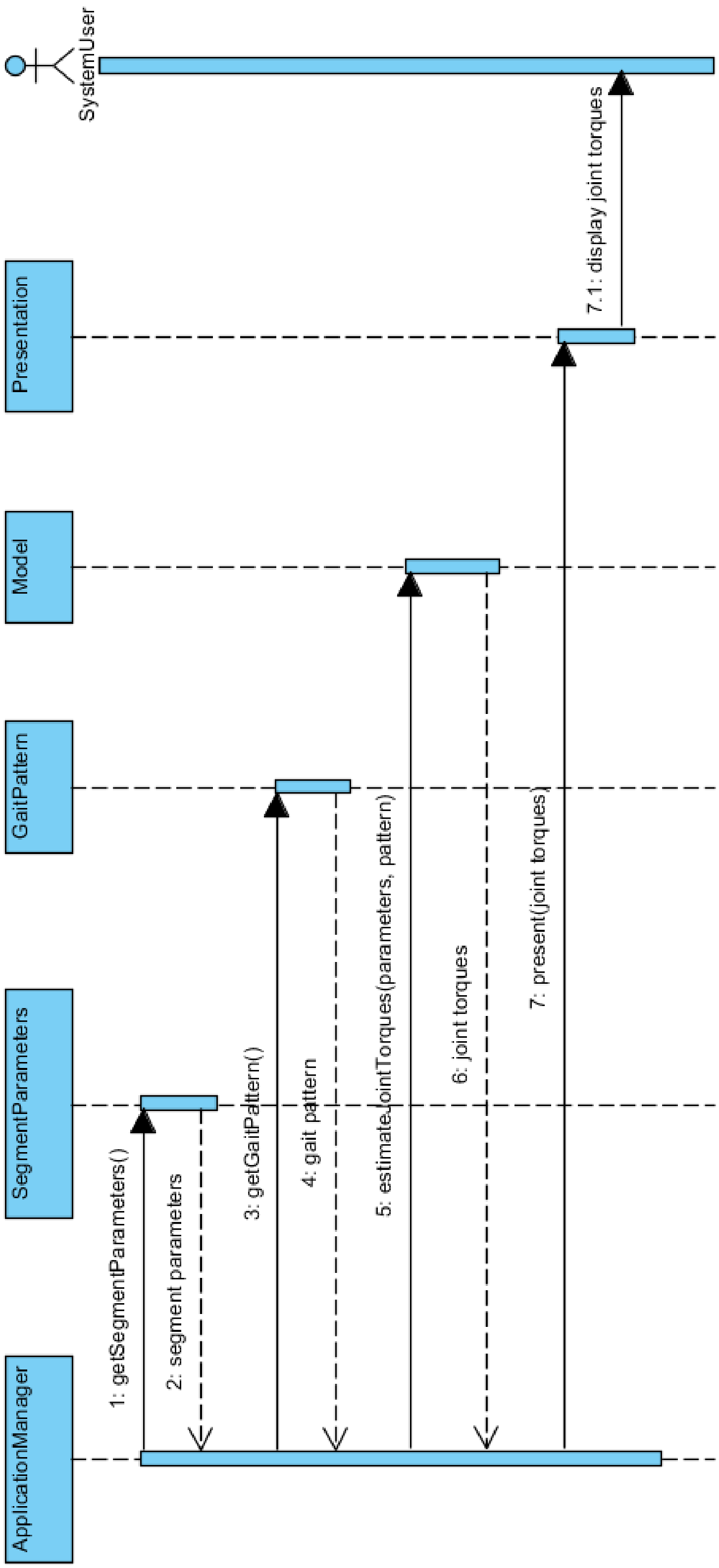

Figure 5.6: Sequence diagram for execution of **Estimate Joint Torques**.

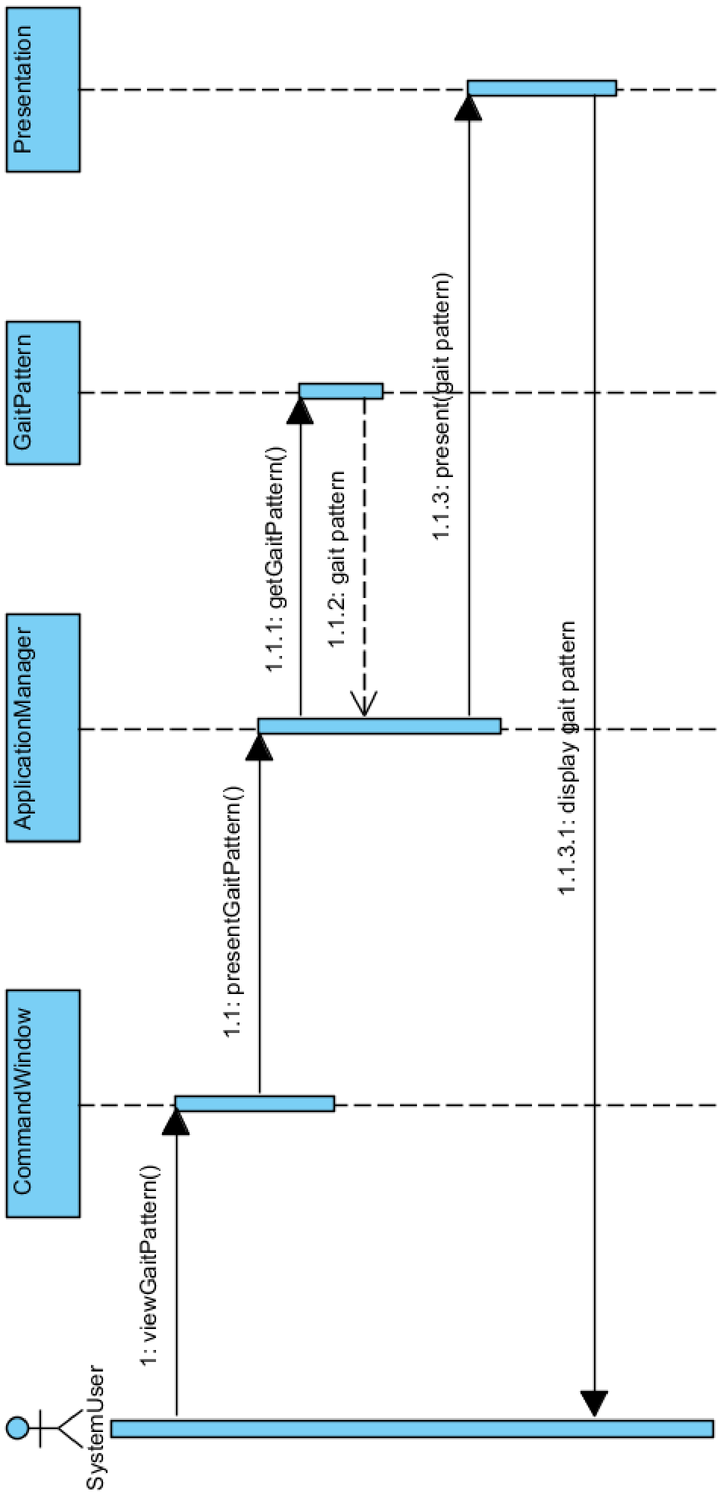

Figure 5.7: Sequence diagram for execution of **Present Gait Pattern**.

Chapter 6

Test

6.1 Test of Model

Joint torques are estimated for one stride starting from heelstrike using a set of model parameters from a healthy subject given in table 6.1 and the gait pattern shown in figure 6.1.

Parameters for the Biomechanical Model	
Thigh	$J_{CT} = 0.104kg \cdot m^2$; $m_T = 7.5kg$; $d_T = 0.17m$; $L_T = 0.405m$
Shank	$J_{CT} = 0.053kg \cdot m^2$; $m_T = 3.48kg$; $d_T = 0.19m$; $L_T = 0.425m$
Foot	$J_{CT} = 0.003kg \cdot m^2$; $m_T = 1.08kg$; $d_T = 0.04m$; $L_T = 0.08m$

Table 6.1: *The set of model parameters from a healthy subject that is used as input for the biomechanical model.*

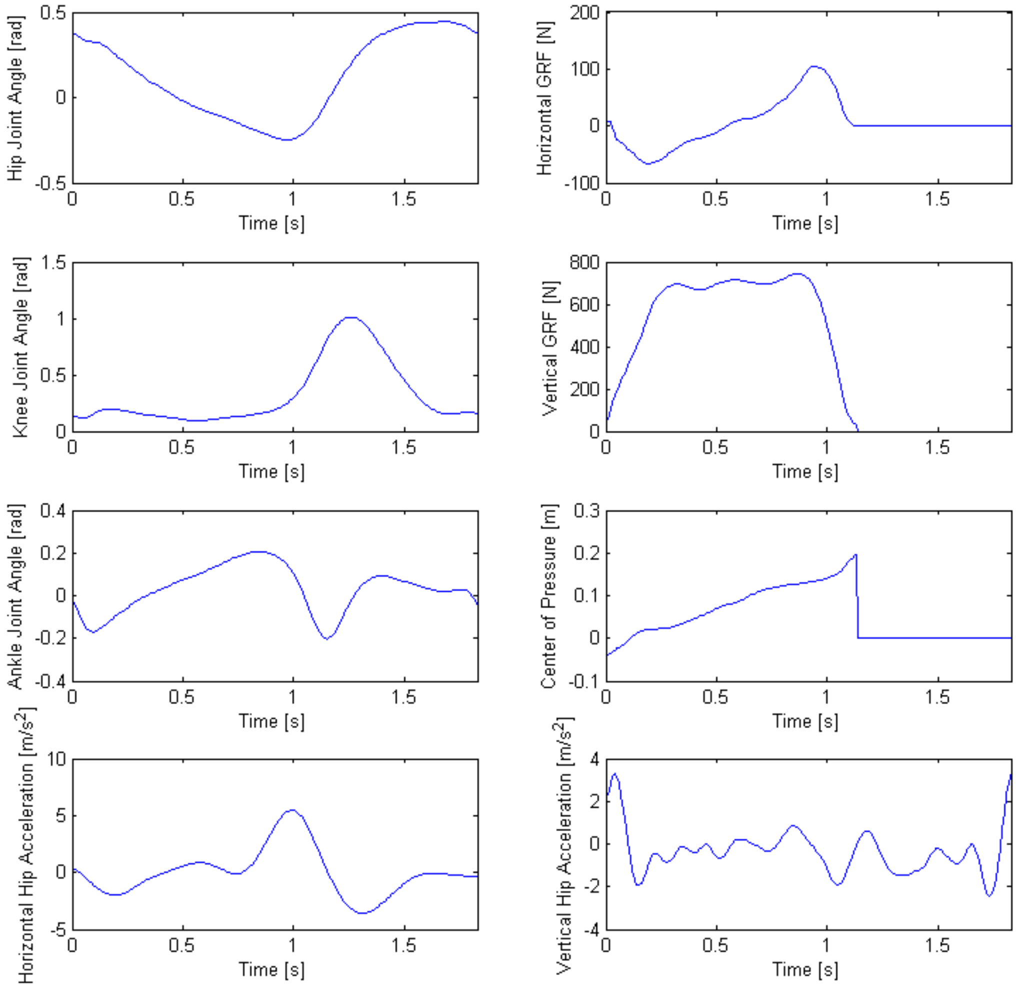

Figure 6.1: The recorded gait pattern used as input for the biomechanical model.

6.1.1 Results

The joint torques estimated by the model are shown in figure 6.2.

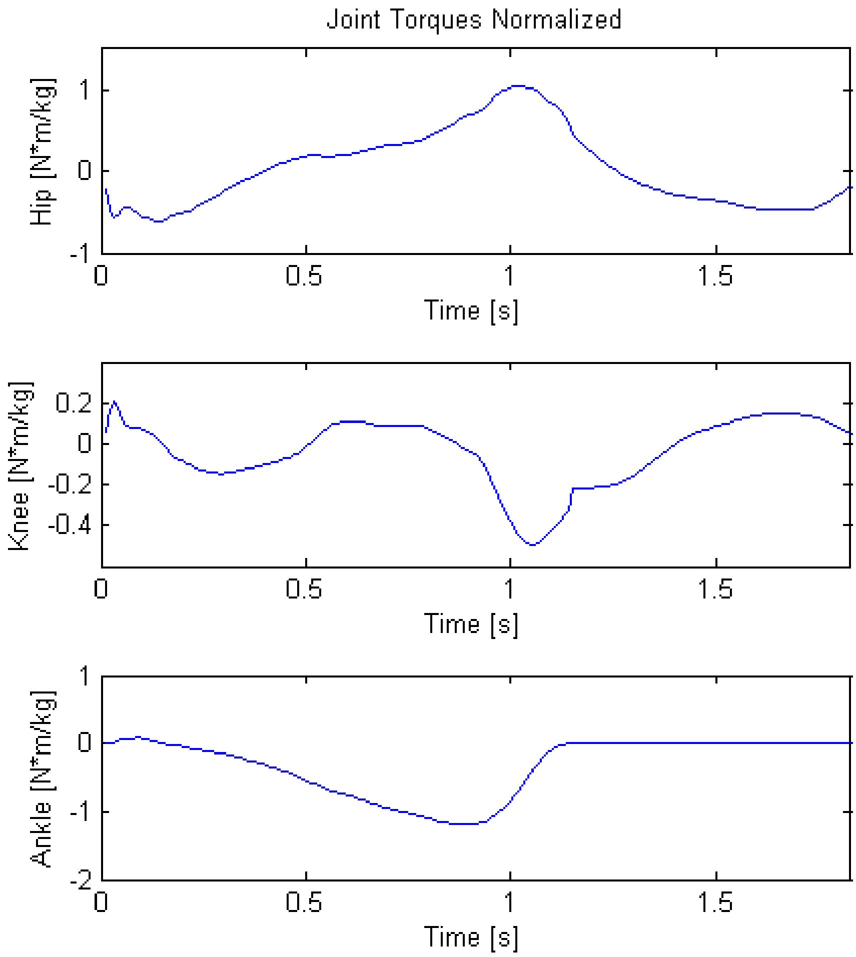

Figure 6.2: Joint torques estimated by the model. Joint flexion is in the positive direction.

6.2 Conclusion

Restoring gait for physically disabled often requires assistance with weight, posture and swing of the leg. A new approach incorporates this using a system called Walkaround, which is a walker with wheels, and a hip orthosis. The Walkaround assists with weight and postural control and a hip orthosis controls the leg. The hip orthosis is controlled by a mechanism comprising a motor and a spring, where the motor produces joint flexion, which loads the spring with energy that it uses to produce joint extension. An analysis tool for designing the control for the hip orthosis has to be developed, to analyze the effects of the spring during walking and the necessary characteristics of the motor drive. This requires the development of an inverse dynamics model to estimate the joint torques necessary to produce a given gait pattern for a given individual.

A link segment model has been constructed comprising three rigid bodies each characterized by the four parameters: moment of inertia, center of mass, mass and length. The model estimates joint torques from a gait pattern described by the angular displacement, velocity and acceleration of the each link segment, the acceleration of the hip, ground reaction force and the center of pressure under the foot sole.

The report has investigated how a system could contribute in performing the estimation of the necessary joint torques. A system has been developed that has been implemented with object oriented Matlab. Three requirements have been identified, which are 1) loading input data for the biomechanical model 2) estimating the joint torques using the loaded input data and 3) presenting the joint torques and input data to the user.

Design classes were identified to provide the functionality that meets the three requirements. The functionality was divided up into different classes that can be reused in the future development of the analysis tool for designing the control for the hip orthosis. The biomechanical model was implemented and tested and it produces joint torques from the given recorded data that are within range and similar in profile to the ones stated in litterature.

Bibliography

[1] Aleksandar Veg and Dejan B. Popovic. Walkaround: Mobile balance support for therapy of walking. *IEEE Trans. Rehab. Eng.*, 16(3):264–269, 6 2008.

[2] Dejan B. Popovic. Dynamics of the self-fitting modular orthosis. *IEEE Transactions on Robotics and Automation*, 6(2):200–207, 4 1990.

[3] Strahinja Dosen and Dejan B. Popovic. Accelerometers and force sensing resistors for optimal control of walking of a hemiplepic. *IEEE Trans. Biomed. Eng.*, 55(8):1973–1984, 2008.

[4] Dejan B. Popović. Professor at institut for medicin og sundhedsteknologi at aalborg university.

[5] David A. Winter. *Biomechanics and Motor Control of Human Movement.* Wiley, 2005. ISBN: 0-471-44989-X.

[6] Dejan B. Popovic et al. Optimal control of walking with functional electrical stimulation: A computer simulation study. *IEEE Trans. Rehab. Eng.*, 7(1):69–79, 3 1999.

[7] Strahinja Dosen. *A New Tool for Designing the Control of Walking in Individuals with Disabilities.* PhD thesis, SMI, Aalborg University, 2008.

[8] Mathworks. Matlab 7 object-oriented programming. `http://www.mathworks.com/help/techdoc/matlab_oop/ug_intropage.html`, 2011. Viewed: May 2011.

[9] Hans-Erik Eriksson et al. *UML 2 Toolkit.* Wiley, 2004.

Chapter 7

Appendix

Reduced Model Equations of Motion

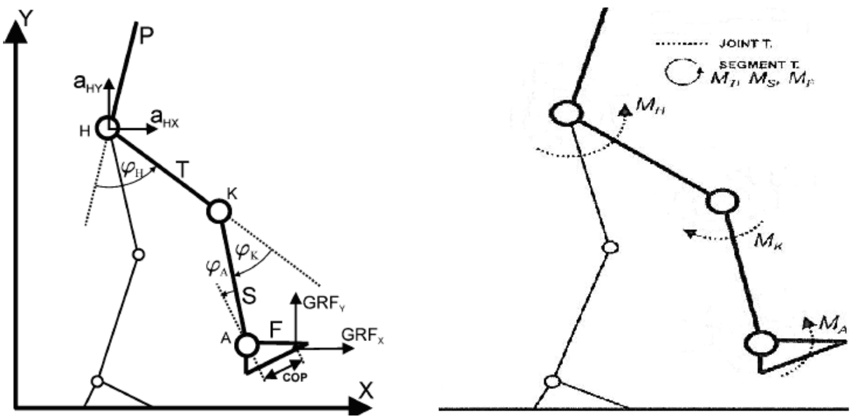

Figure 1: The reduced biomechanical model. [7]

$$A_1\ddot{\varphi}_T + A_2\ddot{\varphi}_S cos(\varphi_T - \varphi_S) + A_3\ddot{\varphi}_F cos(\varphi_T - \varphi_F) + A_4\dot{\varphi}_S^2 sin(\varphi_T - \varphi_S) + A_5\dot{\varphi}_F^2 sin(\varphi_T - \varphi_F) - A_6 a_{HX} sin(\varphi_T) + A_7(a_{HY} + g)cos(\varphi_T) + GRF_X L_T sin(\varphi_T) - GRF_Y L_T cos(\varphi_T) = M_T$$

$$B_1\ddot{\varphi}_S + B_2\ddot{\varphi}_T cos(\varphi_T - \varphi_S) + B_3\ddot{\varphi}_F cos(\varphi_S - \varphi_F) - B_4\dot{\varphi}_T^2 sin(\varphi_T - \varphi_S) + B_5\dot{\varphi}_F^2 sin(\varphi_T - \varphi_F) - B_6 a_{HX} sin(\varphi_S) + B_7(a_{HY} + g)cos(\varphi_S) + GRF_X L_S sin(\varphi_S) - GRF_Y L_S cos(\varphi_S) = M_S$$

$$C_1\ddot{\varphi}_F + C_2\ddot{\varphi}_T cos(\varphi_T - \varphi_F) + C_3\ddot{\varphi}_S cos(\varphi_S - \varphi_F) - C_4\dot{\varphi}_T^2 sin(\varphi_T - \varphi_F) - C_5\dot{\varphi}_F^2 sin(\varphi_S - \varphi_F) - C_6 a_{HX} sin(\varphi_F) + C_7(a_{HY} + g)cos(\varphi_F) + GRF_X L_F sin(\varphi_F) - GRF_Y L_F cos(\varphi_F) + M_{GRF} = M_F$$

$$A_1 = J_{CT} + m_F L_T^2 + m_S L_T^2 + m_T d_T^2;$$

$A_2 = m_F L_S L_T + m_S d_S L_T$;
$A_3 = m_F d_F L_T$; $A_4 = A_2$; $A_5 = A_3$;
$A_6 = m_S L_T + m_F L_T + m_T d_T$; $A_7 = A_6$

$B_1 = J_{CS} + m_F L_S^2 + m_S d_S^2$; $B_2 = A_2$;
$B_3 = m_F d_F L_S$; $B_4 = A_2$; $B_5 = B_3$;
$B_6 = m_F L_S + m_S d_S$; $B_7 = B_6$

$C_1 = J_{CF} + m_F d_F^2$; $C_2 = A_3$; $C_3 = B_3$;
$C_4 = A_3$; $C_5 = B_3$; $C_6 = m_F d_F$; $C_7 = C_6$

For COP > 0:
$M_{GRF} = GRF_X \cdot COP \cdot sin(\varphi_F - \frac{3 \cdot \pi}{2}) - GRF_Y \cdot COP \cdot cos(\varphi_F - \frac{3 \cdot \pi}{2})$

For COP ≤ 0:
$M_{GRF} = -GRF_X \cdot COP \cdot sin(\varphi_F - \frac{\pi}{2}) + GRF_Y \cdot COP \cdot cos(\varphi_F - \frac{\pi}{2})$

$M_T = M_H + M_K$
$M_S = -M_A - M_K$
$M_F = M_A$
where

J_{CT}, J_{CS}, J_{CF}	moments of inertia of the thigh, shank and foot about the central axes perpendicular to the saggital plane.
L_T, L_S, L_F	lengths of the thigh, shank and foot.
d_T, d_S, d_F	distances from the proximal joint to the center of mass for the thigh, shank and foot.
m_T, m_S, m_F	masses of the thigh, shank and foot.
a_{HX}, a_{HY}	horizontal and vertical components of the hip acceleration.
GRF_X, GRF_Y	horizontal and vertical components of the GRF.
g	gravitational acceleration.
$\varphi_T, \varphi_S, \varphi_F$	angles of thigh, shank and foot from the horizontal axis.
$\varphi_H, \varphi_K, \varphi_A$	hip, knee and ankle joint angles.
M_T, M_S, M_F	torques acting at the thigh, shank and foot.
M_H, M_K, M_A	total torques acting at the hip, knee and ankle joints.
COP	center of pressure.
M_{GRF}	the part of the ground reaction force moment that depends on COP.