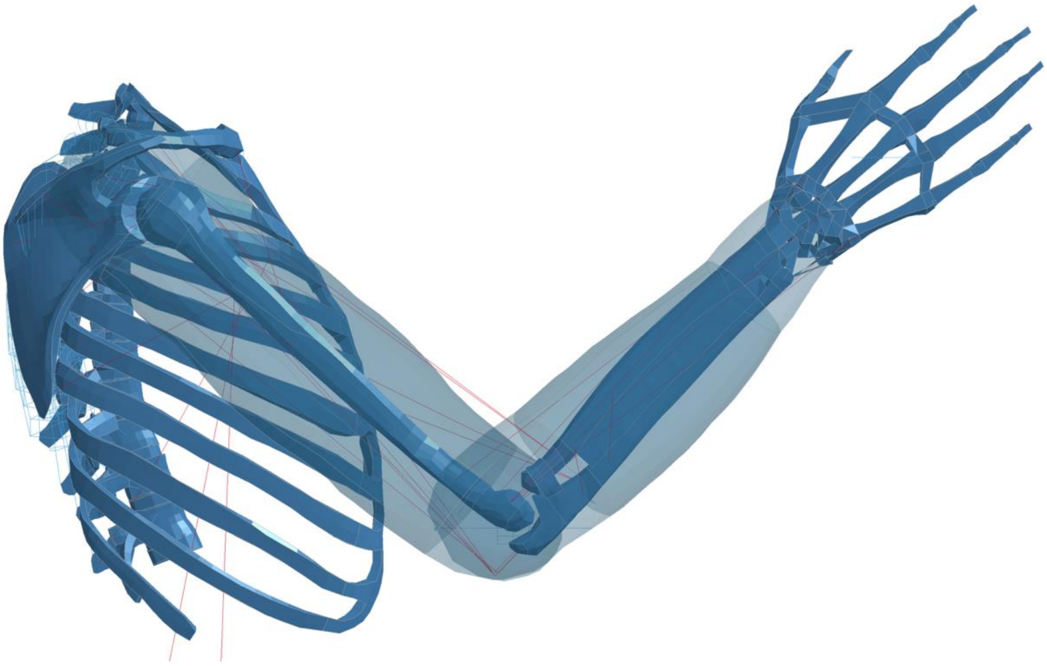
CHALMERS



Active Muscle Control in Human Body Model Simulations

Implementation of a feedback control algorithm with standard keywords in LS-DYNA

Master's Thesis in Automotive Engineering

SAMUEL ANDERSSON

Department of Applied Mechanics
Division of Vehicle Safety
Injury Prevention
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2013

Master's thesis 2013:62

MASTER'S THESIS IN AUTOMOTIVE ENGINEERING

Active Muscle Control in Human Body Model Simulations

Implementation of a feedback control algorithm with standard keywords in LS-DYNA

SAMUEL ANDERSSON

Department of Applied Mechanics
Division of Vehicle Safety
Injury Prevention
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2013

Active Muscle Control in Human Body Model Simulations
Implementation of a feedback control algorithm with standard keywords in LS-DYNA
SAMUEL ANDERSSON

Master’s Thesis 2013:62
ISSN 1652-8557
Department of Applied Mechanics
Division of Vehicle Safety
Injury Prevention
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone: + 46 (0)31-772 1000

Cover:
Elbow model, part of the Chalmers Active Human Body Model, used for validating the feedback control algorithm.

Chalmers Reproservice
Göteborg, Sweden 2013

Active Muscle Control in Human Body Model Simulations
Implementation of a feedback control algorithm with standard keywords in LS-DYNA

Master's Thesis in the Automotive Engineering programme

SAMUEL ANDERSSON
Department of Applied Mechanics
Division of Vehicle Safety
Chalmers University of Technology

ABSTRACT

Simulation of occupant responses in scenarios involving low load levels requires the addition of active muscles to add posture control and muscular response to the human body. A model for active muscle control has previously been implemented in the Chalmers Active Human Body Model (Chalmers AHBM) using subroutines written in the FORTRAN programming language. In order to decrease conflicts with future software updates and increase the usability of the model, the possibility to build the model using standard LS-DYNA keywords has been investigated in this thesis. The keywords need to perform signal retrieval, addition of a time delay, implementation of feedback control algorithms and solution of differential equations for activation dynamics. In addition to the approach already existing as a subroutine to LS-DYNA, alternative ways to implement the areas of functionality has been reviewed in literature and scientific papers. A proposed solution using standard keywords called the Load Curve Method (LCM) has been developed. For basic mathematical operations *DEFINE_CURVE_FUNCTION keyword and its built in functions are used. To solve differential equations a mechanical representation of the equations together with the ability in LS-DYNA to simulate kinetic and kinematic properties of single and multiple nodes is proposed. All areas of functionality except a true time delay have been implemented using standard LS-DYNA keywords. Since it is not possible to retrieve data from earlier time steps a first order low pass filter has been implemented to model the effect of a time delay. Even though it does not exactly replicate the input signal, the filter provides a phase shift and can be used to model closed loop instability. The standard keyword solution has been implemented with the Chalmers AHBM as well as on an elbow joint model using LS-DYNA V971, R5.1.1 or 6.1.0 standard binary. It has been validated against the implementation that uses a subroutine to LS-DYNA; the elbow joint model in a situation similar to the ARMANDA haptic equipment (de Vlugt et al., 2003) and Chalmers AHBM in a seated position under influence of gravity and 1G forward acceleration. In simulations with the Chalmers Active Human Body Model a decrease in computational time of about 14% has been seen for the LCM solution compared to the user defined uctrl1 FORTRAN sub-routine.

Keywords: THUMS; LS-DYNA; active human body model; active muscle; reflexive response; feedback control

Aktiv muskelkontroll i Humanmodellsimulering Implementering av reglerad återkoppling med hjälp av programkort i LS-DYNAs grundutförande

Examensarbete inom Fordonsteknik
SAMUEL ANDERSSON
Institutionen för Tillämpad Mekanik
Avdelningen för Fordonssäkerhet
Chalmers tekniska högskola

SAMMANFATTNING

För att undersöka hur gravitation och muskelkraft påverkar skaderisken hos passagerare i fordon under inverkan av låga yttre krafter pågår arbete i att bygga in aktiv muskelstyrning i humanmodeller. Tidigare arbete inom detta område har innefattat en på förhand bestämd mängd muskelaktivitet (Iwamoto et al., 2009; Choi et al., 2005). En modell har tagits fram av Tekn. Lic. Jonas Östh (Östh, 2010) där muskelaktiviteten är självreglerande och återkopplad baserat på positionen hos vissa delar av kroppen. Fördröjningen i tid på grund av neural aktivering via sensorisk förnimmelse samt biokemiska aktiveringsprocesser i muskeln ingår. Modellen finns implementerad som en användarspecificerad subrutin till LS-DYNA programmerad i FORTRAN som kräver kompilering av en användarbinär. Denna rapport beskriver arbetet med att överföra ovan nämnda funktioner, programmerade så att simuleringen kan exekveras med LS-DYNA standard-binär. Utöver Jonas Ösths modell har en undersökning gjorts i litteratur relaterad till biomekanik, muskelaktivering och humanmodeller för simulering av skaderisk-bedömning i fordon. Den resulterande programmeringen, kallad ”Load Curve Method” (LCM), använder enbart funktioner som finns i LS-DYNAs grundutförande, version 971, utgåva 5.1.1 och 6.1.0. Muskelstyrningen har kunnat implementeras med bibehållen grundfunktionalitet, där endast fördröjning av neural respons har fått en implementering som inte lyckas uppfylla koherens gentemot Jonas Ösths lösning. Frånvaron i LS-DYNA att kunna spara information från ett beräkningssteg till ett senare under en och samma simulering komplicerade implementeringen av fördröjd neural respons. Grundläggande matematiska uttryck har implementerats via *DEFINE_CURVE_FUNCTION programkortet. Differentialekvationer för dödtid och muskelaktivering har tolkats kinetiskt och kinematiskt där LS-DYNAs förmåga att lösa rörelsen hos noder i mekaniska system utnyttjas. Den framtagna lösningen har implementerats på Chalmers Active Human Body Model (Chalmers AHBM) samt på en armbågs-modell. Validering har skett gentemot Jonas Ösths lösning för armbågsmodellen i en uppställning liknande ARMANDA (de Vlugt et al., 2003). För Chalmers AHBM i en sittande position under inverkan av gravitation samt 1G framåtriktad acceleration. Resultaten visar på goda möjligheter att kunna använda den utvecklade modellen men framförallt implementering av muskelaktiveringsdynamik samt fördröjning behöver vidareutvecklas och ytterligare justeras. En minskning i beräkningstid om 14% har noterats vid användande av LS-DYNA standardbinär.

Nyckelord: THUMS; LS-DYNA; aktiv humanmodell; muskel aktivitet; muskelreflex; återkopplad reglering

Contents

Preface

This work was conducted as an M.Sc thesis at the department of Applied Mechanics at Chalmers University of Technology. The work is part of an on-going project concerning Active Human Body Modelling at SAFER - The Vehicle and Traffic Safety Centre at Chalmers in collaboration with Autoliv, Volvo Car Corporation and Volvo Technology.

I would like to thank my supervisor Lic.Eng Jonas Östh for his support and for giving me the opportunity to delve in to a world of sarcomeres, computer coding and the endless staring at your own elbow performing a flexing motion.

I would also like to thank my examiner Associate Professor Karin Brolin and the people at SAFER Vehicle and Traffic Safety Centre whom I have always been able to come and ask for advice and has given me an insight into the work being done at the forefront of vehicle safety research of today and in the future to come.

Göteborg June 2013

Samuel Andersson

Notations

Variabels

$\phi(\omega)$	Phase shift function concerning low pass filter latency
$\sigma_{pe}(l)$	Stress Equation for parallel elastic element
$\sigma_d(v)$	Stress Equation for parallel damping element
τ	Differential notation for time
ω	Angular frequency
$de(t)/dt$	Differentiate of error signal
$e(t)$	Error signal
$f_l(l)$	Hill force-length relationship
$f_v(v)$	Hill force-velocity relationship
$G(s)$	Laplacian transfer function
$ie(t)$	Integrated error signal
l	Instantaneous muscle length
n_1, n_2, n_3	Nodes positioned in elbow model to define vector
n_{1x}, n_{1z}	x- and z-coordinate for upper node, defines vectors in the trunk model
n_{2x}, n_{2z}	x- and z-coordinate for lower node, defines vectors in the trunk model
$N_e(t)$	Muscle excitation level
$N_a(t)$	Muscle activation level
$r(t)$	Reference signal
s	Laplacian variable for frequency
t_n	Time at nth timestep
$u(t)$	Control signal
$u_*(t)$	Control signal from body region controller
$u_m(t)$	Control signal for each muscle
U_{hum}	Vector defining orientation of humerus
U_{uln}	Vector defining orientation of ulna
v	Muscle velocity/rate of contraction
X	Laplacian representation of x(t)
$\dot{x}_n(t)$	Velocity of node n_1 in activation dynamics model
Y	Laplacian representation of y(t)
$y(t)$	Body region angle
$y_d(t)$	Delayed input of body region angle
$\dot{y}_{n(t)}$	Velocity of node n_2 in activation dynamics model
$\ddot{y}_n(t)$	Acceleration of node n_2 in activation dynamics model

Constants

σ_{max}	Maximum isometric contraction
ω_c	Cut off frequency
cc	Muscle cocontraction level
C_{mvl}	Asymptotic value for increased eccentric muscle lengthening
C_{lng}	Transition between concentric and eccentric muscle shortening
C_{pe}	Passive elastic muscle stiffness
C_{sh}	Shape of rate of contraction curve for concentric shortening
ΔT	Time between two consecutive time steps
D	Passive damping muscle stiffness
K_p	Proportional gain
K_i	Integrative gain

K_d	Derivative gain
l_{opt}	Optimal muscle length for maximum isometric contraction
$\bar{r}_*$	Constant to scale control signal depending on flexor or extensor muscle
t_0	Initial time of simulation
T	Low pass filter time constant
T_{de}	Neural delay time
T_f	Lowest level of muscle excitation
T_{ne}	Neural muscle excitation time constant
T_{naa}	Neural muscle activation time constant
T_{nad}	Neural muscle deactivation time constant
x_{dir}	Positive x-direction for simulation coordinate system compared to model coordinate system
y_0	Initial body region angle

Acronyms

AHBM	Active Human Body Model
ATD	Anthropomorphic Test Devices
CE	Contractile Element
CNS	Central Nervous System
DE	Parallel Damping Element
EF	Extrafusal Fibre
FE	Finite element
GTO	Golgi Tendon Organ
HBM	Human Body Model
IF	Intrafusal Fibre
LCM	Load Curve Method
LSTC	Livermore Software Technology Corporation
MB	Multi Body
PE	Parallel Elastic element
PID	Proportional, Integrative and Derivative
PNS	Peripheral Nervous System
THUMS	Total Human Model for Safety

1 Introduction

The amount of fatalities in vehicle accidents has steadily decreased due to the on-going work of understanding the mechanisms of, and reducing, high loads inflicted on a vehicle occupant in vehicle accidents. However, there is still much to be learnt concerning low-load injury inducing mechanisms. Chronic pain caused by vehicle accidents has become a growing concern, leaving many vehicle accident victims with a crippled life caused by an untreatable and sometimes undetectable injury (Örtengren et al., 1996). A large part of injuries causing chronic pain are inflicted in accidents with comparable low loads, far below the expected chronic injury inducing load levels (Örtengren et al., 1996). An essential part of reducing chronic injuries caused in a vehicle accident is therefore improving the ability to understand low load injury inducing mechanisms. The cause of low load injuries is not only external forces in a vehicle accident but could also be the forces generated by the muscles in the body (Iwamoto et al., 2009; Choi et al., 2005). The ability to model muscle activities will therefore greatly contribute to the validity in simulations of humans in vehicle accident. The addition of pre-crash intervention systems in accident simulations also gives a demand to keep a biofidelic response throughout the duration of the simulation. One way of keeping biofidelity is to include muscle activity in the Human Body Models (HBM) used to simulate vehicle occupants. HBMs with active muscles added are referred to as Active Human Body Models (AHBM).

At the Vehicle Safety Division at Chalmers University of Technology, a project has resulted in an HBM with muscle activity (Östh, 2010). Muscles controlling posture in the sagittal plane for lumbar-, cervical and head rotation was implemented on the Total HUman Model for Safety v3.0 (THUMS; see Section 2.1). A feedback control algorithm that determines the level of muscular activity based on the angle of the body region has been developed. During validation of the control algorithm an elbow model constituting of parts of the THUMS upper body, especially the right arm was constructed (see Section 4). The feedback loop used in the subroutine incorporates models for the neural delay of somatic reflexes (Section 2.2.3; Section 4.1.2) as well as a model for the dynamics of muscle activation mechanisms (Section 2.2.2; Section 4.1.4). The feedback is a proportional-, integrative- and derivative (PID) control algorithm (Equation 4.10) commonly used in industrial control applications for its stable and customizable performance. The implemented control algorithm monitors the angle of a certain body region and calculates an activation level for the muscles controlling flexion and extension for that region accordingly (Section 4.1; Östh, 2010). The AHBM resulting from the work by Östh will from here on be referred to as the Chalmers Active Human Body Model (Chalmers AHBM). The THUMS HBM and Chalmers AHBM is modelled using the Finite Element (FE) software LS-DYNA. LS-DYNA is a widely spread FE computational solver software developed by Livermore Software Technology Corporation (LSTC, Livermore CA). The feedback loop developed by Östh is in this report referred to as the uctrl1 subroutine.

The developed method presented in this report, called the Load Curve Method (LCM), is a result of breaking down the programming for the muscle activity of the Chalmers AHBM and reviewing keywords in LS-DYNA able to perform the same functionality. The LCM relies on the functionality provided by defining mathematical functions and operations as load curves by the keyword *DEFINE_CURVE_FUNCTION (LSTC, 2012a). For differentiation and integration, when not provided by load curve functions, the ability to define and monitor the placement, velocity and acceleration of nodes is used. Differential Equations have been given a mechanical interpretation as defined by kinetic relations based on a mass-damper system (Lennartsson, 2002).

1.1 Aim

The aim of this work has been to investigate if it is possible to implement a control algorithm using the programming language and functionality in the conventional FE software LS-DYNA. The work has used the algorithm developed by Jonas Östh at the division of Vehicle Safety at Chalmers University of Technology. The algorithm is used in the modelling of muscular activity in AHBMs for vehicle accident simulations.

The validation of the controller model using standard LS-DYNA keywords was made through a comparison with the control algorithm implemented in the, FORTRAN programmed and user defined, uctrl1 subroutine to LS-DYNA. The control algorithm as a result from this work will be able to be used in exchange of the subroutine to give activation level for the muscles in the Chalmers AHBM.

1.2 Limitations

The work will not involve any modification to the THUMS-model or to the muscles added in the work by Östh (2010). No change is made to the Chalmers AHBM other than to convert the controller for muscle activity into standard LS-DYNA keywords. This means that the same muscle geometry and muscle strength model is used as in previous work. Since the control functions of the present AHBM are restricted to motions in the sagittal plane, the same limitation is applied for validation of the developed model. The result of the work is implemented in the elbow model and in the Chalmers AHBM trunk model. The model resulting from the work is not validated against data from in vivo human experiments but only against the performance of the uctrl1 subroutine implementation of the control algorithm. Work was carried out at Chalmers SAFER Lindholmen using LS-DYNA version 9.7.1, releases 5.1.1 and 6.1.0 and double precision. The project involves calculating a level of muscle activity in accordance with previous work (Östh, 2010).

2 Background

HBMs as defined for this thesis work are used in computer simulations of vehicle accidents. The reason to use numerical HBMs is a continuation of the work done in simulating load levels inflicted on the human body in vehicle accidents. Previously such tests could only be done using real cars in various accident-like conditions where the human body was estimated by using Anthropomorphic Test Devices (ATDs) (Fressmann et al., 2007).

2.1 Finite Element Human Body Models

Due to computers becoming increasingly powerful and, amongst other, a wish to increase biofidelity of the HBMs, research has been done in building numerical models of the human body. Two examples of numerical HBMs are multi-body HBMs (MB-HBMs) and finite-element HBMs (FE-HBMs). An MB-HBM is constructed using rigid, non-deformable parts in a deformable structure. FE-HBM is modelled using deformable elements in a deformable structure which results in a greater level of biofidelity but also increased computational time. By using FE calculations, complex geometries and load paths can be simulated in great detail. The THUMS FE-HBM was developed based on ex vivo data by the Toyota Research and Developments Labs (Iwamoto et al., 2002). Validation was done for each part of the body and the model was compared to the Hybrid III ATD (Fressmann et al., 2007). The THUMS model incorporates several hundreds of thousands of elements with the goal of being able to, in great detail, predict the loads exerted on the human body. One aspect of further increasing bio-fidelity in simulations has been the inclusion of muscular tissue and muscular activity in HBMs, see Section 2.3 and Section 3. The modelling of muscle activity can primarily be divided into predetermined level of activation and an activation level based on feedback from sensory information, see Section 2.3.

2.2 The Musculoskeletal system

The musculoskeletal system plays a vital part in the morphology and functionality of the human body. It serves three main purposes; to maintain the posture, to provide mobility and exertion of power, and to protect vital organs from injury.

Neuromuscular control is performed by a complex system constituted of nerve pattern pathways and execution control regimes located in the brain, the spinal cord and throughout the body. Central to the challenge of implementing active muscular control is the amount of neural excitation and activation to each muscle depending on time and sensory input (van Leeuwen and Spierts, 2002). A brief introduction to skeletal muscles and neuromuscular control is given.

2.2.1 Morphology of the skeletal muscles

A cross-Section of a skeletal muscle reveals a repeated pattern of groups and subgroups of muscle fibres surrounded by a connective tissue as seen in Figure 2-1. Skeletal muscles are part of the striated muscle group which is signified by the lengthwise repeated constitution of sarcomeres, the basic contractile unit of the striated muscles. Several sarcomeres stacked around and after each other form the myofibril. The myofibril together with the connective tissue called sarcolemma and a surrounding tissue sheath called endomysium form the muscle cell, also called muscle fibre. Several muscle cells, surrounded by the perimysium, a connective tissue sheath, form a fascicle which in turn is surrounded by the epimysium. Surrounded by the fascia and several other collagenous, reticular and elastic fibres, several fascicles form a muscle (Epstein et al., 1999).

According to the sliding filaments model, the tension and contraction generated in a muscle comes from the activity in the sarcomere, more specifically the interplay between the thick filament and the thin filament (Marieb and Hoehn, 2010). The filaments are primarily built up of the proteins myosin, actin, troponin, tropomyosin and titin. In the thick filament several myosin molecules group together by attaching alongside each other to form a kind of stem from which upon globular heads are situated, like several stumps of branches on a tree (Epstein et al. 1999). The globular head is often referred to as a cross-bridge. The thin filament surrounds the thick filament in an isotropic manner. It constitutes of two chains of serially linked actin globules. In between the actin globules are strands of tropomyosin and upon these twisted strands lay troponin molecules. Titin act both as a spring and as a structural element to keep the thick filament in place and centred between the thin filaments during sarcomere elongation (Epstein et al., 1999; Marieb and Hoehn, 2010).

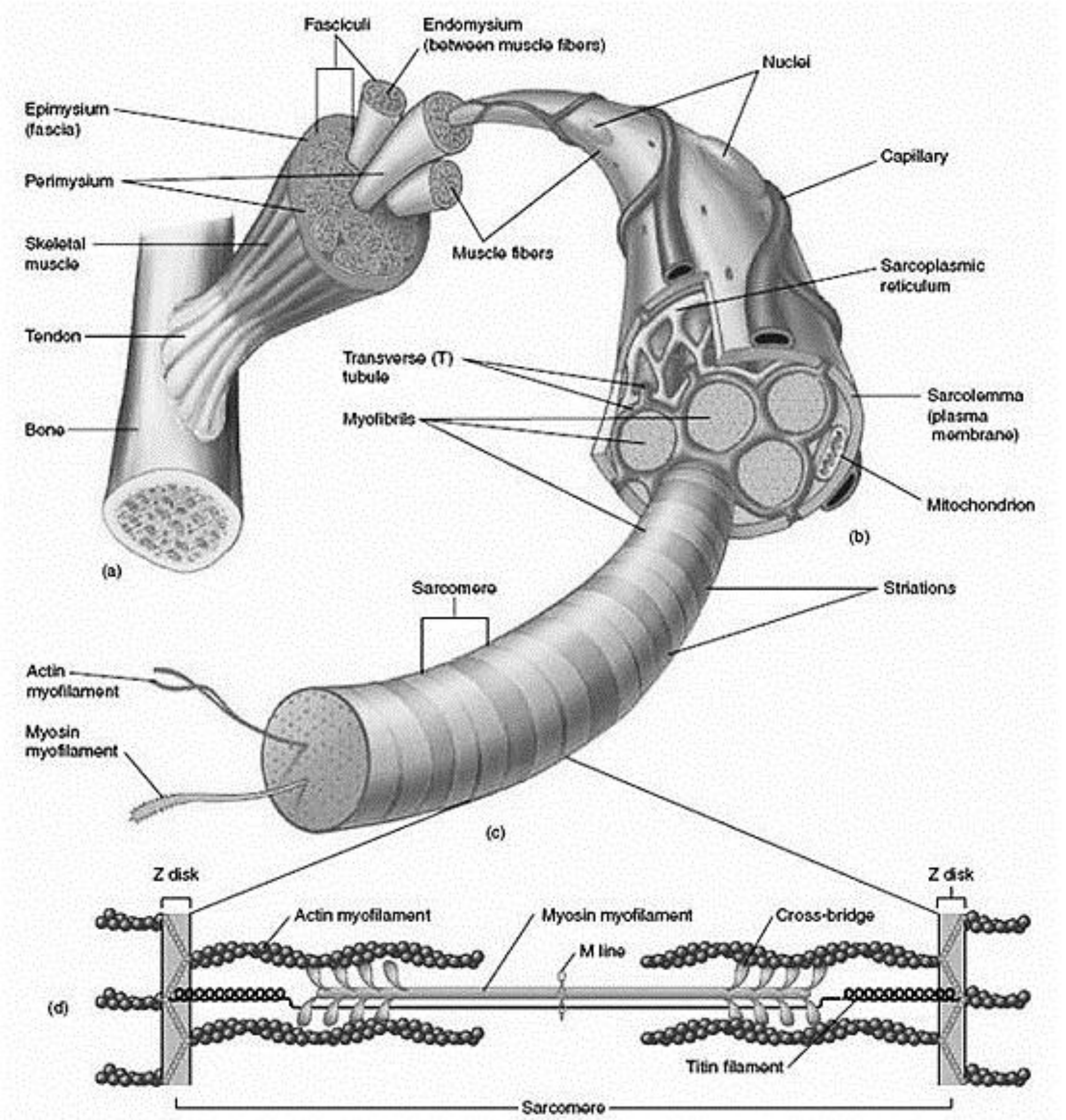


Figure 2-1 Morphology of a human striated muscle showing the constituents of a skeletal muscle down to the smallest contractile unit, the thin and thick filaments of the sarcomere. Image adapted from http://damiane.wikispaces.com/11+Muscle+Physiology.

The filaments are fitted into a structure built up of protein. The thick filament is in the M-line interlinked by accessory proteins, while there is a gap at the same place of the thin filament. The opposite end of the thin filament is in the z-disc attached to a rigid structure. The distance between the thin filament and the M-line as well as the distance between the thick filament and the z-disc stipulates the margin necessary to allow contraction (Marieb and Hoehn, 2010; see Figure 2-1). The tension and contraction is generated by chemical reactions that affect the proteins in the filaments. The cross-bridge in the thick filament bonds with the actin in the thin filament. At the same time energy is released, that causes the cross-bridge to move the thin filament towards the M-line (Epstein et al., 1999). The cross bridge detaches and gets ready to perform a new bonding with actin. These processes provide the basis for either a contraction of the muscle or a static rigidness (Marieb and Hoehn, 2010).

2.2.2 Neuromuscular control

The overall appearance of the sensory-neuromuscular system is that of a feedback loop where activities in the brain constantly try to adapt every motion to the current situation (van Leeuwen and Spierts, 2002). An action conducted by the musculoskeletal system also relies on feed forward control in the form of learned behaviour, but the feedback loop can improve the control of the action if the result is not what was expected. The musculoskeletal system is an undetermined set of variables and Equations, since several muscles can be connected to control a single joint movement (Marieb and Hoehn, 2010).

The Central Nervous System (CNS), where integration of sensory input and commands for control output are generated, consists of the brain and spinal cord. The communication from the CNS to the body is carried out by the Peripheral Nervous System (PNS) which consists of cranial- and spinal nerves. Somatic and visceral sensory nerve fibres are part of the sensory (afferent) division and connect to the PNS. The somatic nervous system conducts commands from the CNS to skeletal muscles, and is part of the motor (efferent) division which in turn is part of the PNS (Marieb and Hoehn, 2010).

Muscle fibres in skeletal muscles are activated by somatic motor neurons, part of the PNS, which via an axon attach to the muscle fibre at the neuromuscular joint. The muscle fibres are grouped into motor units, and each unit is controlled by a motor neuron (Epstein et al., 1999; Marieb and Hoehn, 2010). Apart from the extrafusal fibres, the muscle fibre where the tension and contraction is generated, there are intrafusal fibres that form the muscle spindle and act as a sensory part. Muscle spindles are embedded in between the extrafusal muscle fibres and register the elongation and rate of contraction of the muscle fibre. Golgi tendon organs register the tension in the tendons attached to the muscle in order to limit the overall stress of the muscle and protect the tendinous tissue it is attached to (Epstein et al., 1999; Marieb and Hoehn, 2010). The number of muscle fibres in each motor unit depends on what purpose each unit has. Units with a large number of fibres are slow to react on a nerve impulse but are able to produce a high force. Smaller units react quicker to the nerve impulse but are not able to deliver as high amount of force as the larger groups. This distribution of unit size, among other parameters, makes it possible for the muscle to have a smooth response to the task it needs to perform and do it in an energy efficient way (Epstein et al., 1999).

The muscle response to a single action potential can be separated into the latent period, the period of contraction and the period of relaxation. Muscle activation dynamics (the latent period up until being fully contracted) are governed by the mechanisms initiated by the action potential at the neuromuscular joint (Marieb and Hoehn, 2010). These mechanisms control the flux of potassium, K^{+} and sodium, Na^{2+} ions in and out of the muscle fibre membrane in response to impulses from the motor neuron. The flux of K^{+} and Na^{2+} ions causes an action potential to propagate along the muscle fibre which in turn initiates a release of calcium, a^{2+} , ions throughout the muscle fibre. The release of calcium ions into the sarcomeres initiates the chemical reactions involved in the sliding motion of the filaments in the sarcomere (Marieb and Hoehn, 2010). During the first milliseconds after stimulation, due to the activation dynamics, tension builds up but do not cause any noticeable movement. After these first milliseconds the cross-bridge activity increases, causing contraction, and lasts from 10-100ms. After the peak of contraction the muscle enters a period of relaxation for 10-100ms (Marieb and Hoehn, 2010). The neural control signal is only binary but the amount of tension in a muscle fibre is controlled by the amplitude and frequency of the neural signal. The latency of the muscle fibres reacting to a neural impulse makes it possible to superimpose the response of a series of neural signals. This averts the muscle response from being binary but being smoothly controlled over time in amount of tension and contraction (Marieb and Hoehn, 2010).

The muscle can contract in a number of ways, depending on load and purpose. An isotonic contraction occurs when the load is small enough for the muscle to be able to move it. In contrast, an isometric contraction occurs when the load is greater than the maximum force the muscle is able to generate, thus the muscle can´t contract. An isotonic contraction can either be concentric, thus shorten the muscle while producing force, or an eccentric and produce force while the muscle lengthens. During an eccentric contraction the muscle is able to produce more force compared to a concentric contraction (Marieb and Hoehn, 2010).

2.2.3 Reflex activity

Reflexes concerning skeletal muscles are called somatic reflexes and if they only involve the spinal cord, are called spinal reflexes. Reflexes can be either inborn or learned but the basic concept of a reflex is to perform a motor response where higher CNS activity is unnecessary (Marieb and Hoehn, 2010; van Leeuwen and Spierts, 2002). Actions driven by reflexes can for example be: breathing, staying upright, avoiding a suddenly appearing obstacle or to some extent driving a car (Pick, 2004; Gandevia and Burke, 1992). A reflex is the shortest neural reaction to a certain stimuli. Via the integration centre in the spinal cord, the motor neuron causes the muscle to react on impulses generated by a receptor, such as stimulus from the skin, transmitted by the sensory neuron to the spinal cord, the pathway of the signal is also known as the reflex arc (Figure 2-2; Marieb and Hoehn, 2010). This does not mean that a reflex cannot be intervened and altered by a higher brain activity. In an acute situation the primary stimulus is often followed by several other stimuli which may add up in complexity so that higher brain activity is inevitable in order to adapt, facilitate or inhibit the primary reflex signal response (Marieb and Hoehn, 2010; van Leeuwen and Spierts, 2002). In the same manner, local feedback loops may intervene on the nerve signals coming from the CNS, acting on impulses from the Golgi tendon organs or afferent fibres situated in proximity of the actuated muscle (Epstein et. al, 1999). Studies conducted on living human subjects concerning cervical musculatures reaction to sudden perturbation made to the head concluded a mean reflex time for flexor and extensor muscles in the cervical region of 55-90 ms depending on gender and age (Wittek, 2000). Experiments made on multijoint arm movement concluded a neural delay of 34ms for muscles surrounding the elbow joint (de Vlugt et al., 2003). The results are in line with the works done by Matthews (1994) where the frequency response of human stretch reflexes was studied by using surface electromyography to measure level of activation in biceps brachii and human abductor digiti minimi . The stretch varied from 0.5 to 2mm for abductor digiti minimi and 0.1 to 1 mm for biceps brachii with a frequency ranging from 10Hz to 50Hz. The resulting measured phase lag and amplitude modulation is, because of placement of electromyographic measurments, over the main belly of the muscle, a combination of neural delay, muscle neurochemical activation dynamics and the activation scheme of different muscle fibres in the muscle. Depending on site of measurement the jerk response time could differ by 10ms. Matthews concluded a reflex time of 30ms for biceps brachii and 60ms for abductor digiti minimi (Matthews, 1994).

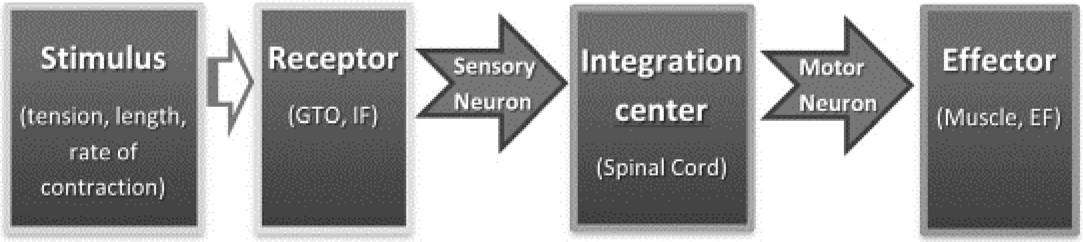


Figure 2-2 Schematic represention of a reflex arc from stimulus to effector (based on information taken from Marieb and Hoehn, 2010). GTO, IF and EF is abbreviations for, respectively, golgi tendon organs, intrafusal muscle fibres and extrafusal muscle fibres.

As with general human motion, much of the primary behaviour when driving a car is done with either inborn or learned reflexes. Sitting upright and keeping your hands on the steering wheel are controlled by adapted inborn reflexes (Pick, 2004). Every move needs to be coordinated between several different muscles and even different motor units within muscles in order to be able to handle the driving tasks. Still the coordination between the muscles referring to posture of body and limbs is no conscious action, but only initiated by a higher motor command (van Leeuwen and Spierts, 2002; Pick, 2004).

2.3 Biomechanics of the Musculoskeletal System and Neuromuscular Control

Biomechanics deals with the mechanical modelling of the human body and human motion. It concerns physical, mechanical, kinetic and kinematic properties in order to understand and numerically explain the human body and human motion. It is a fundamental part in trying to implement computer simulations of how the human body moves and responds to the forces exerted on it, for instance in a car accident.

2.3.1 Hill muscle model

The Hill muscle model was a result from thermodynamic experiments conducted by A.V. Hill in the 1930’s (Hill, 1938) when trying to establish principles concerning the energy relations in muscles. Though several mathematical approaches for modelling the force exerted by a muscle exist, because of the relative simplicity of the model it is still widely used as a model for muscular force based on the level and rate of contraction. The main criticism of the first proposed Hill model has been its inability to replicate the transient behaviour of muscles and the lack of detail when trying to determine the force in separate parts of a single muscle (Winters, 1990). However, for modelling of movement with multiple numbers of muscles, the Hill model is still frequently used (Williams, 2011; Winters, 1990).

The properties of the sarcomeres are modelled with a contractile element (CE) in conjunction with two non-linear elastic models. The passive properties of the muscle fibres and surrounding tissues are modelled with a parallel damping element (DE) while the elastic properties of the tendon are modelled with a parallel elastic element (PE) (Wittek, 2000; Winters, 1990, see Figure 2-3).

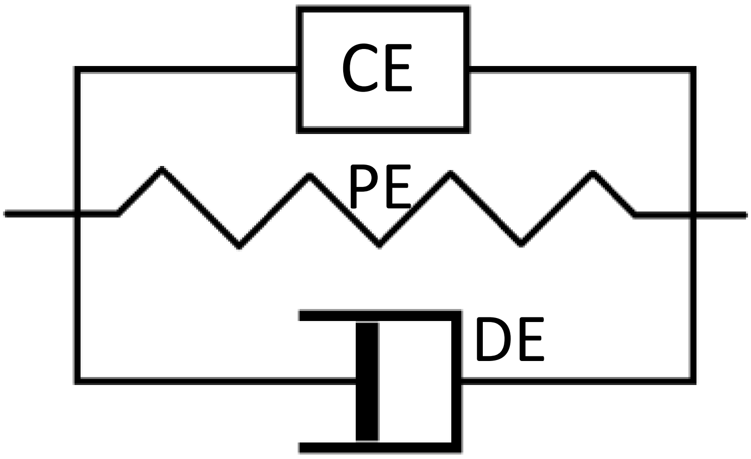


Figure 2-3 Schematics of a Hill-type muscle model as studied by Wittek (2000).

The currently exerted muscle force is determined by the properties of the passive element as well as the current length of the muscle, the muscle activation level and the current rate of contraction, see Equation 2.1. The force-length relationship of the Hill-model stipulates that the maximum force possible by the muscle depends on the elongation of the same, due to the possible number of actin-myosin connections decreasing at the extreme ends of muscle elongation. Though the relationship can alter on individual basis and even alter in one individual depending on exercise level or physical aspects. Thus, in order to mathematically and accurately determine the force-length dependency, the lengths of the thick and thin myofilaments of the muscle fibre concerned must be known (Epstein, et al., 1999). The Hill muscle model has been further developed and used in the works of Winters and Stark (1985) and Zajac (1989). Both in terms of improving the Equations for the force-length, the force-contraction rate relationships and the inclusion of activation dynamics into the Hill proposition of active state (Zajac, 1989). In LS-DYNA a Hill-type approach towards calculating muscle force as described in this Chapter are used in the material model nr. 156, *MAT_MUSCLE (LSTC, 2012b). The muscle properties for the contractile element, the force-length and force-rate of contraction relationships are implemented as dimensionless and normalized. The force vs. length relationship is in *MAT_MUSCLE identified as a tensile stress vs. stretch ratio relationship, see Figure 2-4. The tensile stress vs. strain rate is equivalent to the relationship between force vs. rate of contraction (Figure 2-4).

The muscle force in the Hill muscle model is based on the dynamic active and passive response of the muscle (Wittek, 2000). As described the tension in the CE depend on force-length, force-rate of contraction and activation state, see Figure 2-4. In addition to this the tension is limited by the maximum tension exerted by the muscle, see Equation 2.1-2.5:

$$\sigma_c = N_a(t) f_l(l) f_v(v) \sigma_{max} \tag{2.1}$$

$$f_l(l) = e^{-\left[\left(\frac{l}{l_{opt}}-1\right)/C_{sh}\right]^2} \tag{2.2}$$

$$f_v(v) = \begin{cases} 0 & v \leq -1 \\ (1+v)/(1-v/C_{sh}) & -1 < v \leq 0 \\ \left[1+v\left(C_{mvl}/C_{lng}\right)\right]/\left(1+v/C_{lng}\right) & v > 0 \end{cases} \tag{2.3}$$

In addition to the active properties, the passive kinetic properties of muscle tissue are modelled with the PE and calculated as:

$$\sigma_{pe}(l) = \{\sigma_{max}/[e^{C_{pe}}-1]\}\left\{e^{[(C_{pe}/PE_{max})(l/l_{opt}-1)]}-1\right\} \tag{2.4}$$

$$\sigma_d(v) = \left(D/l_{opt}\right)v \tag{2.5}$$

Where,

$f_l(l)$	Hill force-length relationship
$f_v(v)$	Hill force-velocity relationship
l	Instantaneous muscle length
$N_a(t)$	Muscle activation level
$\sigma_{pe}(l)$	Stress Equation for parallel elastic element
$\sigma_d(v)$	Stress Equation for parallel damping element
v	Instantaneous muscle velocity/rate of contraction
cc	Muscle cocontraction level
C_{mvl}	Asymptotic value for increased eccentric muscle lengthening
C_{lng}	Transition between concentric and eccentric muscle shortening
C_{pe}	Passive elastic muscle stiffness
C_{sh}	Shape of rate of contraction curve for concentric shortening
D	Passive damping muscle stiffness
l_{opt}	Optimal muscle length for maximum isometric contraction
PE_{max}	Maximum force from the passive elastic element, measured at l_{opt}
σ_{max}	Maximum isometric contraction
T_f	Lowest level of muscle excitation

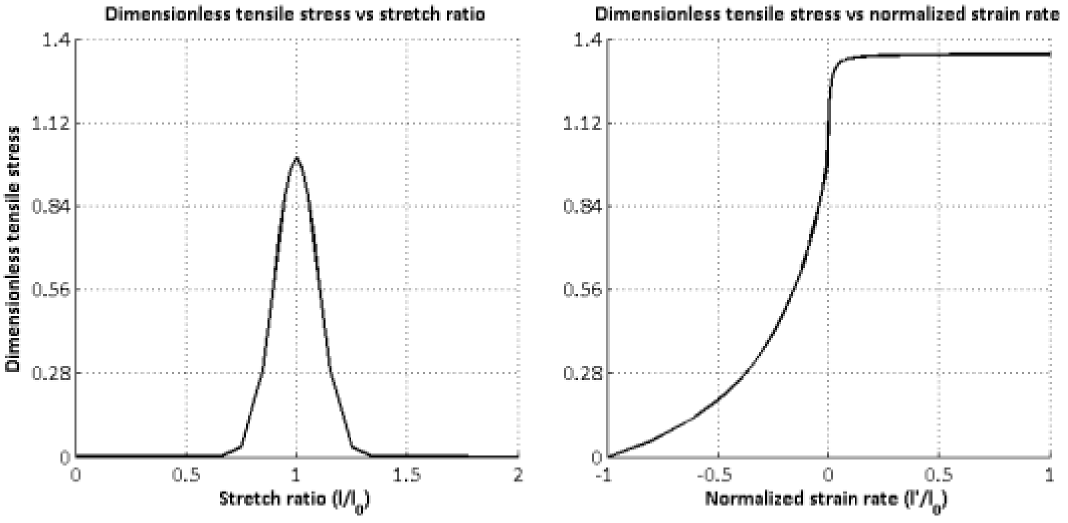


Figure 2-4 Force vs length and force vs rate of contraction equivalent curves as used for Biceps Brachii long head muscle in the Chalmers AHBM elbow model.

2.3.2 Feedback Control Models of CNS Activity

As stated in Section 2.2.2. The CNS relies on feedback control by using sensory information in the process of determining movement of the human body. It plays a vital role in the ability of the human body to act and react in relation to the current environment (Marieb and Hoehn, 2010; van Leeuwen and Spierts, 2002). Also to note is that the parameters in the feedback system show signs of non-linear properties, thus ruling out linear control theory, according to van Leeuwen and Spierts (2002). Several models exist in trying to determine the operational features of the CNSs feedback regime. The main parameters the models try to establish is which sensory information should be used and what approach to use in amplifying the sensory feedback information before activating the muscles (Kistemaker et al., 2013; Feldman and Levin 1995; Winter, 1995; de Vlugt, 2003). The equilibrium point theory builds on the idea that muscular feedback control can be modelled by the use of equilibrium positions as reference points in limb movement. The level of muscular force is generated with a spring-like behaviour in proportion to the distance between actual state and reference state (Feldman and Levin, 1995; Latash 2011). It has been widely used in biomechanics when modelling movements of slow to moderate speed. However, it is not able to accurately model the exact behaviour of limb movement for rapid movement (Bizzi et al. 1992). Modelling rapid movement involves increasing the gain in the feedback control or adding parameters in the control algorithm. When modelling rapid movement, several models have included the integrative- and differential part of the control signal apart from the proportional gain (de Vlugt et al., 2006; Kou 2005). The notion of control parameters other than proportional may be seen for example in the ability in GTOs to perceive not only the amount of force in the tendon but also the length of the muscle the tendon is attached to during contraction (Gandevia and Burke, 1992). There has also been work done in combining feedback- and internal models (Stroeve, 1997; Gandevia and Burke, 1992; Bamford and Marteniuk, 1988) in order to model both in-learned behaviour as well as sensory feedback based behaviour.

2.3.3 Internal models of CNS activity

Internal models deal with modelling in-learnt behaviour and knowledge, gained by the CNS through experience, concerning properties for human body movement (Kawato, 1999). The concept of internal models comes from the need to accurately model rapid and coordinated body movements. It is stated that the gain in feedback control models is too low to replicate such movements (Kawato, 1999; Gerdes and Happee, 1994). According to Haith and Krakauer (2013), in their work concerning models of the CNS learning behaviour, the rapid response in human body movement is due to forward models predicting future positions of body parts and the muscular actions involved in making it happen. The source of rapid movements can therefore not come from a response in a feedback model but be based on inborn or learned forward models. Experiments have shown that evidence supporting the internal model theory can be found in the neural activity of the CNS concerning movement (Kawato, 1999; van Leeuwen and Spierts, 2002; Kuo, 1994). Also, experiments have shown that musculoskeletal movement can be generated in parts of the body without the presence of a feedback loop (Kawato, 1999; van Leeuwen and Spierts, 2002). The construction of internal models in the CNS through a learning period can be studied by altering the force field in which a certain movement task is to be performed. In an established force field through iteration the CNS adapts and movements become progressively smoother and more accurate for each iteration. However, if the force field is changed, a new series of iterations is needed in order to achieve the same amount of smoothness and accuracy of movement as before (Kawato, 1999; van Leeuwen and Spierts, 2002; Bamford and Marteniuk 1988). There is still some uncertainty in what paradigm is used as means of improvement of the movement, whether it be reducing jerkiness or cost in energy or some other parameter (van Leeuwen and Spierts, 2002; Kawato, 1999; Kuo, 1994). Internal models may involve the concept of inverse dynamics as well as neural networks. Through inverse dynamics the amount of muscular force to be applied is calculated based on knowledge of mechanical properties of the body and the proposed trajectory of the movement (Kawato, 1999).

3 Literature review - Muscle Control in Human Body Models

In order to accurately model occupant movement in pre-crash situations and as a means to increase the bio fidelity of HBMs, the need to incorporate muscle elements and its activity has emerged. The field of biomechanics and its models for muscular activity provide strategies and techniques for modelling of human body movement. What differs is the need to mostly model the muscular response on rapid changes on the force field resulting in forces and moments of inertia acting on the body that is normally not inflicted upon it.

3.1 Predetermined Muscle Activation Level

Predetermined activation level builds on the notion of internal models of CNS activity. It also provide means to implement muscle activation in HBMs without having to in real time model CNS activity. In order to capture rapid movement of the human body, activation levels for the muscles are defined prior to the simulation. The methods studied are optimization algorithms, simulating the learning protocols of the CNS (Section 3.1.1), electromyography-data from in vivo human subjects experiments (Section 3.1.2) and muscle activation level based on a validated trajectory of the controlled body region (Section 3.1.3).

3.1.1 Internal Model based Muscle Activation

Iwamoto (Iwamoto et al., 2012) used the THUMS and reinforcement learning written in C++ code, a neural network optimization technique in order to reduce either usage of time or energy. The approach demands several simulations for each run in which the model learns how to move according to some parameter set on a general basis or towards some validation data. Their muscles were modelled with passive and active force-elongation curves and force-velocity curves. 25 neck muscles were implemented with 104 muscle bar elements in a test setup. For each step, after calculating which muscle to contract and using the muscle curves to calculate the force exerted by the muscle on the joint, LS-DYNA was called to calculate the energy and resulting displacement in the nodes of the model. The underlying theory is that this reinforcement learning is able to simulate the behaviour in the basal ganglia where posture control functionality is situated. The outcome of this approach is said to be a muscle activation function that can be used for a multiple type of impact situations. Once muscle activation functions are determined, computational time is decreased by using pre-determined activation curves in the simulations. The model tries to reduce the overall energy consumption as a scheme to handle the muscle redundancy around body joints. Through iteration the energy consumption is optimized and in a way tries to replicate the learning scheme that the human mind and body uses in order to design appropriate activation patterns for certain tasks.

Choi et al. (2005) describes an incorporation of active muscles in the H-model, an FE-model similar to THUMS (Choi et al., 2005). The implementation was done with Hill-type bar elements where the overall level of activation was partly determined from volunteer experiment data of joint torques and partly determined via an optimization algorithm when muscle redundancy was present. The optimization algorithm was to determine the likely distribution of muscle forces by minimizing the active muscle energy for static equilibrium. The authors conclude that the approach as stated works "quite well" but that further research and implementation of a dynamic muscle activation model is needed.

3.1.2 Electromyography based Muscle Activation

An approach, studied and validated for the upper arm of the THUMS model, implements established force-length and force-velocity relationships for Hill-type muscle elements (Iwamoto et al., 2009). Proper level of muscle activity in the model was set according to validation data. In addition to the active implementation, passive solid elements with rubber-like material models were implemented. Based on experiments suggesting that muscle activity increased the risk of injury on lower and upper extremities (Choi et. al, 2005), the model was built in order to investigate whether this relationship could be detected in simulations. The validation was done by mainly recording the properties of biceps brachii in human experiments. Properties investigated were stiffness versus muscle activity and elbow flexion versus muscle activity on a 50th percentile male human subject. The model was used in simulations of lateral impact with and without muscle activity in order to simulate bracing behaviour of a car driver. The conclusion was that the FE model showed signs to indicate that bracing significantly affects the behaviour of the arm and increase the risk of injury.

In an attempt to model active- and passive- mechanical and physical properties of the cervical skeletal-musculature, work has been done (Hedenstierna, 2008) at The Royal Institute of Technology in Stockholm, KTH, where the model incorporates both passive solid muscle elements and active discrete muscle elements. The passive and active elements were added to an existing head and neck model. The muscle geometry is based on MRI-scans of a 50th percentile male, and muscle material properties are based on published data concerning human cervical muscles and validated against experimental data from rabbit muscle tissue. The active biomechanics is based on the Hill muscle model, see Section 2.3.1. The elements have been separated from the solid mesh, divided, serially linked and embedded in a parallel pattern in the solid elements. The activation schemes of the muscles were based on electromyography measurements done on volunteers in experiments as part of the validation of the model. Hedenstierna (2008) concludes that incorporating solid muscular elements increases biofidelity but that further research is needed in order to establish a robust model for the level of muscle activation.

3.1.3 Trajectory based Muscle Activation

Nissan car-corporation has been part in developing a finite element human body model with active and passive muscular characteristics (Murakami et al., 2004). Hill-type muscle elements were applied and the active behaviour was modelled by predefined constants that determined the state of muscle activity, the actual magnitude of force produced versus elongation and versus rate of contraction. A sled test setup experiment was conducted on a living human subject. Movement of the head together with accelerometer data were registered for runs where the subjects cervical muscles were perceived to be in a relaxed state. Based on the data, constants for the muscle activity, force versus rate of contraction and force versus elongation functions was determined for the model. Murakami et al. (2004) concludes that the result of the simulations fit within the corridor of data established from the validation runs.

3.2 Feedback Control Models in FE-simulations

The feedback control model is a way to model part of the CNS activity (Section 2.2.2; Section 2.3.2) and based on that model determine the level of muscular activity. Creating a biofidelic model for CNS activity may lead to the ability to accurately simulate human body movement in pre-crash and crash situations. If there is a lack of experimental data, this may improve the accuracy in the simulations of pre-crash and crash situations and in the end improve the safety for the vehicle occupant.

3.2.1 Matlab Simulink coupled to LS-DYNA

Via an interface that couples LS-dyna to Matlab Simulink, active control based on an online connection that updates the muscle force in real-time was presented (Prüggler et al., 2012). The controller was successfully implemented on an FE-model incorporating a single Hill muscle element connected between two parts made up of shell elements representing the humerus and ulna. When simulating more complex structures the Matlab Simulink functionality was substituted by a C++ coded sub-routine. This approach was implemented on a simplified THUMS model and compared with in vivo human subject sled tests with a maximum acceleration of 2g. The presented results are unfortunately inconclusive due to lack of presenting the load case used and lack in describing what control algorithm is implemented.

3.2.2 Multi Body Model with PID-controller for Muscle Activity

Meijer et al. (2008) describe a MB model with added Hill-type muscle elements used in the MADYMO software. In addition to the muscle elements, sensors, controllers and joint torque actuators were added. Posture control is modelled without neural delay in order for posture maintenance to work during short time periods (<200ms). For longer periods of time and for movements not maintaining posture, neural delay is added, 40ms for neck muscles, 70ms for spine and arm muscles and 100ms for leg muscles. The neural delay is also frequency dependent, allowing lower order frequencies to travel faster than higher frequencies. For each actuator there are six degrees of freedom, three translational and three rotational. The control signal is based on the joint angle that passes through a PID control algorithm. The model gives the opportunity to involve other than posture maintenance, by altering the reference state for the joint angle in the control algorithm, so that motions such as the turning of a steering wheel or the pressing of the brake pedal can be modelled (Cappon et al. 2007; Meijer et al. 2012). The model was evaluated based on in vivo human volunteer experiments. The result show that the response of the MB-HBM thorax maximum deflection is less than for the volunteers, but the feedback model is able to maintain posture of the MB-HBM.

4 Chalmers Active Human Body Model

The Chalmers Active Human Body Model utilizes a feedback control approach, controlling Hill-type muscle elements that are implemented in the user control subroutine, uctrl1, in the user material version of LS-DYNA (Östh et al., 2012). A PID-controller is used to generate muscle activation levels, Figure 4-1.

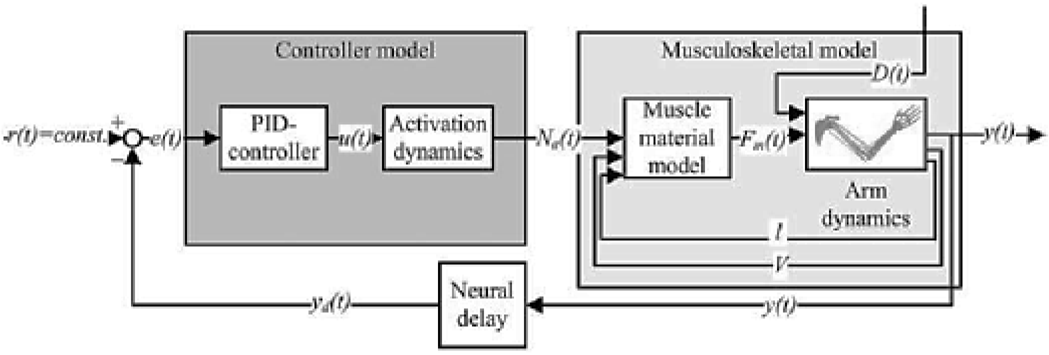


Figure 4-1 Model schematics for Chalmers Active Human Body Model. Figure adapted from Östh et al. (2010) with permission.

The activation model was first implemented on a part of the THUMS model, constituting of the right arm and shoulder complex. The model was used to validate elbow joint flexion and the force exerted by the muscles in the right upper arm (Östh et al., 2012). The model was validated against a 50th percentile living human subject in the validation method denoted Arm Movement and Disturbance Analysis (ARMANDA) developed by de Vlugt (2003; Östh et al., 2011a). Muscular activity was further implemented on the trunk of the THUMS model to give active muscular control and control movement in sagittal plane of the lumbar-, cervical- and head region. Experiments conducted on living human subjects in scenarios involving autonomous braking were used to evaluate the model (Östh et. al, 2011b). Based on the controller output and a muscle activation dynamics model activation levels are provided to flexor and extensor muscles.

For the control of the head angle, the angle of a vector between the head centre of gravity and the occipital condyles and a vector spanning the cervical spine is used. Muscles are implemented using beam elements and a Hill type muscle material model and are grouped according to the body part on which they act. The initial angular relationship between the nodes acts as the reference state for the system. The model allows for a delay to be imposed on the control signal that models the neural delay present in the CNS concerning somatic reflexes when retrieving and processing sensory information (de Vlugt et al., 2003). Delays are also present in the activation dynamics of the muscles that models the neurochemical processes described in Section 2.2.2.

Time constants have been defined based on experimental data (Östh et al., 2011b). The files needed to run the model is separated depending on functionality. The THUMS human body model is used to model the geometry and mechanical properties of the 50th percentile male body. A more detailed representation of the various steps in the uctrl1 subroutine can be found in Appendix B.

4.1.1 Calculating the angle of a joint

The measured body region angle, $y(t)$, is calculated through vector algebra. For the elbow model two vectors are defined, as shown in Figure 4-2. For the trunk model three vectors are defined based on the location of chosen nodes to control rotation in the sagittal plane for the lumbar-, cervical- and head region (see Table 4-1, Table 4-2 and Figure 4-3).

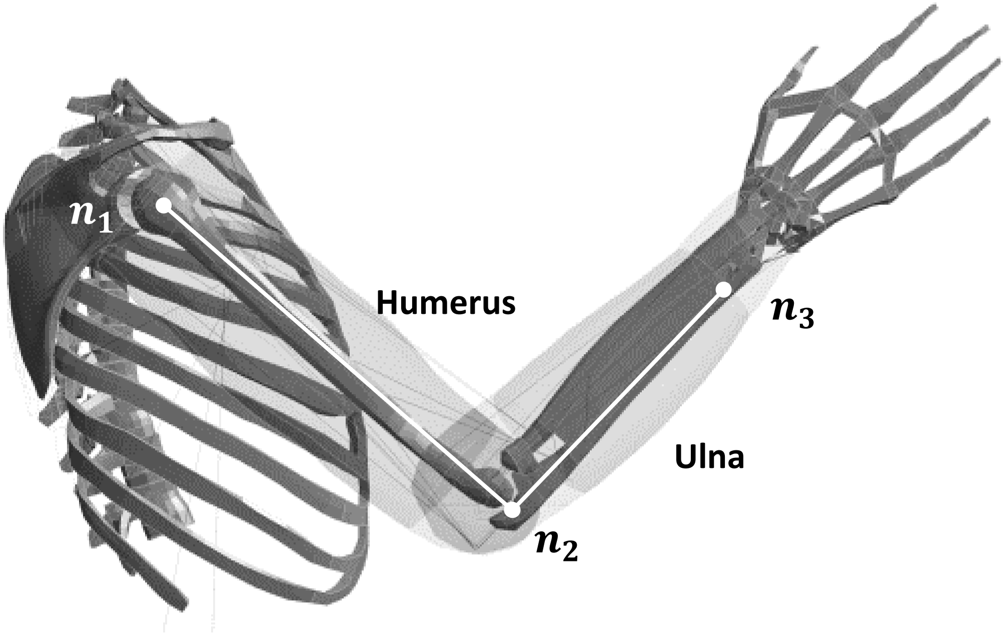


Figure 4-2 View on the right side of the elbow model showing placement of the nodes and vectors used in calculating the elbow joint angle.

The calculations for $y(t)$ can be seen in Equation 4.1-4.6 for the elbow model and 4.7-4.9 for the seated trunk model. In the elbow model the angle is calculated as the angle between two vectors representing the humerus and ulna of the right arm.

$$U_{hum} = \left(n_{1x} - n_{2x}\,, n_{1y} - n_{2y}\,, n_{1z} - n_{2z}\right) \tag{4.1}$$

$$U_{uln} = \left(n_{3x} - n_{2x}\,, n_{3y} - n_{2y}\,, n_{3z} - n_{2z}\right) \tag{4.2}$$

Notation of the nodes in the elbow model, defining the humerus and ulna vectors in Equation 4.1-4.2 is,

$$8540955 = n_1 \tag{4.3}$$

$$8530167 = n_2 \tag{4.4}$$

$$8530319 = n_3 \tag{4.5}$$

the vectors are defined in the global coordinate system of the elbow model and the angle of the elbow joint is calculated according to,

$$y(t) = \cos^{-1}\frac{U_{hum} \cdot U_{uln}}{|U_{hum}||U_{uln}|} \tag{4.6}$$

For the trunk model, vectors are only defined in the sagittal plane. The vectors are defined based on the location of five nodes, the angle is calculated by defining a vector between two nodes and calculating its angle to the global vertical axis (Equation 4.7 - 4.9, Figure 4-3 and Table 4-1). The nodes are for each region used in the calculations according to table 4-2.

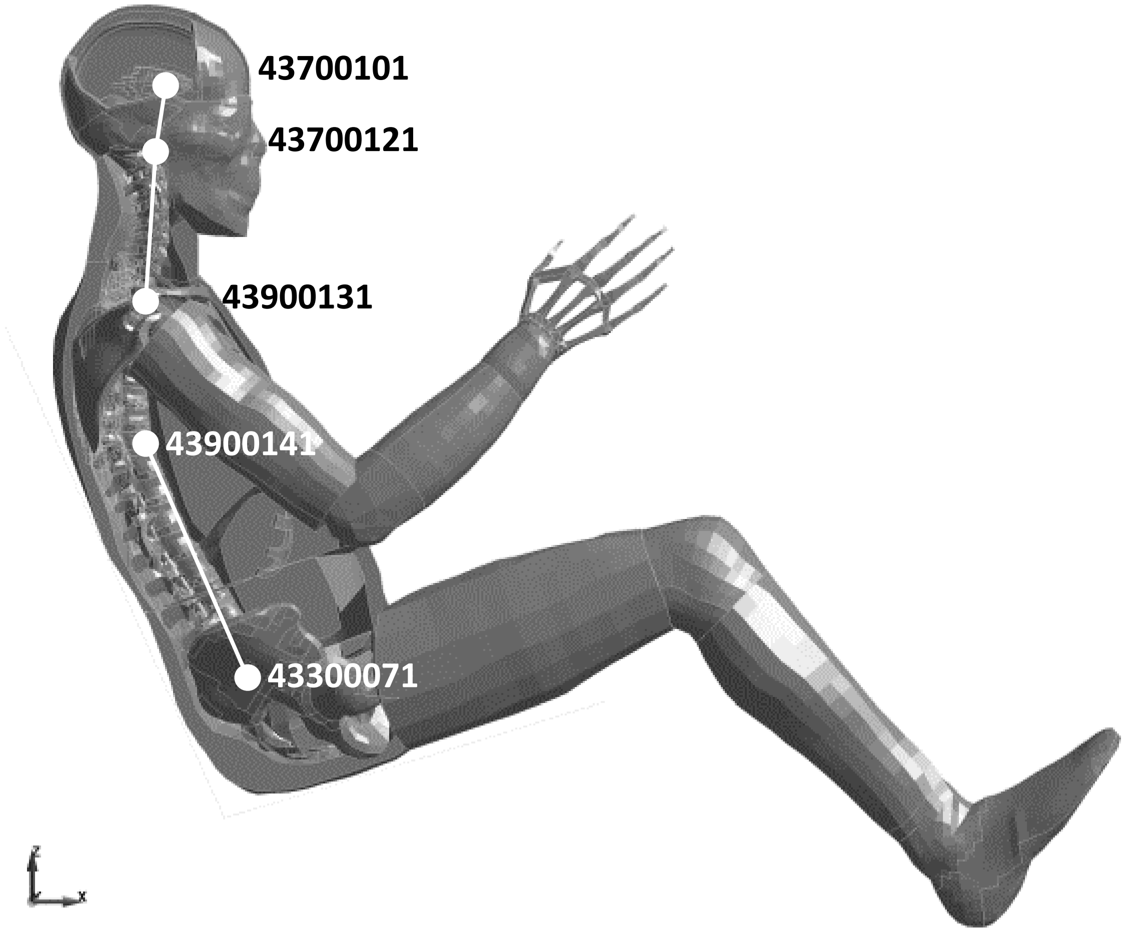


Figure 4-3 Schematic view of the placement of the nodes and vectors used to calculate the lumbar-, cervical- and head rotation. The nodal point in the thorax region is only used to calculate rotation of the lumbar region.

$$U_x = n_{1x} - n_{2x} \tag{4.7}$$

$$U_z = n_{1z} - n_{2z} \tag{4.8}$$

$$y(t) = \tan^{-1}\left(\frac{U_x}{U_z}\right) - y_0 \tag{4.9}$$

Where,

$y_0 =$ Initial angle between the vertical axis and the body region vector.

Table 4-1 Listing the location of the nodes used for the trunk model to define the angle of a body region. See Figure 4-3

Nodal number	Location
43300071	Center of Sacrum
43900141	Center of T10
43900131	Center of T1
43700121	Center of Atlanto-Occipital Joint, on top of C1
43700101	Head center of gravity (HCoG)

Table 4-2 Notation and the nodes used in the calculations of body region angle for the trunk model according to Equations 4.7 to 4.9.

	Lumbar	Cervical	Head
n_1	43900141	43700121	43700101
n_2	43300071	43900131	43700121

4.1.2 Neural Delay

The neural delay implements a circular storing function to sample and hold $y(t)$ in a vector in accordance with the stated time delay, T_{de} (Figure 4-4). At time t_n, $y(t_n)$ is stored at position n in the vector. At the same time, if $t_n \geq T_{de}$, $y(t_n - T_{de})$ is retrieved at position $n - T_{de}$ from the vector and further used as $y_d(t_n)$, input signal to the controller. The function is circular in the sense that the positions in the vector is reused and only samples values in accordance with the time period set by T_{de}, with a resolution of $y(t)$ set by the user. The concept of a neural delay is described in Section 2.2.3. The delay is due to the time it takes for a signal to travel from a sensor through the integration centre to an effector (Figure 2-2). Numerical values for T_{de} depending on model can be seen in Table 4-4.

$$(\dots, y(t_n - T_{de}), \dots, y(t_n), \dots)$$

Figure 4-4 The angle of the joint is stored at $t = t_n$ in a vector where it, at $t = t_n + t_d$, is retrieved and used as $y(t_n + t_d)$ a delayed output of the angle of the joint.

4.1.3 PID control algorithm

The Equations used in the controller to calculate the control signal, $u(t_n)$ is presented in Equations 4.10-4.17. Equation 4.10 is the general description of a PID-control algorithm while Equations 4.12 - 4.16 show the discrete approximations of the analytic error function, its integration equation 4.14 and differentiation equation 4.16. Numerical values for the variables used can be seen in table 4-3 and 4-4. A PID controller is commonly used in order to get a reliable and stable control system output that can react on sudden changes without any error remaining (Lennartsson, 2002).

$$u(t_n) = \left(e(t_n)K_p + \int_{t_0}^{t_n} e(\tau)d\tau\, K_i + \frac{de}{dt}K_d \right) x_{dir} \tag{4.10}$$

Each error function is numerically implemented as

$$e(t_n) = r(t_n) - y_d(t_n) \tag{4.11}$$

$$\int_0^{t_n} e(\tau)d\tau = ie(t_n) = ie(t_{n-1}) + \frac{e(t_{n-1}) + e(t_n)}{2}\Delta t \tag{4.12}$$

$$ie(t_0) \cong 0 \tag{4.13}$$

$$\frac{de}{dt} = \frac{de(t_{n-1})}{\Delta t} + \frac{\Delta t}{\Delta t + T_f}\left(\frac{e(t_n) - e(t_{n-1})}{\Delta t} - \frac{de(t_{n-1})}{\Delta t} \right) \tag{4.14}$$

$$y_d(t_n) = y(t_n - T_{de}) \tag{4.15}$$

$$\Delta t = t_n - t_{n-1} \tag{4.16}$$

x_{dir}, is a constant, either 1 or -1 depending on the environment of which the THUMS model is to be simulated in, to ensure that the action of the controller is counter-acting external disturbances and not amplifying them. If the model is facing forward in the positive x-direction $x_{dir} = 1$, if the model is facing forward in the negative direction, which typically can happen in a simulation with a complete vehicle model, $x_{dir} = -1$. The controller variables and constants used are summarized in Table 4-3 and 4-4.

In addition to Equation 4.10 the controller signal is adapted to each muscle depending on specified cocontraction level and whether it during contraction causes flexion or extension of the joint:

$$u_m(t) = max(u_l(t)\bar{r}_l + u_c(t)\bar{r}_c + u_h(t)\bar{r}_h, 0.0) + (cc_l|\bar{r}_l| + cc_c|\bar{r}_c| + cc_h|\bar{r}_h|) \tag{4.17}$$

Where,

$u_*(t)$ = control signal for each body region,

$\bar{r}_*$ = vector defining the direction for each muscle in the body region with respect to how it affects the joint angle when contracting,

cc_* = cocontraction level (user specified, see Table 4-4).

Cocontraction level is introduced to dampen the oscillations of the controlled body region, it is a comparatively small level of activation used for both flexor and extensor muscles.

Table 4-3 Load curve ID for variables used in the control algorithm and activation dynamics of the seated trunk and elbow flexion simulations

	Seated Trunk Model			Elbow model
Control Parameter	Lumbar control	Cervical control	Head control	Elbow control
y_0	3100000	7100000	8100000	200000
$y(t)$	3100001	7100001	8100001	200001
$r(t)$	3100002	7100002	8100002	200002
$e(t)$	3100003	7100003	8100003	200003
$ie(t)$	3100004	7100004	8100004	200004
de/dt	3100005	7100005	8100005	200005
$u(t)$	3100006	7100006	8100006	200006
$Na(t)$	3100007	7100007	8100007	200007
$Ne(t)$	3100008	7100008	8100008	200008

Table 4-4 Load curve ID and corresponding value for constants used in the PID controller.

	Seated Trunk Model						Elbow Model	
Controller constants	Lumbar control		Cervical control		Head control		Elbow control	
	Load Curve ID	Value	Load Curve ID	Value	Load Curve ID	Value	Load Curve ID	Value
cc	3200001	0.03	7200001	0.03	8200001	0.03	200011	0.05
$ie(t_0)$	3200002	0	7200002	0	8200002	0	200012	0
K_p	3200003	1.25	7200003	1.5	8200003	1	200021	0.307
K_i	3200004	0.625	7200004	0.75	8200004	0.5	200022	0
K_d	3200005	0.15625	7200005	0.2	8200005	0.133	200023	0.068625
T_f	3200006	0.005	7200006	0.005	8200006	0.005	200024	0.005
T_{de}	3200007	0.025	7200007	0.02	8200007	0.02	-	0.02
T_{ne}	1100001	0.035	1100001	0.035	1100001	0.035	-	0.030
T_{naa}	1100002	0.01	1100002	0.01	1100002	0.01		0.005
T_{nad}	1100003	0.04	1100003	0.04	1100003	0.04		0.03

4.1.4 Activation dynamics

The neurochemical reactions taking place after an action potential, see Section 2.2.2, is described by two low-pass first order filters in series (Winters and Stark, 1985), Equation 4.18 and 4.19:

$$T_{ne}\frac{dN_e}{dt} = m_{fe} - N_e(t), \tag{4.18}$$

$$T_{na*}\frac{dN_a}{dt} = N_e(t) - N_a(t), \tag{4.19}$$

where T_{ne} is the muscle activation dynamics time constant for neural excitation and $T_{na*}$is the muscle activation dynamics time constant for activation or deactivation.

$$m_{f,e} = \frac{u(t)}{M_{f,e}} \tag{4.20}$$

$M_{f,e}$ = Maximum moment of extension or flexion about the joint

Numerically implemented through discretization in the uctrl1 sub-routine this equates to a series of two Equations (Equations 4.21 and 4.22).

$$N_e(t_n) = \min\left(\max\left(N_e(t_{n-1}) + \left(u(t_n) - \frac{N_e(t_{n-1})}{T_{ne} + \Delta t}\right)\Delta t, 0\right), 1\right) \tag{4.21}$$

$$N_a(t_n) = \min\left(\max\left(N_a(t_{n-1}) + \left(N_e(t_n) - \frac{N_a(t_{n-1})}{T_{na*} + \Delta t}\right)\Delta t, 0\right), 1\right) \tag{4.22}$$

$N_a(t)$ behaves differently depending on if the activity is increasing or decreasing. If $N_a(t)$ is increasing, the time constant is six times lower than if $N_a(t)$ is decreasing (see Table 4-4). The level of neural excitation and activation is limited according to,

$$0.005 \leq N_e(t) \leq 1 \tag{4.23}$$

$$0.005 \leq N_a(t) \leq 1 \tag{4.24}$$

4.1.5 Executing the Sub-Routine

The uctrl1 sub-routine is one of many sub-routines present in the dyn21.f file. The uctrl1 code found in the dyn21.f file after unpacking a downloaded LS-DYNA binary package performs no functionality of its own but lets a user add functionality (LSTC, 2012a). The ability to add functionality in the uctrl1 part of dyn21.f is used to add the algorithms and functions needed to calculate the muscle activity level according to Equations 4.10-4.17 and 4.18-4.22. In order to let the uctrl1 run as a sub-routine to the simulation, a binary need to be compiled that incorporate the uctrl1 sub-routine of the dyn21.f file. When moving between different versions of LS-DYNA, experience during the work of this thesis has shown that it is advised to only replace the uctrl1 sub-routine and leave other parts of the dyn21.f file unaltered in order to avoid any conflicts or incompatibilities.

4.1.6 Analyzing the result

The load curves generated in the sub-routine cannot be logged in a curvout file. Instead data is written to comma separated text files, .csv-format, (listed and explained in Table 4-5). Post processing was achieved in Matlab (The Mathworks, Natick MA), Excel (Microsoft, Redmond WA), or LS-PREPOST (LSTC, Livermore CA) which have tools to interpret the text file into identifiable numbers arranged in rows and columns from where the result could be further plotted and analysed.

Table 4-5 Table of files and their content, that contain the result from the uctrl1 sub-routine. Lhs stands for left hand side, rhs stands for right hand side.

Seated Trunk model

ctrl_var.csv	yd(t), r(t), e(t), ie(t), de(t)/dt, u(t) for each controller
mctrl.csv (_lhs, _rhs)	u(t) for each muscle
mexc.csv (_lhs, _rhs)	Ne(t) for each muscle
mact.csv (_lhs, _rhs)	Na(t) for each muscle
mlen.csv (_lhs, _rhs)	Muscle length
mstr.csv (_lhs, _rhs)	Muscle strain

5 Method

The main part of this thesis has involved breaking down the functionality inside the uctrl1 subroutine or inside the .key-files associated with the muscle activation model of the Chalmers AHBM. For each functionality, existing LS-DYNA keywords have been reviewed in the manuals for the LS-DYNA software (LSTC, 2012a) in order to find the same functionality as in the user defined uctrl1 subroutine. The process of deconstructing the functionality and searching for keywords has been an iterative process where possible candidates for performing the functionality has been conceptualized and tested.

Early on it became apparent that the ability to define load curves that can take input data and perform a mathematical operation on that data would solve much of the functionality needed or otherwise be helpful when used as part of the functionality. The ability to prescribe the motion of a single node was another LS-DYNA feature very helpful in solving the muscle activation model differential Equations. Mechanical representations of the differential Equations were used in order to use the solution capacity of LS-DYNA. Differentiation and integration can be performed by letting LS-DYNA solve kinematic Equations of the freely moving, or dynamically-constrained, nodes with prescribed motion. Much of this functionality was solved by defining node movements as variables to Equations, defined by standard keywords. The solution was implemented both in the Elbow model as well as in the Seated Trunk model. Though the functionality involved is basically the same, in some areas they vary and are therefore when needed presented in separate Sections.

5.1 Calculating the joint angle

Although the standard input deck/cards in LS-DYNA provides the possibility to define coordinate nodes that can be used to determine the rotational displacement of a node, a vector algebra approach is used in the proposed solution, similar to that in the uctrl1 subroutine (Section 4.1.1). The keyword *DEFINE_CURVE_FUNCTION contain the functionality needed to represent the mathematical expressions covering Equations 4.1-4.6 for the elbow model and 4.7-4.9 for the seated trunk model. The initial angle of the joint is retrieved by initializing a simulation and used as reference value for the controller. An alternative method is to define a local coordinate system and by using a load curve function, measure the rotational angle of a joint. When evaluating the LCM, vector calculation was used in order to minimize unnecessary differences between the LCM and the uctrl1 subroutine model.

5.2 Neural delay

A time delay, that only delays and does not distort the input signal, is the most challenging function to implement with the standard LS-DYNA keywords. The goal is to have a sample and hold mechanism without affecting the sampled signal during the hold sequence. Since no keywords or method has been found that can perform such functionality the delay is modelled, using the ability to simulate latency in the kinematics of mechanical systems in LS-DYNA. A delay is implemented through the use of a low pass filter, similar as for the activation dynamics (see Figure 5-1). The filter is interpreted as a mechanical low-pass filter that establishes a phase shift of the signal related to the viscous properties of a discrete linear damper connecting two nodes. As described in Section 5.4, two nodes with one degree of translational freedom are separated by a distance of 10mm and connected using a discrete element with purely viscous material properties. The viscous properties are in turn related to the defined time constant of the filter (Equation 5.4; Figure 4-3; Table 4-4). A low pass filter is defined in the frequency domain by the laplacian transfer function:

$$G(s) = \frac{1}{Ts + 1} \tag{5.1}$$

where, the time constant, T, defines the cut off frequency, ω_c, of the low pass filter,

$$\omega_c = \frac{1}{T} \tag{5.2}$$

The time constant also defines the time it takes for the output signal to reach 63% of the input signal so that $y(T) = 0.63\sigma(T)$, when subjected to a step function defined as (Lennartson, 2002):

$$\sigma(t) = \begin{cases} 0 & t < 0 \\ 1 & t \geq 0 \end{cases} \tag{5.3}$$

The transfer function, Equation 5.1, give an understanding of the effect on the output of the model based on the angular frequency, ω, of the input signal. The response of the system depends on the cut-off frequency (Equation 5.2). The cut-off frequency defines the frequency at which a negative gain of -3dB is achieved. For the Elbow model, where $T = 0.034\ s$, $\omega_c = 29{,}41 \frac{rad}{s}$ $(\Rightarrow f_c = 4{,}68\ Hz)$. The resulting delay is calculated according to,

$$T_{de}(\omega) = \frac{\phi(\omega)}{\omega} \tag{5.4}$$

where, $\phi(\omega)$ is the phase shift in relation to input angular frequency, ω, of the filter. A plot was made in Matlab (The Mathworks, Natick MA), using Equation 5.1 and Equation 5.4 that can be seen in Figure 7-3.

5.3 Control Algorithm

The control algorithm works in the same way as presented in Section 4.1.3, Equation 4.10, with the exception that no separate discrete approximation is defined but relies on the discrete differential Equations solver built into LS-DYNA. The control algorithm was implemented using load curves that perform the mathematic operations needed as well as receiving input from the calculations of the angle of the joint being controlled (see Appendix A). The mathematic operations available in *DEFINE_CURVE_FUNCTION in LS-DYNA incorporates no method for integration of a signal other than that it is possible to retrieve and use kinematic and kinetic properties of nodes as input. The method chosen is that a node is used as holder of the error signal, $e(t)$. This is done by creating a node only used for the controller, which velocity is defined through *BOUNDARY_PRESCRIBED_ MOTION_NODE. The node is allowed to move in x-direction only. Integration of error, $ie(t)$, is defined as the position of the node, which is retrieved as a curve function. In the same manner, the differentiation of the error, de/dt, is defined as the acceleration of the node, also retrieved as a curve function using *DEFINE_CURVE_FUNCTION.

5.4 Activation Dynamics

Solving the Equations of activation dynamics differs from the sub-routine since retrieval of data from former time steps is not possible. To define the differential equations in a physical sense that is simulated by LS-DYNA, the solution to the differential equation is modelled as a mechanical system with two nodes connected by a discrete beam with purely viscous material properties (Figure 5-1, Equation 5.5-5.13). Via *BOUNDARY_ PRESCRIBED_MOTION_NODE, the controller output, $u(t)$ defines the one-dimensional velocity of node n_1 in the excitation model (see Figure 5-1). The velocity of node n_2 is read as $N_e(t)$ which in turn defines the input to the activation model. Material used for $N_e(t)$ and $N_a(t)$ is the simplest possible viscous material, where only viscosity is defined. The non-linear viscous material incorporates the possibility to define different viscosity properties depending on if the material is experiencing loading or unloading. A force equilibrium Equation of the system described in Figure 5-1 translates into Equation 5.2. By substituting x and y into the variables N_e and N_a (Equation 5.11, 5.13) this representation is able to represent the first order low pass filter described in Section 4.1.4 (Equation 4.18, 4.19). As this approach is also used for the neural delay, the same frequency dependency applies for the activation dynamics according to Equations 5.1-5.4, with a cut off frequency in relation to the time constant used and thus the viscous properties chosen for the discrete element (Equation 5.14-5.16).

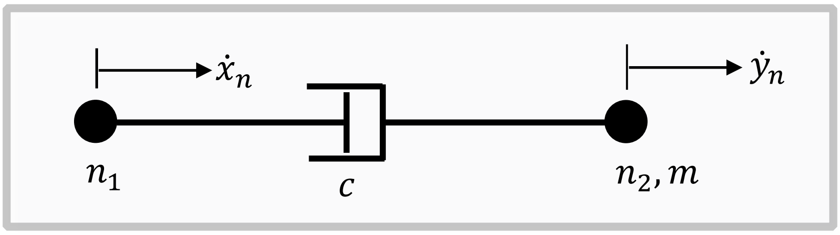


Figure 5-1 Design of two nodes connected with a truss element with viscous material properties in order to model a first order low pass filter

$$m\ddot{y}_n = c(\dot{x}_n - \dot{y}_n) \tag{5.5}$$

$$\frac{m}{c}\ddot{y}_n + \dot{y}_n = \dot{x}_n \tag{5.6}$$

$$\{\mathcal{L}\} \qquad Y\left(\frac{m}{c}s^2 + s\right) = Xs \tag{5.7}$$

$$\Rightarrow \qquad G(s) = \frac{Y}{X} = \frac{s}{\left(\frac{m}{c}s^2 + s\right)} \tag{5.8}$$

$$T = \frac{m}{c} \Rightarrow c = \frac{m}{T} \tag{5.9}$$

$$\dot{x}_n = u(t), \dot{y}_n = N_e(t) \tag{5.10}$$

$$\Rightarrow \qquad T_{ne}\dot{N}_e(t) = T_{ne}\frac{dN_e}{dt} = u(t) - N_e(t) \tag{5.11}$$

$$\dot{x} = N_e(t), \dot{y} = N_a(t) \tag{5.12}$$

$$\Rightarrow \qquad T_{na*}\dot{N}_a(t) = T_{ne}\frac{dN_a}{dt} = N_e(t) - N_a(t) \tag{5.13}$$

The viscous properties of the material used in the modelling of $N_a(t)$ is different depending on if the excitation level, $N_e(t)$, is increasing or decreasing. The non-linearity is defined by a tabulated list of relational values that translates into the curve seen in Figure 5-2. The resulting response of the system depending on level of input signal is presented in Section 7. With T_{ne}, T_{naa} and T_{nad} defined in Table 4-4 and with the mass of n_2, $m = 1.0E\text{-}6$ $[ton]$, the viscous material property, c, is calculated according to Equation 5.9, as seen in Equation 5.14-5.16. The viscous material property becomes, in the simulations, relationships relating velocity, v $[mm/s]$, and force, f $[N]$.

$$c_{ne} = \frac{m}{T_{ne}} = 2.8571E\text{-}5\ [ton/s] \tag{5.14}$$

$$c_{naa} = \frac{m}{T_{naa}} = 2.0E\text{-}4\ [ton/s], f > 0, v > 0 \tag{5.15}$$

$$c_{nad} = \frac{m}{T_{nad}} = 3.33E\text{-}5\ [ton/s], f < 0, v < 0 \tag{5.16}$$

Equation 5.15 and 5.16 describes a non-linear viscous relationship that is visualized in Figure 5-2.

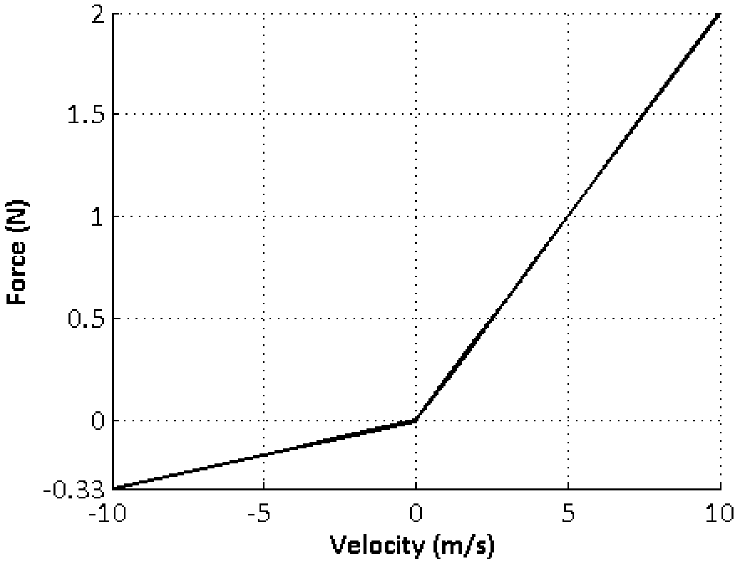


Figure 5-2 Force versus velocity response curve for non-linear viscous material properties used in modeling of $N_a(t)$ (taken from the elbow model).

Due to the implementation of cocontraction, the neural excitation level has to be allowed to be negative and is therefore defined as,

$$-(cc-0.005) \leq N_e(t) \leq 1-cc \tag{5.17}$$

5.5 Signal Output

Since the muscle activation level in the Hill-type muscle material is defined as,

$$0.005 \leq N_a(t)_{muscle} \leq 1, \tag{5.18}$$

the same has to be true for the output from the activation dynamics. When an initial cocontraction level is desired, as in the elbow model, it has to be initiated after the output from the activation dynamics, $N_a(t)_{activation}$, such that,

$$N_a(t)_{muscle} = N_a(t)_{activation} + cc \tag{5.19}$$

This implementation of cocontraction initiation is done in order to avoid the initial step response of $N_a(t)_{muscle}$ from a level of zero activation to the level of cocontraction. An initial step response might interfere with the response of the muscles and increase instability of the model. When $N_a(t)_{muscle}$ is defined as in Equation 5.19, it means that in order for $N_a(t)_{muscle}$ to reach an activation level below the cocontraction level, $N_a(t)_{activation}$ must be allowed to be negative.

Thus,

$$-(cc-0.005) \leq N_a(t)_{activation} \leq 1-cc \tag{5.20}$$

$N_a(t)$ is given to the *MAT_MUSCLE material model, defined by a load curve function representing the instantaneous velocity of node n_2 in the calculation of $N_a(t)_{activation}$. If cocontraction is used, $N_a(t)_{activation}$ is added to that level of activation. If no cocontraction is used, $N_a(t)_{activation}$ correlates with the muscle activation level (Equations 4.23-4.24). Muscles are selected for performing flexion or extension of the body regions as described by Östh (2010).

6 Simulations

Simulations has mainly been done using LS-DYNA release V971 R5.1.1 and R6.1.0 on a Red Hat based computer system equipped with Intel Xeon processors. Post processing has been done in LS-PREPOST version 4.0 and Matlab (The Mathworks, Natick MA). Unit set used during simulation of the elbow model and of the trunk model was mm, s, ton, N. Apart from simulating all parts of muscle activity feedback loop together with the Elbow model or the Trunk model, simulations have also been done in order to reassure that each part of the feedback loop is working as intended. This was done as part of the Elbow model simulation. When comparing the calculation of joint angle and the calculation of control signal, the LCM was run in the background of the simulation. Activity level was calculated by the uctrl1 subroutine, thus the LCM calculated an activity level based on the instantaneous elbow joint angle. This approach was used to reduce a cumulative error between the two methods and to exclude the errors imposed from implementation of the neural delay and activation dynamics equations. When modelling a step behaviour, for evaluation of the response of the low pass filter, in LS-DYNA (Equation 5.3), in order to minimize any transient behaviour, the step was introduced at $t = 10\ (ms)$ and a slight ramping with a duration of $t_{ramp} = 0.01\ (ms)$ of the function. The result from the simulations are taken from the curvout-logfile, concerning the control algorithm and the elout-logfile concerning muscle force properties.

6.1 Elbow Model

The Elbow Model, comprising of parts of the trunk with muscles added for the elbow and wrist of the right arm is used in the simulations. The initial constrained elbow motion, seen in Figure 6-1, and setup is denoted as simulation 31 in the works of Östh (Östh, 2010; 2011). The simulation setup is based on an in vivo human experiment using the ARMANDA haptic manipulator (de Vlugt et al., 2003, Östh et. al, 2011a). Since a neural delay in the sense of a sample and hold mechanism was not implemented in the LCM two simulations were made, one in which modelling of the neural delay was excluded. The results of the simulations are presented in Section 7.1, Figure 7-6 to Figure 7-11.

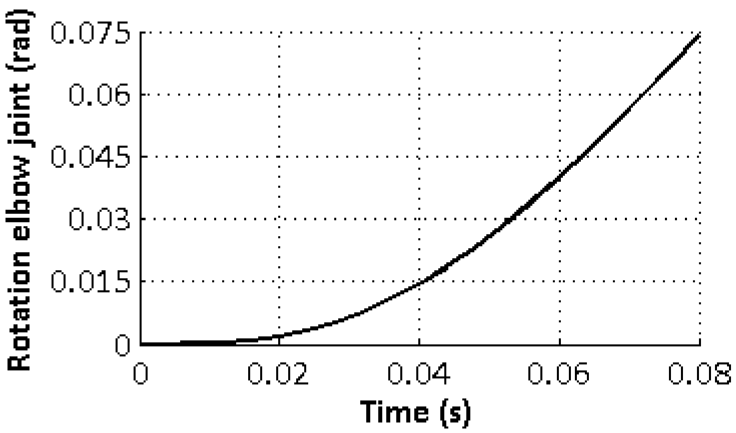


Figure 6-1 Initial constrained rotation of the elbow joint as used in simulation 31 of the ARMANDA volunteer experiment (Östh, 2010;2011).

6.2 Trunk Model

The validation of the Trunk model is done by measuring the response of the Chalmers AHBM with muscular activity level determined either by the LCM or the user defined uctrl1 subroutine. Simulation setup with the defined coordinate system is shown in Figure 6-2. The AHBM is positioned slightly above a seat construction comprised of two flat rigid surfaces that act as support for the back, hip and thigh of the AHBM. During the simulation a gravity load was applied to the AHBM in the z direction, denoted g in Equation 6.1, and the seat was accelerated according to Equation 6.2 in x direction, where positive x is defined in the forward facing direction of the AHBM.

$$g = -9.82\, m/s^2 \tag{6.1}$$

$$\ddot{x}_{seat}(t) = \begin{cases} 0 & t < 500ms \\ 9.82\, m/s^2 & 500ms \leq t \leq 800ms \\ 0 & t > 800ms \end{cases} \tag{6.2}$$

The gravity load is applied to the Chalmers AHBM while the x-directional load is applied to the rigid seat. Since there are no muscles implemented in the arms or shoulders, the hands are locked in position relative to the rigid seat. The results of the simulations are presented in Section 7.2, divided in accordance to each body region controller and compared with a simulation using the uctrl1 subroutine.

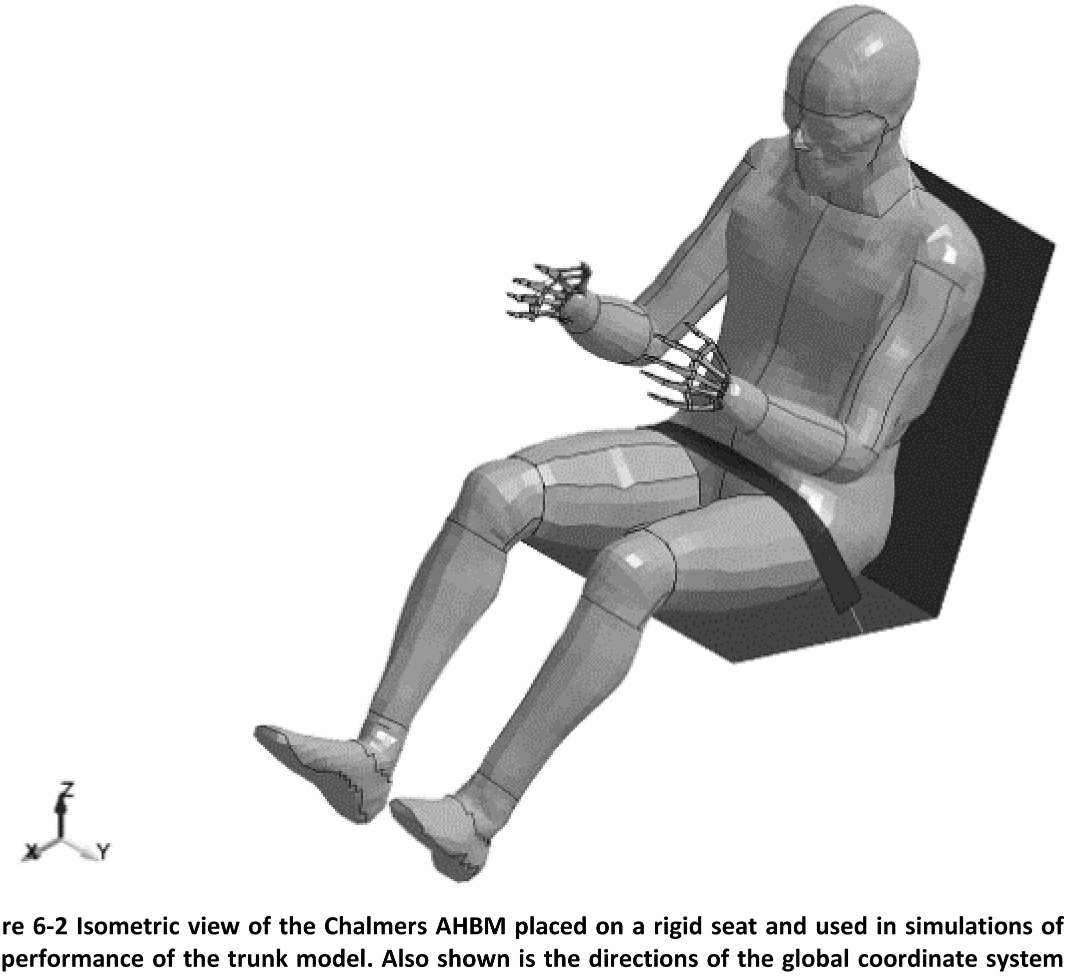


Figure 6-2 Isometric view of the Chalmers AHBM placed on a rigid seat and used in simulations of the performance of the trunk model. Also shown is the directions of the global coordinate system used for the Chalmers AHBM.

7 Results

As can be seen in Figure 7-1 there is a minor difference between calculating the angle between two vectors using load curve functions and monitoring node rotation in a locally defined coordinate system though the overall result is almost identical. The measured nodal rotation is able to deliver a higher maximum rotational angle than the calculated vector angle.

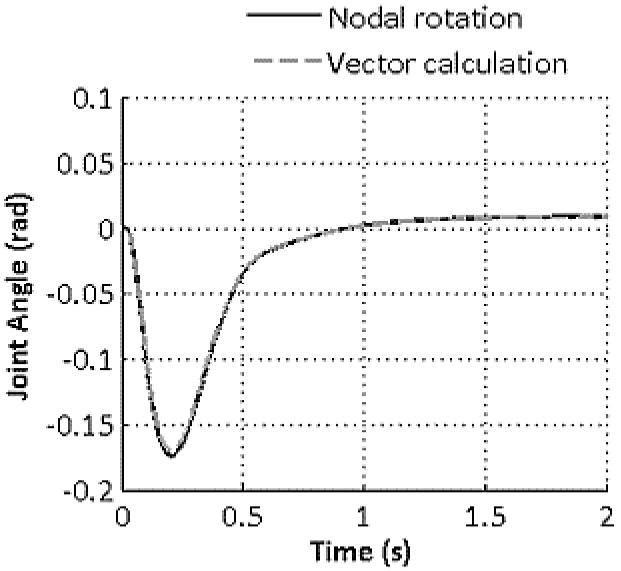


Figure 7-1 Comparison between the angle calculation for the elbow joint in the LCM using defined vectors as described in Section 4.2.1 and measuring the rotation of a node defined by a local coordinate system.

By defining a low pass filter, a cut-off frequency is being defined at the same time. For frequencies below the cut-off frequency the phase shift is constant and the signal strength is unaffected. Above the cut-off frequency, signal strength is decreasing with frequency and the time delay decreases (Figure 7-2). With $T = 0.034$ (corresponding to $c \approx 2.94E - 5$ [ton/s] for a nodal mass of 1e-6 ton), the frequency response and time delay versus frequency is plotted in Figure 7-2. The resulting time delay and signal gain is approximately constant for frequencies below 1Hz (see Figure 7-2). The frequency response of the neural delay is valid for frequencies below 1Hz, where the delay is constant and signal level is preserved. As seen in Figure 7-2 signal amplitude is reduced by 50% at 8Hz, consistent with the calculations in Section 5.4.

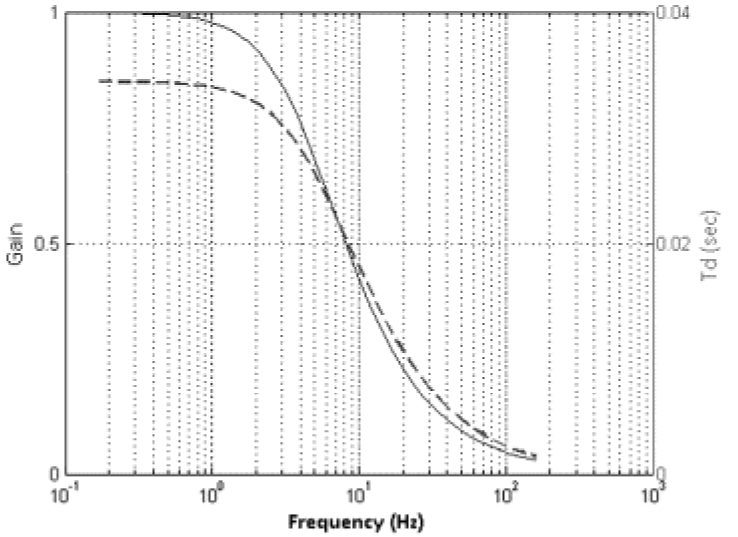


Figure 7-2 Time delay versus frequency (dashed line, right Y-axis) and gain versus frequency (solid line, left Y-axis) for the first order low pass filter where T=0.034, used for modeling neural delay in the elbow model. Logarithmic X-axis.

In Figure 7-3 the calculation of joint angle with defined vectors and the output from the control signal is compared between the uctrl1 subroutine and the LCM. The comparative curves in Figure 7-3 are close to identical, showing that the calculations performed concerning the control algorithm in the LCM give the same result as the uctrl1 subroutine.

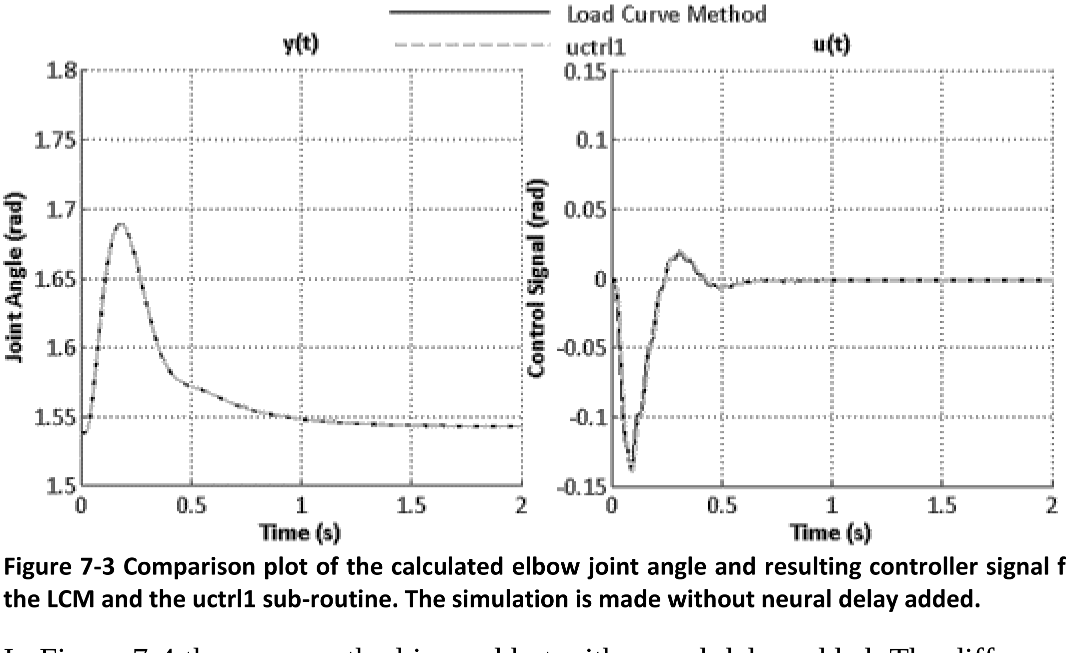


Figure 7-3 Comparison plot of the calculated elbow joint angle and resulting controller signal from the LCM and the uctrl1 sub-routine. The simulation is made without neural delay added.

In Figure 7-4 the same method is used but with neural delay added. The difference in signal peak level for $y_d(t)$ and $u(t)$ is due to the negative gain imposed by the neural delay in the LCM, as seen in Figure 7-2. It is also noted in Figure 7-4 the difference at the start of the simulation between the sample and hold delay used in the uctrl1 subroutine and the delay filter used in the LCM. Since the neural delay in the LCM is a dynamic filter, the delay is defined by the time constant, referring to the time it takes for the output signal to reach 63% of the input signal as described in Section 5.2. The smoother result of $u(t)$, when using the LCM, is due to the low pass filtering characteristics of the neural delay.

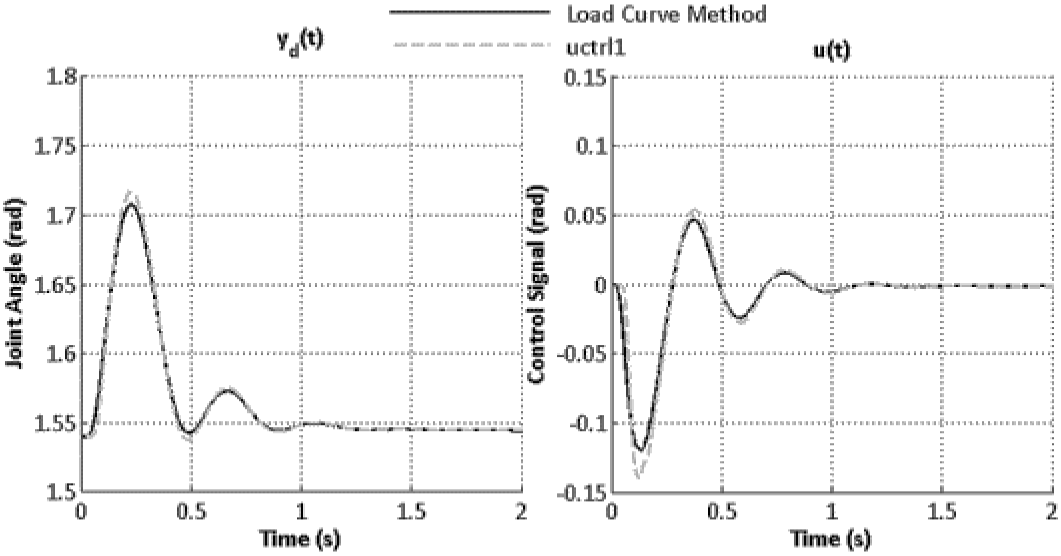


Figure 7-4 Comparison plot, as Figure 7-2 but with the different approaches to neural delay added

The step response for the mechanical representation of the activation dynamics system is as expected. There is a clear distinction in the response time for $N_a(t)$ depending on whether the response is to a positive or negative step. In Figure 7-3 the response for $N_a(t)$ is 63% at $t = 0.015\ s$ which is $0.005\ s$ after the step, and in accordance with $T_{naa} = 0.005$. For the negative step, at $0.2\ s$, the response of $N_a(t)$ is slower, 63% of the response occurring at $0.23\ s$, equivalent to $T_{nad} = 0.03$. $N_e(t)$ has a constant response time of $T_{ne} = 0.035$, 63% response is achived at $t = 0.045\ s$ and $t = 0.235\ s$.

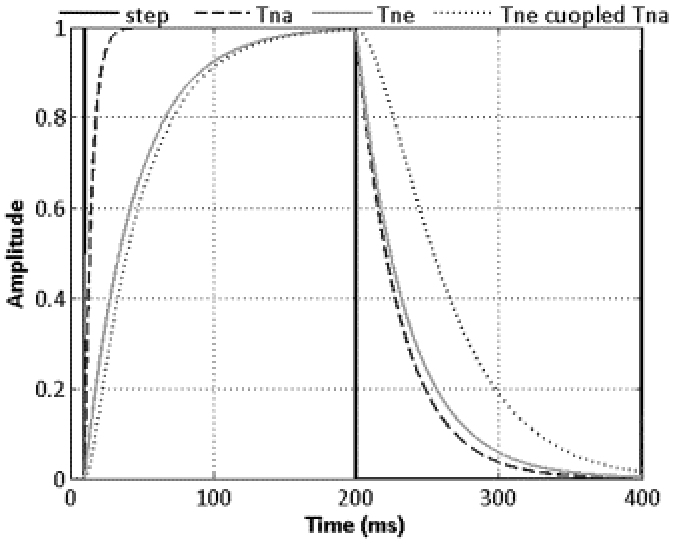


Figure 7-5 Simulated step response of the activation dynamics model in LS-DYNA. The difference in response time for $N_a(t)$ depending on if the step is positive or negative is as expected relative to T_{naa} and T_{nad}.

7.1 Simulation of Elbow Model

The comparative simulations show a great deal of resemblance between the LCM and the sub-routine method in the performance of the functions and in the overall angle of the elbow joint. In Figure 7-6 and 7-7, activation level sent to the flexor and extensor muscles are displayed. Figure 7-6 is taken from a simulation using the uctrl1 subroutine as provider of muscle activity. As seen in Figure 7-6 to 7-11, the addition or absence of a neural delay affect the number of overshoots before the signal is stabilizing. Due to the smoothening effect of the low pass filter used as neural delay in the LCM, this behaviour is reduced in that approach.

7.1.1 Neural Delay Excluded

As seen in Figure 7-3, the calculation of the angle of the joint and the calculation of control angle results in similar results for the LCM and sub-routine when using the same joint motion as input to the calculations. In Figure 7-6 to 7-9, results are presented from simulations as described in Chapter 6.1 without the implementation of a neural delay filter.

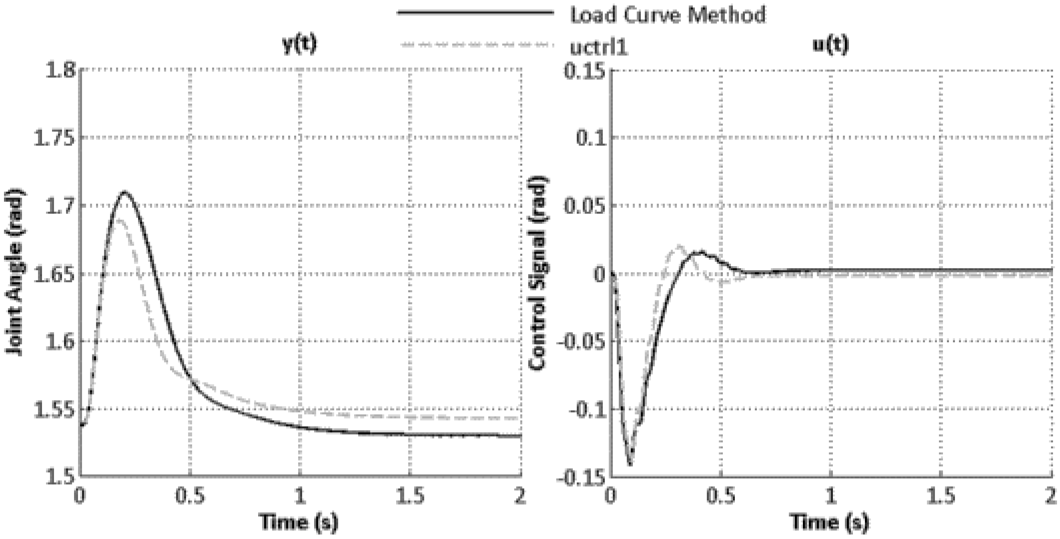


Figure 7-6 Resulting joint angle and controller signal when simulating the elbow response in the ARMANDA setup.

The difference in neural excitation in Figure 7-7 is due to the difference in the definition of $N_e(t)$ according to Equation 4.24 and Equation 5.12. Still it is seen that the LCM reacts slower in excitation when switching from extensor to flexor activity. In some part due to extensor peak excitation occurring later for the LCM than for the uctrl1 subroutine and due to the de-excitation being slower for the LCM than for the subroutine.

As described in Section 4.1.3 and Section 5.4. The neural de-excitation of the flexor muscles is much more rapid for the subroutine than in the LCM. This difference is due to the control signal being limited in the LCM not to be negative. If the control signal was allowed to be negative, the de-excitation of the muscles in the LCM would probably be faster.

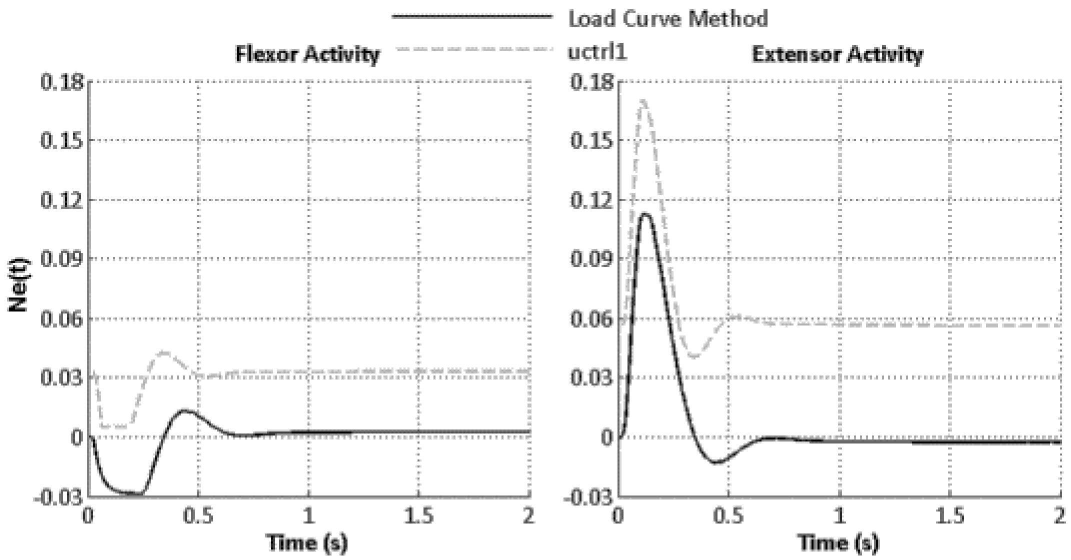


Figure 7-7 Level of Neural excitation during simulated elbow motion in the ARMANDA simulation setup. The apparent difference of signal level is due to cocontraction being added later for the LCM signal.

The level of extensor activity seen in Figure 7-8 differs between the LCM and the uctrl1 subroutine, similar to the difference in calculated final angle of the elbow joint (Figure 7-6). The LCM achieves a somewhat slower response but with a reduced overshoot, in part due to the reduced peak level and smoothening effect of the neural delay low pass filter.

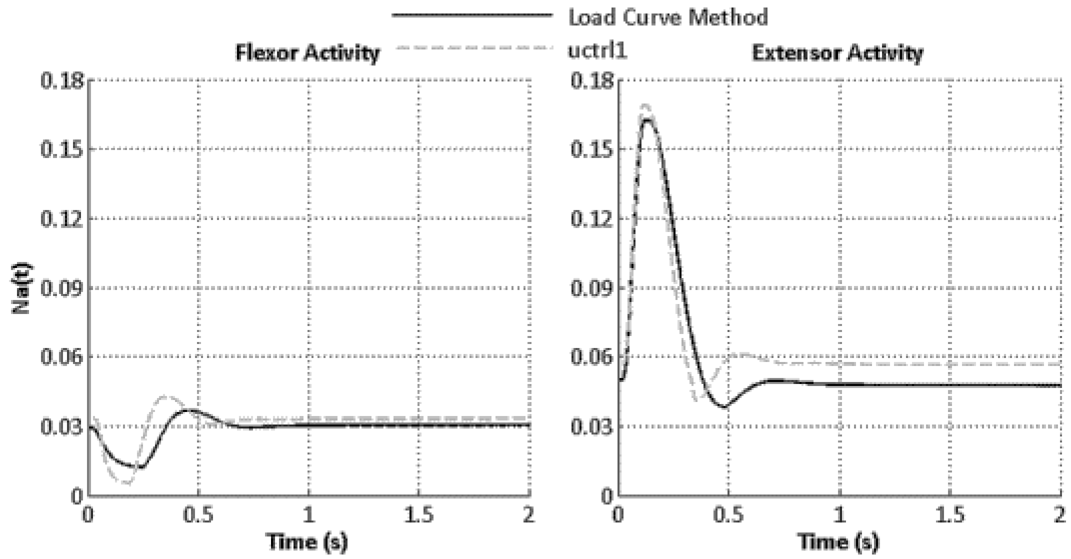


Figure 7-8 Level of muscle activation simulated to resemble the results from the uctrl1 subroutine based on simulation of the ARMANDA haptic test setup.

The total force in Figure 7-9 is based on the passive and active forces in the muscle. When comparing the shape of the curves in Figure 7-8 and 7-9 it is seen that the level of muscular activity plays a vital part in the production of muscle force, as seen in the Hill-model, Equation 2.1.

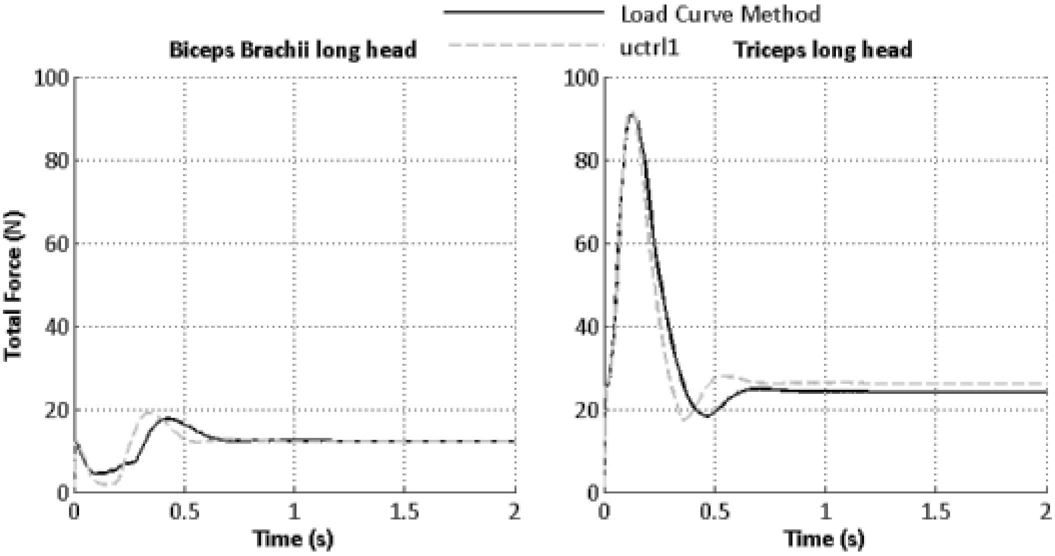


Figure 7-9 Total force exerted by Biceps Brachii long head and Triceps long head comparing the LCM with the uctrl1 subroutine with neural delay excluded.

The results show that even though the neural delay low pass filter is excluded from the LCM, the result still differs from the uctrl1 subroutine.

7.1.2 Neural Delay Included

Neural delay adds a further level of complexity in the model. The effect of the different approaches to model the neural delay is seen in Figure 7-10 to 7-13. The LCM has a slower response with reduced peak level and therefore less over-shoot, especially seen in the result of $u(t)$, $N_e(t)$ and $N_a(t)$.

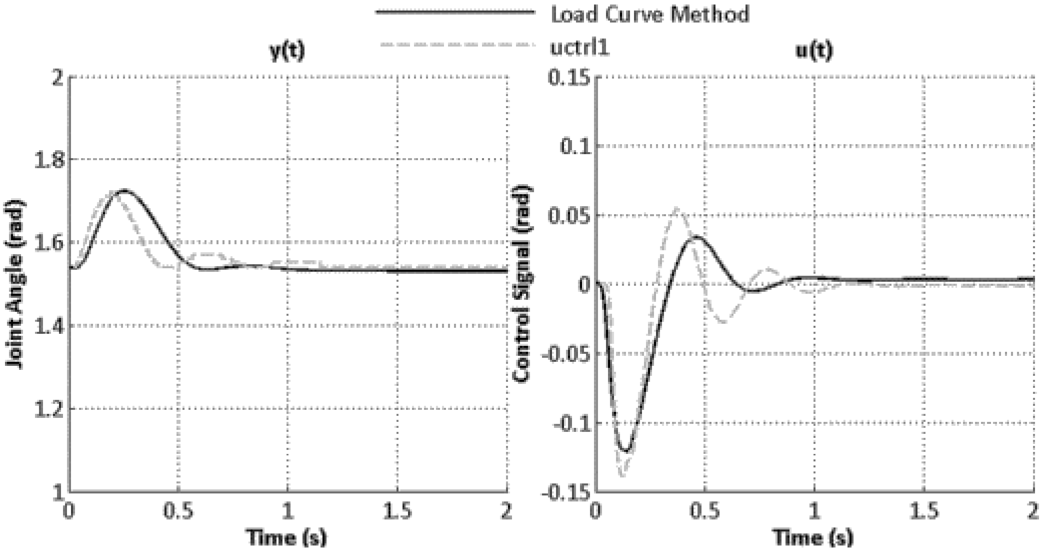


Figure 7-10 Calculated joint angle and control signal with delay added to the uctrl1 and LCM models.

The faster de-excitation of the uctrl1 subroutine is seen in Figure 7-11 concerning flexor activity. This response is attributed to $-1 \leq u(t) \leq 1$ for the uctrl1 subroutine while $0 \leq u(t) \leq 1$ was used for the LCM, resulting in an increased negative gradient for the uctrl1 subroutine de-excitation.

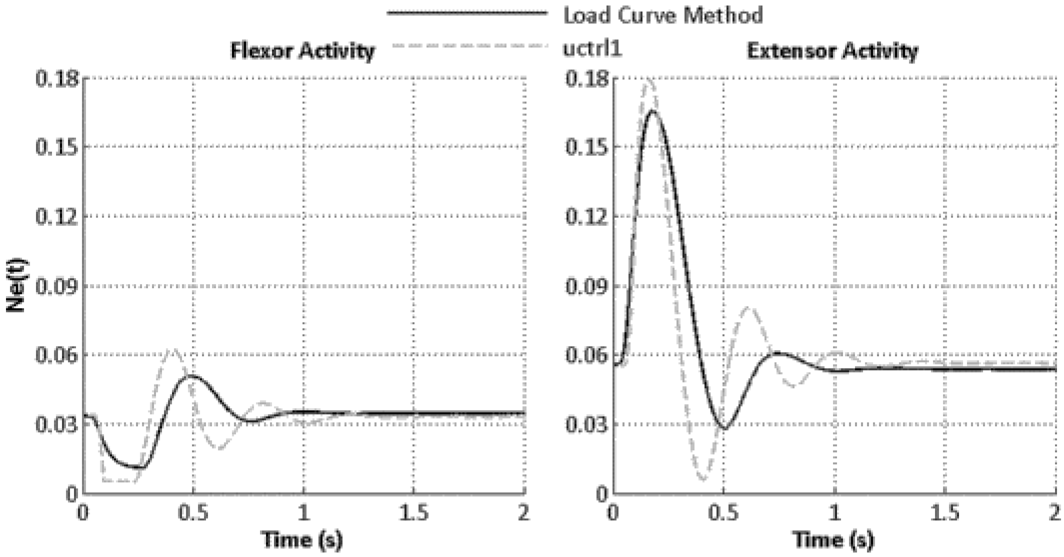


Figure 7-11 Neural excitation level with neural delay added in the LCM, compared to the uctrl1 subroutine.

The difference shown in Figure 7-11 is further enhanced in Figure 7-12, showing activation level sent to the flexor and extensor muscles in the elbow model. Though, since basically the same gradient is used as input, the difference is much less than for the level of excitation. The LCM appears to show a slightly enhanced stability.

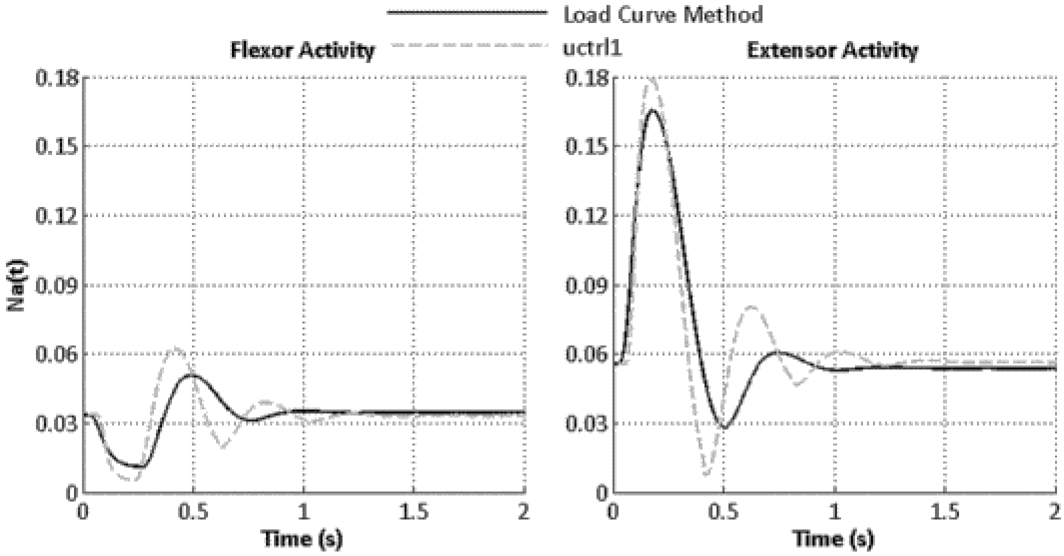


Figure 7-12 ARMANDA simulation of muscle activity level with neural delay model added. Flexion on the left and extension on the right.

Concerning the force exerted by the muscles in the LCM when delay is included (Figure 7-13), the behavior resembles that of no delay present, that is, it is closely tied to the muscle activity level (Figure 7-12). As the muscle activity level differs from the result from the uctrl1 subroutine so does the total exerted force.

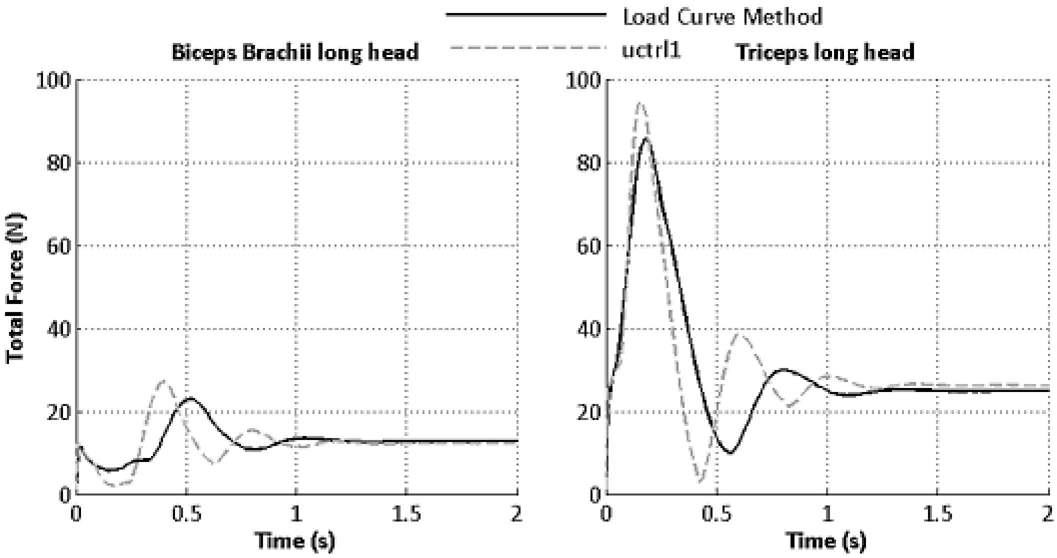


Figure 7-13 Total force produced by Biceps Brachii long head and Triceps long head in the elbow model simulation with neural delay included in both methods.

7.2 Simulation of seated Active Human Body Model

The test setup is described in Section 6.2. Presented are the results according to each body region controller (Figure 7-16 to 7-21). Due to the Chalmers AHBM being supported by the backrest, the lumbar rotation is much less than the cervical- and head rotation. This results in a much lower level of muslce activity, perhaps also attributed to the size of the muscles controlling lumbar rotation compared to the muscles controlling cervical- and head rotation. Since the backrest is in level with the clavicle (Figure 6-2), the cervical region may have some support from the backrest while the head has none. The motion of head centre of gravity relative to T1 vertebrae can be seen in Figure 7-14 (see Figure 4-3 for reference to placement of head centre of gravity and T1 vertebrae). In Figure 7-15 the rotation along the global y-axis for the head centre of gravity is seen.

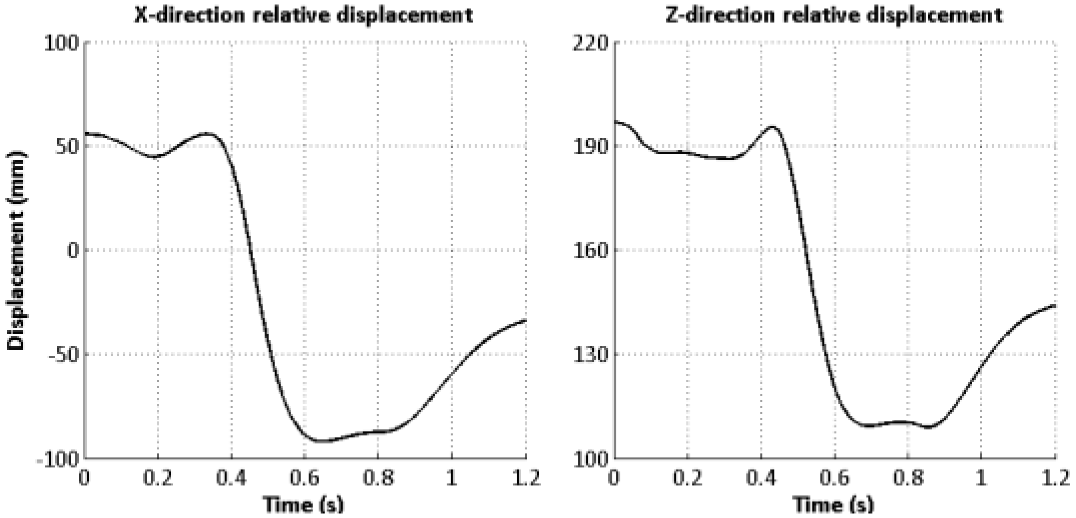


Figure 7-14 X- and Z-displacement of head centre of gravity relative to T1 vertebrae displacement using geometrical data provided in LS-PrePost.

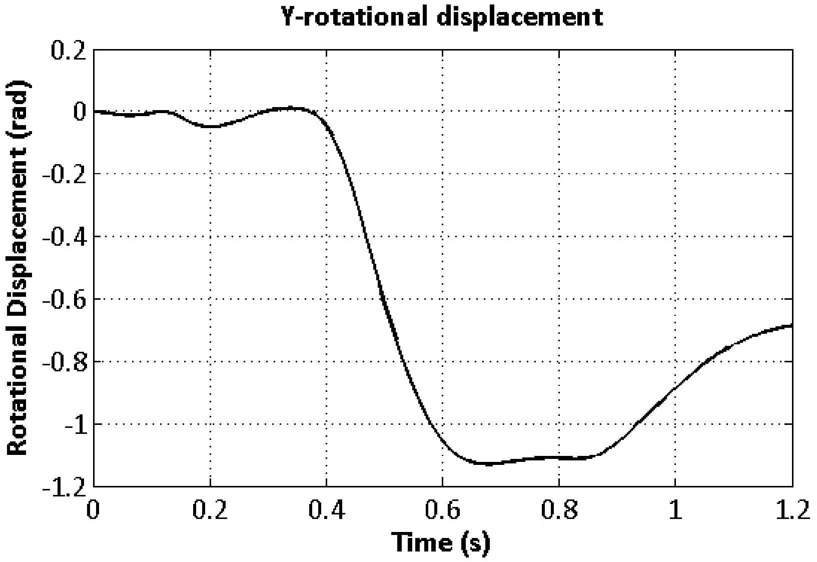


Figure 7-15 Rotational displacement of head centre of gravity around the global Y axis of the Chalmers AHBM

7.2.1 Lumbar Rotation

The result in lumbar rotation is close to identical between the two methods compared in Figure 7-16 and 7-17. This show the effect of the support from the rigid backrest. Again, the high frequency filtering properties of the neural delay filter of the LCM is shown in Figure 7-16 concerning $u(t)$.

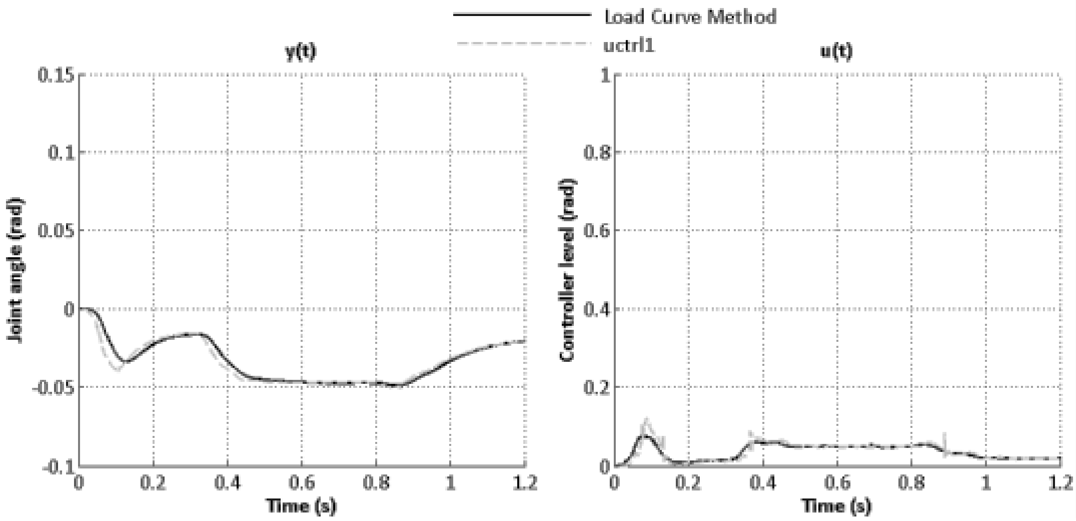


Figure 7-16 Input and output signal taken from the lumbar controller. Note that the axis scale is different when referring to plots from other controllers or from plot of Na(t).

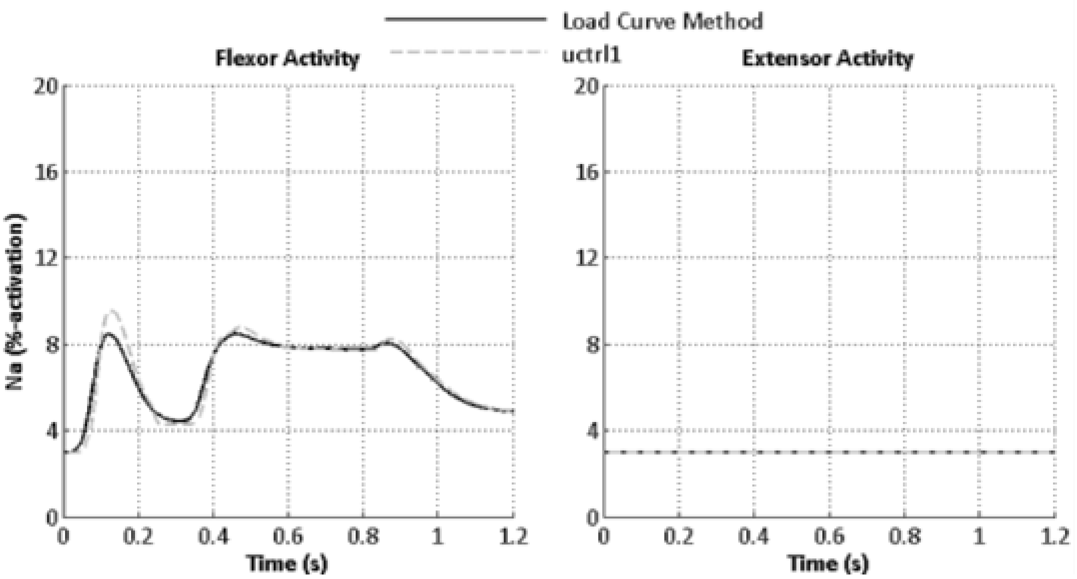


Figure 7-17 Activation level for flexor and extensor muscle in the lumbar region. Comparison between uctrl1 model and LCM model. Take notice of the properties of the axis scaling.

7.2.2 Cervical Rotation

Rotation of the cervical region, since unsupported, show a higher level of muscle activity and a greater peak rotational deviation from initial state as seen in Figure 7-18. The slight delay of the LCM compared to the uctrl1 subroutine seen at 0.6 s in Figure 7-19 reveals itself as a slightly increased peak rotational angle for the LCM as seen in Figure 7-18.

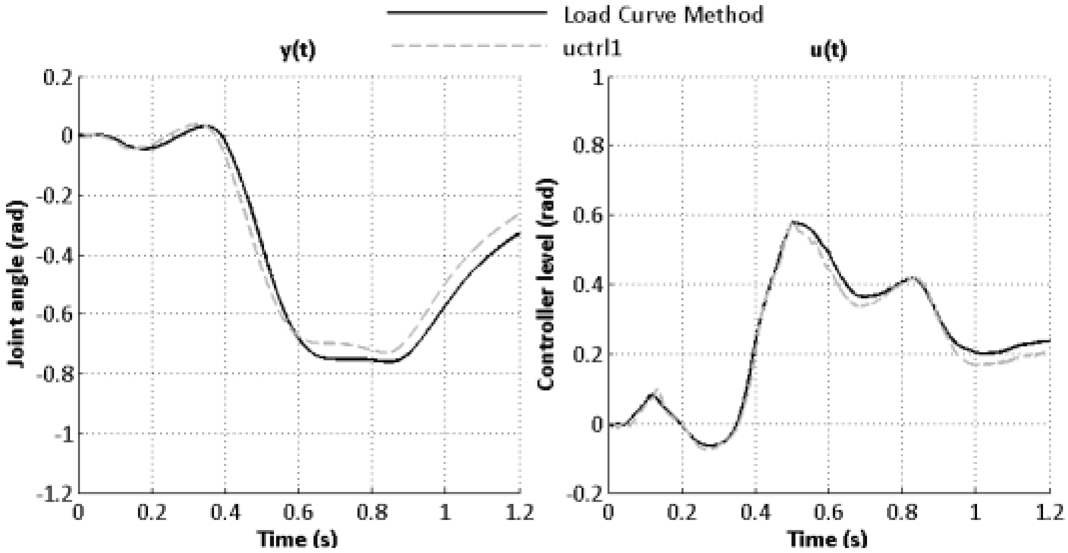


Figure 7-18 Input and output signal taken from the cervical controller. Note that the axis scale is different when referring to plots from other controllers or from plot of Na(t).

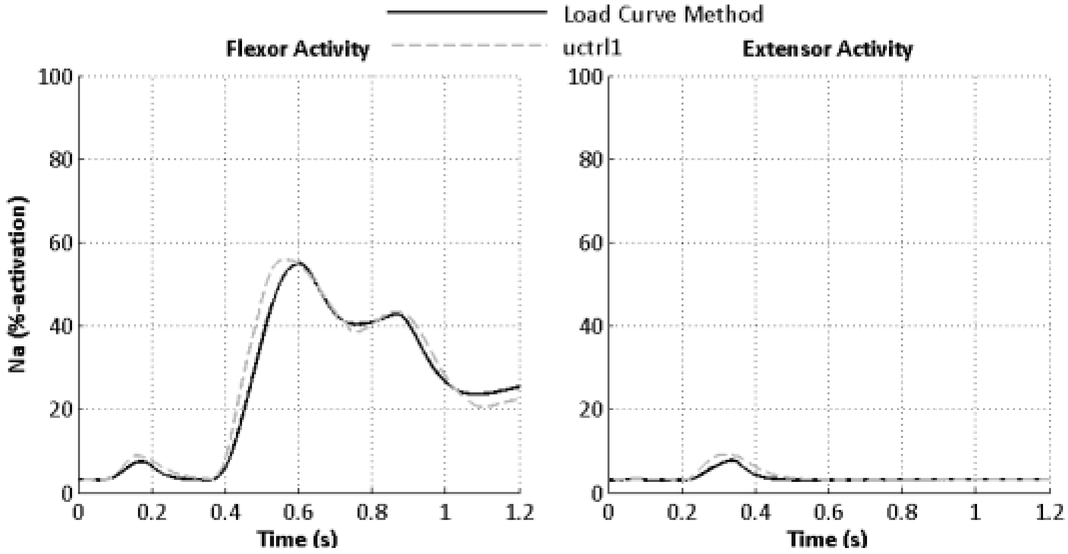


Figure 7-19 Activation level for flexor and extensor muscles in the cervical region. Comparison between uctrl1 model and LCM model.

7.2.3 Head Rotation

Even though the level of activity is increased in peak flexor activity in Figure 7-21, the difference is not shown in the same regard in resulting head rotational angle in Figure 7-20. As for the cervical region, the peak flexor activation level for the LCM is slightly lagging behind the uctrl1 subroutine. Since the resulting head rotational angle seem to deviate less between the two methods (Figure 7-20) than for cervical rotation (Figure 7-18) there is a clear indication that the support from passive body structures are limiting the rotation of the head, perhaps in greater regard than the muscular activity.

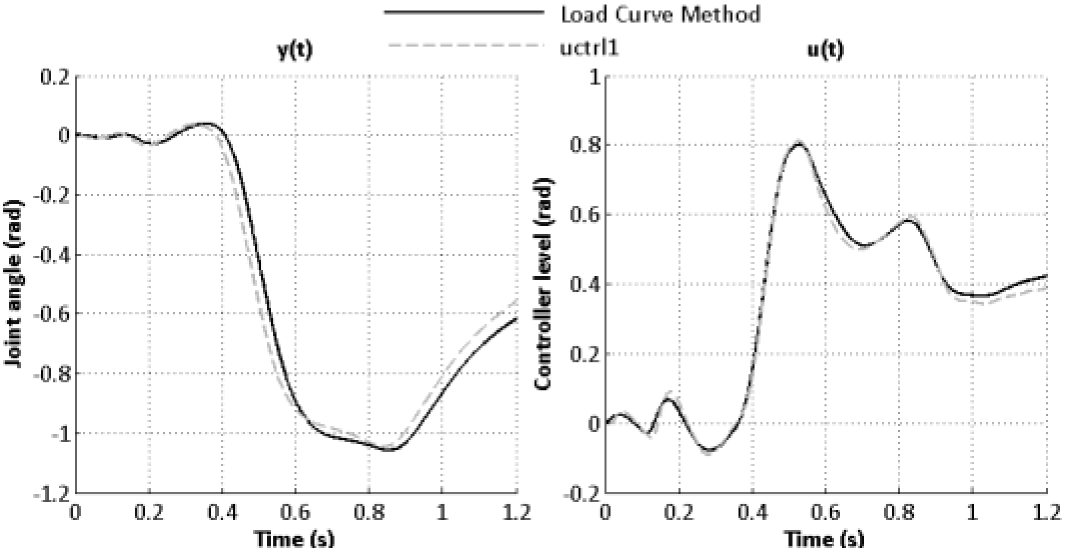


Figure 7-20 Input and output signal taken from the head controller. Note that the axis scale is different when referring to plots from other controllers or from plot of Na(t).

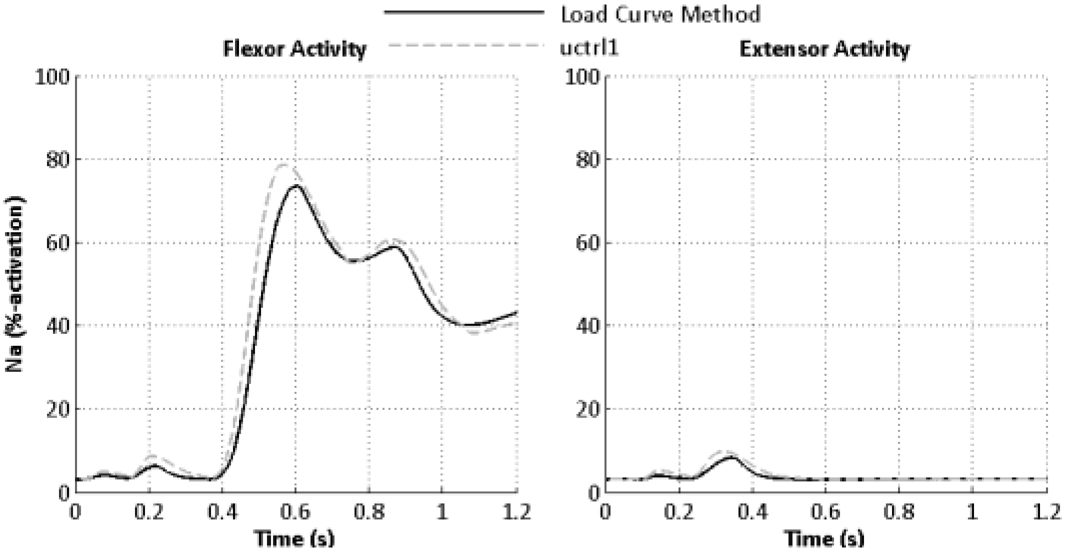


Figure 7-21 Activation level for flexor and extensor muscles in the cervical region controlling head rotation in the sagittal plane. A comparison between uctrl1 model and LCM model.

8 Discussion

The proposed LCM provides the means to implement feedback control using standard LS-DYNA keywords for the calculation of muscle activity for HBMs It may give the possibility for the Chalmers AHBM to be executed with the use of standard LS-DYNA binaries. The performance of the LCM has been compared to the uctrl1 subroutine method in simulations of elbow joint angle and Chalmers AHBM trunk muscle reactions. Alternative implementations together with remarks of the implemented LCM are discussed in this Section.

8.1 Comments to the proposed method

Alternatives are presented that have been considered but rejected for various reasons when investigating the various functions and how to model them in LS-DYNA. These discussions refer to the results in Section 7, where the performance of the functionality of the LCM is presented.

8.1.1 Input Signal

An alternative method of measuring the angle of a joint is present in LS-DYNA. The rotational displacement of a node can be read during simulation via function defined load curves, by defining the rotation in relation to another node or to a locally defined coordinate system. What proved difficult was to determine which node to use and the ability to define a local coordinate system that coincides with the rotation of the body part. The functionality in *DEFINE_CURVE_FUNCTION concerning rotation of a node allows to retrieve not only the rotational angle of a node but also the rotational velocity and rotational acceleration. This functionality might be used as means to implement a control algorithm without having to define a one degree of freedom node to perform the integrative task in the control algorithm.

Comparative plots between the two methods can be seen in Figure 7-1. The method was only tested on the elbow model and not on the seated trunk model. The results show that measuring nodal rotation is able to show the same rotational deviation from initial state as the use of vector calculations. Since nodal rotation is given as displacement from initial position, the reference angle is zero. The benefit is that the initial angle does not need to be calculated manually. When using vector calculation the angle is given in relation to the vertical axis and is initially non-zero.

8.1.2 Neural delay

The proposed neural delay implementation consisting of a first order filter is signal gain and time delay frequency dependant. While one can wish that this is in line with the characteristics of the human neural signal regime no papers have been found to support that such is the case. Meijer et al. (2007; 2012) describe a neural delay consisting of a low pass first order filter, however in their work this frequency dependant time delay is placed as part of the activation dynamic but the implementation is also thought to model the delay caused by the length of the neural axons. The basis for the choice of frequency dependence has not been further investigated but may be due to the modelling of activation dynamics. More complex filters that model a frequency independent time delay should be able to improve on the performance, however, the difficulty lies in determining an appropriate mechanical interpretation of the filter design.

The negative gain imposed on the signal by the filter in the LCM is seen in lower peak values (Figure 7-4). The lower peak values will contribute to a slower and more unstable response than compared to the uctrl1 subroutine. Several concepts other than the proposed method have been considered but rejected, due to the lack of keywords providing one or several parts of the solution. Sensor keywords (*SENSOR) in LS-DYNA provide means to determine for how long a sensor shall remain being on or off, however, in order to delay a semi-continuous signal, there would need to be as many sensors as time steps. To reduce the number of sensors, the location of a sensor could be defining when the magnitude of a certain node should be read or written to. However, since there are no means to create nodes during a simulation, all nodes would have to be predefined and would require as many nodes as time steps.

A way of improvement to the model would be to add a keyword function in LS-DYNA that defines a circle vector that can be used to store data in order to model a delay. The size of the vector and resolution of the data stored is to be user defined. An implementation possible would be to add a simple circular vector functionality to the *DEFINE_CURVE_FUNCTION keyword that is able to store information from load curves based on a defined duration of time steps or simulated time.

When defining load curves in the form of tables an offset can be introduced to the abscissa and ordinate values. An easy implementation for a delay would be if this could be a general function independent on whether the load curve is defined in the form of a table or in the form of a mathematical expression. This would allow the offset to be used as a delay to the load curve affected by the offset. Another approach would be to add the definition of a delay time in conjunction to birth- and death time in *BOUNDARY_PRESCRIBED_MOTION. Also necessary for this implementation to work would be to allow any type of defined load curve to be affected by birth-, death- and delay time, since according to the manual only predefined in the form of tables load curves are allowed to be affected by birth- and death time.

8.1.3 Control Signal

The results from the simulations indicate that the implemented approach to calculation of a control signal in the LCM is able to achieve the same result as the uctrl1 subroutine. The use of one degree of freedom nodes for $ie(t)$ and de/dt seem to impose no errors in the calculations though errors might occur if a load is applied onto the node in the direction of motion. This has not been modelled since loads in the simulations has been applied to rigid bodies only. The absence of a filter for de/dt has not brought any inconsistency or singularity of the control signal in the LCM, since higher frequencies are filtered in the current implementation of the neural delay filter.

Concerning alternative models, no forward modelling, as implied by Section 2.3.3, has been implemented. Though, the articles in the literature review point to the fact that rapid and often repeated tasks such as posture stability involve an internal, forward model in the CNS activity.

8.1.4 Activation Dynamics

Care must be taken when defining the physical appearance of the activation dynamics systems. Since the element is discrete and no contact is defined between the nodes, they are able to move past one another which means that an unloading scenario can become a loading scenario if the slave node move past the master node. For level of excitation this brings no errors in the calculations, while for the activation dynamics this characteristic affects the latency concerning loading and unloading scenarios.

The control signal output is allowed to be negative which affects the performance of the speed of change in the excitation and activation level. A negative control signal results in faster de-excitation.

In the current implementation, N_e and N_a, because of the cocontraction level being added after the $N_a(t)$ dynamics, is allowed to be negative in the LCM. The resulting activation level applied in the muscles is then correct but there are several ways in if and how to implement a cocontraction level that affect the dynamic systems of $N_e(t)$ and $N_a(t)$. The result is that the reaction is too slow for the resulting level of muscle activation. A solution would be to instead of defining Na(t) as in Equation 5.10, Na(t) is defined between zero and one. The tension in the damper connecting the nodes should have an initial level of stress according to the level of cocontraction, and that the stress should not be allowed to be below 0,5% as set in the uctrl1 subroutine for Ne and Na, seen in Figure 7-7. In the activation dynamics system, an initial velocity of the Na node can also be used to initiate a cc level, however, the cc level should be maintained throughout the simulation, as a raised ground level, but allowed to decrease for one set of muscles if the other set is increasing.

8.2 Elbow Model simulation

In Figure 7-4 it can be seen that the calculations referring to joint angle and controller output coincide with the result of the uctrl1 subroutine. This stands in contrast to the results presented in Figure 7-6 to 7-8, where activation dynamics where included and muscle activation was determined for each method accordingly, without a neural delay. Therefore, part of the reason to the different response of the LCM is probably found in the activation dynamics and the implementation of cocontraction. Even though the same time constants have been used, it might be needed to use different time constants in order to get the same response and appearance as for the subroutine.

The deactivation of $N_a(t)$ seem to be slower for the LCM than for the uctrl1 method. The reason for this is not clearly understood but shows that the model would require further adjustments. Figure 5-4 show that the implemented method is working for a step function but when comparing the output from the LCM and uctrl1 subroutine method the output differs. There are differences present in the way the cocontraction level is implemented and how the excitation and activation level is calculated but in theory the methods should achieve the same functionality. As described in Section 8.1.4 and seen in Appendix A, the input and output signal to and from the N_a- and N_e calculations are filtered as to allow different ranges for the signals. This filtering can be done in a number of ways to achieve similar output of muscle activation and it is probably this filtering that causes the muscle activation in the LCM and the uctrl1 subroutine to differ. Apart from difference in muscle activation it was seen that the computational time was decreased with approximately 30% when using the LCM compared to uctrl1 computation with neural delay modelled.

8.3 Seated Chalmers AHBM simulation

Due to the model being exposed to a greater level of applied external load, the calculation of the level of muscle activity contributions to the overall motion of the body part is less than for the elbow model. Still, because of the different approaches to neural delay and implementation in how the signals are filtered and handled in the activation dynamics, there are some differences in the LCM and the uctrl1 subroutine. As in the elbow model the response of the LCM show decreased peak level and partly therefore a slower response in the activation level. A benefit from using a low pass filter as a model for neural delay is that signal transients are effectively smoothened which makes it unnecessary to add a filter to the derivative part of the control algorithm, de/dt. Elapsed computational time was decreased by 14% when using the LCM.

9 Conclusion

The aim of this work has been to investigate the possibility to implement a feedback control algorithm using standard keywords, specifically for simulation of AHBM muscle activity in pre-crash situations. The resulting approach, called the LCM show some differences compared to the uctrl1 sub-routine controlled model. The differences are in the modelling of neural delay in LS-DYNA with the use a low-pass filter and in the model of the activation dynamics. A suggestion of how to implement a set of keywords that could provide a time delay functionality, similar to that in the uctrl1 model has been made. Such a feature would benefit the functionality in the LCM. Further work is needed in order to have the LCM model perform as the uctrl1 subroutine in the current Chalmers AHBM especially concerning the modelling of muscle activation dynamics, specifically in the modelling of muscle deactivation. The LCM show a great promise in decreasing elapsed computational time compared to the uctrl1 method. In the simulations made a decrease of 30% for the elbow model and 14% for the seated Chalmers AHBM model has been seen.

10 References

Bamford, R.B., Marteniuk, R.G. (1988). Visual Feedback, Movement Learning and the Representation of Movement. *Human Movement Science*, Vol. 7, pp. 1–26.

Bizzi, E., Hogan, N., Mussa-Ivaldi, F.A., Giszter, S. (1992). Does the Nervous System Use Equilibrium-Point control to Guide Single and Multiple Joint Movements?. *Behavioral and Brain Sciences*, Vol. 15, pp. 603–613.

Cappon, H., Mordaka, J., Van Rooij, L., Adamec, J., Praxl, N., Muggenthaler, H. (2007). A Computational Human Model with Stabilizing Spine: A Step Towards Active Safety. *Society of Automotive Engineers (SAE) 2007 World Congress*, Paper no. 2007-01-1171

Choi, H. Y., Sah, S. j., Lee, B., Cho, H. S., Kang, S. J., Mun, M. S., Lee, J. (2005). Experimental and Numerical studies of muscular activations of bracing occupant. *Proceedings 19th International Technical Conference on the Enhanced Safety of Vehicles*, Paper no. 05-0139.

de Vlugt, E., Shouten, A.C., van der Helm, F.C.T., Teerhuis, P.C., Brouwn, G.G., (2003). A force-controlled planar haptic device for movement control analysis of the human arm. *Journal of Neuroscience Methods*, Vol. 129, pp. 151–168.

Epstein, M., Frank, C. B., Herzog, W., Nigg, B. M., Shrive, N. G., van den Bogert, T. J., and Zernicke, R. F. (1999). Biomechanics of the Musculo-skeletal System (2nd ed.). (B. M. Nigg, and W. Herzog, Eds.) Calgary, Alberta, Canada: John Wiley and Sons Ltd.

Feldman, A.G., Levin, M.F., (1995). The Origin and Use of Positional Frames of Reference in Motor Control. *Behavioral and Brain Sciences*, Vol. 18, pp. 723–806.

Fressmann, D., Münz, T., Graf, O.,Schweizerhof, K., (2007). FE Human Modelling in Crash – Aspects of the numerical Modelling and Current Applications in the Automotive Industry.[Elektronisk], 6th German LS-DYNA Forum 2007, Frankenthal, Germany 11-12 October. Retrievable: http://www.dynamore.de/de/download/papers/forum07/passivesafety01/fe-human-modelling-in-crash-2013-aspects-of-the [2013-05-10]

Gandevia, S.C., Burke, D, (1992). Does the nervous system depend on kinesthetic information to control natural limb movements?. *Behavioral and Brain Sciences,* Vol. 15, pp. 614–632.

Gerdes, V.G.J, Happee, R., (1994). The use of an internal representation in fast goal-directed movements: A modelling approaSection *Biological Cybernetics*, Vol. 70, pp. 513–524.

Haith, A.M., Krakauer, J.W., (2013). Model-Based and Model-Free Mechanisms of Human Motor Learning. *Progress in Motor Control Advances in Experimental Medicine and Biology*, Vol. 782, pp. 1–21.

Hedenstierna, S. (2008). *3D Finite element modeling of cervical musculature and its effect on neck injury prevention*. Stockholm: Royal Institute of Technology, School of Technology and Health, Division of Neuronic Engineering.

Hill, A. V. (1938). The heat of shortening and the dynamic constants of muscle. London: Royal Society of London.

Iwamoto, M., Kisanuki, Y., Watanabe, I., Furusu, K.,Miki, K. (2002). Development of a Finite Element Model of the Total Human Model for Safety (THUMS) and Application to Injury Reconstruction. Proceedings of the International IRCOBI Conference, Session I, Munich, Germany, September 18–20.

Iwamoto, M., Nakahira, Y., Kimpara, H,. Sugiyama, T. (2009). Development of a human FE model with 3-D Geometry of muscles and lateral impact analysis for the arm with muscle activity. *SAE International*, 2009-01-2266. www.sae.org [2009-06-09]

Iwamoto, M., Nakahira, Y., Kimpara, H., Suguyama, T., (2012). Development of a Human Body Finite Element Model with Multiple Muscles and their Controller for Estimating Occupant Motions and Impact Responses in Frontal Crash Situations. *Stapp Car Crash Journal*. Vol. 56, pp. 231–268.

Kawato, M. (1999). Internal Models for Motor Control and Trajectory Planning. *Current Opinion in Neurobiology,* Vol.9, pp.718–727.

Kistemaker, D. A., Soest, A. J., Wong, J. D., kurtzer, I., and Gribble, P. L. (2013). Control of position and movement is simplified by combined muscle spindle and Golgi tendon organ feedback. *Journal of Neurophysiology*, Vol. 109(4), pp. 1126–1139.

Kuo, A. D. (1994). A Mechanical Analysis of Force Distribution Between Redundant, Multiple Degree-of-freedom Actuators in the Human: Implications for the central Nervous System. *Human Movement Science,* Vol.13, pp. 635–663.

Kuo, A. D. (2005). An Optimal State Estimation Model of Sensory Integration in Human Postural Balance. *Journal of Neural Engineering,* Vol. 2(3), pp. 235–249

Latash, M. L. (2011). Anticipatory control of Voluntary action - Merging the ideas of equilibrium-point control and synergic control. in Motor Control - Theories, experiments and applications, pp. 3–26.

Lennartsson, B. (2002). Reglerteknikens grunder. Studentlitteratur, Lund

LSTC, (2012a). LS-DYNA Keyword User's Manual, Vol. I, Version 971 R6.1.0.

LSTC, (2012b). LS-DYNA Keyword User's Manual, Vol. II, Material Models, Version 971 R6.1.0.

Marieb, E. N., and Hoehn, K. (2010). *Human Anatomy and Physiology* 8th issue. San Francisco: Pearson Benjamin Cummings.

Matthews, P., (1994). The simple frequency response of human stretch reflexes in which either short- or long-latency components predominate. *Journal of Physiology*, Vol. 481(3), pp. 777–798.

Meijer, R., Rodarius, C., Jiri Adamec, E. V., and Rooij, L. V. (2008). A first step in computer modelling of the active human response in a far-side impact. *International journal of crashworthiness*, vol. 13, Issue 6, special issue Advanced Protection Systems (APROSYS): Part 1, pp. 643–652.

Meijer, R., van Hassel, E., Broos, J., Elrofai, H., van Rooij, L., van Hooijdonk, P. (2012). Development of a multi-Body Human Model that Predicts Active and Passive Human Behaviour. International Research Council on Biomechanics of Injury (IRCOBI) Conference 2012, paper no. IRC-12-70.

Murakami, D., Kitagawa, Y., Kobayashi, S., Kent, R., and Crandall, J. (2004). Development and validation of a finite element model of a vehicle occupant. SAE Technical Paper series, 2004-01-0325. www.sae.org [2004-03-08]

Pick, J. A., (2004). Neuromuscular Dynamics and the Vehicle Steering Task. Diss. Cambridge University, Engineering Department.

Prüggler, A., Huber, P., Rieser, A., Steiner, K., Kirschbichler, S., and Eichberger, A. (2012). Implementation of reactive human behavior in a numerical human body model using controlled beam elements as muscle element substitutes. *Proceedings of the 22nd International Technical Conference on the Enhanced Safety of Vehicles*, Paper no. 11-0154-O.

Stroeve, S. (1997). A learning feedback and feedforward neuromuscular control model for two degrees of freedom human arm movements. *Human Movement Science* vol. 16, pp. 621–651.

van Leeuwen, J., Spierts, I., (2002). *Encyclopedia of Life Sciences; Nervous Control of Movement*. John Wiley and Sons

William, W. O. (2011). Huxley's model of muscle contraction with compliance. Journal of elasticity, 105(1–2), 365–380.

Winters, J. M., Stark, L., (1985). Analysis of Fundamental Human Movement Patterns Through the Use of In-Depth Antagonistic Muscle Models. *IEEE Transactions on Biomedical Engineerin,* Vol.32(10), pp. 826–839

Winters, J. M., Woo, S. L-Y., (1990). *Hill-Based Muscle Models: A Systems Engineering Perspective*. New York: Springer. pp. 69–93.

Wittek, A., Kajzer, J., Haug, E., (2000). Hill-type Muscle Model for Analysis of Mechanical Effect of Muscle Tension on the Human Body Response in a Car Collision Using an Explicit Finite Element Code. *The Japan Society of Mechanical Engineers: International Journal,* Vol.43(1), pp. 8–18.

Zajac, F.E., (1989). Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control. *Critical Reviews in Biomedical Engineering*, Vol. 17(4), pp. 359–411.

Örtengren, T., Hansson, H.-A., Lövsund, P., Svensson, M.Y., Suneson, A., Säljö, A., (1996). Membrane Leakage in Spinal Ganglion Nerve Cells Induced by Experimental Whiplash Extension Motion: A Study in Pigs. Journal of Neurotrauma, Vol. 13(3), pp 171–180.

Östh, J., (2010). *Active Muscle Responses in a Finite Element Human Body Model*. Lic.Eng Thesis. Department of Applied Mechanics, Chalmers University of Technology, Göteborg, Sweden.

Östh, J., Brolin, K., Happee, R., (2011a). Active muscle response using feedback control of a finite element human arm model. *Computer Methods in Biomechanics and Biomedical Engineering,* Vol. 15(4), pp. 347–361.

Östh, J., Brolin, K., Carlsson, S., Wismans, J., Davidsson, J., (2011b) The occupant response to autonomous braking: A modelling approach that accounts for active musculature. *Traffic Injury Prevention*, Vol. 13(3), pp. 265–277.

Appendix A

Code used for calculation of muscular activity level for flexor- and extensor muscles controlling elbow motion based on the elbow joint angle. The code is referred to as the LCM for the elbow model in the report.

```
*KEYWORD
*DATABASE_CURVOUT
$#      dt    binary      lcur     ioopt
  0.000100         1         0         1
*DATABASE_BNDOUT
$#      dt    binary      lcur     ioopt
  0.000100         0         0         1
*DATABASE_NODOUT
$#      dt    binary      lcur     ioopt   option1   option2
  0.000100         0         0         1     0.000         0
*DATABASE_HISTORY_NODE
$#     id1       id2       id3       id4       id5       id6       id7       id8
    200501    200503    200504    200021    200022    200601    200603    200604
$Controller node whos velocity is equal to e(t)
*BOUNDARY_PRESCRIBED_MOTION_NODE_ID
$#      id                                                               heading
         1Controller node
$#     nid       dof       vad      lcid        sf       vid     death     birth
    200020         1         0    200003  1.000000         01.0000E+28     0.000
*BOUNDARY_SPC_NODE
$#     nid       cid      dofx      dofy      dofz     dofrx     dofry     dofrz
    200021         0         0         1         1         1         1         1
$Local coordinate system measuring the flex of the lower arm versus upper arm initial
flex=0.0
*DEFINE_COORDINATE_NODES
$#     cid        n1        n2        n3      flag       dir
    200090   8530167   8530355   8530169         1         X
$------------------------------------------------------------------------------
$----------------------------Calculation of y(t)-------------------------------
$#Vector calculation of angle between humerus and ulna of arm
*DEFINE_CURVE_FUNCTION_TITLE
U_humX
$#    lcid      sidr       sfa       sfo      offa      offo    dattyp
    200041         0  1.000000  1.000000     0.000     0.000         0
$# function
CX(8530167)-CX(8540955)
*DEFINE_CURVE_FUNCTION_TITLE
```

```
U_humY
$#    lcid      sidr       sfa       sfo      offa      offo    dattyp
    200042         0  1.000000  1.000000     0.000     0.000         0
$# function
CY(8530167)-CY(8540955)
*DEFINE_CURVE_FUNCTION_TITLE
U_humZ
$#    lcid      sidr       sfa       sfo      offa      offo    dattyp
    200043         0  1.000000  1.000000     0.000     0.000         0
$# function
CZ(8530167)-CZ(8540955)
*DEFINE_CURVE_FUNCTION_TITLE
U_ulnX
$#    lcid      sidr       sfa       sfo      offa      offo    dattyp
   200044         0  1.000000  1.000000     0.000     0.000         0
$# function
CX(8530319)-CX(8530167)
*DEFINE_CURVE_FUNCTION_TITLE
U_ulnY
$#    lcid      sidr       sfa       sfo      offa      offo    dattyp
    200045         0  1.000000  1.000000     0.000     0.000         0
$# function
CY(8530319)-CY(8530167)
*DEFINE_CURVE_FUNCTION_TITLE
U_ulnZ
$#    lcid      sidr       sfa       sfo      offa      offo    dattyp
    200046         0  1.000000  1.000000     0.000     0.000         0
$# function
CZ(8530319)-CZ(8530167)
*DEFINE_CURVE_FUNCTION_TITLE
L_hum
$#    lcid      sidr       sfa       sfo      offa      offo    dattyp
    200047         0  1.000000  1.000000     0.000     0.000         0
$# function
SQRT((lc200041*lc200041)+(lc200042*lc200042)+(lc200043*lc200043))
*DEFINE_CURVE_FUNCTION_TITLE
L_uln
$#    lcid      sidr       sfa       sfo      offa      offo    dattyp
    200048         0  1.000000  1.000000     0.000     0.000         0
$# function
SQRT((lc200044*lc200044)+(lc200045*lc200045)+(lc200046*lc200046))
```

```
*DEFINE_CURVE_FUNCTION_TITLE
sp_hum_uln
$#    lcid      sidr       sfa       sfo      offa      offo    dattyp
    200049         0  1.000000  1.000000     0.000     0.000         0
$# function
lc200041*lc200044+lc200042*lc200045+lc200043*lc200046
*DEFINE_CURVE_FUNCTION_TITLE
Theta
$#    lcid      sidr       sfa       sfo      offa      offo    dattyp
    200050         0  1.000000  1.000000     0.000     0.000         0
$# function
ACOS(lc200049/(MAX(1.0E-20,lc200047*lc200048)))
*DEFINE_CURVE_FUNCTION_TITLE
Theta
$#    lcid      sidr       sfa       sfo      offa      offo    dattyp
    200052         0  1.000000  1.000000     0.000     0.000         0
$# function
AY(8530167)
*DEFINE_CURVE_FUNCTION_TITLE
Check if activation dynamics for extension is ok
$#    lcid      sidr       sfa       sfo      offa      offo    dattyp
    200053         0  1.000000  1.000000     0.000     0.000         0
$# function
cx(200503)-cx(200504)
*DEFINE_CURVE_FUNCTION_TITLE
Check if activation dynamics for flexion is ok
$#    lcid      sidr       sfa       sfo      offa      offo    dattyp
    200054         0  1.000000  1.000000     0.000     0.000         0
$# function
cx(200603)-cx(200604)
$-----------------------------------------------------------------------
$------------------------------DELAY, yd(t)-----------------------------
$ Delay is implemented in the same way as the activation dynamics, a first order
filter
$ The initial position is turned to reference by subtracting the reference angle.
Otherwise
$ the muscle will react on the initial error of y(t) and yd(t) instead of the
difference
$ between y(t) and r(t)
*DEFINE_CURVE_FUNCTION_TITLE
Theta
$#    lcid      sidr       sfa       sfo      offa      offo    dattyp
```

```
    200051         0  1.000000  1.000000     0.000     0.000         0
$# function
lc200050-lc200031
*BOUNDARY_PRESCRIBED_MOTION_NODE_ID
$#      id                                                               heading
         0y(t)
$#     nid       dof       vad      lcid        sf       vid     death     birth
    200021         1         0    200051  1.000000         01.0000E+28     0.000
*BOUNDARY_SPC_NODE
$#     nid       cid      dofx      dofy      dofz     dofrx     dofry     dofrz
    200021         0         0         1         1         1         1         1
    200022         0         0         1         1         1         1         1
*SECTION_SPRING_DAMPER
$#   secid       dro        kd        v0        cl        fd
         5         0     0.000     0.000     0.000     0.000
$#     cdl       tdl
         0         0
*ELEMENT_DISCRETE
$#   eid     pid      n1      n2     vid               s      pf          offset
       9       5  200021  200022       0        1.000000       0           0.000
*ELEMENT_MASS
$#   eid     nid            mass     pid
      10  200022   1.000000e-006       5
*MAT_DAMPER_VISCOUS_TITLE
c_yd(t)
$#     mid        dc
         5   2.00E-5
$ c=m/Td, Td=50E-3 [s] m=1E-6 [kg]=> c=2E-5, Td=100E-3 m=1E-6 => c=1E-5
*PART
$# title
yd(t)
$#     pid     secid       mid     eosid      hgid      grav    adpopt      tmid
         5         5         5         0         0         0         0         0
$-----------------------------------------------------------------------
$-----------------------------PID CONTROLLER----------------------------
$
$---Reference signal r(t)
$
*DEFINE_CURVE
$#    lcid      sidr       sfa       sfo      offa      offo    dattyp
    200031         0  1.000000  1.000000     0.000     0.000         0
```

```
$#                a1                  o1
               0.000           1.5383000
          10.0000000           1.5383000
$
$---Controller gains and initial conditions
$
$ 200032 - Cocontraction excitation
*DEFINE_CURVE_FUNCTION
$#    lcid      sidr       sfa       sfo      offa      offo    dattyp
    200032         0  1.000000  1.000000     0.000     0.000         0
$# function
0.0500000
$
$ 200033 - Controller gain kp
*DEFINE_CURVE_FUNCTION
$#    lcid      sidr       sfa       sfo      offa      offo    dattyp
    200033         0  1.000000  1.000000     0.000     0.000         0
$# function
$1.000
0.307000000
$
$ 200034 - Controller gain ki
*DEFINE_CURVE_FUNCTION
$#    lcid      sidr       sfa       sfo      offa      offo    dattyp
    200034         0  1.000000  1.000000     0.000     0.000         0
$# function
$0.700
0.000000000
$
$ 200035 - Controller gain kd
*DEFINE_CURVE_FUNCTION
$#    lcid      sidr       sfa       sfo      offa      offo    dattyp
    200035         0  1.000000  1.000000     0.000     0.000         0
$# function
$0.200
0.06865285
$-----------------------------------------------------------------------
$------------------------CONTROL ALGORITHM----------------------------
$y(t) is defined as positive for flexion and negative for extension from initial position
*DEFINE_CURVE_FUNCTION_TITLE
```

```
yd(t)
$#    lcid      sidr       sfa       sfo      offa      offo    dattyp
    200001         0  1.000000  1.000000     0.000     0.000         0
$# function
$lc200050
vx(200022)
$AY(8530167)
*DEFINE_CURVE_FUNCTION_TITLE
r(t)
$#    lcid      sidr       sfa       sfo      offa      offo    dattyp
    200002         0  1.000000  1.000000     0.000     0.000         0
$# function
lc200031
$0.0
*DEFINE_CURVE_FUNCTION_TITLE
e(t)=r(t)-y(t)
$#    lcid      sidr       sfa       sfo      offa      offo    dattyp
    200003         0  1.000000  1.000000     0.000     0.000         0
$# function
-lc200001
$lc200002-lc200001
*DEFINE_CURVE_FUNCTION_TITLE
ie(t)
$#    lcid      sidr       sfa       sfo      offa      offo    dattyp
    200004         0  1.000000  1.000000     0.000     0.000         0
$# function
dx(200020)
*DEFINE_CURVE_FUNCTION_TITLE
de/dt
$#    lcid      sidr       sfa       sfo      offa      offo    dattyp
    200005         0  1.000000  1.000000     0.000     0.000         0
$# function
accx(200020)
*DEFINE_CURVE_FUNCTION_TITLE
u(t)=Kp*e(t)+Ki*ie(t)+Kd*de/dt
$#    lcid      sidr       sfa       sfo      offa      offo    dattyp
    200006         0  1.000000  1.000000     0.000     0.000         0
$# function
lc200033*lc200003+lc200034*lc200004+lc200035*lc200005
$-----------------------------END OF PID CONTROLLER----------------------------
$------------------------------------------------------------------------------
```

```
$-----------------------------ACTIVATION DYNAMICS------------------------------
$------The signal is negated since u(t)<0 when the arm flexes (thus provide positive
signal to the extensors)
$       lc200300 and 400 is sent to the coupled first order filters to add activation
dynamics to the signal
*DEFINE_CURVE_FUNCTION_TITLE
u for extensors
$#    lcid      sidr       sfa       sfo      offa      offo    dattyp
    200300         0  1.000000  1.000000     0.000     0.000         0
$# function
-lc200006
*DEFINE_CURVE_FUNCTION_TITLE
u for flexors
$#    lcid      sidr       sfa       sfo      offa      offo    dattyp
    200400         0  1.000000  1.000000     0.000     0.000         0
$# function
lc200006
$----------------------------------------------------------------------------------
$                      COUPLED 1ST ORDER FILTER EXTENSION
$----------------------------------------------------------------------------------
*BOUNDARY_PRESCRIBED_MOTION_NODE_ID
$#      id                                                                 heading
         0u(t)
$#     nid       dof       vad      lcid        sf       vid     death     birth
    200501         1         0    200300  1.000000         01.0000E+28     0.000
*BOUNDARY_PRESCRIBED_MOTION_NODE_ID
$#      id                                                                 heading
         0Ne(t)
$#     nid       dof       vad      lcid        sf       vid     death     birth
    200503         1         0    200310  1.000000         01.0000E+28     0.000
*BOUNDARY_SPC_NODE
$#     nid       cid      dofx      dofy      dofz     dofrx     dofry     dofrz
    200501         0         0         1         1         1         1         1
    200502         0         0         1         1         1         1         1
    200503         0         0         1         1         1         1         1
    200504         0         0         1         1         1         1         1
*DEFINE_CURVE_FUNCTION_TITLE
Ne(t)
$#    lcid      sidr       sfa       sfo      offa      offo    dattyp
    200310         0  1.000000  1.000000     0.000     0.000         0
$Func
min(max(VX(200502),-lc200032-0.005),1.0)
```

```
*DEFINE_CURVE_TITLE
F-v
$function for loading and unloading scenarios of Tna, c=m/Tnaa (F>0,v>0) and c=m/Tnad
(F<0,v<0)
$#    lcid      sidr       sfa       sfo      offa      offo    dattyp
    200314         0  1.000000  1.000000     0.000     0.000         0
$#                a1                  o1
             -1.00E4              -0.333
                   0                   0
              1.00E4               2.000
*SECTION_SPRING_DAMPER
$#   secid       dro        kd        v0        cl        fd
         1         0     0.000     0.000     0.000     0.000
$#     cdl       tdl
         0         0
*SECTION_SPRING_DAMPER
$#   secid       dro        kd        v0        cl        fd
         2         0     0.000     0.000     0.000     0.000
$#     cdl       tdl
         0         0
*ELEMENT_DISCRETE
$#   eid     pid      n1      n2     vid               s      pf          offset
       1       1  200501  200502       0        1.000000       0           0.000
       2       2  200503  200504       0        1.000000       0           0.000
*ELEMENT_MASS
$#   eid     nid            mass     pid
       3  200502   1.000000e-006       1
       4  200504   1.000000e-006       2
*MAT_DAMPER_VISCOUS_TITLE
TNe=M/DC
$#     mid        dc
         1 2.8571E-5
*MAT_DAMPER_NONLINEAR_VISCOUS_TITLE
TNa=M/DC
$#     mid      lcdr
         2    200314
$ If time is defined in ms, the dc constants should be 2.8571E-8 and 2.0000E-7
$ In accordance with Tne=35ms and Tnaa=5ms, Tnad=30ms
*PART
$# title
TNe
```

```
$#     pid     secid       mid     eosid      hgid      grav    adpopt      tmid
         1         1         1         0         0         0         0         0
*PART
$# title
TNa
$#     pid     secid       mid     eosid      hgid      grav    adpopt      tmid
         2         2         2         0         0         0         0         0
$-------------------------------------------------------------------------------
$                      COUPLED 1ST ORDER FILTER FLEXION
$-------------------------------------------------------------------------------
*BOUNDARY_PRESCRIBED_MOTION_NODE_ID
$#      id                                                             heading
         0u(t)
$#     nid       dof       vad      lcid        sf       vid     death     birth
    200601         1         0    200400  1.000000         01.0000E+28     0.000
*BOUNDARY_PRESCRIBED_MOTION_NODE_ID
$#      id                                                             heading
         0Ne(t)
$#     nid       dof       vad      lcid        sf       vid     death     birth
    200603         1         0    200410  1.000000         01.0000E+28     0.000
*BOUNDARY_SPC_NODE
$#     nid       cid      dofx      dofy      dofz     dofrx     dofry     dofrz
    200601         0         0         1         1         1         1         1
    200602         0         0         1         1         1         1         1
    200603         0         0         1         1         1         1         1
    200604         0         0         1         1         1         1         1
*DEFINE_CURVE_FUNCTION_TITLE
Ne(t)
$#    lcid      sidr       sfa       sfo      offa      offo    dattyp
    200410         0  1.000000  1.000000     0.000     0.000         0
$Func
min(max(VX(200602)/1.73,-lc200032-0.005),1.0)
*DEFINE_CURVE_TITLE
F-v
$function for loading and unloading scenarios of Tna, c=m/Tnaa (F>0,v>0) and c=m/Tnad
(F<0,v<0)
$#    lcid      sidr       sfa       sfo      offa      offo    dattyp
    200414         0  1.000000  1.000000     0.000     0.000         0
$#                a1                  o1
             -1.00E4              -0.333
                   0                   0
```

```
              1.00E4               2.000
*SECTION_SPRING_DAMPER
$#   secid       dro        kd        v0        cl        fd
         3         0     0.000     0.000     0.000     0.000
$#     cdl       tdl
         0         0
*SECTION_SPRING_DAMPER
$#   secid       dro        kd        v0        cl        fd
         4         0     0.000     0.000     0.000     0.000
$#     cdl       tdl
         0         0
*ELEMENT_DISCRETE
$#   eid     pid      n1      n2     vid               s      pf          offset
       5       3  200601  200602       0        1.000000       0           0.000
       6       4  200603  200604       0        1.000000       0           0.000
*ELEMENT_MASS
$#   eid     nid            mass     pid
       7  200602   1.000000e-006       3
       8  200604   1.000000e-006       4
*MAT_DAMPER_VISCOUS_TITLE
TNe=M/DC
$#     mid        dc
         3 2.8571E-5
*MAT_DAMPER_NONLINEAR_VISCOUS_TITLE
TNa=M/DC
$#     mid      lcdr
         4    200414
$ If time is defined in ms, the dc constants should be 2.8571E-8 and 2.0000E-7
$ In accordance with Tne=35ms and Tnaa=5ms, Tnad=30ms
*PART
$# title
TNe
$#     pid     secid       mid     eosid      hgid      grav    adpopt      tmid
         3         3         3         0         0         0         0         0
*PART
$# title
TNa
$#     pid     secid       mid     eosid      hgid      grav    adpopt      tmid
         4         4         4         0         0         0         0         0
$-----------------------------------------------
```

```
$lc200105 The result from the activation dynamics is called and used to provide ALM
for the muscles in the arm
$------
$(cocon+Na(t)>0.0)
*DEFINE_CURVE_FUNCTION_TITLE
Extension filter ALM
$#    lcid      sidr       sfa       sfo      offa      offo    dattyp
    200301         0  1.000000  1.000000     0.000     0.000         0
$# function
max(vx(200504)+lc200032+0.005,0.0)
*DEFINE_CURVE_FUNCTION_TITLE
Extension filter ALM<1.0
$#    lcid      sidr       sfa       sfo      offa      offo    dattyp
    200302         0  1.000000  1.000000     0.000     0.000         0
$# function
MIN(lc200301,1.0)
$(cocon+Na(t)>0.0)
*DEFINE_CURVE_FUNCTION_TITLE
Flexion ALM
$#    lcid      sidr       sfa       sfo      offa      offo    dattyp
    200401         0  1.000000  1.000000     0.000     0.000         0
$# function
max(vx(200604)+lc200032+0.005,0.0)
*DEFINE_CURVE_FUNCTION_TITLE
Flexion ALM
$#    lcid      sidr       sfa       sfo      offa      offo    dattyp
    200402         0  1.000000  1.000000     0.000     0.000         0
$# function
MIN(lc200401,1.0)
$------------------------END OF ACTIVATION DYNAMICS----------------------------
$------------------------------------------------------------------------------
*NODE
$#   nid               x               y               z      tc      rc
$  200090      61.0000229    -206.4037170     209.0350952       0       0
  200020             0.0        1003.000           210.0       0       0
  200021             0.0        1002.000           210.0       0       0
  200022             1.0        1002.000           210.0       0       0
  200501           0.000        1000.000           210.0       0       0
  200502        1.000000        1000.000           210.0       0       0
  200503          10.000        1000.000           211.0       0       0
  200504        0.000000        1000.000           211.0       0       0
  200601          0.000        1001.000          210.0       0       0
  200602       1.000000        1001.000          210.0       0       0
  200603         10.000        1001.000          211.0       0       0
  200604       0.000000        1001.000          211.0       0       0
*END
```

Appendix B

The uctrl1 subroutine is used to add a PID regulator for the muscular activity level in the Chalmers AHBM. First and foremost to add a possibility to keep the posture during simulations with low loads such as in evasive maneuvers and with a time duration greater than a few hundred milliseconds.

The structure is as follows:

1. Uctrl1
    - 1.1. Memory allocation, variable initiation
    - 1.2. Vector initiation.
        - 1.2.1.External load curve id for left and right hand side
            - 1.2.1.1. Reference
            - 1.2.1.2. Length
            - 1.2.1.3. Strain
            - 1.2.1.4. Controller signal
            - 1.2.1.5. Excitation
            - 1.2.1.6. Activation
        - 1.2.2.External and internal rostral and caudal node ids
            - 1.2.2.1. External rostral left hand side
            - 1.2.2.2. Internal rostral right hand side
            - 1.2.2.3. External caudal left hand side
            - 1.2.2.4. Internal caudal right hand side
        - 1.2.3.Initial length of each left and right hand side muscle
        - 1.2.4.Muscle recruitment vector
    - 1.3. Dump AHBM data control, to allow restart
        - 1.3.1.If not restart, don't load dump data
        - 1.3.2.If restart, load dump data from ahbm_dump
    - 1.4. Calculation of control signals (r.791)
        - 1.4.1. If cycle = 1 then calculate the current angles to be used as reference for the coming cycles
        - 1.4.2. Else calculate the angular difference between current angle and the reference angle from first cycle
    - 1.5. Neural delays for controller signal
        - 1.5.1.Calculate the sample interval that the delayed signals are updated at.
        - 1.5.2. Calculate the number of cycles that the muscles shall not react due to neural delay given the sample interval of the delay vector, nds=Tde in model keyword file.
        - 1.5.3.If cycle = 1, the current time is set as reference for start of counting down to the next update of the delayed signal. The first value in the delay vector is set accordingly. The delay vectors for each controller is updated with the angular difference as calculated in 1.4.2.
        - 1.5.4.If time is such that the delay sample time has passed since the last update then the delay vectors for each controller is updated with the difference calculated in 1.4.2. Delay_time is updated and the reference position in the delay vectors steps one step up.

1.5.4.1. If the current node value of the delay vector is greater than the delay vector length then the node value of the delay vector is set to 1.

1.5.5. If the number of delay samples is greater than the node value of the delay vectors then the delayed control signal is set to the value in the delay vector that corresponds to length of the delayed vector plus current node value of the delayed vector minus number of delayed samples (as calculated in 1.5.2)

1.5.5.1. Else the delayed control signal is set to the value in the delay vector that corresponds to the current node value of the delayed vector minus the number of delayed samples

1.6. Calculate controller errors, $e(t) = y_d(t) - r(t)$

1.6.1. The difference between the calculated change in nodal position and the calculated delayed control signal value

1.7. Set initial integral of error, $ie(t)$

1.7.1. Calculate and store the error for the first cycle for each controlled region

1.8. Calculate integral of error, $ie(t)$

1.8.1. The error from the last calculation plus the old integration plus the new integration times the time step size between n and n+1.

1.9. Calculate $\frac{de}{dt}$ with 1st order filter, $\frac{d}{dt}e(k) = \frac{d}{dt}e(k-1) + \frac{dt}{dt+T_f}\left(\frac{e(k)-e(k-1)}{dt} - \frac{d}{dt}e(k-1)\right)$

1.10. Calculate the control signal, $u(t) = \left(e(t)K_p + ie(t)K_i + \frac{de}{dt}K_d\right)x_{dir}$

1.10.1. For the first cycle the positive X-direction needs to be checked so that coming updates of nodes are done in the correct direction, x_{dir} , stored as 43999996.

1.10.2. Update fvalold for use in the next calculation

1.11. Compute and update muscle length and muscle strain

1.11.1. Store initial length as reference

1.11.1.1. Extract the muscle length from the u_*hs vectors

1.11.1.2. Compute reference muscle length from initial vectors

1.11.1.3. Update current muscle lengths to external muscle load curve length vector

1.11.1.4. Extract vectors u_*hs between the positions of the muscle nodes

1.11.1.5. Compute muscle length

1.12. Compute controller signals, same for both sides of the body $ctrl_{new} = max(u_l(t)\overline{recr_l} + u_c(t)\overline{recr_c} + u_h(t)\overline{recr_h}, 0) + (cc_l|\overline{recr_l}| + cc_c|\overline{recr_c}| + cc_h|\overline{recr_h}|)$

1.12.1. $\overline{recr_x}$, is the muscle recruitment vector that decide which muscle should act and in what direction. cc, is a constant that currently is set to 0.

1.13. Solve differential Equation for muscle excitation levels for right hand- and left hand side

1.13.1. If cycle =1, excitation=controller

1.13.2. else

$$exc_{new} = \min\left(\max\left(exc_{old} + \left(ctrl_{new} - \frac{exc_{old}}{t_{exc} + dt}\right)dt, 0\right), 1\right)$$

1.14. Solve differential Equation for muscle activation levels

1.14.1. If cycle = 1, Activation =excitation

1.14.1.1. Else, if new activation is greater than the old, then set the activation level between 0 and 1

$$act_{new} = \min\left(\max\left(act_{old} + \left(exc_{new} - \frac{act_{old}}{t_{act} + dt}\right)dt, 0\right), 1\right)$$

1.14.1.2. Else, set new activation according to

$$act_{new} = \min\left(\max\left(act_{old} + \left(exc_{new} - \frac{act_{old}}{t_{deact} + dt}\right)dt, 0\right), 1\right)$$

1.15. Update fvalold vectors

1.16. Write data to files

2. Csvfile_initialize