

# Contents

<b>I</b>	<b>Introduction</b>	<b>5</b>
1	Motivation	7
2	State of the art and aims	9
3	Own contributions	11
4	Overview	13
<b>II</b>	<b>Digital human</b>	<b>15</b>
<b>5</b>	<b>Comparative Study: Modeling Upper Arm Movement</b>	<b>17</b>
5.1	Introduction . . . . .	18
5.2	Experimental Study . . . . .	22
5.3	Models . . . . .	25
5.4	Results and Discussion . . . . .	32
5.5	Conclusion . . . . .	40
<b>6</b>	<b>Formulation of electrophysiology and muscle contraction</b>	<b>43</b>
6.1	State of the art . . . . .	43
6.2	discretization . . . . .	43
6.3	Standard for formulation of models . . . . .	43
6.4	Bidomain . . . . .	43
6.5	Multidomain . . . . .	43
6.6	Formulation of solid mechanics . . . . .	44
6.7	Simulation workflow . . . . .	44
<b>7</b>	<b>Numeric</b>	<b>45</b>
7.1	Fundamentals of the discretization . . . . .	45
7.2	Finite element method . . . . .	45
7.3	Incompressible nonlinear solid mechanics . . . . .	45
7.4	Numerical schemes . . . . .	45
7.5	time stepping schemes . . . . .	45
7.6	linear system solvers . . . . .	45
7.7	newton algorithm . . . . .	45

<b>8</b>	<b>Simulation software</b>	<b>47</b>
8.1	General aspects	47
8.2	Detail	47
<b>9</b>	<b>Simulation software</b>	<b>49</b>
9.1	State of the art	49
9.2	OpenCMISS	49
9.3	The simulation framework <i>opendihu</i>	49
9.4	Overview of discretization features	49
9.5	Input and output	50
9.6	In-situ visualization	50
9.7	Details on the implementation	50
9.8	Representation of structured meshes	50
9.9	Parallel data handling	50
9.10	Implementation of boundary conditions	50
9.11	Implementation of material formulation and automatic derivation of derivatives	50
9.12	efficient and configurable data transfer	50
9.13	Mapping between meshes	51
9.14	An efficient solver to the Monodomain equation	51
9.15	Details on the partitioning	51
9.16	Parallelization of the multidomain system	51
9.17	load balancing	51
<b>10</b>	<b>External coupling</b>	<b>53</b>
10.1	coupling with FEBio	53
10.2	coupling with AceFEM	53
<b>11</b>	<b>An algorithm for mesh generation</b>	<b>55</b>
11.1	State of the art	55
11.2	Data preprocessing	55
11.3	Outline of the algorithm	55
11.4	Plane mesh smoothing	55
11.5	Parallelization	55
11.6	Generation of fiber meshes	55
<b>12</b>	<b>Simulation scenarios</b>	<b>57</b>
12.1	Simulation of electromyography with fiber-based formulation	57
12.2	Simulation of electromyography with multidomain formulation	57
12.3	Muscle contraction with fiber-based formulation	57
12.4	Muscle contraction with multidomain formulation	57
<b>13</b>	<b>Technical simulation results</b>	<b>59</b>
13.1	Runtime evaluation	59
13.2	parallel partitioning	59
13.3	memory consumption	59

CONTENTS	3
<b>14 biomechanical simulation results</b>	<b>61</b>
14.1 Simulation of fatigue . . . . .	61
14.2 Simulation of EMG . . . . .	61
<b>III Soft tissue robotics</b>	<b>63</b>
<b>15 ...</b>	<b>65</b>
<b>IV Conclusion</b>	<b>67</b>



# **Part I**

## **Introduction**



# 1 Motivation

- great picture, digital human vision, connection to soft tissue robotics
- EMG, parameter estimation, see motivation of IRTG paper
- MU estimation





## **2 State of the art and aims**



## 3 Own contributions

- own contributions



## 4 Overview

- structure of following sections



# **Part II**

## **Digital human**





## 5 Comparative Study: Modeling Upper Arm Movement

Moving one's upper arms and forearms is an action that is performed unnoticed every day. What seems like a trivial task involves a sophisticated interplay of muscles, tendons, bones and joints. Macroscopic behaviour, such as mechanical properties of fibers and tissues, as well as microscopic mechanisms, such as molecular-scale processes inside biological cells and changes in electric potential across muscle fiber membranes contribute to the overall, versatile, human musculoskeletal system.

Understanding this system well allows to use observations to make predictions. Using observations and predictions of the musculoskeletal system, we can design safe assistive robotic devices. Such robotic devices, in the form of exoskeletons, can potentially support humans in strenuous, unhealthy tasks that pose high loads on the human skeleton. An example is the precise handling of heavy objects, which is required in various industries and can only be done by human. Moreover, they can help to restore muscle function in a rehabilitation therapy.

When developing models for such predictions, various choices have to be made. Relevant properties of the muscular system have to be identified. Based on a physiological understanding, essential relationships have to be selected. Phenomenologic relations can be incorporated. Mathematical formulations and numerical algorithms have to be found.

Model formulations can be differentiated by how much they are based on biophysical insights compared to raw experimental observations. This chapter presents two different approaches that use a relatively high proportion of experimental observations combined with some biophysically justified relations. The two approaches are compared in an experimental study of forearm movement.

After the models described in this chapter, the remainder of this thesis is mainly concerned with models that have a higher proportion of biophysical insight. These models include more details of the physiology of the musculoskeletal system.

## 5.1 Introduction

Actuated orthoses and prosthesis help rehabilitation patients to regain their ability to move arms and legs when muscles have lost their full function. The dysfunction can be a result of muscular or nervous diseases, e.g., after stroke, or originate from amputation of part of the limb. The rehabilitation or prosthetic device is firmly attached to the body. Powered actuators at the joints support or replicate the natural movement of the limb.

Exoskeletons are similar devices that typically extend to a larger part of the human body. Apart from rehabilitation they are used as assistive, haptic or teleoperation device. Details can be found in [\[Per07\]](#).

For the actuation to be supportive and helpful, the device has to determine the intended movement of the limb. If muscles are functioning at least residually, EMG signals can be captured from the skin surface. They can be interpreted to determine finely graduated levels of force.

For this purpose, mathematical models are required that, given EMG measurements, predict joint torques for the system of limb segments and muscles. Because of the variety of muscle characteristics among humans, such models have to be patient-specific in order to be safe and effective for the particular individual.

In the following study, two different approaches for formulating such models, A and B, are developed. Instances of these two models are parameterized for a particular healthy subject.

The specific task in this study is to predict movements of a human upper arm. The arm is flexed and extended under varying loads and with varying velocities. Measured EMG signals on the agonist and antagonist muscles should be used to predict the torque in the elbow joint. Considering the application of a supportive orthosis or exoskeleton, the predicted torque value is the control input to the actuator at the joint.

Prior to online application of the models, an offline training-phase is carried out, where all required parameter values get identified for the subject. After the two computational

models have been trained, we perform validation experiments and compare the output of the models with measurements from the real system.

The first model approach, A, is a non-parametric, data-driven model. It uses the captured information from the training phase to construct a map between input and output values. It is based on Gaussian Process Regression.

The second approach, B, uses biophysically informed models of individual muscles together with the kinematics of the overall system. This approach requires a set of subject-specific parameters which is determined in the training phase. The model is based on the commonly used Hill-type muscle model.

### 5.1.1 Related Works

Numerous experimental studies of flexion and extension of the upper arm with the aim to predict elbow torques can be found in the literature. The studies presented in the following all include a Hill-type muscle model; such a model is also present in approach B of the present study.

In [Ros99], an exoskeleton across the elbow joint on the forearm is used as a passive measurement device. Experiments with lifting weights are performed and EMG is captured. Two different models are evaluated to compare their performance in predicting moments and, thus, their fitness for exoskeleton control. The first model is Hill-based, similar to model B in the study of this work. The second model is data-driven, as is model A in our study. However, the method is different, the authors use a neural network.

The study reveals that the neural network is easier to setup but only works for the space defined by the learning data set. The advantage of the Hill-based model is that it is universal and not task dependend.

Further studies using neural networks to estimate muscle activations and elbow torques are presented by [Lin02] and [Son05]. The paper of [Ros01] focuses on an exoskeleton that supports the forearm in lifting heavy weights. Different control strategies are investigated. A naturally feeling human machine interface is achieved when control input is taken from processed EMG measurements and moment feedback of the external load. The predictions are based on a generic Hill-type model. This result is promising as it shows that neural control of exoskeletons is possible, even using non-customized models.

The goal and setup of all these studies is similar to the work presenting in the following sections. Differences are, apart from different setups, that they use state-less Hill-type models instead of the more advanced Hill-based models in our study. Furthermore, they do not use subject specific parametrizations. Both improvements can lead to better predictions of the moments in the elbow.

Several studies of predicting joint torques using Hill-type models exist in the literature that optimize model parameters to fit a specific subject. The authors of [Cav05]; [Cav06] study a scenario where a weight is lifted by the forearm. They use a genetic algorithm to find subject specific parameters to the models. Similar studies are given in [Llo03]; [Ven05]; [Pon09]; [Sar12].

One study is performed by [Hei03]. The authors include models of activation dynamics, Hill-type muscle contraction and musculo-skeletal geometry and restrict the scenario to isometric tasks. They optimize parameters for different subjects and determine the importance of parameters for good model predictions. It is found that the predictive quality of the model decreases with its complexity, however a model with seven parameters still has reasonable validity. In contrast to this study that only predicts static cases, our study also includes muscle dynamics and has more parameters to describe all required muscle properties.

Another detailed study is performed by [Fal16]. They estimate muscle model parameters of the knee joint actuators involving 23 degrees of freedom considering eight flexors and four extensors. Just like our study with model approach B, EMG signals and motion capture data is used to solve an optimization problem to fit the model. A difference is that three-element Hill-type models are used, whereas our study is based on more detailed, four-element Hill-type models but includes a smaller number of muscles.

Hill-based models of the muscle-tendon complex can also be parameterized without using EMG data. [Gar03] estimate characteristic parameters of 26 major muscles around shoulder, elbow and wrist in a two-phase optimization procedure. This approach uses individual experiments to identify different parameters. A method that requires less experiments is the ISOFIT method presented by [Wag05]. They use non-linear regression to fit Hill-type model parameters for various muscles from only 6-8 isovelocity contractions.

The authors of [Van14] develop a new method for estimating a subject-specific model of muscles around the knee which achieves higher accuracy than [Gar03] and is robust with respect to noisy data. Two improvements are that they use physiological constraints in the parameter optimization process and a heuristic for the initial guess of the parameters.

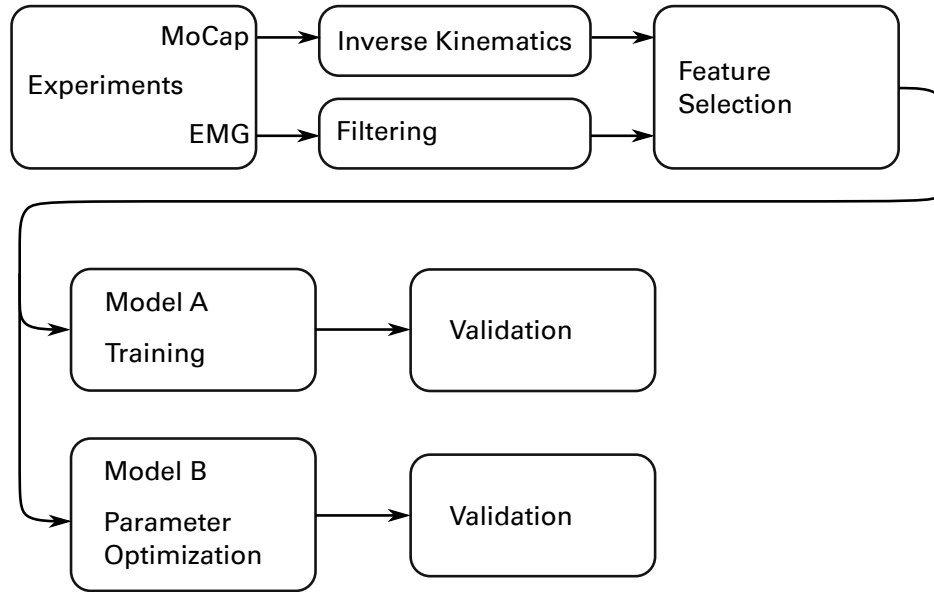


Figure 5.1: Schematic workflow of data processing. In experiments, Motion Capture (MoCap) data and EMG signals were recorded. They were processed using inverse kinematics and filtering techniques, respectively. The sizes of the large datasets were reduced by selecting specific features. Those were used as training inputs for the two models A and B. The trained models were validated using a validation dataset that also originated from experiments.

The former is also done in the present study. Concerning the latter, we base our initial guess on literature values.

### 5.1.2 Contribution Statement

The work presented in this Sec. was performed in collaboration by several members of the IRTG “Soft Tissue Robotics”. A summary can be found in the publication [Wal20], where also the list of contributors can be seen. The experiments were conducted during the IRTG’s Summer School in 2019 at the University of Auckland, New Zealand, where all involved researchers were present. Derivation and programming of the models as well as evaluation and discussion of the results was done in smaller expert groups as well as plenary video calls before and after this event at both the Universities of Stuttgart and Auckland.

An overview of the workflow from experiments to the models A and B and their validation is visualized in Fig. 5.1. This author contributed mainly to the fields of data processing, especially filtering of EMG signals and feature selection, derivation and training

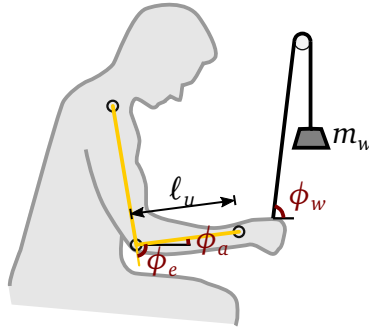


Figure 5.2: Experimental setup for the triceps trials. The subject pulls down a rope over a pulley which is connected to a weight with mass  $m_w$ . Angles required for the kinematic formulation are the elbow angle,  $\phi_e$ , the forearm angle,  $\phi_a$ , and the angle of the weight,  $\phi_w$ . The length of the ulna bone is denoted by  $\ell_u$ .

of both models, A and B, their validation and the overall programming and visualization of the results. The respective fields are presented more detailed in the following, corresponding sections.

### 5.1.3 Structure of this Chapter

[Section 5.2](#) gives an overview over the experiments, data processing and feature selection, which resulted in the required datasets. In [Sec. 5.3](#), the two models, A and B are described. Results including the validation of the models and a discussion are given in [Sec. 5.4](#). A conclusion follows in [Sec. 5.5](#).

## 5.2 Experimental Study

At first, details about the experimental study are given. Experiments are required to identify the model parameters for the particular subject. First, the experimental setup is described, second, details on the processing of the measured values are given.

### 5.2.1 Experimental Trials

In a series of experiments, eight different actions of flexing and extending the elbow were performed by the subject. Weights of 3 kg and 5 kg were hold in the hand during the elbow flexion trials. For the elbow extension trials, a pulley system was installed that

redirected the force of the weight such that the downward movement of the forearm acted against the direction of the force. This is shown in Fig. 5.2. A detailed description of the experimental trials can be found in [Wal20].

Time series of position and velocity of upper arm and forearm were recorded using a Motion Capture system. It consisted of eight cameras that tracked three markers placed on shoulder, elbow and wrist of the subject.

The elbow torque,  $\tau$ , was computed as

$$\tau = m_w g \ell_u \sin(\phi_w) - m_a g \frac{\ell_u}{2} \sin(\phi_a),$$

where  $(m_w g)$  is the force of the weight,  $m_a$  is the mass of forearm and hand,  $\ell_u$  is the length of the ulna bone and  $\phi_a$  and  $\phi_w$  are the angles of the forearm and rope, as visualized in Fig. 5.2.

### 5.2.2 Data Processing

From the captured data, derived quantities of biceps ( $B$ ) and triceps ( $T$ ) muscles were estimated using a geometric model of the upper arm. The geometric model is available in the software OpenSim [Del07] and was customized for the particular subject. The inverse kinematics module of OpenSim was used to estimate the muscle tendon unit lengths,  $\ell_{\text{MTU},M}$ , contraction velocities,  $v_M = \dot{\ell}_M$  and moment arms  $r_M$  of the two muscles,  $M \in \{B, T\}$ .

EMG signals were captured by two electrodes on the skin above the biceps and triceps muscles. For both signals, several preprocessing steps were applied to obtain the inputs for the two models, A and B. The raw signal was filtered according to the procedure of [Fal16]. First a fourth-order Butterworth high pass filter with cutoff frequency of 30 Hz was applied to reduce non-zero average voltages. The resulting signal was full-wave rectified by taking the absolute value of every measured data point. Then, application of a fourth-order Butterworth low pass filter with 10 Hz cutoff frequency yielded a smoothed signal. The resulting filtered EMG signals were normalized to the interval  $[0, 1]$ , such that the value of 1 corresponds to the experimentally determined value of maximum voluntary contraction.

The measured EMG signals on the skin directly correspond to the electric excitation level in the muscle,  $u$ . Excitation leads to the release of free calcium ions within the sarcomere.

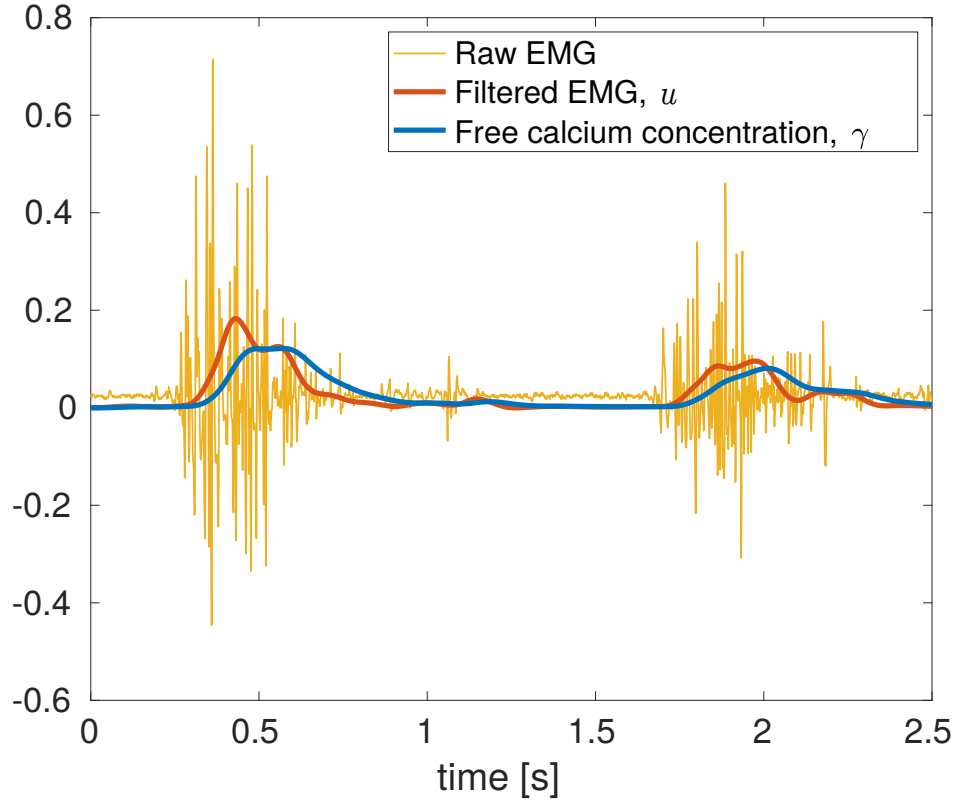


Figure 5.3: From raw EMG data of the biceps (yellow) to the filtered signal,  $u$ , (red) to the free calcium concentration,  $\gamma$  (blue). The data is taken from the beginning of the first elbow flexion experiment. It can be seen that the filtering smooths out the initial signal and removes the constant offset. The free calcium ion concentration follows the filtered EMG with a short delay.

Binding of calcium ions to myosin increases the concentration of cross-bridges, which is commonly known as the muscular activation,  $\alpha$ . The muscular activation directly corresponds to the produced force of the muscle [Bay17].

The concentration of free calcium ions is denoted as  $\gamma$  and can be computed from excitation,  $u$ , by the following first order differential equation [Hat77],

$$\dot{\gamma} = m(u - \gamma).$$

We used the filtered EMG signal,  $u$ , to obtain values for  $\gamma$ . Figure 5.3 shows the raw and filtered EMG signals and the resulting free calcium concentration for a sample of the experimental data.

The activation of the muscle,  $\alpha$ , does not only depend on the free ion concentration,  $\gamma$ , but also on the current state of muscle contraction. This excitation-contraction coupling has to be described by a dynamic system of ODEs and is included in model B. Therefore,



preprocessing is completed with computing the free calcium concentration  $\gamma$  and not the activation  $\alpha$ .

The experimental trials were split into  $n_{\text{trials}} = 7$  experiments to be used for model identification and one for validation. The total captured number of values in the training experiments,  $N$ , was too large, such that not all points could be used for training of models A and B. To reduce the amount of data and, thus, speed up the computation, we selected  $n \ll N$  featured values with the aim that they are representative for the whole data set. For every experimental trial, we choose the same fixed number of data points,  $n_{\text{per\_trial}}$ . Our selection algorithm identifies  $n_{\text{per\_trial}}$  time steps,  $t_i, i = 1, \dots, n$  such that the summed values of the free calcium concentrations for biceps and triceps,  $\gamma_B(t_i) + \gamma_T(t_i)$ , are evenly distributed along the value range. This leads to  $n = n_{\text{trials}} \cdot n_{\text{per\_trial}}$  selected data points.

The set of training data  $\mathcal{D} = \mathcal{X} \times \mathcal{Y}$  consists of the  $n$  selected vectors of the experimental values that are the input to the system of muscles,  $\mathbf{x}_i \in \mathcal{X}, i = 1, \dots, n$ , together with the observed output values,  $y_i \in \mathcal{Y}, i = 1, \dots, n$ . The input vectors contain values for muscle tendon unit lengths, contraction velocities, moment arms and free calcium ion concentrations for biceps and triceps each,  $\mathbf{x}_i = (\ell_{\text{MTU},B}, \ell_{\text{MTU},T}, v_B, v_T, r_B, r_T, \gamma_B, \gamma_T)^\top(t_i)$ . The output values consist of the elbow torques,  $y_i = \tau(t_i)$ . This data set,  $\mathcal{D}$ , serves as training input for both models A and B.

## 5.3 Models

The current section describes the two model approaches that can predict elbow torques from experimental input data. [Section 5.3.1](#) introduces the non-parametric data driven model, A. Afterwards, [Sec. 5.3.2](#) presents the biophysically based model, B. It requires a parameter optimization, which will be described in [Sec. 5.3.3](#).

### 5.3.1 Data-driven Model A

The first modeling approach uses a non-parametric model. Such model approximates the function  $f$  that maps from input to output data points. The function  $f$  is learned by using the training data sets. Regression is used to obtain predictions for new data points. In our case, we use a probabilistic model that considers the propability distribution of the model function.

We use the method of *Gaussian Process Regression*. A Gaussian process is a collection of random variables such that the joint distribution of every finite subset of these random variables is multivariate normal (Gaussian). In our example, each input data point in the space of measured values,  $\mathbf{x} \in \mathcal{X}$  has an associated random variable  $f(\mathbf{x})$  that describes the output of the model for this point.

A Gaussian process,  $\mathcal{GP}$ , is characterized by a mean function,  $m(\mathbf{x})$ , and a kernel function,  $k(\mathbf{x}, \mathbf{x}')$  that models the covariance between any pair of points,  $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ . Different choices of kernel functions are possible and can depend on hyperparameters,  $\psi$ . Describing observed values,  $y$ , by a Gaussian process distribution can be expressed as

$$y \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}', \psi)).$$

This representation is non-parametric in the sense that no particular parametric form of the function  $y = f(\mathbf{x})$  is assumed whose parameters would be determined. Instead, the probabilistic model is constructed using the observed function values at measured inputs,  $\mathbf{x} \in \mathcal{X}$ .

Gaussian Process Regression is based on *Bayesian Inference* to update a prior belief of the model to a posterior model using information contained in observations of the process. The observed data is the set of measurements,  $\mathcal{D}$ .

The *prior* distribution,  $p(\mathbf{f} | \mathcal{X}, \psi)$ , is described by the Gaussian process as

$$p(\mathbf{f} | \mathcal{X}, \psi) = \mathcal{N}(\mathbf{f} | \mathbf{m}, \mathbf{K}),$$

with mean values  $\mathbf{m} = (m(\mathbf{x}_i))_{i=1, \dots, n}^\top$  and covariance matrix  $\mathbf{K}$  with  $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j, \psi)$ .

The *likelihood*,  $p(\mathbf{y} | f(\mathbf{x}), \boldsymbol{\theta})$ , describes the probability of an observation  $\mathbf{y}$  given a particular model,  $f$ . The vector  $\boldsymbol{\theta}$  denotes additional parameters of the likelihood.

Using Bayes' rule, the *posterior* distribution,  $p(\mathbf{f} | \mathcal{D})$ , of the function values,  $\mathbf{f}$ , can be computed from prior and likelihood as

$$p(\mathbf{f} | \mathcal{D}, \boldsymbol{\theta}, \psi) = \frac{p(\mathbf{y} | \mathbf{f}, \boldsymbol{\theta}) p(\mathbf{f} | \mathcal{X}, \psi)}{p(\mathcal{D} | \boldsymbol{\theta}, \psi)}.$$

This results in a measure for the uncertainty of the model,  $f$ , at unobserved points  $\mathbf{x}_* \notin \mathcal{D}$ .

Additionally, the fact that the measured quantities in the experiments are subject to measurement noise can be incorporated into the model. The assumption  $y = f(\mathbf{x}) + \varepsilon$  adds

a normally distributed value of observational noise,  $\varepsilon \sim \mathcal{N}(0, \sigma_n^2)$ , to the formulation. The noise variance,  $\theta = \sigma_n^2$ , is an additional parameter of the likelihood.

It is also possible to explicitly model the mean function,  $m(\mathbf{x})$ . By replacing the model  $f(\mathbf{x})$  by

$$g(\mathbf{x}) = f(\mathbf{x}) + \mathbf{h}(\mathbf{x})^\top \boldsymbol{\beta},$$

we allow for a global trend in the data that is formulated in terms of a vector of explicit basis functions,  $\mathbf{h}(\mathbf{x})$ , and corresponding coefficients,  $\boldsymbol{\beta}$ .

The algorithm for Gaussian Process Regression involves estimating several values from the given data. The hyperparameters of the covariance function,  $\boldsymbol{\psi}$ , the noise variance,  $\theta$ , and the coefficients of the fixed basis functions,  $\boldsymbol{\beta}$ , are determined by solving an optimization problem. The computation involves matrix inversions and has a computational complexity that is cubic in the number of data points,  $\mathcal{O}(n^3)$ . For details, the reader is referred to the literature [Ras05]; [Kus06].

### 5.3.2 Biophysical Model B

During elbow flexion, three muscles are involved: biceps brachii, brachialis and brachioradialis. For simplicity, only biceps brachii, which contributes most, is considered. In the current study, the effects of the other two muscles are lumped and contained in the biceps brachii model.

Thus, the biophysical model consists of two Hill-type muscle models, for biceps and triceps, respectively. The muscle models are arranged around a hinge joint for the elbow angle. The muscle forces contribute to the torque at the elbow over their respective moment arms.

Hill-type models describe the macroscopic, dynamic mechanical behaviour of an entire muscle along a one-dimensional line of action. The behaviour is formulated by mathematical functions that have to be parameterized to fit experimental observations.

Multiple variants of Hill-type models exist that use various configurations of mechanical elements to consider different effects of the muscle. The original model was proposed in [Hil38]. It contains a contractile element (CE) and two elastic elements, arranged in series and in parallel to the CE. The authors of [Sie08] compare two different approaches using these three elements. The effect of tension in eccentric contractions is added to

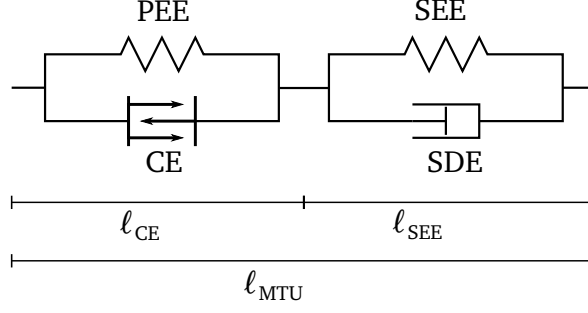


Figure 5.4: Mechanical structure of the Hill-type muscle model. The force generating contractile element (CE) is parallel-connected to the parallel elastic element (PEE) and connected in series to a second parallel-connected structure of the serial elastic element (SEE) and the serial damping element (SDE). The length of the whole muscle tendon unit,  $\ell_{\text{MTU}}$ , is composed of the common length of CE and PEE,  $\ell_{\text{CE}}$ , and the common length of SEE and SDE,  $\ell_{\text{SEE}}$ . The variable  $\ell_{\text{CE}}$  is an internal state of the model.

the Hill-type model by [Til08]. The authors of [Gün07] add a forth, damping element to account for high-frequency damping of the muscle tissue. In [Mör12], electro-mechanical delay is investigated with and without the additional damping element.

We employ the four-element Hill-type muscle model that is described by [Hae14]. The structure is visualized in Fig. 5.4. It consists of four components: the contractile element (CE), the parallel elastic element (PEE), the serial elastic element (SEE) and the serial damping element (SDE). Inputs to the model are the muscular activation,  $\alpha(t)$ , and length and contraction velocity of the muscle tendon unit (MTU),  $\ell_{\text{MTU}}(t)$  and  $\dot{\ell}_{\text{MTU}}(t)$ . The output of the model is the muscle force,  $f_{\text{MTU}}(t)$ . The model contains one internal state variable, the length of the CE,  $\ell_{\text{CE}}(t)$ . The muscle dynamics determine this internal length and its time derivative, the contraction velocity of the CE,  $\dot{\ell}_{\text{CE}}(t)$ .

The force of the MTU is computed as sum of the forces of the respective parallel elements as visualized in Fig. 5.4,

$$F_{\text{MTU}} = F_{\text{CE}}(\ell_{\text{CE}}, \dot{\ell}_{\text{CE}}, \alpha) + F_{\text{PEE}}(\ell_{\text{CE}}) = F_{\text{SEE}}(\ell_{\text{CE}}, \ell_{\text{MTU}}) + F_{\text{SDE}}(\ell_{\text{CE}}, \dot{\ell}_{\text{CE}}, \dot{\ell}_{\text{MTU}}, \alpha). \quad (5.1)$$

The force terms of the four elements,  $F_{\text{CE}}$ ,  $F_{\text{PEE}}$ ,  $F_{\text{SEE}}$  and  $F_{\text{SDE}}$  are described by analytical functions that use a total of 19 parameters. The second identity of (5.1) can be solved for  $\dot{\ell}_{\text{CE}}$  to get an evolution equation for the length of the CE,  $\ell_{\text{CE}}$ , which is an internal state of the model. Together with the activation dynamics, which describes the relation between the free calcium ion concentration,  $\gamma$ , and the muscle activation,  $\alpha \in [0, 1]$ , we get the

following coupled system of differential-algebraic equations,

$$F_{\text{MTU}} = F_{\text{MTU}}(\ell_{\text{MTU}}, \ell_{\text{CE}}, \dot{\ell}_{\text{CE}}, \alpha), \quad (5.2)$$

$$\dot{\ell}_{\text{CE}} = f_{\text{CE}}(\ell_{\text{CE}}, \ell_{\text{MTU}}, \dot{\ell}_{\text{MTU}}, \alpha), \quad (5.3)$$

$$\alpha = f_{\alpha}(\gamma, \ell_{\text{CE}}). \quad (5.4)$$

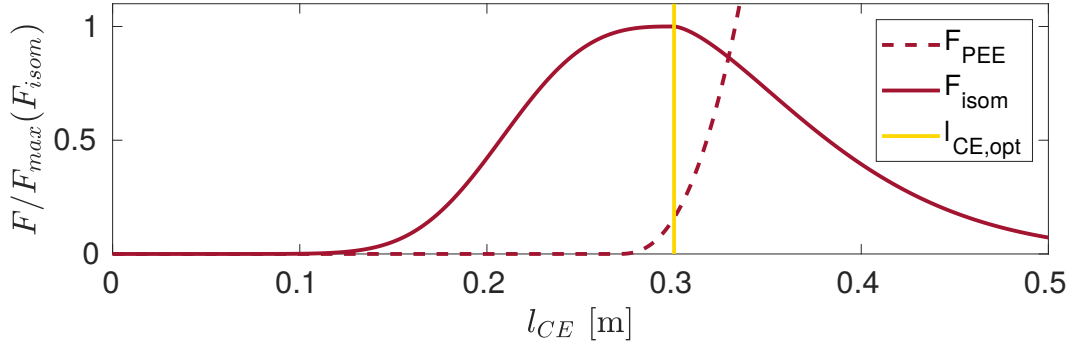
The detailed equations of the force terms in [Equations \(5.2\) and \(5.3\)](#) can be found in [\[Hae14\]](#). [Figure 5.5](#) visualizes characteristic curves of the model. [Figure 5.5a](#) shows how the generated force depends on the length of the CE. The active force, shown by the solid red line, has its maximum at an optimal length,  $\ell_{\text{CE,opt}}$ . This is explained by the exposure of the actin and myosin filaments in the sarcomere relative to each other, which is lower when the actin filaments are pulled apart or pushed together. The dashed line in [Fig. 5.5a](#) represents the passive force of the elastic muscular tissue which is essentially generated by the titin proteins in the sarcomere. Only starting from a certain length, the structure exerts reaction forces against lengthening forces to avoid overstretching of the muscle.

The force-velocity relation of the Hill-type model is shown in [Fig. 5.5b](#). The curve is composed of two branches: The concentric branch for shortening contraction with  $\dot{\ell}_{\text{CE}} \leq 0$  and the eccentric branch for lengthening contraction with  $\dot{\ell}_{\text{CE}} > 0$ . It can be seen that the generated force increases monotonically over the lengthening velocity. It approaches a limit for maximum positive and negative velocity. These limits are given by parameters and are exemplary for how the shape of the curves of a Hill-type model can be parametrized.

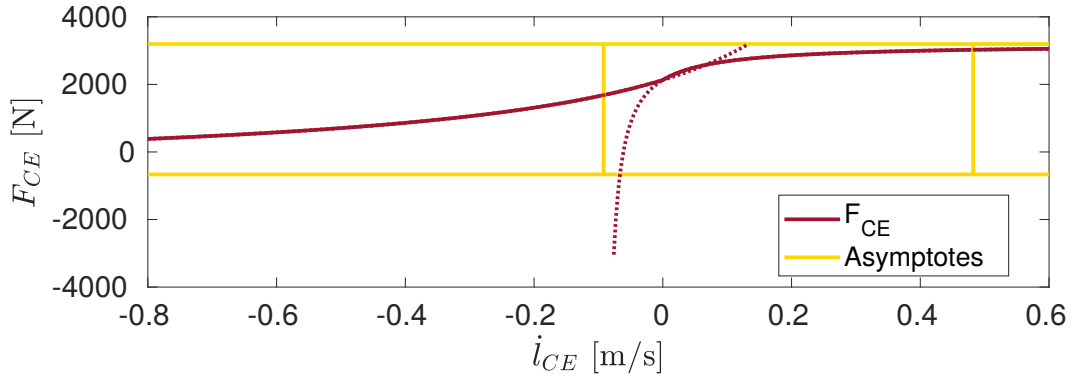
The model of activation dynamics, [Eq. \(5.4\)](#), is the one of Hatze et al. [\[Hat77\]](#) with parameters taken from [\[Bay17\]](#).

### 5.3.3 Parameter Identification for Model B

The process of model identification finds the parameters that make the model B predict correct values for the specific subject. An optimization scheme finds the parameters that minimize the error in the predicted outcome for the training data set.



(a) Force-length curves of the PEE (dashed red line) and CE (solid red line), for isometric condition ( $\dot{l}_{CE} = 0$ ) and normalized to the maximum isometric force. The optimal length of the CE,  $l_{CE,opt}$ , is shown as yellow vertical line. The force of the PEE,  $F_{PEE}(l_{CE})$ , is zero for  $l_{CE} < 0.9l_{CE,opt}$ . The isometric contraction force,  $F_{isom} = F_{CE}(l_{CE})$  is formulated by two branches for separated by  $l_{CE,opt}$ .



(b) Force-velocity curve of the CE,  $F_{CE}(\dot{l}_{CE})$ , at optimal length,  $l_{CE} = l_{CE,opt}$ , and for activation level  $\alpha = 0.5$ . The function (red solid line) is formulated piecewise, the base functions of the two branches are shown as red dotted lines. Their limits are visualized by the yellow horizontal and vertical asymptotes.

Figure 5.5: Force-length and Force-velocity relations for the muscle model with generic parameters taken from literature [Hae14].

The following minimization is performed,

$$\min_{\substack{\boldsymbol{\theta}_M, \ell_{CE,M}(t), \\ M \in \{B, T\}, t \in \mathcal{T}}} \sum_{t \in \mathcal{T}} |\tau(t) - \hat{\tau}(t)|^2 \quad (5.5)$$

$$\text{s.t. } \forall t \in \mathcal{T} : \quad \tau(t) = F_{MTU,B}(t, \ell_{CE,B}, \boldsymbol{\theta}_B) \cdot \hat{r}_B(t) \\ - F_{MTU,T}(t, \ell_{CE,T}, \boldsymbol{\theta}_T) \cdot \hat{r}_T(t), \quad (5.6)$$

$$\dot{\ell}_{CE,M}(t) = \dot{\ell}_{MTU,M}(t), \quad M \in \{B, T\}, \quad (5.7)$$

$$\boldsymbol{\theta}_B, \boldsymbol{\theta}_T \in \Theta, \quad (5.8)$$

$$\ell_{CE,M}(t) \in [0, \ell_{MTU,M}(t)], \quad M \in \{B, T\}. \quad (5.9)$$

The optimization variables are the parameters for the biceps and triceps Hill-type models,  $\boldsymbol{\theta}_B$  and  $\boldsymbol{\theta}_T$ , and the lengths of the contractile elements for both models at every point in time,  $\ell_{CE,B}$  and  $\ell_{CE,T}$ . The variables designated as  $\hat{\square}$  are the measured quantities from the training experiments. The objective function, Equation (5.5), penalizes the difference between computed torque,  $\tau$ , and measured torque,  $\hat{\tau}$ , at every timestep of the training data,  $t \in \mathcal{T}$ . The torque is computed in Eq. (5.6) by multiplying the predicted forces,  $F_{MTU,B}, F_{MTU,T}$ , with the measured moment arms,  $\hat{r}_B, \hat{r}_T$ .

In Eq. (5.7), the contraction velocities are constraint to the measured values. Because the lengthening velocity of the CE,  $\dot{\ell}_{CE}$ , is an internal quantity and, thus, cannot be observed in experiments, we assume it to be equal to the lengthening velocity of the whole muscle,  $\dot{\ell}_{CE} \approx \dot{\ell}_{MTU} = \dot{\ell}_{CE} + \dot{\ell}_{SE}$ . This requires the assumption  $\dot{\ell}_{SE} \approx 0$  which can be justified given the low dynamic nature of the experiments.

By Eq. (5.8) we bound each of the parameters,  $\boldsymbol{\theta}_B, \boldsymbol{\theta}_T$ , to a range between half and twice the generic value from literature. The lengths of the CEs are constraint by Eq. (5.9) to be positive and smaller than the length of the MTU.

All optimization variables are normalized to improve the numeric conditioning of the optimization problem. The parameters  $\boldsymbol{\theta}_B, \boldsymbol{\theta}_T$  are normalized with respect to generic values from literature, that were taken from [Gün07]; [Mör12]; [Hae14]. The initial values are set to these generic parameters, which corresponds to the value one for the normalized variables. The internal states,  $\ell_{CE,B}, \ell_{CE,T}$ , are normalized with respect to the measured MTU lengths,  $\hat{\ell}_{MTU,B}, \hat{\ell}_{MTU,T}$  and initialized to zero.

## 5.4 Results and Discussion

In the following, results of combining the experimental data with the two models, A and B are presented. At first, [Sec. 5.4.1](#) gives details on the preprocessed data. The training phase is described in [Sec. 5.4.2](#). Applying the trained models to the validation data is done in [Sec. 5.4.3](#). For model A, [Sec. 5.4.4](#) tests a simplified version. For model B, [Sec. 5.4.5](#) shows some insights into the optimized parameter values.

### 5.4.1 Feature Selection

The experimental data is split into a training and a validation dataset. [Figure 5.6](#) shows the processed data of the training data set. In total we captured  $N = 34,934$  data points for the seven experimental trials. Out of these, we select  $n_{\text{per\_trial}} = 40$  feature points in every trial, leading to a total of  $n = n_{\text{per\_trial}} \cdot n_{\text{trials}} = 280$  points. The selected points are visualized by crosses in the top plot of [Fig. 5.6](#). It can be seen that the algorithm distributes the feature points equally along the  $\gamma$  axis.

### 5.4.2 Training of the Models

Training of the Gaussian Process of model A was performed using the ready to use implementation provided by MATLAB. We parametrized the covariance by a squared exponential kernel and used constant basis functions,  $H(x) = 1$ . We enabled observational noise, its standard deviation,  $\sigma_n^2$ , was found by optimization during training of the model.

For training of model B, the optimization problem for the parameters needed to be solved. We implemented the Hill-type models and the constraints also in MATLAB and used the nonlinear programming implementation, `fmincon`, to minimize the given bounded and nonlinear constrained, multivariable function.

The total number of optimization variables can be computed by  $2 \cdot 19 + 2n = 598$ , as each of the parameter vectors  $\theta_B, \theta_T$  had 19 entries. The resulting parameter values and their relation to the initial values are summarized in [Tab. 5.1](#). It can be seen that none of the final parameter values are limited by the constraints which would be  $-50\%$  and  $+100\%$ . However, setting the constraints helps the optimizer to stay in the valid range of meaningful model parameters.



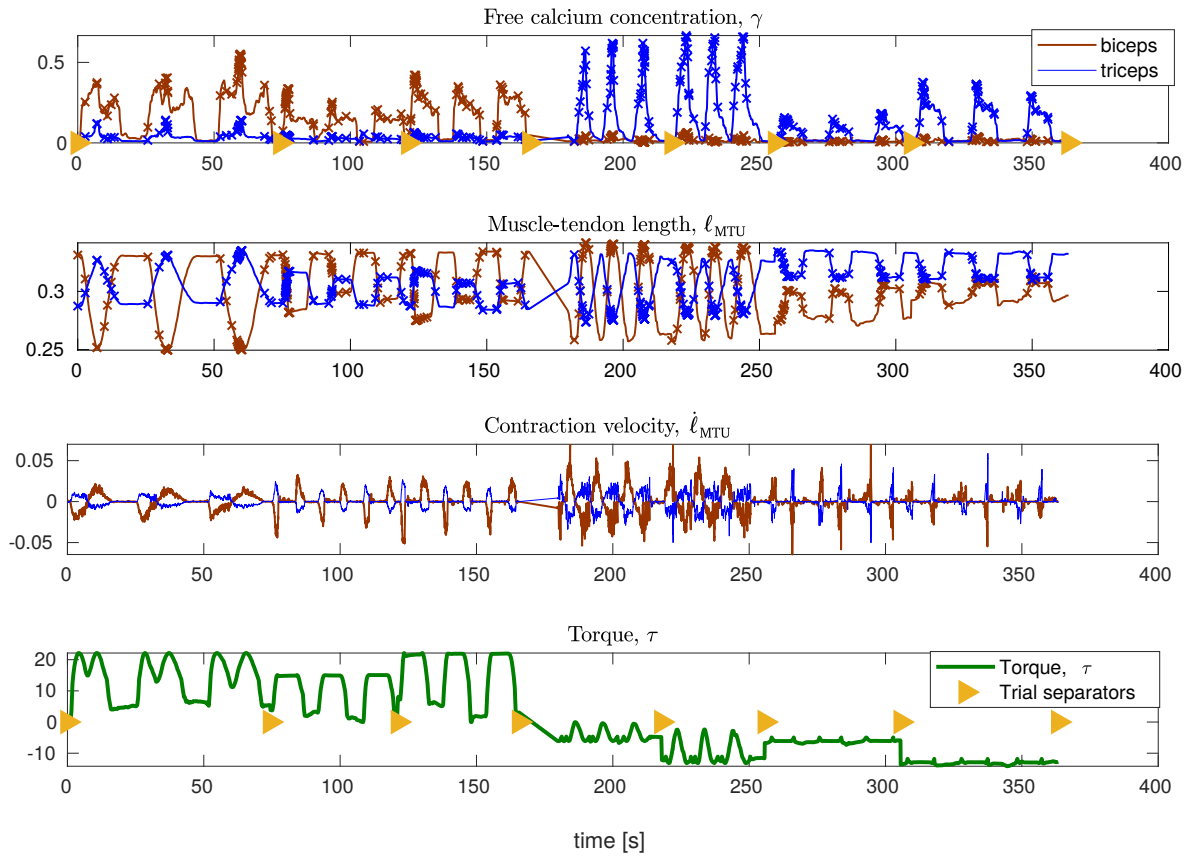
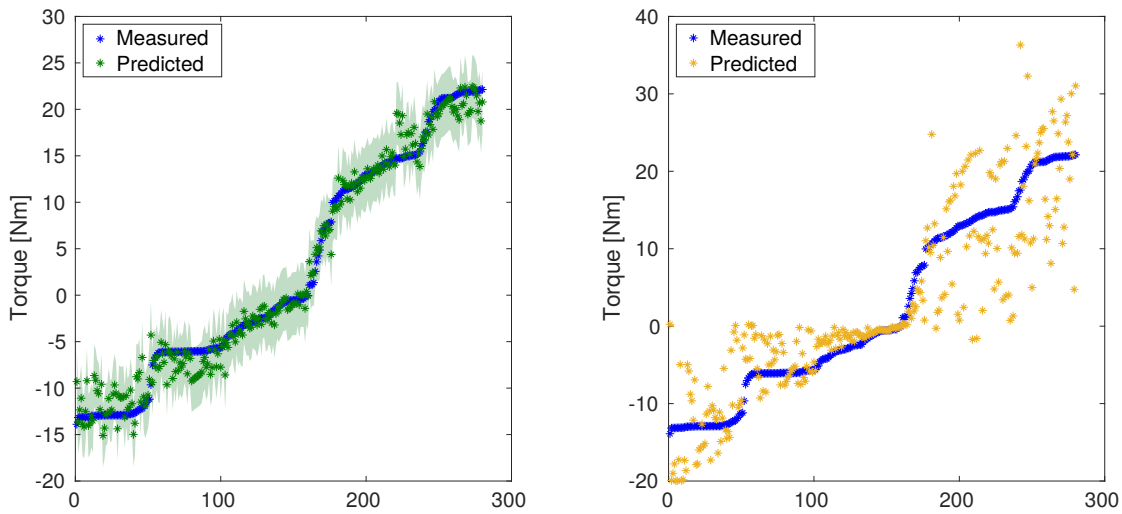


Figure 5.6: Processed experimental values over time that were used for training of both models. The concatenated data of seven trials is shown which yields a total end time of 363.32 s. The individual trials are separated by the yellow triangles on the  $x$ -axis. The three upper plots show the values of  $\gamma$ ,  $\ell_{MTU}$  and  $\dot{\ell}_{MTU}$  for both biceps (brown) and triceps (blue), the bottom plot shows the elbow torque,  $\tau$ . The selected feature points are visualized as crosses in the two top plots. In the upper-most plot, it can be seen that the first three trials, which correspond to elbow flexion, mainly activated the biceps muscle, whereas in the last four trials, corresponding to elbow extension, the triceps is more active.

CE	$F_{\text{max}} [\text{N}]$	$\ell_{\text{CE,opt}} [\text{m}]$	$\Delta W_d []$	$\Delta W_a []$	$\nu_{\text{CE},d} []$
Generic	4260	0.3	0.35	0.35	1.5
Biceps	+11.0%	+31.7%	+10.9%	+91.6%	+10.9%
Triceps	−49.0%	−25.4%	+10.9%	+5.1%	+10.9%
CE	$\nu_{\text{CE},a} []$	$A_{\text{rel},0} []$	$B_{\text{rel},0} []$	$S_{\text{ecc}} []$	$F_{\text{ecc}} []$
Generic	3.0	0.25	2.25	2	1.5
Biceps	−46.4%	+14.0%	+77.5%	−4.1%	−30.1%
Triceps	+95.9%	−20.3%	+41.4%	+22.3%	+36.8%
PEE	$L_{\text{PEE},0} []$	$\nu_{\text{PEE}} []$	$F_{\text{PEE}} []$		
Generic	0.9	2.5	2.0		
Biceps	+10.9%	+10.9%	+10.9%		
Triceps	+10.9%	+10.9%	+10.9%		
SDE	$D_{\text{SDE}} []$	$R_{\text{SDE}} []$			
Generic	0.3	0.01			
Biceps	+10.9%	+8.6%			
Triceps	+10.9%	−11.0%			
SEE	$\ell_{\text{SEE},0} [\text{m}]$	$\Delta F_{\text{SEE},0} []$	$\Delta U_\ell []$	$\Delta U_{n\ell\ell} []$	
Generic	0.172	0.0425	0.017	568	
Biceps	−25.0%	−42.4%	+63.3%	+59.2%	
Triceps	−10.3%	+64.75%	+19.0%	+23.6%	

Table 5.1: Hill-type muscle model parameters of the four elements, CE, PEE, SDE and SEE, initial values given in literature and relative changes of the optimized values. Further explanations of the parameters and references to their initial values are given in [Hae14].

After training of the models A and B using the selected points of the training dataset, both models were tested by a *resubstitution prediction*, i.e. predicting output from the training input data. The results are shown in Fig. 5.7a for model A and Fig. 5.7b for model B. As this uses only the selected subset of experimental values, the data points have no natural ordering and therefore were sorted for better visibility. It can be seen that for both models the predicted value are a good fit to the measured values. For model A, the predicted 95% confidence interval includes the real values almost everywhere. For model B, the predicted values have a higher variance, especially for high torque values.



(a) Measured values (blue) and values predicted by Model A (dark green), with 95% confidence interval (light green). (b) Measured values (blue) and predicted values by Model B (orange).

Figure 5.7: Measured and predicted torque values for the training data set. The measured points are ordered by magnitude, the order of the predicted points matches the order of the measured points.

### 5.4.3 Validation

The next evaluation is to use the validation dataset and compare the predicted outputs of the model with the real, experimental values. In contrast to the training data, where a small number,  $n$ , of points was selected, we use all captured values, in total  $54 \cdot 10^3$  data points for a time span of  $t = 54$ s. The results are shown in Fig. 5.8.

Comparing the green curve for model A with the blue curve for the experimental data, it can be seen that the predicted values match qualitatively for most of the time span. The predicted torque values are slightly smaller than the real values. Only for the two intervals  $[11\text{s}, 12\text{s}]$  and  $[40\text{s}, 42\text{s}]$  the predicted value is far off. The 95% confidence interval that was computed by the Gaussian Process spans a large range for these time intervals which implies that the model prediction is not to be trusted in this area.

The biophysical model approach, model B was tested in two variants. First, with the generic parameters from literature (yellow curve), second with the subject-specific, optimized parameters (red curve). It can be seen that the generic model fails to predict the torque values whereas the trained model computed reasonable values. These values are

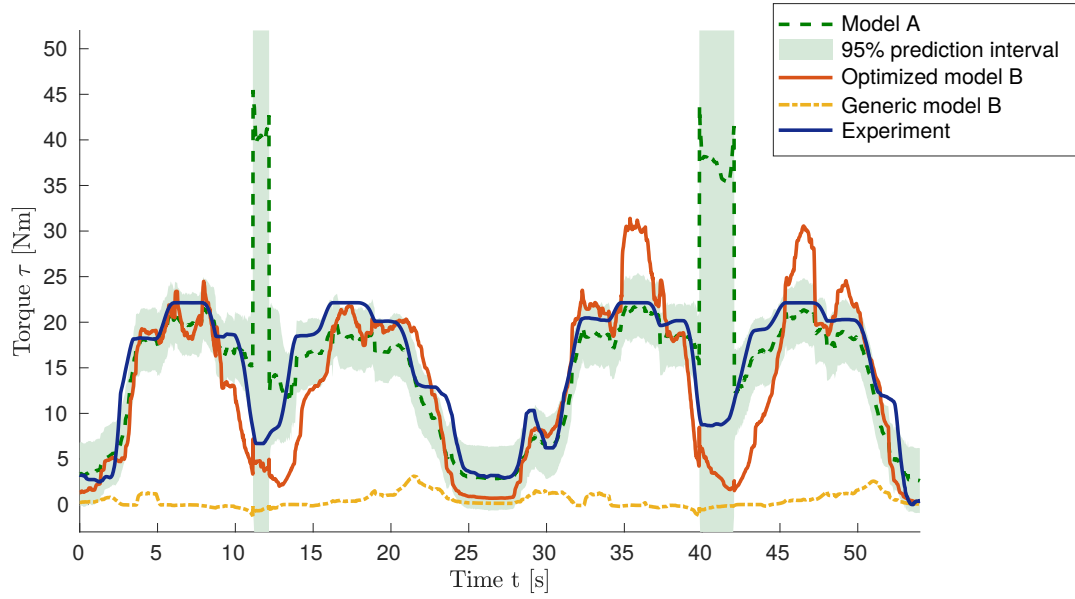


Figure 5.8: Predicted torque values by model A (green), trained model B (red) and un-trained model B (yellow), in comparison to the experimentally measured values (blue), for the validation data set.

worse than most of the predictions from model A, but they succeed in giving a qualitative estimate about a low, medium or high torque output.

The match between models and experiment can be quantified using the normalized root-mean-square error (NRMSE), which is defined as

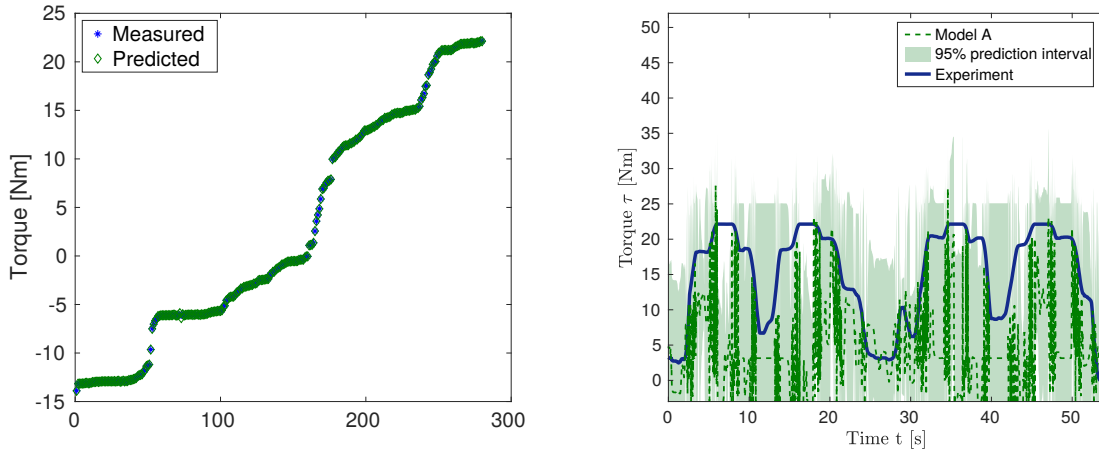
$$\text{RMSE} = \sqrt{\sum_{i=1}^N (\theta_i - \hat{\theta}_i)^2 / N},$$

$$\text{NRMSE} = \frac{\text{RMSE}}{\max_i \{\hat{\theta}_i\} - \min_i \{\hat{\theta}_i\}}.$$

The NRMSE for model A is 0.267 which is worse than the value of 0.163 for the trained model B. The generic model B has the worst NRMSE of 0.547.

#### 5.4.4 Simplified Model A

An advantage of model approach A is that it forgoes any biophysical description and the associated type of model error. It is a generic approach that does not require expert knowledge about the physiological structure. In the present study, however, some level of expert knowledge and physiological model was required in preprocessing the MoCap



(a) Measured and predicted torque values of the training dataset, ordered by magnitude. The measured values (blue) and the values predicted by the Gaussian Process (green) lie on each other.

(b) Predicted torque values for the validation trials (green dotted line), 95% confidence interval (light green) and the reference values of the experiment (blue), which reveal bad prediction capabilities of the simplified model A.

Figure 5.9: Result for the simplified model A, where only the elbow angles,  $\phi_e$ , are used as training input instead of MTU lengths, velocities and moment arms,  $\ell_{\text{MTU},B}$ ,  $\ell_{\text{MTU},T}$ ,  $v_B$ ,  $v_T$ ,  $r_B$  and  $r_T$

data, i.e. solving the inverse kinematics of the observed forearm movements to get the kinematic quantities of muscle lengths, velocities and moment arms. Therefore, the next study tries to apply the approach of model A using only the elbow angle instead, which is easier to measure experimentally. Thus, training data consist of input vectors  $\mathbf{x}_i = (\phi_e(t_i), \gamma_B(t_i), \gamma_T(t_i))^T \in \mathcal{X}$ . In the following, this model is named “Simplified model A” in contrast to the “Full model A” that used the extended input data.

The results are shown in Fig. 5.9. It can be seen that the resubstitution prediction in Fig. 5.9a where the trained model is used to predict the training values shows a perfect fit. In contrast to the full model A, Fig. 5.7a, here, the learned input-output mapping shows no variance. However, the prediction for the validation dataset in Fig. 5.9b shows a high error relative to the experimental data. The curve for the experimental data even lies outside the 95% confidence interval of the prediction at some points. The NRMSE value has a high value of 0.461. The NRMSE values of the full and simplified model A and the generic and optimized model B are summarized in Fig. 5.11. This means that simplified model A gives no useful results where the training input is too scarce. Instead, preprocessing of the measurements using a subject specific geometric model, as done for the full model A gives the desired results.

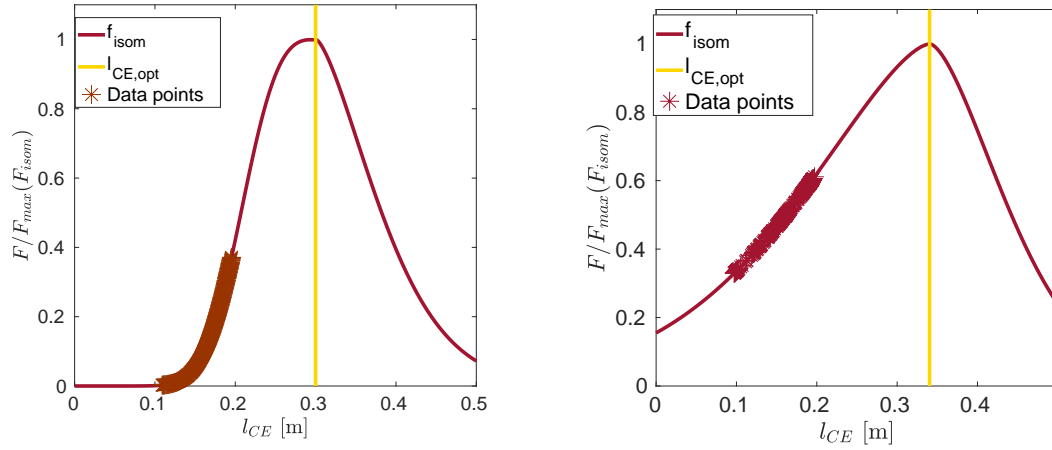
### 5.4.5 Insights of Model B

An advantage of model approach B is that the trained parameters are physically meaningful and allow insight into the properties of the subject specific model. Furthermore, the quality of the training data can be assessed. [Figure 5.10](#) shows the force-length relation of the biceps muscle model using the generic and the optimized parameter values. It can be seen that the subject-specific model has a smaller slope of the force curve. All points of the training data set are indicated by red crosses on the curves and show the operating range of the muscle in which the model has been trained. It can be seen that the experimental training data is restricted to a small range of the muscle length below its optimal CE length,  $\ell_{CE,opt}$ . In order to improve the quality of the model predictions for this subject, additional experimental trials can be designed for training that fills in the missing range of operation for larger muscle extensions.

A low computational time of the offline parameter identification and the online evaluation of the two models would be an important measure for their practical applicability. In the present study, the training phase of Model A, i.e. optimization of the quantities for the Gaussian Process Regression using 280 training data points took 2.24 s. The evaluation of Model A for the validation data set of  $54 \cdot 10^3$  took 116 ms. The parameter optimization of model B had a duration of 25min16s and the evaluation for the validation data 13 s.

The large difference in runtime between models A and B can be explained by the inefficient implementation of the biophysical model using the MATLAB programming language. During parameter identification, this model needs to be evaluated iteratively by the optimization algorithm. In contrast, the optimization within model A works on an internal implementation of the Gaussian Process which could be optimized during development of the particular MATLAB functionality. In general, the training for Gaussian Process Regression has cubic time complexity whereas for the parameter optimization of model B, iterative solvers with linear time complexity exist.

If EMG is used to control an exoskeleton that supports the movement of the limb, the measured signals are ahead of the human movement for a small duration. This is a result of the time delay in the neuromusculoskeletal system. This property gives the assistive exoskeleton a small time to predict the intended movement, and thereby allows a seamless integration.



(a) Model with generic parametrization and training data points. (b) Model with subject-specific parametrization and training data points.

Figure 5.10: Isometric force-length relation of the CE for the biceps model, analogous to Fig. 5.5a.

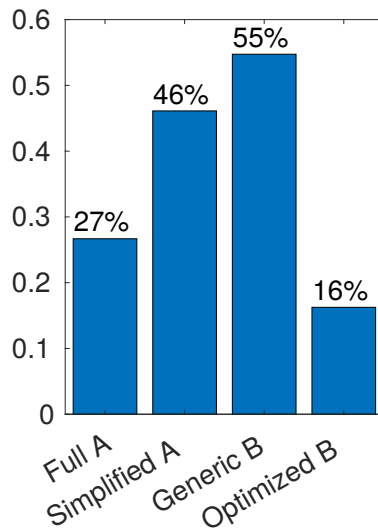


Figure 5.11: Normalized Root Mean Square errors (NRMSE) of the validation trials between the respective models and the measured values. A lower error value means a better fit.

## 5.5 Conclusion

In this study, elbow torques during flexion and extension of the upper arm was predicted from motion capture information and EMG measurements. Two models, A and B, were developed. Model A is non-parametric and uses Gaussian Process Regression. Model B is biophysically informed and involves two Hill-type muscle models for biceps and triceps. Experiments were conducted to generate training data that was used for model parameter identification. The two models were evaluated using experimental validation data.

Regarding the implementation, model A requires low effort and no special knowledge about the model, except when motion data is preprocessed for a specific subject. Model B needs expert knowledge about the biophysical structure and the implementation of all comprised models.

Similar holds for the offline training phase. Here, an advantage of model A is that additional information, e.g. from neighbouring muscles or additional sensors can easily be added. Instead, Model B incorporates a-priori information about the functioning of the muscular system and allows to aid the parameter identification step by enforcing physiological constraints  $\Theta$  for the parameters.

Considering the prediction error, both models were able to predict the level of torque reasonably. [Figure 5.11](#) showed the best score for model A, followed by model B. It was also shown that the generic parametrization of model B does not yield a useful prediction. The same is true for a simplified application of model A, where the elbow torque was used as training input instead of derived quantities from the motion capture system that required a complex preprocessing step.

Both models provide possibilities to assess the level of confidence of their predictions. With model A, confidence intervals can be computed directly from the Gaussian Processes. Their usefulness was shown in the validation where regions with large errors also had a large confidence interval. Model B allows insight into force-length and force-velocity characteristics of the two involved muscles. The operating range of the muscle during the experiments can be visualized and allows to assess whether the desired model features were covered by the training phase and, thus, promise a good prediction.

In our study, runtimes were low for model A and high for model B in both offline and online phases. However, this is due to our prototypical implementation of model B. For larger data sizes and more sophisticated implementation, the reverse effect is expected. The runtime complexity for the training phase is better for model B (linear in time)



compared to model A (cubic in time). For the online phase, costly integration over data points is needed for model A whereas model B directly provides a differential equation of the system that can be solved efficiently.

When targeted at real-time application, such as for powered orthoses and prostheses or exoskeletons, both models could be considered. Model A fits the use case of a device that could be (re-)calibrated by the patient itself. Because of the built-in confidence estimation, compliance and safety could be ensured more easily even for imperfect training. Model B would need a controlled environment such as a specialist's laboratory and careful assistance for the calibration process. After calibration it would promise a more natural and more responsive experience because of the subject-specific model and possibly smaller computation times.

This study revealed some advantages of biophysically informed models for prediction of human movements. The kinematics and the individual muscle dynamics were modeled close to the understanding of the system in-vivo. However, several aspects were not modeled as detailed as possible. The pathway from neural stimulation to excitation and activation of the muscle, the recruitment strategies including different motor units, neural feedback loops as well as effects stemming from the 3D geometry of the muscle were not considered. A more detailed, biophysically informed model including these properties will be described in the following sections of the chapter.



## **6 Formulation of electrophysiology and muscle contraction**

- electrophysiology introduction

### **6.1 State of the art**

- Hodgkin-Huxley, shorten, advanced models, how it gets solved

### **6.2 discretization**

- discretization of electrophysiology, operator splitting

### **6.3 Standard for formulation of models**

- cellml introduction

### **6.4 Bidomain**

- EMG

### **6.5 Multidomain**

- multidomain formulation

## **6.6 Formulation of solid mechanics**

- basics solid mechanics framework, material modeling,

## **6.7 Simulation workflow**

- simulation workflow of skeletal muscle

# **7 Numeric**

## **7.1 Fundamentals of the discretization**

## **7.2 Finite element method**

- basics FEM formulation of Laplace operator and diffusion equation, boundary conditions

## **7.3 Incompressible nonlinear solid mechanics**

- incompressibility, penalty formulation vs. mixed formulation
- governing nonlinear eq.
- jacobian

## **7.4 Numerical schemes**

## **7.5 time stepping schemes**

## **7.6 linear system solvers**

## **7.7 newton algorithm**



## **8 Simulation software**

### **8.1 General aspects**

### **8.2 Detail**





## 9 Simulation software

### 9.1 State of the art

- review on bioengineering simulation frameworks

### 9.2 OpenCMISS

- OpenCMISS introduction, iron zinc cmgui, unstructured
  - Strang splitting, solvers 1D
- improved interpolation
- parallelization, issue with 1D
- 3D parallelization strategies
- design flaws, memory

### 9.3 The simulation framework *opendihu*

- modularity of opendihu, concepts (python c++ build system)

### 9.4 Overview of discretization features

- feature overview, implemented equations, ansatz functions, meshes

## **9.5 Input and output**

- file i/o parallel, i: only local data important for high parallelism, o: paraview

## **9.6 In-situ visualization**

- in-situ

## **9.7 Details on the implementation**

## **9.8 Representation of structured meshes**

## **9.9 Parallel data handling**

- data structures, petsc variables, partitioning, ghost communication

## **9.10 Implementation of boundary conditions**

- Neumann, Dirichlet boundary conditions

## **9.11 Implementation of material formulation and automatic derivation of derivatives**

- implementation of solid mechanics with analytic differentiation

## **9.12 efficient and configurable data transfer**

- data layout details, efficient transfer of variables between solvers

## **9.13 Mapping between meshes**

- parallel mapping between meshes, invertability of index space representation

## **9.14 An efficient solver to the Monodomain equation**

- cellml code generator and its optimizations - openmp, simd, vc

## **9.15 Details on the partitioning**

- fibers emg structure and partitioning

## **9.16 Parallelization of the multidomain system**

- multidomain parallelization, reordering of matrix

## **9.17 load balancing**

- load balancing on fiber level



# **10 External coupling**

- introduction to external coupling, precice

## **10.1 coupling with FEBio**

## **10.2 coupling with AceFEM**



# **11 An algorithm for mesh generation**

- introduction, need for fiber and 3d mesh

## **11.1 State of the art**

- literature, laplace potential flow idea

## **11.2 Data preprocessing**

- visual human dataset, methods to extract surface, approximation by spline surfaces

## **11.3 Outline of the algorithm**

- pseudo code

## **11.4 Plane mesh smoothing**

## **11.5 Parallelization**

## **11.6 Generation of fiber meshes**

- generation of muscle mesh, 3D and fibers, also in parallel





## **12 Simulation scenarios**

### **12.1 Simulation of electromyography with fiber-based formulation**

- scenario static-biceps-emg, composite meshes

### **12.2 Simulation of electromyography with multidomain formulation**

### **12.3 Muscle contraction with fiber-based formulation**

- fibers with contraction

### **12.4 Muscle contraction with multidomain formulation**



# **13 Technical simulation results**

## **13.1 Runtime evaluation**

- results and experiments, runtime, peak performance
- monodomain runtime comparison OpenCMISS opendihu

## **13.2 parallel partitioning**

- parallel partitioning and weak scaling, opencmis and opendihu

## **13.3 memory consumption**

- memory scaling OpenCMISS



# **14 biomechanical simulation results**

## **14.1 Simulation of fatigue**

- monodomain, hh shorten, fatigue effects

## **14.2 Simulation of EMG**

- static biceps emg
- fibers with contraction
- multidomain with fat layer
- multidomain with contraction



# **Part III**

## **Soft tissue robotics**





# 15 ...

- introduction, handling flexible objects, overview of different object types
- basics Lagrange formulation of cable
- cable with friction on surface
- placing cable
- peg-in-hole, with sparse grids
- simulation of 2d tissue
- gripping point estimation

[\[Mai21\]](#)



# **Part IV**

## **Conclusion**



# Bibliography

- [Bay17] **Bayer**, A. et al.: *The influence of biophysical muscle properties on simulating fast human arm movements*, Computer Methods in Biomechanics and Biomedical Engineering 20.8, 2017, PMID: 28387534, pp. 803–821, doi:[10.1080/10255842.2017.1293663](https://doi.org/10.1080/10255842.2017.1293663), eprint: <https://doi.org/10.1080/10255842.2017.1293663>, <https://doi.org/10.1080/10255842.2017.1293663>
- [Cav05] **Cavallaro**, E. et al.: *Hill-based model as a myoprocessor for a neural controlled powered exoskeleton arm - parameters optimization*, Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005, pp. 4514–4519
- [Cav06] **Cavallaro**, E. E. et al.: *Real-time myoprocessors for a neural controlled powered exoskeleton arm*, IEEE Transactions on Biomedical Engineering 53.11, 2006, pp. 2387–2396
- [Del07] **Delp**, S. L. et al.: *Opensim: open-source software to create and analyze dynamic simulations of movement*, IEEE Transactions on Biomedical Engineering 54.11, 2007, pp. 1940–1950
- [Fal16] **Falisse**, A. et al.: *Emg-driven optimal estimation of subject-specific hill model muscle-tendon parameters of the knee joint actuators*, IEEE Transactions on Biomedical Engineering PP, 2016, pp. 1–1, doi:[10.1109/TBME.2016.2630009](https://doi.org/10.1109/TBME.2016.2630009)
- [Gar03] **Garner**, B. A.; **Pandy**, M. G.: *Estimation of musculotendon properties in the human upper limb*, Annals of Biomedical Engineering 31.2, 2003, pp. 207–220, ISSN: 1573-9686, doi:[10.1114/1.1540105](https://doi.org/10.1114/1.1540105), <https://doi.org/10.1114/1.1540105>
- [Gün07] **Günther**, M.; **Schmitt**, S.; **Wank**, V.: *High-frequency oscillations as a consequence of neglected serial damping in hill-type muscle models*, Biological Cybernetics 97.1, 2007, pp. 63–79, ISSN: 1432-0770, doi:[10.1007/s00422-007-0160-6](https://doi.org/10.1007/s00422-007-0160-6), <https://doi.org/10.1007/s00422-007-0160-6>
- [Hae14] **Haeufle**, D. et al.: *Hill-type muscle model with serial damping and eccentric force–velocity relation*, Journal of Biomechanics 47.6, 2014, pp. 1531–1536, ISSN: 0021-9290, doi: <https://doi.org/10.1016/j.jbiomech.2014.02.009>, <http://www.sciencedirect.com/science/article/pii/S0021929014001018>
- [Hat77] **Hatze**, H.: *A myocybernetic control model of skeletal muscle*, Biological cybernetics 25.2, 1977, pp. 103–119
- [Hei03] **Heine**, R.; **Manal**, K.; **Buchanan**, T. S.: *Using hill-type muscle models and emg data in a forward dynamic analysis of joint moment*, Journal of Mechanics in Medicine and Biology 03.02, 2003, pp. 169–186, doi:[10.1142/S0219519403000727](https://doi.org/10.1142/S0219519403000727), eprint: <https://doi.org/10.1142/S0219519403000727>, <https://doi.org/10.1142/S0219519403000727>

- [Hil38] **Hill, A. V.:** *The heat of shortening and the dynamic constants of muscle*, Proceedings of the Royal Society of London. Series B - Biological Sciences 126.843, 1938, pp. 136–195, doi:10.1098/rspb.1938.0050, eprint: <https://royalsocietypublishing.org/doi/pdf/10.1098/rspb.1938.0050>, <https://royalsocietypublishing.org/doi/abs/10.1098/rspb.1938.0050>
- [Kus06] **Kuss, M.:** *Gaussian process models for robust regression, classification, and reinforcement learning*, PhD thesis, echnische Universität Darmstadt Darmstadt, Germany, 2006
- [Lin02] **Lin Wang; Buchanan, T. S.:** *Prediction of joint moments using a neural network model of muscle activations from emg signals*, IEEE Transactions on Neural Systems and Rehabilitation Engineering 10.1, 2002, pp. 30–37
- [Llo03] **Lloyd, D. G.; Besier, T. F.:** *An emg-driven musculoskeletal model to estimate muscle forces and knee joint moments in vivo*, Journal of Biomechanics 36.6, 2003, pp. 765–776, ISSN: 0021-9290, doi:[https://doi.org/10.1016/S0021-9290\(03\)00010-1](https://doi.org/10.1016/S0021-9290(03)00010-1), <http://www.sciencedirect.com/science/article/pii/S0021929003000101>
- [Mai21] **Maier, B.; Stach, M.; Mehl, M.:** *Real-time, dynamic simulation of deformable linear objects with friction on a 2d surface*, Mechatronics and Machine Vision in Practice 4, 2021, doi:10.1007/978-3-030-43703-9
- [Mör12] **Mörl, F. et al.:** *Electro-mechanical delay in hill-type muscle models*, Journal of Mechanics in Medicine and Biology 12.05, 2012, p. 1250085, doi:10.1142/S0219519412500856, eprint: <https://doi.org/10.1142/S0219519412500856>, <https://doi.org/10.1142/S0219519412500856>
- [Per07] **Perry, J. C.; Rosen, J.; Burns, S.:** *Upper-limb powered exoskeleton design*, IEEE/ASME Transactions on Mechatronics 12.4, 2007, pp. 408–417
- [Pon09] **Pontonnier, C.; Dumont, G.:** *Inverse dynamics method using optimization techniques for the estimation of muscles forces involved in the elbow motion*, International Journal on Interactive Design and Manufacturing (IJIDeM) 3.4, 2009, p. 227, ISSN: 1955-2505, doi:10.1007/s12008-009-0078-4, <https://doi.org/10.1007/s12008-009-0078-4>
- [Ras05] **Rasmussen, C. E.; Williams, C. K. I.:** *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*, The MIT Press, 2005, isbn:026218253X
- [Ros01] **Rosen, J. et al.:** *A myosignal-based powered exoskeleton system*, IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans 31.3, 2001, pp. 210–222
- [Ros99] **Rosen, J.; Fuchs, M. B.; Arcan, M.:** *Performances of hill-type and neural network muscle models—toward a myosignal-based exoskeleton*, Computers and Biomedical Research 32.5, 1999, pp. 415–439, ISSN: 0010-4809, doi:<https://doi.org/10.1006/cbmr.1999.1524>, <http://www.sciencedirect.com/science/article/pii/S001048099915240>
- [Sar12] **Sartori, M. et al.:** *Emg-driven forward-dynamic estimation of muscle force and joint moment about multiple degrees of freedom in the human lower extremity*, PLOS ONE 7.12, 2012, pp. 1–11, doi:10.1371/journal.pone.0052618, <https://doi.org/10.1371/journal.pone.0052618>
- [Sie08] **Siebert, T. et al.:** *Nonlinearities make a difference: comparison of two common hill-type models with real muscle*, Biological Cybernetics 98.2, 2008, pp. 133–143, ISSN: 1432-0770, doi:10.1007/s00422-007-0197-6, <https://doi.org/10.1007/s00422-007-0197-6>
- [Son05] **Song, R.; Tong, K. Y.:** *Using recurrent artificial neural network model to estimate voluntary elbow torque in dynamic situations*, Medical and Biological Engineering and Computing 43.4, 2005, pp. 473–480, ISSN: 1741-0444, doi:10.1007/BF02344728, <https://doi.org/10.1007/BF02344728>

- [Til08] **Till**, O. et al.: *Characterization of isovelocity extension of activated muscle: a hill-type model for eccentric contractions and a method for parameter determination*, Journal of Theoretical Biology 255.2, 2008, pp. 176–187, ISSN: 0022-5193, doi:<https://doi.org/10.1016/j.jtbi.2008.08.009>, <http://www.sciencedirect.com/science/article/pii/S0022519308004244>
- [Van14] **Van Campen**, A. et al.: *A new method for estimating subject-specific muscle–tendon parameters of the knee joint actuators: a simulation study*, International Journal for Numerical Methods in Biomedical Engineering 30.10, 2014, pp. 969–987, doi:[10.1002/cnm.2639](https://doi.org/10.1002/cnm.2639), eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/cnm.2639>, <https://onlinelibrary.wiley.com/doi/abs/10.1002/cnm.2639>
- [Ven05] **Venture**, G.; **Yamane**, K.; **Nakamura**, Y.: *Identifying musculo-tendon parameters of human body based on the musculo-skeletal dynamics computation and hill-stroeve muscle model*, 5th IEEE-RAS International Conference on Humanoid Robots, 2005. 2005, pp. 351–356
- [Wag05] **Wagner**, H. et al.: *Isofit: a model-based method to measure muscle–tendon properties simultaneously*, Biomechanics and Modeling in Mechanobiology 4.1, 2005, pp. 10–19, ISSN: 1617-7940, doi:[10.1007/s10237-005-0068-9](https://doi.org/10.1007/s10237-005-0068-9), <https://doi.org/10.1007/s10237-005-0068-9>
- [Wal20] **Walter**, J. R. et al.: *Comparative study of a biomechanical model based and black-box approach for subject-specific movement prediction*, 42nd Annual International Conferences of the IEEE Engineering in Medicine and Biology Society, 2020