




## ORIGINAL ARTICLE

# Coupled simulations and parameter inversion for neural system and electrophysiological muscle models

Carme Homs-Pons<sup>1</sup> | Robin Lautenschlager<sup>2</sup>  | Laura Schmid<sup>3</sup>  |  
Jennifer Ernst<sup>4</sup> | Dominik Göttsche<sup>2,5</sup> | Oliver Röhrle<sup>3,5</sup>  | Miriam Schulte<sup>1,5</sup>

<sup>1</sup>Institute for Parallel and Distributed Systems, University of Stuttgart, Stuttgart, Germany

<sup>2</sup>Institute of Applied Analysis and Numerical Simulation, University of Stuttgart, Stuttgart, Germany

<sup>3</sup>Institute for Modeling and Simulation of Biomechanical Systems, University of Stuttgart, Stuttgart, Germany

<sup>4</sup>Department of Trauma Surgery, Hannover Medical School, Hannover, Germany

<sup>5</sup>Stuttgart Center for Simulation Science, University of Stuttgart, Stuttgart, Germany

## Correspondence

Robin Lautenschlager, Institute of Applied Analysis and Numerical Simulation, University of Stuttgart, Pfaffenwaldring 57, Stuttgart 70569, Germany.

Email: [robin.lautenschlager@mathematik.uni-stuttgart.de](mailto:robin.lautenschlager@mathematik.uni-stuttgart.de)

## Funding information

Deutsche Forschungsgemeinschaft, Grant/Award Numbers: 390740016, 465243391

## Abstract

The functioning of the neuromuscular system is an important factor for quality of life. With the aim of restoring neuromuscular function after limb amputation, novel clinical techniques such as the agonist-antagonist myoneural interface (AMI) are being developed. In this technique, the residual muscles of an agonist-antagonist pair are (re-)connected via a tendon in order to restore their mechanical and neural interaction. Due to the complexity of the system, the AMI can substantially profit from *in silico* analysis, in particular to determine the prestretch of the residual muscles that is applied during the procedure and determines the range of motion of the residual muscle pair. We present our computational approach to facilitate this. We extend a detailed multi-X model for single muscles to the AMI setup, that is, a two-muscle-one-tendon system. The model considers subcellular processes as well as 3D muscle and tendon mechanics and is prepared for neural process simulation. It is solved on high performance computing systems. We present simulation results that show (i) the performance of our numerical coupling between muscles and tendon and (ii) a qualitatively correct dependence of the range of motion of muscles on their prestretch. Simultaneously, we pursue a Bayesian parameter inference approach to invert for parameters of interest. Our approach is independent of the underlying muscle model and represents a first step toward parameter optimization, for instance, finding the prestretch, to be applied during surgery, that maximizes the resulting range of motion. Since our multi-X fine-grained model is computationally expensive, we present inversion results for reduced Hill-type models. Our numerical results for cases with known ground truth show the convergence and robustness of our approach.

## KEYWORDS

Bayesian parameter inference, coupled simulations, electrophysiological skeletal muscle models, Hill-type muscle models, In-Silico Laboratory, neural feedback

# 1 | INTRODUCTION

Diseases of the musculoskeletal system can result in a severe restriction of the quality of life. The simulation and identification of individual parameters of human skeletal muscle action is emerging as a valuable tool for individualized treatment of diseases or surgical planning. It constitutes a complex task typically involving the coupling of several muscles and the nervous system. Humans are able to activate their muscles in a highly controlled way and to sense the position of their limbs even with eyes closed. These skills are summarized under the term ‘proprioception’ and are enabled by sophisticated interactions within the neuromuscular system, that is, between skeletal muscles, the nervous system and mechanosensors. Movement commands are generated in the brain and transferred to the muscles via the neural system. Specialized sensory organs monitor the resulting movement and provide the nervous system with information on the position of the muscle or the limb (cf. Figure 1A) [55]. The nervous system then adjusts the control command [55].

Skeletal muscles are only able to contract in one direction. Therefore, they typically work in pairs of an agonist and an antagonist, in order to allow for the multi-directional movement of the limbs. Agonist-antagonist muscle pairs are coupled mechanically by bones, tendons and other structures. In addition, muscle pairs are coupled by nervous pathways, which transmit sensory information. For instance, if one muscle is activated, the nervous system processes sensory information from the agonist as well as from the antagonist to generate a targeted control command [55]. Any intervention to this system is prone to disturb its sensitive control principle.

Surgical interventions, in particular amputations, are a major intrusion into this system. Conventional amputations destroy the mechanical link between residual muscles and thereby heavily impair the exchange of proprioceptive sensory information [35, 75]. Further, little attention has been given to tendons so far, even knowing that many proprioceptive sensory organs are located at the musculo-tendinous junctions [52]. The agonist-antagonist myoneural interface (AMI) is an emerging surgical approach that aims to overcome these drawbacks [44]. The procedure is based on establishing a mechanical connection between residual muscles with the goal to maintain the mechanical and sensory coupling of naturally linked muscles (Figure 1B) [44]. First results show that the control of the residual muscles is improved and pain is reduced in patients after receiving an AMI [90, 94]. In the future, the AMI procedure has the potential to deliver neural feedback to the user and, thereby, facilitate the control of advanced prosthetic limbs [45].

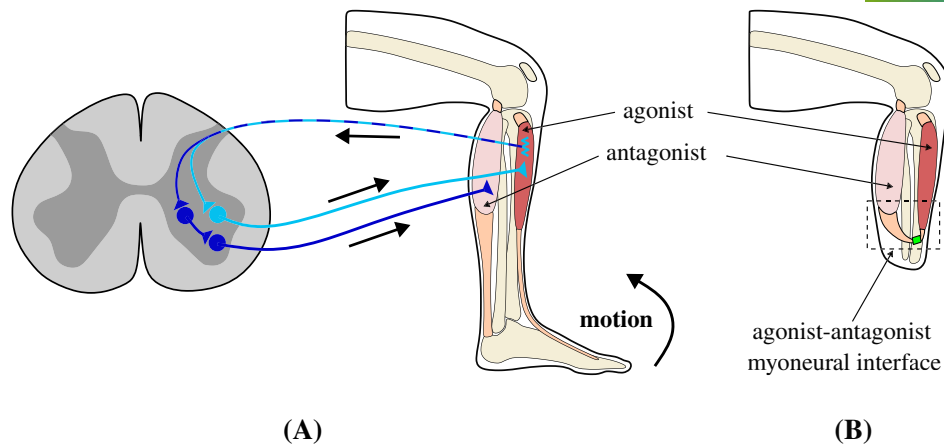
However, such a complex surgery requires a careful pre-surgery evaluation and planning, for example, to decide which muscles are suitable and how exactly their mechanical connection should be designed in surgery. Under undisturbed conditions, all muscles in the body are prestretched by a certain amount via their connections to joints and bones, that is, they would be shorter in a completely relaxed state. Since muscle stretch is directly related to the muscle’s force-production capability, the prestretch directly influences the force that can be applied to the joint and the antagonistic muscle, respectively and, thus, the quality of the muscle functionality [33]. Further, damaged or degenerated muscle tissue influences the active and passive mechanical properties of the muscles and thus, the mechanical and sensory interaction between them.

## 1.1 | Goals and challenges

We propose and pursue a close co-design approach for the development of a simulation-based optimization framework (*in silico* laboratory) to support the identification of optimal physiological and anatomical parameters for the surgical amputation procedure. We address the prestretch of the residual muscles, and thus the resulting range of motion, as our first use case. By co-design, we mean that we have to consider and develop three methodological components: (i) clinical experiments, (ii) detailed mathematical models for muscle physiology and neural feedback loops, (iii) an inversion approach allowing us to identify parameters under uncertainty and based on sparse data.

This requires complex coupled simulations involving at least two muscles and their mechanical connection by a tendon, each of the muscles being described by multi-scale multi-physics hierarchical muscle models. These span from 0D sarcomere models over 1D activation potential diffusion and reaction along muscle fibers to 3D (nonlinear) mechanical deformation. Also, models of sensory organs that provide the central nervous system with information on the muscle’s state and neuron models that activate the muscles after integrating central activation commands and information from sensory organs must be included.

A simulation framework for such a multi-X model needs to provide numerical and software coupling among strongly different temporal and spatial scales and dimensions of the model parts as well as between muscle and tendon in addition to optimized numerical methods for the individual scales. To achieve the desired accuracy, high resolutions and, thus, high performance computing techniques are required.



**FIGURE 1** Our target application—agonist-antagonist muscle pairs and the associated sensory feedback loop in a healthy lower leg (A) and after AMI (agonist-antagonist myoneural interface) amputation (B). In more detail, the left picture (A) shows the stretch reflex as an exemplary sensory reflex. A lengthening of the agonist muscle (red) is registered by sensory organs within the muscle. In response, the sensory organs activate populations of neurons (circles) in the spinal cord via synapses (triangles). Consequently, the monosynaptic stretch reflex pathway (light blue) leads to an activation of the agonist muscle. Additionally, the disynaptic reciprocal inhibition pathway (dark blue) leads to the inhibition of the antagonist muscle (pink). Tendons are shown in orange. The right picture (B) shows the anatomical situation after transtibial amputation. The residual muscles are surgically connected by their tendons (orange) to build an agonist-antagonist muscle pair. The connected tendons are placed through an artificial guide (green).

To identify parameters of interest in the model, that is, to infer their values and to ultimately optimize those parameters, statistical methods can be used. To do so, our multi-X models need to be solved many times. For instance, we want to infer the prestretch given the targeted range of motion of the two-muscle-one-tendon system in the AMI. Solving such severely ill-posed inversion problems under significant uncertainty of measurements is a formidable challenge. In terms of clinical experiments, access to data and observations that allow to derive a clinically meaningful and computable objective function is mandatory.

## 1.2 | Contributions, limitations, and paper structure

In this paper, we approach the envisioned *in silico* laboratory for the AMI from two sides and present first results. Currently, some limitations are unavoidable, which we sketch in the following alongside our concrete contributions. Also, we do not employ clinical data in a more than qualitative way yet.

Section 2 provides the necessary background in terms of neuromuscular physiology, clinical procedures and clinical requirements. In Section 3, we first provide the state-of-the-art in neuromuscular system modeling and simulation as it applies to our AMI setting. We then present our full multi-X model of the muscle-tendon-muscle setting, comprising these three so-called participants. Their coupling is outlined in Section 4, along with implementation details and the numerical approaches for the full forward model in the OpenDiHu and preCICE software frameworks. The implementation currently does not support full neural coupling between the participants, but only the mechanical coupling.

As explained above, the simulation of the full multi-X model is computationally challenging. Thus, to simultaneously make progress toward parameter inference and, ultimately, parameter optimization, we need to resort to a simplified forward model. For this, we adapt well-known Hill-type models in Section 5. Section 6 then provides background on Bayesian statistics, inversion and parameter inference, followed by a description of Markov Chain Monte Carlo and Hamiltonian Monte Carlo methods.

Section 7 presents and discusses our results. We start with a brief validation of the full multi-X single muscle setup to proceed to the two-muscle-one-tendon simulation. The experiments highlight the performance of our numerical coupling between muscles and tendon and demonstrate a qualitatively correct dependence of the range of motion of muscles on their prestretch. They also indicate the need for careful choice of various numerical thresholds to stabilize the coupling. For the parameter inference based on the reduced, less costly Hill-type models, we assume that we know the actual

range of motion and invert for the corresponding prestretch. Compared to the known ground truth, we demonstrate the convergence and robustness of our approach.

In Section 8, we conclude, summarize and discuss our plans for future work.

## 2 | MUSCLE PHYSIOLOGY AND CLINICAL PROCEDURES

In this section, we provide the necessary background in terms of neuromuscular physiology that is required for our simulation and optimization approach. In addition, we briefly describe clinical procedures and clinical requirements that motivate the formulation of our approaches and show their clinical relevance and feasibility.

### 2.1 | Basic anatomy and physiology of the neuromuscular system

The neuromuscular system, which enables movement, is composed of skeletal muscles, neurons and sensory organs. The neurons are the control elements that activate muscles, while the sensory organs monitor the state of the muscle. The sensory information is forwarded to the neurons, which, in turn, adjust their activation command to the muscle, which eventually contracts and produces a joint movement (cf. Figure 1A) [55].

Skeletal muscles are structured hierarchically. The basic unit of a skeletal muscle is the sarcomere. Nearby sarcomeres are arranged together in myofibrils. Several parallel myofibrils constitute a skeletal muscle fiber, which corresponds to a muscle cell and can extend to the full length of the muscle. Muscle fibers are grouped together in fascicles, that constitute the entire muscle [62].

The basic functional unit for muscle activation is the motor unit. A motor unit consists of a nerve cell (motor neuron) and all the muscle fibers that it innervates. The muscle fibers are activated by an electrical signal, the so-called action potential, which is produced by the motor neuron and transmitted to the muscle fibers at the so-called neuromuscular junction [26]. The action potential propagates through the muscle fibers toward both ends of the muscle, leading to the generation of force in the sarcomeres, which builds up and results in the contraction of the muscle fiber. Thereby, every action potential triggers a force twitch of defined amount. Consequently, muscle force can be regulated by two means: (i) by increasing the number of motor units, and thus fibers, that are recruited; and (ii) by increasing the frequency with which recruited muscle fibers are activated. The total muscle force is the sum of the individual force responses of the recruited fibers. Finally, a movement is created by transmitting muscle force to the joint via tendons.

Overall, the activity of the motor neurons and the respective motor units is determined by central input from the brain and the information provided by sensory organs. The feedback provided from sensory organs is organized in reflex pathways [55]. Thereby, the most important sensory organs are muscle spindles, which are sensitive to muscle length and length changes [61]. They also provide the sensory information for the most direct reflex pathway, which is the monosynaptic stretch reflex (Figure 1A, light blue). Thereby, muscle spindle activity increases the activity of the motor neurons and consequently leads to contraction of the agonist muscle [89, 104]. In contrast, the reciprocal inhibition reflex delivers the same stretch information to the motor neurons of the antagonistic muscle and decreases their activity (Figure 1A, dark blue) [20, 58].

The reflex pathways induce an interaction between agonist and antagonist via the neural system. In addition, muscles are mechanically coupled via joints and their maximum force production capability is length and velocity dependent [33, 38]. Thus, for an optimal control and performance of a two-muscle system, mechanical parameters such as the prestretch of the components are decisive. Prestretch results from the fact that the muscle is always slightly stretched in all joint positions by its connection to tendons and bones. Prestretch is defined as the length difference between the (theoretical) fully relaxed muscle length and its actual maximally relaxed length within the human body. The prestretch in combination with the passive material properties also restricts the maximal range of motion of muscles and joints on a purely mechanical basis.

### 2.2 | Clinical procedures and requirements

Lower limb amputation still affects more than 60 000 people per year in Germany [102]. Conventional amputation techniques wrap the residual muscle bellies around the transected bone [30]. This disrupts the natural mechanical connection

between agonist and antagonist muscles. Since muscles can only actively contract and not elongate, two muscles, that is, an agonist and an antagonist, are necessary to move a joint and their control is largely determined by feedback from the muscle's sensory organs [55, 79]. Consequently, a disruption of the natural connection between agonist and antagonist heavily distorts natural sensory feedback and poses a major challenge for muscle-controlled prostheses [44]. To overcome these drawbacks, a new surgical technique is currently introduced, the so called agonist-antagonist myoneural interface (AMI) [44]. Thereby, two residual muscles are surgically connected by a tendon (or artificial tissue) within the residual limb (Figure 3) [44]. This technique restores the natural mechanical and sensory connection of the agonist-antagonist muscle pair.

While the results from the first patients look promising, some questions remain unanswered [90, 94]. In the healthy joint, the mechanical coupling between a muscle pair is determined by multiple factors, for example, the lever arm, the weight of the limb and the mechanical properties of the entirety of the tissue that is connected to the joint. In an AMI muscle pair, the mechanical coupling is mainly determined by the amount by which the remaining muscles are prestretched when connected as well as their active and passive tissue properties. The mechanical coupling not only determines the range of motion of the remaining muscles but also directly influences the sensory information about length and tension in the muscle that is delivered to the central nervous system. The range of motion of the AMI can clinically be determined by ultrasound tracking of implanted tantalum beads [94]. These measurements deliver relative fascicle strain measures or the absolute muscle excursion during contraction [16, 91, 94]. First results indicate that a large range of motion of the AMI is beneficial for both, limb perception and control of the residual muscles [91, 95]. However, the optimal prestretch to achieve this is not known a priori. This motivates our approach for aiming at a model-based framework to calculate the optimal prestretch for a maximal range of motion of the AMI construct.

### 3 | THE ELECTROPHYSIOLOGICAL MUSCLE MODEL

The mathematical modeling of skeletal muscles is a nontrivial task due to their complex multi-scale nature. In this section, we give a short overview of the state-of-the-art presented in literature in Section 3.1, followed by a detailed description of the multi-scale mixed-dimensional coupled model we use in this paper as a high fidelity forward model in Section 3.2.

#### 3.1 | State-of-the-art in neuromuscular system modeling and simulation

Existing computational models of the neuromuscular system vary widely in the components considered and their level of detail. Most simulations of (parts of) the musculoskeletal system appeal to multi-body simulation frameworks, in which the mechanical behavior of skeletal muscles and the corresponding tendons is modeled with substantially simplified muscle models. For instance, multi-body musculoskeletal simulations are mostly based on inverse dynamics using an optimization criterion such as the minimal squared muscle activation sum or the minimal metabolic cost of transport to determine muscle activation given a particular motion [2, 77]. Although such models have been used to investigate movement disorders [87], neglecting the physiology and spatial heterogeneity of the neural system considerably limits their application as predictive *in silico* laboratories. On the other hand, the activity of neurons and sensory organs can be included into the model to determine the activation command to the muscles. Such neuronal models considering spinal circuits and basic reflex pathways have been successfully coupled with, for example, 1D Hill-type muscle models and multi-body simulation frameworks [e.g., 24, 27, 51, 56, 93, 96].

Although these models are useful to investigate motor control strategies, 1D muscle models and multi-body simulations fail to fully capture the effect of anatomical and/or surgical adaptations within the musculoskeletal system and cannot correlate kinematic *in silico* predictions with other projections of neuromuscular activity such as electromyogram (EMG) recordings. Further, the spatial distribution of sensory organs within the muscles cannot be represented [86].

Those limitations can be overcome by using continuum mechanical models, based on classical field theories. While there exist several approaches for continuum mechanical muscle models [e.g., 10, 22, 54, 64, 82], system models considering 3D muscle models, neurons and sensory feedback are rare. Volk et al. [100] integrated a pool of motor neurons within a finite element framework. Therein, they modeled the bony structure of the joint in 3D, but used one-dimensional Hill-type models to represent the muscles.

Very promising are so-called multi-scale multi-physics muscle models, which combine continuum mechanics models with electrophysiology models [e.g., 41, 43, 81]. These models are able to capture many relevant physical phenomena that



occur at different temporal and spatial scales: They model the subcellular ion dynamics that lead to force generation at the scale of milliseconds and micrometers, as well as the action-potential propagation and mechanical deformation in the scale of seconds and centimeters. Such models are capable of providing a detailed physiological description of the process of force generation in the muscle and its deformation. Further, the inherent structure of multi-scale models offers the possibility to extend the system by further components, such as sensory organs and reflex pathways.

### 3.2 | An agonist-antagonist two-muscle-one-tendon model

We now discuss an electrophysiological model of a muscle pair for the particular setting where the agonist and the antagonist are mechanically connected by a tendon in the course of the AMI amputation of a limb. The anatomical setup of the model is shown in Figure 1B. For this, we build on the scalable electrophysiological muscle model based on the works in [e.g., 11, 41, 83].

Each participant is represented by a continuum mechanics three-dimensional model. The problem's system of equations originates from the conservation of mass and momentum for each participant:

$$\rho_{M1}\ddot{\mathbf{x}}_{M1} = \nabla \cdot (\mathbf{P}_{\text{passive}}(\mathbf{F}_{M1}) + \mathbf{P}_{\text{active}}(\mathbf{F}_{M1}, \gamma_{M1}) - p_{M1}\mathbf{F}_{M1}^{-T}), \quad \text{div } \dot{\mathbf{x}}_{M1} = 0 \quad \text{in } \Omega_{M1} \times [0, T_{\text{end}}), \quad (1)$$

$$\rho_{M2}\ddot{\mathbf{x}}_{M2} = \nabla \cdot (\mathbf{P}_{\text{passive}}(\mathbf{F}_{M2}) + \mathbf{P}_{\text{active}}(\mathbf{F}_{M2}, \gamma_{M2}) - p_{M2}\mathbf{F}_{M2}^{-T}), \quad \text{div } \dot{\mathbf{x}}_{M2} = 0 \quad \text{in } \Omega_{M2} \times [0, T_{\text{end}}), \quad (2)$$

$$\rho_T\ddot{\mathbf{x}}_T = \nabla \cdot (\mathbf{P}_{\text{passive}}(\mathbf{F}_T) - p_T\mathbf{F}_T^{-T}), \quad \text{div } \dot{\mathbf{x}}_T = 0 \quad \text{in } \Omega_T \times [0, T_{\text{end}}). \quad (3)$$

At the interfaces  $\partial\Omega_{M1-T}$  and  $\partial\Omega_{M2-T}$  between the muscle and the tendon, kinematic and dynamic conditions must be satisfied:

$$\mathbf{x}_{M1} = \mathbf{x}_T, \quad \dot{\mathbf{x}}_{M1} = \dot{\mathbf{x}}_T, \quad \mathbf{P}(\mathbf{F}_{M1}) = \mathbf{P}(\mathbf{F}_T) \quad \text{on } \partial\Omega_{M1-T}, \quad (4)$$

$$\mathbf{x}_{M2} = \mathbf{x}_T, \quad \dot{\mathbf{x}}_{M2} = \dot{\mathbf{x}}_T, \quad \mathbf{P}(\mathbf{F}_{M2}) = \mathbf{P}(\mathbf{F}_T) \quad \text{on } \partial\Omega_{M2-T}. \quad (5)$$

For simplicity, we omit the subscripts M1, M2 and T in our explanation of the Equations (1)–(5). The equations describe the evolution of the muscles' and the tendon's material point displacements  $\mathbf{x}$ , velocities  $\dot{\mathbf{x}}$ , and pressure  $p$ . The material density of each participant is indicated by  $\rho$  while  $\mathbf{F}$  represents the deformation gradient. The right hand sides of the equations comprise the divergence of the first Piola Kirchhoff stress tensor  $\mathbf{P}$ , which can be decomposed in an active and a passive part. The active part contribution only appears in the muscle participants. A term  $p\mathbf{F}^{-T}$  enters the equation to ensure the incompressibility constraint ( $\det(\mathbf{F}) = 1$ ), with  $p$  being a Lagrange multiplier that can be interpreted as pressure.

The passive response of the muscle and the tendon tissue is modeled by an incompressible transversely isotropic material model. For the tendon, we use the formulation proposed in [15], which describes the high stiffness in fiber direction. In case of the muscle, the passive behavior is modeled by the sum of an isotropic and an anisotropic contribution as presented in [41]. We use the Mooney–Rivlin material formulation for the isotropic part, and a function of the fiber stretch  $\lambda_f$  to model the anisotropic behavior in the muscle fiber direction  $\mathbf{a}_0$ . Overall, the model for the passive response of the muscle has four material parameters that are set as in [39].

The active stress tensor  $\mathbf{P}_{\text{active}}(\mathbf{F}, \gamma)$  represents stresses generated by active muscle contraction. It is defined as a function of the deformation gradient tensor  $\mathbf{F}$  and the lumped activation parameter  $\gamma$ . We assume that the active stress contribution acts only along the fiber direction  $\mathbf{a}_0$  as presented in [41].

#### 3.2.1 | Subcellular model

The lumped activation parameter  $\gamma = H(\mathbf{y}, \lambda_f, \dot{\lambda}_f)$  required in Equations (1) and (2) is determined by the so-called subcellular model.  $H$  is a function of the internal state variables summarized in the vector  $\mathbf{y}$ , the fiber stretch  $\lambda_f$  and the fiber contraction velocity  $\dot{\lambda}_f$  [40]. It describes the active muscle stress generation on a microscopic scale, that is, within sarcomeres of the muscle fibers. To compute the vector of state variables  $\mathbf{y}$ , we use the Hodgkin-Huxley Razumova model [40, 47, 78]. It models the activation of ion channels in response to changes in the muscle fiber's transmembrane

potential  $V_m^f$  and is given by a system of differential-algebraic equations (DAEs) for each sarcomere:

$$\frac{\partial \mathbf{y}}{\partial t} = G(\mathbf{y}, V_m^f) \text{ and } I_{\text{ion}} = I_{\text{ion}}(V_m^f, \mathbf{y}), \quad (6)$$

where  $G$  is a function given by the Hodgkin-Huxley Razumova model and  $I_{\text{ion}}$  is the ionic current across the membrane of the muscle fibers.

### 3.2.2 | Action potential propagation model

Equation (6) considers isolated processes that take place in one sarcomere. Another equation is required to account for the propagation of the action potential, that is, to describe the evolution of the value of  $V_m^f$  at each sarcomere's location. To do so, we model each muscle fiber explicitly by solving the monodomain equation

$$\frac{\partial V_m^f}{\partial t} = \frac{1}{C_m^f} \left( \frac{1}{A_m} \sigma_{\text{eff}} \frac{\partial^2 V_m^f}{\partial s^2} - I_{\text{ion}}(V_m^f, \mathbf{y}) + S(V_m^s) \right) \text{ in } \Omega_f \quad (7)$$

in each fiber  $f$ , where  $\Omega_f$  denotes the one-dimensional fiber domain and  $s$  the spatial position along the fiber [41].  $A_m$  is a factor for the membrane area to domain volume relation,  $C_m^f$  is the membrane's electrical capacitance and  $\sigma_{\text{eff}}$  is the effective conductivity.  $S(V_m^s)$  is a source term that models the activation of the fiber by motor neurons, described below.

### 3.2.3 | Motor neuron pool model

The activation of muscle fibers through the source term  $S(V_m^s)$  in Equation (7) takes place at the so-called neuromuscular junctions, see Section 2.1. At the neuromuscular junction, a motor neuron's nerve fiber attaches to the muscle fiber membrane. Each motor neuron innervates multiple fibers of the same muscle and each muscle is innervated by one motor neuron pool. For the definition of the neuromuscular junction and the allocation of fibers to motor neurons, see [64] and [81].

We model each motor neuron by an equivalent electric circuit model based on [19] and [73, 83]. The model considers two anatomical compartments, which represent the cell body (soma) and the dendrite. Each compartment considers a number of ion channels, which allow current exchange between the intra- and extracellular space. The time points of the action potentials, which finally activate the muscle fibers, are determined from the time course of the soma's membrane potential  $V_m^s$ . That means, whenever the soma's membrane potential exceeds a threshold,  $S(V_m^s)$  is temporarily set to a supra-threshold value, causing a nonzero action potential in the respective muscle fibers. The membrane potentials  $V_m^s$  and  $V_m^d$  in the soma and dendrite compartments are described by two coupled ordinary differential equations (ODEs):

$$\frac{dV_m^d}{dt} = \frac{1}{C_m^d} (-I_{\text{ion}}^d(V_m^d) - I_C^d(V_m^d, V_m^s)), \quad (8)$$

$$\frac{dV_m^s}{dt} = \frac{1}{C_m^s} (-I_{\text{ion}}^s(V_m^s) - I_C^s(V_m^d, V_m^s) + I_{\text{spindle}}(\lambda_f, \dot{\lambda}_f) + I_{\text{ext}}). \quad (9)$$

The capacitance of the membrane in each compartment is denoted by  $C_m^d$  and  $C_m^s$ , respectively. The two equations are linked to each other through coupling currents  $I_C^d$  and  $I_C^s$ , where  $I_C^d = -I_C^s$  due to the conservation of electric charge.  $I_{\text{ion}}$  summarizes the ionic currents crossing the respective membrane.  $I_{\text{ion}}^s(V_m^s)$  is composed of a leakage current and the currents through sodium and two types of potassium channels. The variables describing the ionic currents are determined from four ODEs [73, 83]. In the dendrite the ionic current is solely composed of a leakage current.  $I_{\text{ext}}$  represents inputs from the central nervous system,  $I_{\text{spindle}}$  the current originating from the sensory organs, that is, the summed muscle spindle activity.  $\lambda_f$  and  $\dot{\lambda}_f$  denote the fiber stretch and stretch velocity, respectively. In summary, each motor neuron is described by a system of six coupled ordinary ODEs.

### 3.2.4 | Sensory organ model

Sensory organs, that is, muscle spindles are located at a few positions in each muscle. They detect macroscopic length changes in the muscle and provide the motor neurons with this information. The muscle spindle firing rate  $Ia_j$  of a spindle  $j$  is modeled according to [68]. We summarize the contributions of all spindles within one muscle in  $I_{\text{spindle}}$ :

$$I_{\text{spindle}}(\lambda_f, \dot{\lambda}_f) = \sum_{j=1}^{N_{\text{spindle}}} w_j Ia_j(\lambda_f^j, \dot{\lambda}_f^j), \quad (10)$$

where  $N$  is the number of spindles within the muscle,  $\lambda_f^j$  and  $\dot{\lambda}_f^j$  denote the fiber stretch and stretch velocity at the location of spindle  $j$ . The weight  $w_j$  can account for the different strengths and signs of the monosynaptic and the reciprocal inhibition reflex pathway (cf. Figure 3).  $I_{\text{spindle}}$  is used as an input to the motor neuron model (Equation 9). So far, we only consider the spindle feedback of the same muscle, that is, the monosynaptic stretch reflex.

## 4 | IMPLEMENTATION OF THE MULTI-X FORWARD MODEL

In this section we discuss the implementation of the electrophysiological model presented in Section 3.2. A skeletal muscle can contain up to a few hundred thousand fibers and each fiber has an average of 50 000 sarcomeres [62]. Thus, electrophysiologically-based simulations of skeletal muscles with a realistic description of the activation process and the force generation require the use of high performance computing frameworks. Besides, given the multi-X nature of the presented muscle model, a highly modular software is desired. The open-source neuromuscular simulation framework OpenDiHu fulfills these requirements and is our software of choice to implement the full muscle's forward model. To create the two-muscle-one-tendon AMI model, we use a partitioned coupling approach and couple three independent OpenDiHu instances using the open-source coupling library preCICE. In the following, we first present the implementation of the model for a single muscle in OpenDiHu (Section 4.1) before explaining the coupling between muscles and tendon to establish the simulation environment for the two-muscle-one-tendon system via preCICE (Section 4.2).

### 4.1 | Implementation of the electrophysiological muscle model in OpenDiHu

OpenDiHu is an open-source software framework for scalable biophysical simulations of the neuromuscular system.<sup>1</sup> It provides an extensive list of physics-specific solvers for electrophysical processes, generation of forces and mechanical muscle deformation among others, which can be combined in a tree-fashion to create a custom-made solver.

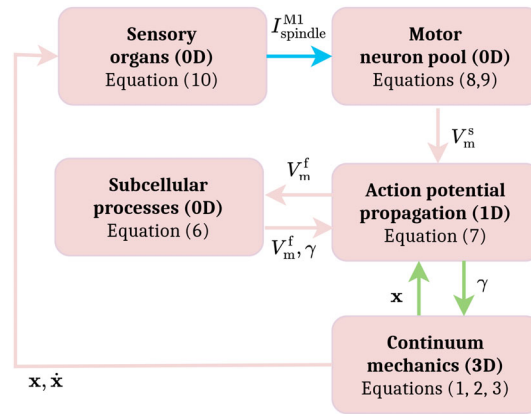
In our case, we combine five physics-specific solvers as shown in Figure 2. We select the physics-specific solvers according to the models described in Section 3.2, for example, the continuum mechanics solver uses an incompressible transversely isotropic Mooney-Rivlin model, the action-potential propagation solver computes the monodomain equation, etc. OpenDiHu's subcellular model equations are not hard-coded, but imported via a code generator from a CellML file to mitigate errors and facilitate reproducibility. CellML is a language based on XML markup that has become a standard to exchange and store reaction models in the biological community [21]. The CellML files used by OpenDiHu and our case are stored online under an open-access policy.<sup>2</sup>

For each physics-specific solver we have to specify a spatial and time discretization. The muscle is represented by the 3D mesh of the finite element continuum mechanics solver. We use 3D hexahedral elements to discretize the muscle domain in space. Quadratic ansatz functions are used for  $\mathbf{x}$  and  $\dot{\mathbf{x}}$ , while we use linear ansatz functions for  $p$ . Embedded in the continuum mechanics domain, we create  $n$  1D fiber meshes. The fiber meshes are identical and have a higher spatial resolution than the continuum mechanics mesh. To exchange information between the fibers and the mechanics mesh, the volumetric data mapping feature of OpenDiHu based on linear interpolation is used. The remaining physics-specific solvers are not mesh-based. The subcellular model solver is called at every point of the 1D fiber meshes, but the solver does not use geometrical information and simply takes an electrical signal as an input.

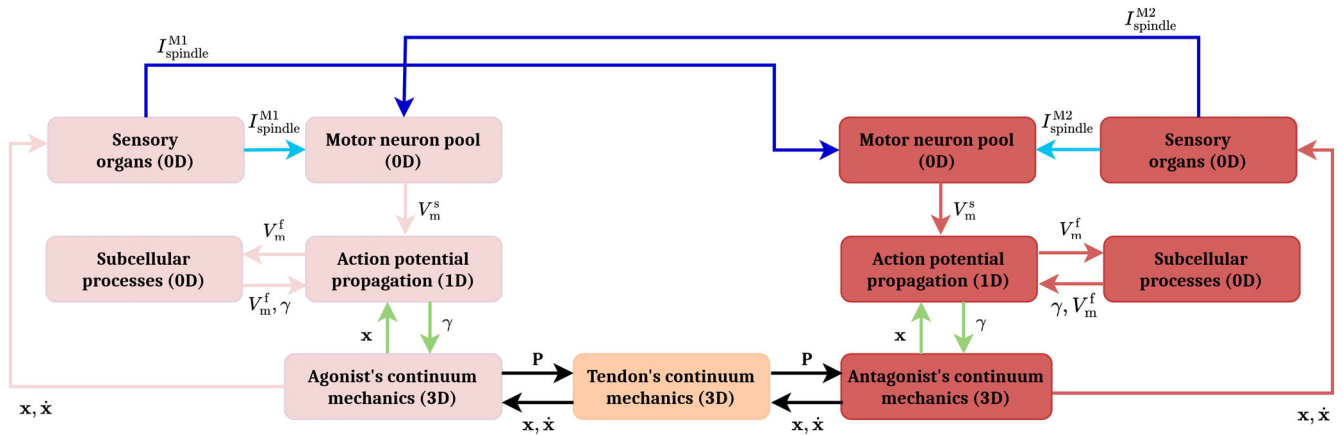
<sup>1</sup><https://github.com/pendihu/pendihu>.

<sup>2</sup><https://doi.org/10.5281/zenodo.4705944>.





**FIGURE 2** Full muscle's forward model, consisting of five different physics-specific solvers here represented by boxes. The arrows between boxes show the data dependencies between sub-models. In light blue, we highlight the transfer of data between the muscle spindles and the motor neuron pool (the monosynaptic reflex pathway). In green, we show the volumetric coupling that takes place between the fibers model and the mechanics model.



**FIGURE 3** Forward model of the AMI amputation. The dark blue arrows show the neural coupling between the two muscles (the reciprocal inhibition pathway), while the black arrows show the mechanical coupling at the contact surface of the muscle participants and the tendon.

Regarding time discretization, OpenDiHu allows users to select different time step and time integration methods for each physics-specific solver. We use a Strang splitting to solve the monodomain Equation (7) in each fiber, splitting between the reaction term (subcellular processes) and the diffusion term (action potential propagation). The reaction term covers a smaller time scale and uses the Heun integrator, whereas the diffusion term occurs at a larger time scale and uses the implicit Euler scheme. The continuum mechanics solver also uses the implicit Euler scheme.

## 4.2 | Coupling with preCICE

For the realization of the two-muscle-one-tendon AMI model, it is necessary to couple three instances of OpenDiHu, called participants in the following. They are distinguished by color in Figure 3. The agonist and antagonist muscle participants are identical and consist of the multi-X forward muscle model described previously, while the tendon consists of a single continuum mechanics physics-specific solver. Each participant is defined by a 3D mesh with no overlap between the meshes of different participants and shared interfaces between the agonist and the tendon as well as between the tendon and the antagonist.

Figure 3 displays the flux of information. Most of the arrows connecting boxes show interactions between physics-specific modules of the same OpenDiHu participant. However, the arrows painted in black and dark blue depict interactions between different OpenDiHu participants, which we implement using the open-source coupling library preCICE.<sup>3</sup> preCICE is a library for partitioned multi-physics simulations that allows for a minimally invasive integration and for the coupling of multiple participants and combinations of strong and weak coupling options [18]. Besides, it provides ready-to-use state of the art data mapping methods as well as acceleration methods. The preCICE library is used by each participant of a coupled simulation via so-called adapters, pieces of additional software that call preCICE via its application user interface (API). In the following, we describe our specific choices for the coupling instances in our setting.

#### 4.2.1 | Mechanical coupling

The mechanical coupling between the tendon and each of the muscles is realized as a partitioned Dirichlet–Neumann coupling, that is, the participants solve their equations separately and exchange the following data after each coupling iteration: At the muscle-tendon interface, one participant sends the values of the displacement  $x$  and velocity  $\dot{x}$  while the other sends the values of stresses  $P$ . Incoming displacement and velocity values are read and enforced by a Dirichlet boundary condition, whereas incoming values of traction are applied using a Neumann boundary condition. A proof-of-concept of a muscle-tendon complex using the preCICE surface adapter has already been presented in [65]. In this paper, we present the three participant coupling connecting two muscles via a tendon for the first time. Using explicit coupling, that is, calling each participant only once per coupled time step is enough in some cases, however, prone to instabilities and inaccuracies. To improve the stability and accuracy of the coupling, we added implicit coupling in the preCICE adapter for OpenDiHu, that is, to iterate between participants until convergence of all coupling variables, that is, tractions, velocities and displacements at both muscle-tendon interfaces. To accelerate these iterations, we use the quasi Newton acceleration methods offered by preCICE [18, 59].

#### 4.2.2 | Neural coupling

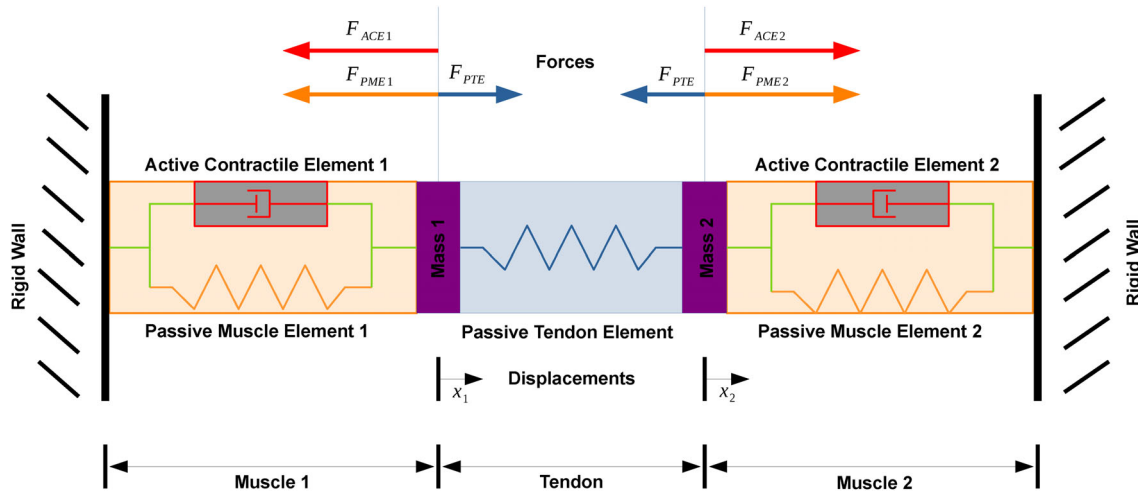
The dark blue arrows in Figure 3 show the reciprocal inhibition pathway, or in other words, the neural coupling. Note that this is an example of unidirectional coupling and the data to be sent by each muscle to the other muscle transferred are essentially the same transferred by the monosynaptic stretch reflex pathway within a single muscle, which is already implemented in OpenDiHu. The implementation of the neural coupling is still missing, but can be done with tools provided by preCICE.

## 5 | A SIMPLIFIED FORWARD PROBLEM FOR INVERSION

The computational model presented in Sections 3.2 and 4.1 is a forward problem, determining deformations and forces in a very accurate and detailed manner, due to its multi-X nature. As described in Section 2, AMI as clinical application ultimately requires optimal physiological parameters or, in general, the identification of (patient specific) parameters based on experimental data. This is an inverse problem, that is, we need to determine plausible input parameters of the simulation based on a known set of output data, given by experimental observations. In our particular case, we want to determine the unknown muscle prestretch from mechanical experiments.

Obviously, solving inverse problems is costly as it requires repeated solutions of forward problems with varying inputs. Ultimately, for very fine-scale reference solutions, the multi-scale hierarchical muscle model from Section 3.2 must be employed. However, to facilitate and speed up the development process of our optimization approach, the fully resolved model (temporarily) needs to be replaced by a simplified model. Thus, we simulate the muscle behavior based on Hill-type muscle models. These models provide a less complex computation by only solving a system of ODEs. The downside is that these models do not account for microscale information about the muscle, and also do not account for spatial extensions of the muscle. Such effects thus need to be modeled in full dimension. We adapt such a model to the AMI setup of a two-muscle-one-tendon system, which calculates the range of motion of the system for given prestretch.

<sup>3</sup><https://github.com/precice/precice>.



**FIGURE 4** Simple Hill-type model approach for a two-muscle-one-tendon system. The forces are divided into active muscle forces  $F_{ACE}$ , passive muscle forces  $F_{PME}$  and a passive force  $F_{PTE}$  caused by the tendon.

## 5.1 | A Hill-type two-muscle-one-tendon model

Hill-type models are a class of muscle models which aim to predict the lumped passive and active muscle forces during movement on a macroscopic scale [14, 46]. Therefore, the behavior of the muscle system is divided into single distinct components, that is, active contractile elements for force generation and passive elements as response to stretch and contraction [84]. The passive stress-strain relation for the muscle follows a highly nonlinear hyperelastic behavior and we assume a muscle-tendon system without external forces and friction [66].

For simplicity, Hill-type muscle models combine the influence of excitation, activation and motor unit recruitment dynamics of the neuromuscular system into a single activation parameter [14]. Figure 4 shows the basic components of our Hill-type two-muscle model with a connecting tendon. The muscles are fixed at the outer ends, the output of the model is the movement of the two muscle ends connected to the tendon (i.e., the positions  $x_1$  and  $x_2$ ). While the tendon is described by a simple spring-type mechanical behavior, the two muscles have an additional active component. The final model is given by a system of ODEs derived from the balance of forces at the muscle ends:

$$m_1 \ddot{x}_1(t) = F_{PTE}(t) - F_{ACE1}(t) - F_{PME1}(t), \quad (11)$$

$$m_2 \ddot{x}_2(t) = -F_{PTE}(t) + F_{ACE2}(t) + F_{PME2}(t). \quad (12)$$

Here,  $t$  is the simulation time and  $x_1$  and  $x_2$  are the displacements at the muscle-tendon connections compared to the stress-free positions. The effective mass  $m_i$  for  $i \in \{1, 2\}$  is derived from the mass of each muscle and the tendon [36, 80, 97].  $F_{ACEi}$  is the active force of the contractile element and  $F_{PMEi}$  the passive muscle force for muscle  $i$ .  $F_{PTE}$  is the passive force induced by the tendon.

In the following, we introduce the equations for all force components, following [84]. We omit the muscle numbering 1 and 2 as all formulas hold for both muscles.

### 5.1.1 | Active force $F_{ACE}$ generated by the contractile element

The active force  $F_{ACE}$  generated by the contractile element depends on a force-length relationship given by a function  $f_L$  depending on the muscle length  $\ell_M$ , a force-velocity relationship given by a function  $f_v$  depending on the contraction velocity  $v_M$ , and an activation function  $a$ ,

$$F_{ACE}(t) = F_0^M \cdot a(t) \cdot f_L(\ell_M(t)) \cdot f_v(v_M(t)), \quad (13)$$

with the maximum isometric force  $F_0^M$ . The activation  $a$  with values between zero and one is defined by a first order ODE [76]:

$$a'(t) = \frac{1}{\tau_{\text{rise}}} (1 - a(t)) \cdot u(t) + \frac{1}{\tau_{\text{fall}}} (a_{\text{min}} - a(t)) \cdot (1 - u(t)). \quad (14)$$

$u \in \{0, 1\}$  is the neural input at time  $t$ , and the time activation and deactivation constants  $\tau_{\text{rise}}$  and  $\tau_{\text{fall}}$  account for the time delay between the neural stimulation of the muscle and the actual force generation through cross-bridge cycling. The functions  $f_L$  and  $f_V$  are chosen according to the muscle model proposed by Thelen in [84].  $a_{\text{min}}$  is a constant safety factor for numerical stability.

### 5.1.2 | Passive force $F_{\text{PME}}$

The passive force generation of the muscle depends on the ratio between the actual muscle length  $\ell_M$  and the stress-free length  $\ell_M^{\text{slack}}$ :

$$F_{\text{PME}}(\ell_M(t)) = F_0^M \cdot \frac{\exp\left(\frac{k_{\text{PE}}}{\epsilon_0^M} \left(\frac{\ell_M(t)}{\ell_M^{\text{slack}}} - 1\right)\right) - 1}{\exp(k_{\text{PE}}) - 1}. \quad (15)$$

This addresses the nonlinear hyperelastic stress-strain behavior.  $k_{\text{PE}}$  is an exponential shape factor and  $\epsilon_0^M$  the passive muscle strain due to the maximum isometric force  $F_0^M$ .

### 5.1.3 | Passive tendon force $F_{\text{PTE}}$

The passive tendon force is described in [84, Eq. 8], where the specific numbers are fitted to experimental studies on human tendons and is given by

$$F_{\text{PTE}}(\ell_T(t)) = \begin{cases} F_0^M \cdot 0.10377(\exp(91 \cdot \epsilon_T(t)) - 1) & \text{if } 0 \leq \epsilon_T(t) \leq 0.01516, \\ F_0^M \cdot (37.526 \cdot \epsilon_T(t) - 0.26029) & \text{if } 0.01516 < \epsilon_T(t) < 0.1 \end{cases} \quad (16)$$

with tendon strain  $\epsilon_T(t) = \frac{\ell_T(t) - \ell_T^{\text{slack}}}{\ell_T^{\text{slack}}}$ . Here is  $\ell_T^{\text{slack}}$  the stress-free length of the tendon. The passive tendon force is divided in an initial nonlinear toe region followed by a linear function.

More detailed information about Hill-type muscle models is given in the literature [8, 9, 23, 37, 74, 85, 98].

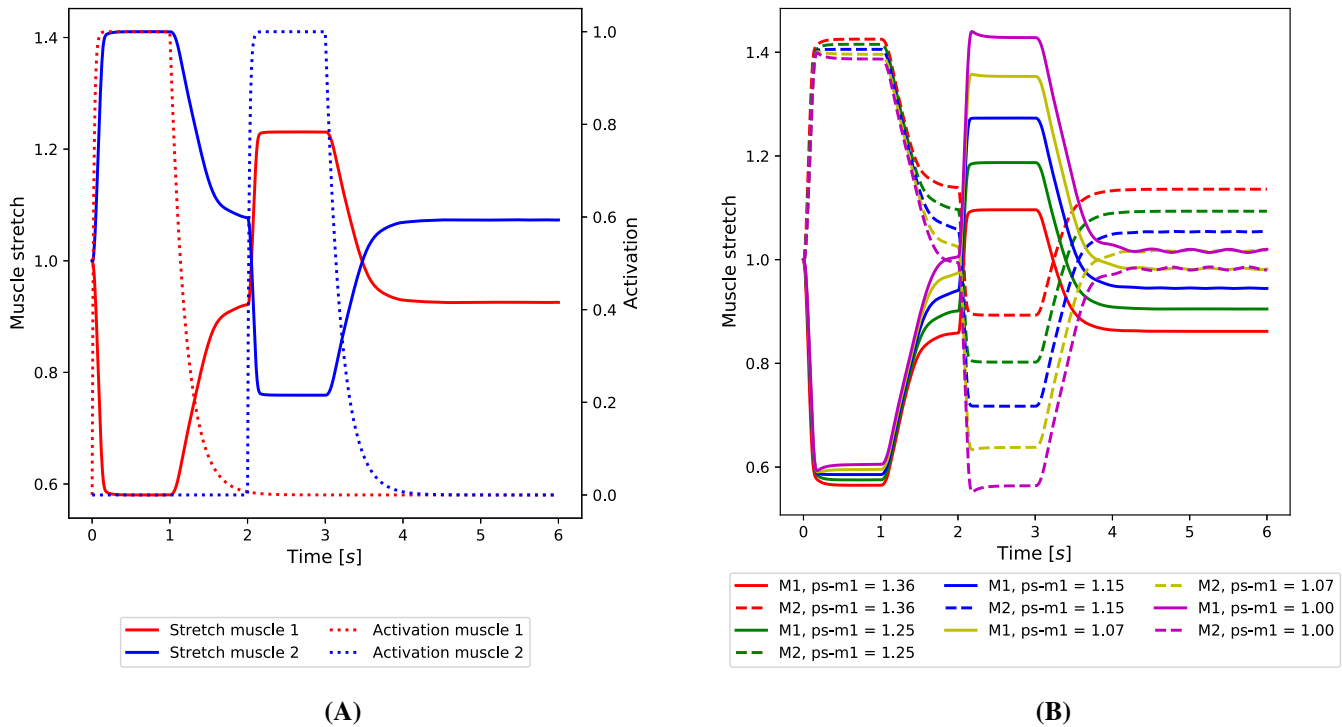
## 5.2 | Incorporating prestretch as optimisation parameter

As discussed in Section 2.2 a parameter of special interest for clinical applications like AMI is the prestretch of the muscle for fixed observed physiological properties. Adapted to our two-muscle-one-tendon system, the prestretches are related to the stretch of the muscles and tendon at the start of the simulation and are therefore defined by the displacements  $x_1(0)$  and  $x_2(0)$  of the muscles at time zero and stress-free length  $\ell_{M1}^{\text{slack}}$ ,  $\ell_{M2}^{\text{slack}}$ , and  $\ell_T^{\text{slack}}$  of the muscle or tendon:

$$\lambda_{M1}^{\text{PS}} = \frac{x_1(0) + \ell_{M1}^{\text{slack}}}{\ell_{M1}^{\text{slack}}}, \quad \lambda_{M2}^{\text{PS}} = \frac{-x_2(0) + \ell_{M2}^{\text{slack}}}{\ell_{M2}^{\text{slack}}}, \quad \lambda_T^{\text{PS}} = \frac{x_2(0) - x_1(0) + \ell_T^{\text{slack}}}{\ell_T^{\text{slack}}}. \quad (17)$$

### 5.3 | Example

To demonstrate that the simplified model works as expected and shows a substantial sensitivity to prestretch, we depict the solution of the system of ODEs given by Equations (11) and (12) in Figure 5A. For this example, muscle 1 is prestretched



**FIGURE 5** Model validation example: Changes of muscle stretch of the two muscles relative to the prestretched muscle length at time zero compared to the activation of the muscles (A) and for a comparison for different prestretches of muscle 1 (ps-m1) (B).

with  $\lambda_{M1}^{PS} = 1.2$  and muscle 2 is less prestretched with  $\lambda_{M2}^{PS} = 1.03$ , but both muscles have the same absolute length in their prestretched states, before neural stimulation is applied. The changes in length over time are induced by the distinct activation of both muscles and by the fact that the system is not in equilibrium at time zero: The tendon has no prestretch. Each muscle receives neural stimulation for one second, starting from time zero, that is, immediately at the beginning of the simulated time, for muscle 1 and after two seconds of simulated time for muscle 2. Therefore we set the neural input  $u$  for each muscle to

$$u_{M1}(t) = \begin{cases} 1 & \text{if } 0 \leq t \leq 1 \\ 0 & \text{else} \end{cases} \quad \text{and} \quad u_{M2}(t) = \begin{cases} 1 & \text{if } 2 \leq t \leq 3 \\ 0 & \text{else} \end{cases}. \quad (18)$$

That is, first muscle 1 acts as agonist and muscle 2 as antagonist, and then vice versa. We observe that the activation of the muscles increases and decreases with a small delay compared to the neural stimulations due to the time delay between stimulation and the resulting force generation in the muscle. Further, we observe that when the agonist is activated, its length decreases and the antagonist is elongated until the opposing force of the antagonist is similar to the active contractile force. The same behavior occurs when muscle 2 is activated, that is, becomes the agonist. Comparing the length changes for the activation of muscle 1 and muscle 2, we observe deviating results. These differences result from the larger prestretch of muscle 1 compared to muscle 2. This also leads to different lengths in the final steady-state when no muscle is activated any more.

To emphasize the direct connection between prestretch and the muscle behavior, we compare the results of the system of ODEs for different prestretches of muscle 1 with a fixed prestretch of muscle 2 with  $\lambda_{M2}^{PS} = 1.03$  in Figure 5B. We observe that only slight changes occur in the contraction of muscle 1 because the prestretch of muscle 2 is fixed and, therefore, the opposing force generation remains the same. On the other hand, when muscle 2 contracts, muscle 1 is stretched more the smaller the prestretch of muscle 1 is. In this case the point where the induced forces by the muscles are equal is reached for a larger absolute stretch of muscle 1. In the time between 4 and 6 seconds oscillations occur for the cases where the prestretch is small in both muscles. This is expected since the small prestretch in the whole system corresponds to a small physical damping after releasing the system at the end of the activation of muscle 2.



## 6 | BAYESIAN INFERENCE

Models like those presented in Sections 3.2 and 5 can calculate the behavior of muscles given characteristics and parameters of the muscle(s), the tendon and the neuromuscular system. In clinical applications like the AMI amputation procedure, the problem is that these characteristics are typically unknown, and only an intended behavior of the muscle is known. An inverse problem of the given models has to be solved, as certain input parameters for the forward problems are unknown. Ultimately, we not only want to determine these parameters, but rather solve the optimization problem arising in AMI surgeries related to the unknown parameters. Specifically, we aim to determine the optimal prestretch of the two muscles that leads to a maximal range of motion. In this paper, we limit ourselves to inverting for the prestretch from a given range of motion. Measurements and the realization of prestretching a muscle are prone to various uncertainties, such as measurement errors and model inaccuracies. In the following, we introduce the background of the inverse problem formulation and our inversion approach.

### 6.1 | The inverse problem

The setup for our inverse problem is a two-muscle-one-tendon system as shown in Figure 4, for our Hill-type model that we have to employ for reasons of computational efficiency. The goal of this example is to find the prestretches of both muscles or, more specifically, their stress-free length to reach a requested range of motion. Based on our model formulation presented in Section 5, we can calculate the maximal and minimal extension of both muscles given stress-free lengths and initial lengths by solving Equations (11) and (12). To formulate the inverse problem, we introduce a so-called forward operator

$$g : \mathbb{R}^2 \rightarrow \mathbb{R}^4, \quad (19)$$

where the input for  $g$  consists of the stress-free lengths of the two muscles, and the output is the minimal and maximal length for each of the two muscles.

In the Bayesian setting that we propose to pursue, measurement errors and model inaccuracies are considered by a forward problem with uncertainties [103],

$$g(\Theta) = F(\Theta) + \xi, \quad (20)$$

where  $F$  denotes the observed values, that is, in our case minimal and maximal muscle extension, and  $\xi \in \mathbb{R}^4$  the measurement errors. The entries of  $\xi$  are assumed to be additive, and normally distributed, that is,  $\xi_j \sim \mathcal{N}(0, \sigma_{\text{err},j})$  for  $j \in \{1, 2, 3, 4\}$ .  $\Theta$  is the vector of investigated physiological input parameters, for example, the prestretch of the first and second muscle.

The basic idea of the inverse problem is to determine values for the input parameters of the forward problem, such that these input parameters lead to solutions close to the observed experimental data in some norm,

$$\min_{\Theta} \|g(\Theta) - F_{\text{obs}}\|^2, \quad (21)$$

where  $F_{\text{obs}}$  denotes measurement data. In our case, this inverse problem is particularly challenging due to the uncertainties and measurement errors, and thus, we pursue a Bayesian approach.

### 6.2 | Bayesian inference as a statistical inversion approach

The Bayesian statistical approach is particularly attractive for our inverse problem setting, as Bayesian inference models both the observables and the parameters as random quantities to address the uncertainties of the involved data [53, 103]. Instead of delivering the one and only correct set of parameters  $\Theta^*$  for the observed output, Bayesian inference calculates a so-called posterior probability density  $\pi(\Theta|F_{\text{obs}})$  [92] of the parameters from a given prior (i.e., initial estimate) density.  $F_{\text{obs}}$  denotes observed model data (measurements).

The theoretical justification of the approach stems from Bayes' theorem of conditional probability describing the posterior density distribution  $\pi$  of the input parameters  $\Theta$  given the observed output values  $\mathcal{F}_{\text{obs}}$  [92]:

$$\pi(\Theta|\mathcal{F}_{\text{obs}}) = \frac{\mathcal{L}(\mathcal{F}_{\text{obs}}|\Theta)\pi_0(\Theta)}{\tilde{\pi}(\mathcal{F}_{\text{obs}})}, \quad (22)$$

Here,  $\pi_0$  is the prior distribution of the input parameters. It contains the statistical initial estimate derived, for example, from expert knowledge or averages over many individuals, for the parameters  $\Theta$ . Via Equation (22), it is improved with the help of observed data  $\mathcal{F}_{\text{obs}}$  [42].  $\mathcal{L}$  is the likelihood distribution of the observed data, given the input parameters. For a normally distributed uncertainty parameter  $\xi$  of the forward problem (20), the likelihood distribution is also normally distributed with

$$\mathcal{L}(\mathcal{F}_{\text{obs}}|\Theta) = \frac{1}{\sqrt{2\pi} \prod_{j=1}^4 \sigma_{\text{err},j}} \cdot \exp \left[ -\frac{1}{2} \sum_{j=1}^4 \left( \frac{g_j(\Theta) - \mathcal{F}_{\text{obs},j}}{\sigma_{\text{err},j}} \right)^2 \right]. \quad (23)$$

Note that this also justifies, from a computational perspective, the necessity of the simplified model from Section 5, as each evaluation of the likelihood function  $\mathcal{L}$  entails one execution of the forward simulation.  $\tilde{\pi}$  is the unconditional distribution of the observed output as a constant scaling factor, which is in general unknown, but ensures the condition that the integral of the posterior density  $\pi(\Theta|\mathcal{F}_{\text{obs}})$  over the parameters  $\Theta$  is one. Since the forward operator  $g$  appearing in the formula for the posterior in Equation (22) is not explicitly given, but requires solving our model equations, it is also not possible to give a closed formula for the posterior and plot the respective probability density. Thus, sampling strategies are required in order to approximate the posterior density with the help of bin counts as depicted, for example, in Figure 10B,D.

Within this approach, we choose the Markov Chain Monte Carlo (MCMC) method [67] for sampling, due to its wide scope for realistic statistical modeling [32]. However, MCMC is quite simple, and can be inefficient as it rejects many samples after the expensive solution of the respective forward problem. The Hamiltonian Monte Carlo (HMC) method [25, 63, 72] improves the MCMC methods by including Hamiltonian dynamics to the method, motivated by molecular dynamics [1]. In the following, we describe the idea of the MCMC method, and how we can improve its efficiency by introducing the improved sampling strategy with the HMC method.

### 6.2.1 | The Markov Chain Monte Carlo method

The Markov Chain Monte Carlo method draws samples from the sought distribution with a cleverly constructed Markov chain: We start with an initial parameter sample  $\Theta^0$  and generate a 'chain' of further samples  $\Theta^n$ ,  $n = 1, 2, \dots$  following an iterative process [92]: (i) Generate a new sample candidate  $\Theta'$  by sampling from a preselected proposal distribution  $q(\Theta'|\Theta^n)$ ; (ii) compute an acceptance criterion  $\alpha(\Theta'|\Theta^n) = \min \left\{ 1, \frac{\pi(\Theta'|\mathcal{F}_{\text{obs}})q(\Theta^n|\Theta')}{\pi(\Theta^n|\mathcal{F}_{\text{obs}})q(\Theta'|\Theta^n)} \right\}$ ; (iii) draw a random variable  $u$  from an equal distribution  $\mathcal{U}(0, 1)$ ; (iv) if  $u < \alpha$ , accept the sample candidate and set  $\Theta^{n+1} = \Theta'$ , else reject it and set  $\Theta^{n+1} = \Theta^n$ .

Note that the evaluation of the posterior distribution for the computation of the acceptance criterion is not explicitly needed because of the quotient we can neglect the constant distribution  $\tilde{\pi}$  of Equation (22). New sample candidates are accepted with a rate motivated by the Monte Carlo integration theory [17]. In essence, the number of iterations spent in a particular region of the parameter space is proportional to the posterior density in that region. Thus, the density of the MCMC samples represents the posterior distribution of the investigated physiological input parameters [92]. If sample candidates are drawn from a normal distribution, that is,  $\Theta'_i = \Theta_i^n + v_i$  with  $v_i \sim \mathcal{N}(0, \sigma_{\text{prop},i})$  with  $i \in \{1, 2\}$ , the MCMC method is called random-walk Metropolis-Hastings algorithm. The basic structure of the MCMC method is sketched in Algorithm 1 [92].

The problem with the Metropolis-Hastings algorithm is that, because of the normally distributed proposal distribution, the samples are highly correlated and the proposal is chosen independently of the underlying muscle-tendon model.

---

**Algorithm 1.** Sketch of the Markov Chain Monte Carlo method
 

---

**Input:**  $\mathcal{F}_{\text{obs}}$ , start proposal  $\Theta^0$  for input data, prior information  
**Output:** Samples for input values drawn from the posterior distribution

```

1: function MCMC
2:   for  $n = 0 : (\text{MCMC-iterations} - 1)$  do
3:     Propose  $\Theta' \sim q(\Theta'|\Theta^n)$ 
4:     Compute  $\alpha(\Theta'|\Theta^n) = \min \left\{ 1, \frac{\pi(\Theta'|\mathcal{F}_{\text{obs}})q(\Theta^n|\Theta')}{\pi(\Theta^n|\mathcal{F}_{\text{obs}})q(\Theta'|\Theta^n)} \right\}$ 
5:     Draw  $u$  from  $\mathcal{U}(0, 1)$ 
6:     if  $u < \alpha$  then
7:        $\Theta^{n+1} = \Theta'$ 
8:     else
9:        $\Theta^{n+1} = \Theta^n$ 
10:    end if
11:  end for
12: end function
  
```

---

### 6.2.2 | The Hamiltonian Monte Carlo method

We propose to employ an improved MCMC, the Hamiltonian Monte Carlo method, with less correlated samples, which are chosen using existing information about the posterior distribution of the physiological parameters. As an improved proposal mechanism, the Hamiltonian Monte Carlo method (HMC) uses Hamiltonian dynamics to explore the parameter space more efficiently by using gradient information of the posterior density and therefore suppressing the diffusive behavior of simple random-walk proposals [101]. It has been demonstrated that HMC outperforms many existing MCMC algorithms, especially in high-dimensional, continuous, and correlated problems [53]. For proposing a new sample, the HMC method executes time steps of a second order ordinary differential equation [72]

$$\ddot{\Theta} = \mathbf{M}^{-1} \nabla_{\Theta} U(\Theta) \text{ with } U(\Theta) = -\log[\pi_0(\Theta) \mathcal{L}(\mathcal{F}_{\text{obs}}|\Theta)], \quad (24)$$

where  $\mathbf{M}$  is a symmetric, positive definite covariance matrix. Thus, new samples are chosen in a direction related to the steepest ascent of the posterior density. Equation (24) is equivalent to the first order system

$$\frac{d\Theta_i}{dt} = \frac{\partial H}{\partial p_i}, \quad (25)$$

$$\frac{dp_i}{dt} = -\frac{\partial H}{\partial \Theta_i}, \quad (26)$$

with the so-called Hamiltonian function

$$H(\Theta, \mathbf{p}) = K(\mathbf{p}) + U(\Theta) \text{ with } K(\mathbf{p}) = \frac{\mathbf{p}^T \mathbf{M}^{-1} \mathbf{p}}{2}. \quad (27)$$

The value of the Hamiltonian  $H$  is an invariant of the solution trajectories of Equations (25) and (26). We preserve this property by using the Leapfrog scheme, a second order symplectic time stepping scheme to solve Equations (25) and (26) [6, 72, 105]. As initial values, we choose the current newest sample  $\Theta^n$  and a random momentum  $\mathbf{p} = \mathbf{M}\dot{\Theta} \in \mathbb{R}^2$  by drawing a sample from the normal distribution  $\mathcal{N}(0, 1)$  for each entry. As the time stepping scheme preserves the value of  $H$  only approximately, we choose the deviation in  $H$  as acceptance criterion  $\alpha$ . A sketch of one HMC step for proposing a new sample inside the MCMC iteration (Algorithm 1) is shown in Algorithm 2, as well as an adapted calculation of the acceptance criterion  $\alpha$  [72].

One downside of the HMC method is that we need to calculate the gradient of the function  $U$ , that is, the gradient of the multiplication of the likelihood distribution and the prior distribution. This gradient especially includes the calculation

---

**Algorithm 2.** Sketch of one Hamiltonian Monte Carlo step
 

---

**Input:**  $\epsilon, L, \Theta^n$   
**Output:**  $\Theta^{n+1}$

```

1: function HMC-STEP
2:   Choose  $\mathbf{p} \sim \mathcal{N}(0, 1)^2$ 
3:   Calculate  $\Theta'$  and  $\mathbf{p}'$  by executing  $L$  Leapfrog discretisation steps
4:     with time step size  $\epsilon$  and initial conditions  $\Theta = \Theta^n$  and  $\dot{\Theta} = \mathbf{M}^{-1}\mathbf{p}$  for Equation (24).
5:   Compute  $\alpha = \exp(U(\Theta^n) - U(\Theta') + K(\mathbf{p}) - K(\mathbf{p}'))$ 
6:   Generate  $u \sim \mathcal{U}(0, 1)$ 
7:   if  $u < \alpha$  then
8:      $\Theta^{n+1} = \Theta'$ 
9:   else
10:     $\Theta^{n+1} = \Theta^n$ 
11:  end if
12: end function

```

---

of the gradient of the forward problem itself. This is much easier for our simple model presented in Section 5 than for the full model from Section 3.2.

The choice of the number of Leapfrog steps and the time step size  $\epsilon$  for the solution of the Hamiltonian system (25) and (26) is a crucial part of the method. The number of steps is estimated by using an adaptive HMC method as extension of HMC, called No-U-Turn-Sampler (NUTS). NUTS calculates the number of steps, such that if we use more steps the distance between the actual sample and the new proposal no longer increases [48]. This technique aims to efficiently chart the whole investigated parameter domain which means that we do not have to hand-tune the number of Leapfrog steps. However, with respect to simulation runtime we cap the maximum number of Leapfrog steps for one iteration by setting a threshold.

The optimal values of HMC parameters can be different for the starting phase of the simulation, which still depends on the randomly chosen starting sample, and the later stationary phase of the simulation. Therefore, a dual averaging method can be included in the NUTS for adaptively adjusting the step size  $\epsilon$ , especially in the starting phase of the simulation [48]. This adaption is important because if  $\epsilon$  is too small we waste computational time on executing too many tiny steps and if  $\epsilon$  is too large the rejection rates of the algorithm increase, therefore again wasting computational time.

For more detailed information about Bayesian inference, the MCMC method and the HMC method we refer to the literature [5–7, 13, 28, 34, 42, 50, 60, 70, 71, 88].

## 7 | NUMERICAL RESULTS

In this section, we present numerical results for both our detailed biophysiological model and for Bayesian inversion based on the simplified Hill-type model. The purpose of our numerical experiments is, first, to show that both models are sensitive to the amount of prestretch. This means the range of motion changes depending on prestretch and the optimal amount of prestretch is not trivial to determine. Second, we show that our Bayesian inversion approach is a powerful tool to identify the distribution of an unknown prestretch for a given range of motion.

### 7.1 | Coupled simulation of the forward model

For our full three-dimensional forward model, we start with simulations of a single muscle scenario to validate the dependence of the simulated range of motion on the muscle prestretch. In a second experiment, we present results for the two-muscle-one-tendon scenario and show that our numerical coupling of different muscle and tendon participants via quasi-Newton methods in preCICE leads to the expected results, if a fully implicit scheme is used for all participants.

**TABLE 1** Full model simulations: resolution of the mesh in the mechanics solver and the action-potential propagation solver for the muscle simulation (left), resolution of the mesh in the mechanics solver for the tendon simulation (center), choice of time steps for the muscle participant (right).

Muscle size (cm)	$3.0 \times 3.0 \times 15.0$	Tendon size (cm)	$3.0 \times 3.0 \times 55.0$	$dt_{\text{mechanics}}$	0.01 ms
Finite elements	$4 \times 4 \times 36$	Finite elements	$4 \times 4 \times 12$	$dt_{\text{Strang}}$	$1e-3$ ms
Fibers	$6 \times 6$			$dt_{\text{reaction}}$	$0.5e-3$ ms
Sarcomeres per fiber	60			$dt_{\text{diffusion}}$	$1e-3$ ms

### 7.1.1 | General setup and parameters

Before explaining the single muscle and the two-muscle-one-tendon scenario in detail, we present parameters used for both scenarios. For the single muscle setup, we solve Equations (1) and (6)–(7), whereas Equations (1)–(7) are used for the two-muscle-one-tendon scenario. As a simplification in both scenarios, we assume rectangular muscles and a rectangular tendon, where the muscles have a size of  $15 \text{ cm} \times 3 \text{ cm} \times 3 \text{ cm}$  each, and the tendon has a size of  $5 \text{ cm} \times 3 \text{ cm} \times 3 \text{ cm}$ . Details on the spatial discretization of the muscles and the tendon are shown in Table 1. Muscle fibers and sarcomeres are distributed uniformly within the muscle. Details on the choice of material parameters for the muscle and the tendon can be found in Tables A1–A3 in the Appendix.

The muscle and tendon mechanics solvers (Equations 1–3) use a time step  $dt_{\text{mechanics}} = 0.01 \text{ ms}$ , which is also used as coupling time step in the two-muscle-one-tendon case. The time steps for the other solvers involved in each muscle are listed in Table 1. We use Strang splitting as described in Section 4 with time step size  $dt_{\text{Strang}}$ . For the diffusion term in Equation (7) we use  $dt_{\text{diffusion}}$ , while for the reaction term (Equation 6) we use  $dt_{\text{reaction}}$ .

### 7.1.2 | Single muscle simulation

In this first experiment we split the simulation in two parts: first we compute the initial prestretch state of the muscle, and then we compute the contraction of the muscle.

The initial prestretch state of the muscle is computed by applying an external pulling force on one end, while the other end is fixed. To do so, we use OpenDiHu to solve the steady state simplification of Equation (1).

In the second part of the simulation, we use OpenDiHu's physics-specific solvers for the continuum mechanics (Equation 1), subcellular processes (Equation 6) and action-potential propagation (Equation 7). The boundary conditions are changed with respect to the first part of the simulation: where we had a pulling force, we apply a homogeneous Dirichlet boundary condition, while a homogeneous Neumann boundary condition (zero forces) is used for the other end, that was previously fixed. We simulate the contraction of the muscle for a time span of 20 ms, which is the approximated time that the action potential takes to propagate from the neuromuscular junctions to the fibers' ends in our muscle.

The two parts of the simulation are englobed in the same OpenDiHu solver, which couples the physics-specific solvers used in both parts. At the start of the first part of the simulation, we assume that the initial displacement and velocity are zero, whereas for the second part of the simulation, the initial conditions are given by the final state of the first part of the simulation.

As shown in Table 2, we have experimented with multiple scenarios, for example, by varying the magnitude of the applied external force. In addition to the scenarios presented in Table 2, we have also considered a scenario with no prestretch. In the case of no prestretch, running the first part of the simulation is redundant, since the muscle remains unaltered.

#### *Results of the single muscle setup*

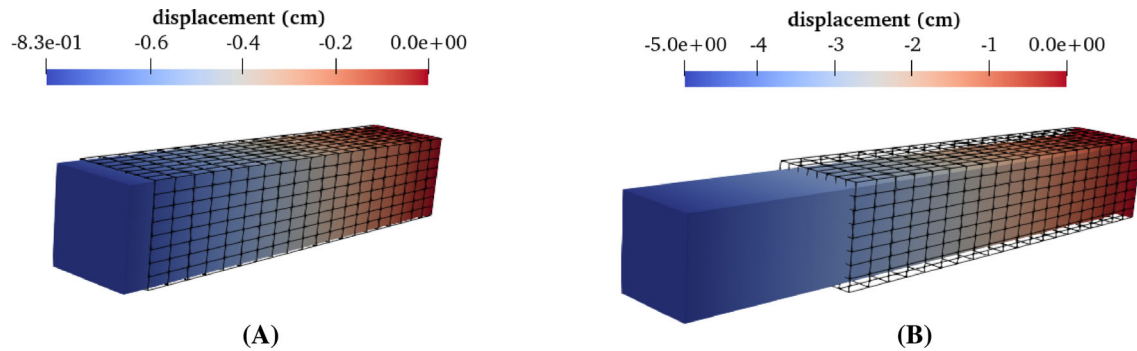
The resulting elongation is computed by averaging over the values at the single elements of each boundary to compute the total length of the muscle and then subtracting the initial length of the muscle. Table 2 shows the results for the first part of the simulation. We can visualize the results in Figure 6 for the case of an external pulling force of 5 and 20 N. Note that forces in Table 2 and in Figure 6 are total forces, which are divided by the area of the boundary surface to generate stress (Neumann) boundary conditions.



**TABLE 2** Single-muscle scenario: different prestretch scenarios for a  $15\text{ cm} \times 3\text{ cm} \times 3\text{ cm}$  muscle.

Applied force	Resulting elongation
1 N	0.15 cm
5 N	0.83 cm
10 N	1.86 cm
20 N	4.95 cm

Note: The table shows the applied force and the resulting length difference.

**FIGURE 6** Single-muscle scenario: Geometry of the muscle after the prestretch simulation for an applied force of 5N (A) and 20N (B). The reference configuration is shown in black.

The results of the contraction part of the simulation for the different prestretch scenarios from Table 2, as well as for the no prestretch scenario, are shown in Figure 7. The initial absolute length of the muscle is different for each case, due to the different results of the first part of the simulation.

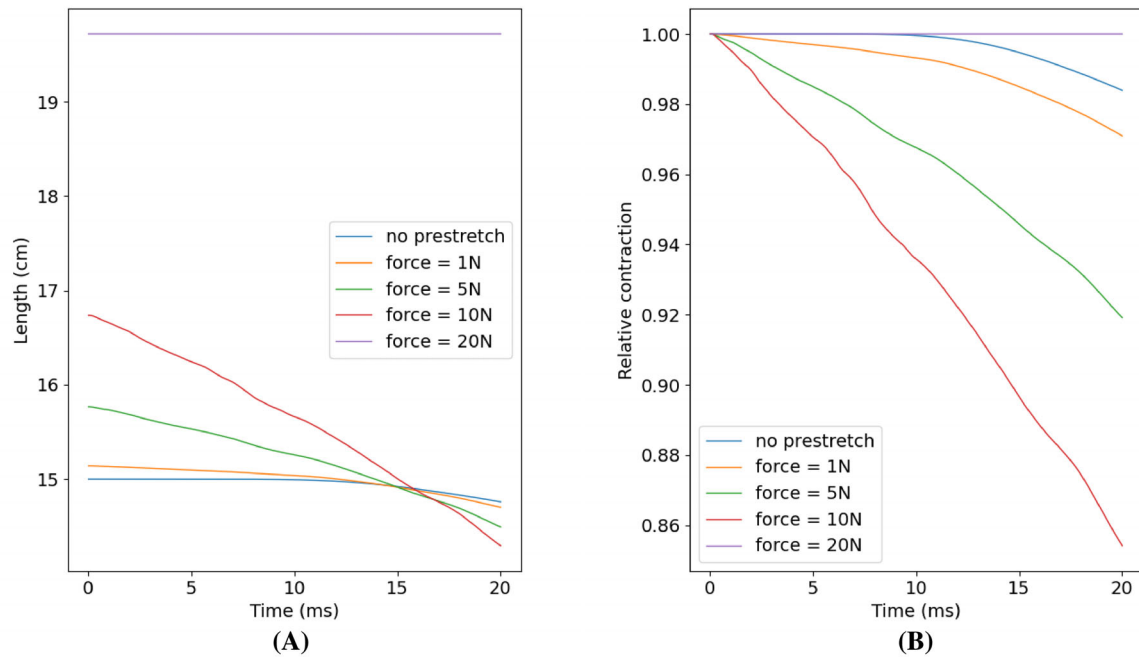
#### Observations and discussion of the single-muscle setup

Because we are considering an incompressible material, the result of applying a force to generate prestretch is a longer but thinner muscle, as observed in Figure 6. The more force we apply, the more the muscle elongates. In Figure 7, we see that the muscle's ability to contract depends on the initial prestretch. The muscle contracts increasingly more for prestretch forces of 1, 5, and 10 N, but for a prestretch of 20 N, the muscle does not contract. The simulation results presented in Figure 7 are consistent with experimental observations. The active muscle force has been reported to first increase with stretch and then decrease, such that no measurable contraction is possible for muscles with very large prestretch [33].

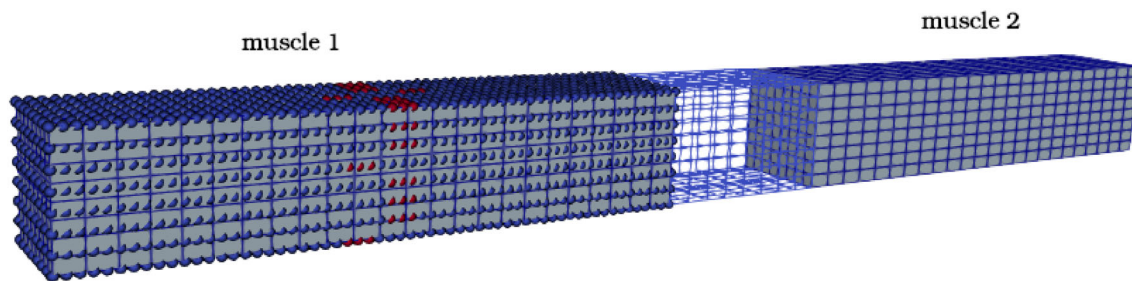
### 7.1.3 | Two-muscle-one-tendon simulations

In this three-participant scenario, we mechanically couple two muscles and a connecting tendon as shown in Figure 8. The figure shows the initial geometry for the simulation. In this setup, we impose an activation in muscle 1 via the source term  $S(V_m)$ , while muscle 2 is not activated and can thus only move in response to the contraction of muscle 1. Both muscles are fixed with homogeneous Dirichlet boundary conditions at the left or the right boundary, that is, the end that is not connected to the tendon. To couple the two muscles with the tendon, we use Dirichlet–Neumann coupling. More specifically, we transfer stresses from muscle 1 to the tendon and displacements and velocities from the tendon to muscle 1, whereas we transfer stresses from the tendon to muscle 2 and velocities and displacements from muscle 2 to the tendon. The muscles' and tendon's meshes have been chosen to ensure matching meshes at the muscle-tendon interfaces. Thus, we do not need sophisticated mapping methods, but we can use nearest neighbor mapping, which is the fastest mapping method provided by preCICE.

In terms of coupling schemes, preCICE allows us to choose between a composition of bi-coupling schemes and the multi-coupling scheme if we couple more than two participants. This means that we can either apply the same scheme,



**FIGURE 7** Single-muscle scenario: Development of the muscle contraction over time for different forces used to prestretch the muscle. The absolute muscle length (A) and the relative muscle length in percentage of the fully relaxed muscle length of 15 cm (B) are shown.

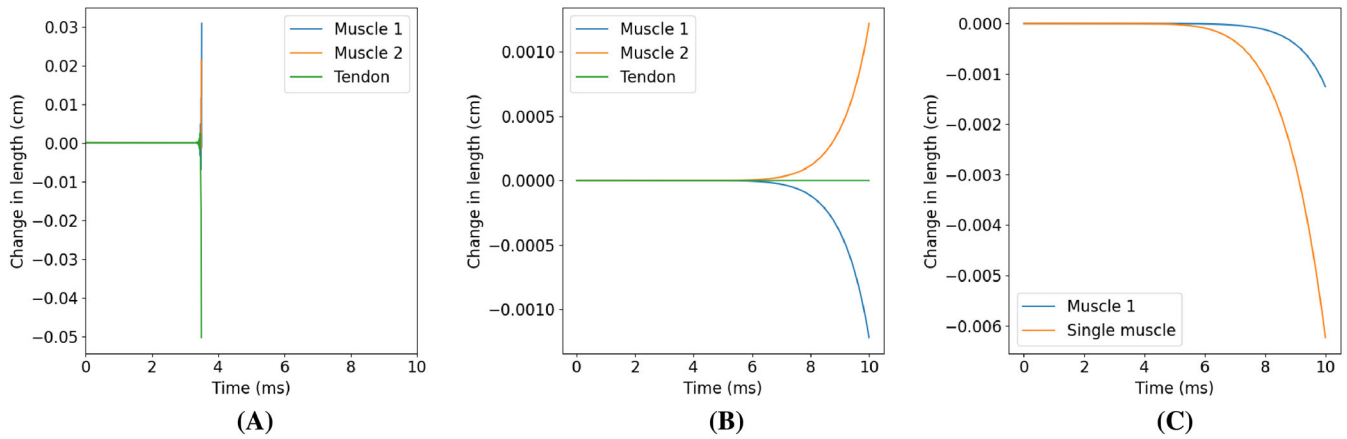


**FIGURE 8** Two-muscle-one-tendon scenario: initial setup, where muscle 1 and muscle 2 are mechanically connected by the tendon. In muscle 1, we show the fiber mesh and mark the first sarcomeres that are electrically stimulated in the simulation in red.

for example, quasi-Newton to all coupling variables at all interfaces between all participants or define separate coupling schemes for each interacting pair of participants. It is known that using implicit coupling is necessary to avoid numerically induced nonphysical behavior, particularly if we send stresses from a stiff material (e.g., the tendon) to a soft material (e.g., muscle 2). Thus, in case of a combination of bi-coupling schemes, we select explicit coupling to couple muscle 1 to the tendon and implicit coupling to couple the tendon to muscle 2. Using the multi-coupling scheme provided by preCICE is equivalent to using implicit coupling both at the muscle 1—tendon interface and at the tendon—muscle 2 interface and to handle all coupling variables in a single quasi-Newton solver. The parameters to configure implicit coupling with preCICE, including the stopping criteria (maximal number of coupling iterations and relative/absolute tolerances) are given in Table A4 in the Appendix of this paper.

To accelerate the implicit coupling iterations within each time step for the coupled simulation, we apply quasi-Newton acceleration using a residual-sum preconditioner for both coupling options—the implicit part of bi-coupling and the multi-coupling. For exact parameter choices, see Table A5 in the Appendix. For the exact definition of these parameters, we refer to the preCICE documentation.<sup>4</sup>

<sup>4</sup><https://precice.org>.



**FIGURE 9** Two-muscle-one-tendon scenario: Results for the change in length of the muscles and the tendon with respect to initial length without prestretch. Muscle 1 contracts due to activation, which results in the elongation of muscle 2. We compare two different coupling configurations for preCICE: a composition of bi-coupling schemes, which crashes after less than 4 ms (A), and the multi-coupling scheme, which yields stable results (B). We also compare the contraction of the muscle within the AMI setup to that of a single muscle.

### Results for the two-muscle-one-tendon setup

Figure 9 shows the change in length for both muscles throughout the first 10 ms simulated time. If a composition of bi-coupling schemes is used, the simulation becomes unstable and crashes as shown in Figure 9A, whereas, if the multi-coupling scheme is used, the simulation is stable as can be seen in Figure 9B. In Figure 9B, we observe how muscle 1 contracts, muscle 2 elongates, and the tendon's length remains almost constant. Figure 9C compares the contraction of an isolated muscle to the contraction of the muscle 1 in the AMI setup. We observe that the isolated muscle contracts more than muscle 1. Besides, the contraction starts much earlier than in muscle 1. Note that in both cases the simulation starts from a no-prestretch state.

### Observations and discussion of the two-muscle-one-tendon setup

Figure 9A shows the expected results: when muscle 1 contracts, muscle 2 elongates. That the length of the tendon remains constant can be explained because tendon tissue is much stiffer than muscles. In terms of stability of the numerical coupling between the three participants, Figure 9A,B shows that the quasi-Newton coupling via preCICE leads to stable results only if we choose fully implicit coupling between all participants. The multi-coupling scheme is more stable than the composition of bi-coupling schemes, since only this option in preCICE avoids explicit coupling completely. When we use explicit coupling, the contraction of muscle 1 is computed without considering the additional mass of the tendon and of muscle 2 that is accelerated, nor the fixed end of muscle 2. Thus, the displacements and the velocities sent to the tendon are overestimated, which quickly leads to oscillations and finally the crash of the simulation.

The reduced contraction of muscle 1 compared to the single muscle scenario in Figure 9C, can be explained due to the additional mass of the tendon and the second muscle and its inertia. The activation of the muscle in both scenarios is performed identically, but in the AMI setup the muscle is attached to a tendon and a second muscle and the whole system is fixed with homogeneous Dirichlet boundary conditions at both ends, while the single muscle has an open end, that is, a homogeneous Neumann boundary condition. This means that the muscle activation in the two-muscle-one-tendon scenario has to overcome a much larger resistance than in the single muscle case.

## 7.2 | Bayesian inference results

In this section, we validate the feasibility of the Markov Chain Monte Carlo methods presented in Section 6 for the estimation of model parameters in a simplified Hill-type muscle model introduced in Section 5.

## 7.2.1 | Description of the muscle setup

The muscle setup is a two-muscle-one-tendon system as shown in Figure 4. At time zero, the observed length of both muscles is 15 cm and the tendon has a length of 5 cm. The tendon is assumed to be stress-free at time zero, but the muscles are assumed to be prestretched. Note that this initial state is not an equilibrium state of the system since stresses in the two muscles and in the stress-free tendon are not balanced. Starting at time zero first muscle 1 is activated for one second, acting as agonist, followed by another second with no activation and a third second with activation of muscle 2, now acting as agonist. Then the activation is stopped completely and, thus, the system approaches its steady state.

The parameters of interest for this example are the prestretches of both muscles or, equivalently, their stress-free lengths required to reach a requested range of motion for each muscle. Hereby, the range of motion is given by the minimal (contraction) and maximal (extension) length of each muscle, that results from the given activation of both muscles. Therefore, the forward operator for our inversion problem is a mapping from  $\mathbb{R}^2$  to  $\mathbb{R}^4$  and we use the simplified Hill-type muscle model presented in Section 5 to obtain the solution of the forward problem. In the investigated case, we set the targeted maximal length of muscle 1 to 18.3 cm and its minimal length to 8.7 cm, for muscle 2, we analogously choose 21.0 and 11.5 cm. These values represent the observed data of our use case for the solution of the inverse problem. The exact stress-free muscle lengths to generate these extensions are known for validation purposes and are 12.5 cm for muscle 1 and 14.5 cm for muscle 2.

The effective muscle masses  $m_1$  and  $m_2$  are set to one for simplicity. The maximum isometric force was arbitrarily chosen in the range of experimental data as  $F_0^M = 6280$  N [98]. All further model parameters of the Hill-type muscle model are chosen similar to [66, 84] and are displayed in the Appendix A.2 in Tables A6 and A7. We mention that all values of the muscle-tendon setup are chosen in a generic way and do not represent specific muscles as we only demonstrate the concept of the Bayesian inference models.

To calculate the estimated distribution of the stress-free length for each muscle (or equivalently their prestretch), we use the Hamiltonian Monte Carlo method presented in Section 6. The prior information about the stress-free muscle lengths of both muscles is a normal distribution given as  $\mathcal{N}(13.5, 6.0)$ . For the calculation of the gradient required in the HMC method, we use automatic differentiation. This allows for the accurate evaluation of derivatives at machine precision and ideal asymptotic efficiency [3, 49].

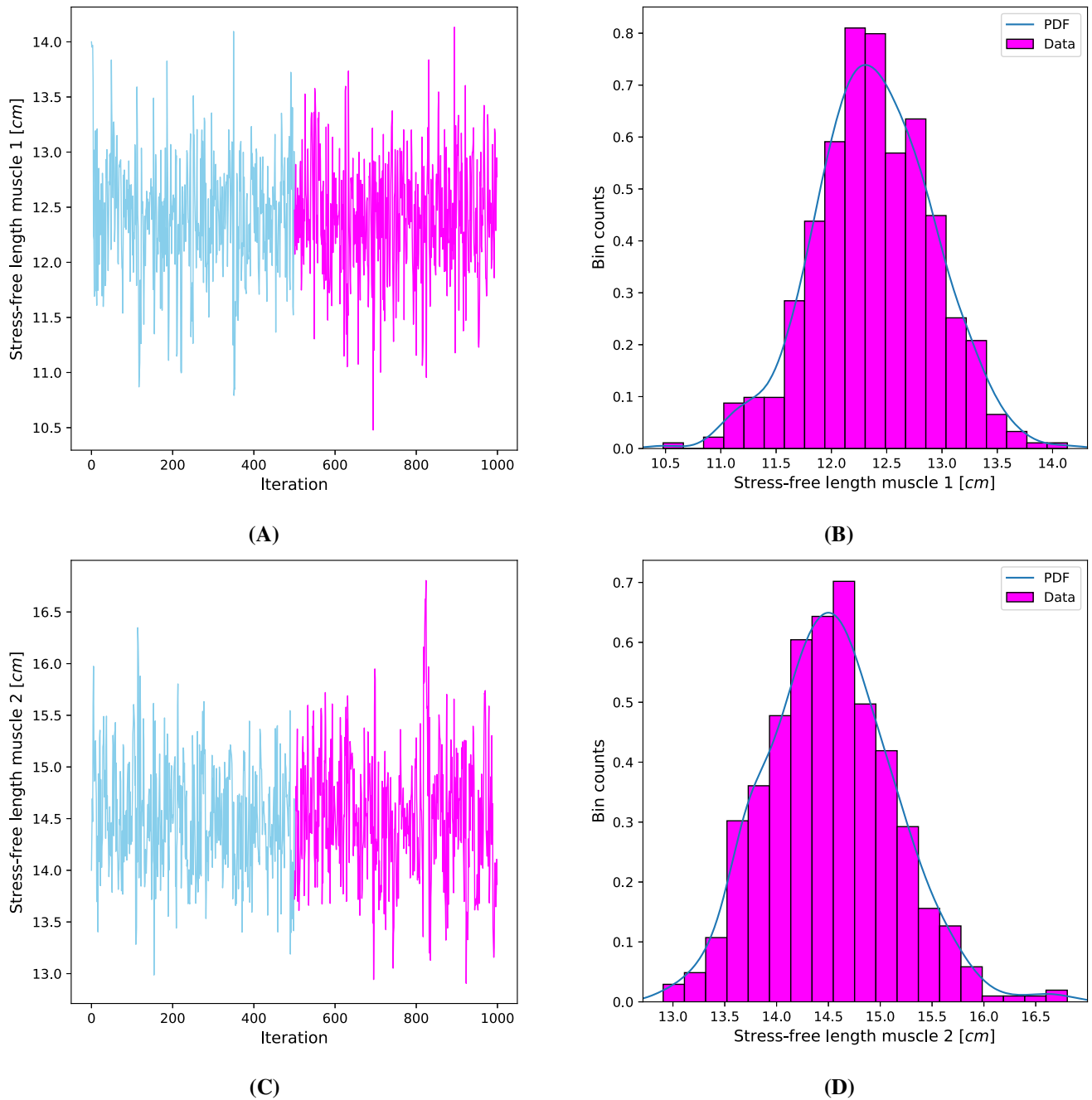
For the adaptive calculation of the number of Leap-frog steps we use the NUTS presented in Section 6. With respect to simulation runtime we set the cap of the maximum number of steps of the NUTS algorithm to 12. To reach a feasible step size  $\epsilon$  we use the adaptive dual averaging method in a pre-run of HMC for 1000 iterations, where the maximum number of steps of the NUTS is capped at 8 to save runtime in the pre-run, and  $\epsilon$  is adapted in each iteration. We observe that in this pre-run  $\epsilon$  converges to an order of magnitude between 0.01 and 0.02. Therefore we set  $\epsilon = 0.02$  for the final run. Additionally, we add noise to the fixed step size  $\epsilon$  for each iteration of the HMC algorithm to prevent the HMC algorithm from getting stuck in one fixed setting, such that  $\epsilon$  is finally uniformly distributed with  $\epsilon \sim \mathcal{U}(0.01, 0.03)$ . At last, we run the HMC algorithm for 1000 iterations with a burn-in of 500 iterations. The samples generated during the burn-in phase are dropped later for the calculation of the HMC results. As start sample for the stress-free muscle lengths we set for each muscle a length of 14 cm. Additional parameters set for the simulations are displayed in the Appendix A.2 in Tables A8 and A9.

## 7.2.2 | Simulation results

The HMC results are presented by showing both all sampling values over the sampling iterations in Figure 10A,C and histograms of all samples  $\Theta^n$  for  $n \in \{500, \dots, 1000\}$  in Figure 10B,D. The histograms show relative frequencies of samples in given intervals and, thus, an approximation of the respective posterior probability density for the stress-free muscle lengths. We drop the first 500 samples, which stem from the burn-in phase of the algorithm. Figure 11 shows the distribution of samples in the two-dimensional space spanned by the stress-free lengths of both muscles. In Figure 12, we compare the results for different start samples.

In Figure 13, we compare the so-called effective sample size (ESS) for different sampling strategies as a measure for potential autocorrelations within a chain. The ESS captures how many independent samples are required to obtain the same amount of information as generated by the MCMC algorithm [99]. The effective sample size is defined as

$$N_{\text{eff}} = \frac{N}{1 + 2 \sum_{t=1}^{\infty} r_t}, \quad (28)$$

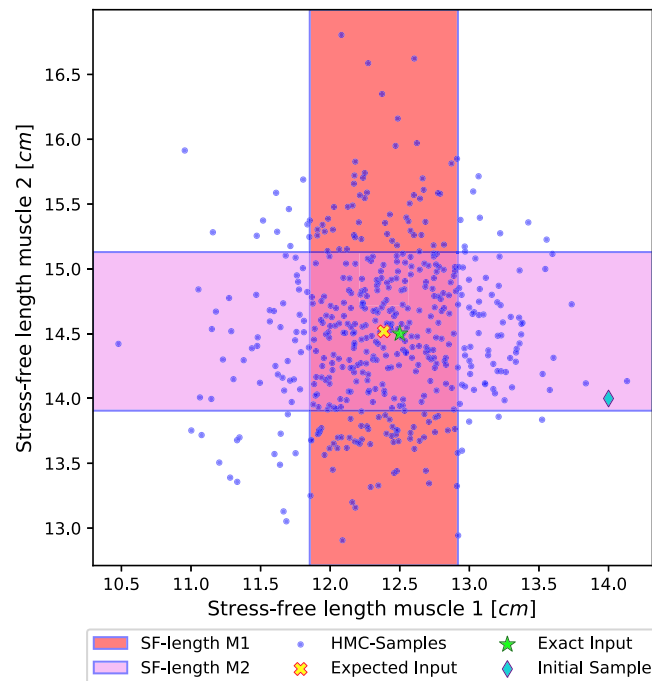


**FIGURE 10** Bayesian inversion experiments: results of the Hamiltonian Monte Carlo sampling method for the stress-free length of muscle 1 (A,B) and muscle 2 (C,D). These results are displayed as trace plots (blue: dropped samples due to the burn-in phase, magenta: used samples) (A,C) and histograms derived from all used samples (Data) with associated probability density functions (PDF) (B,D).

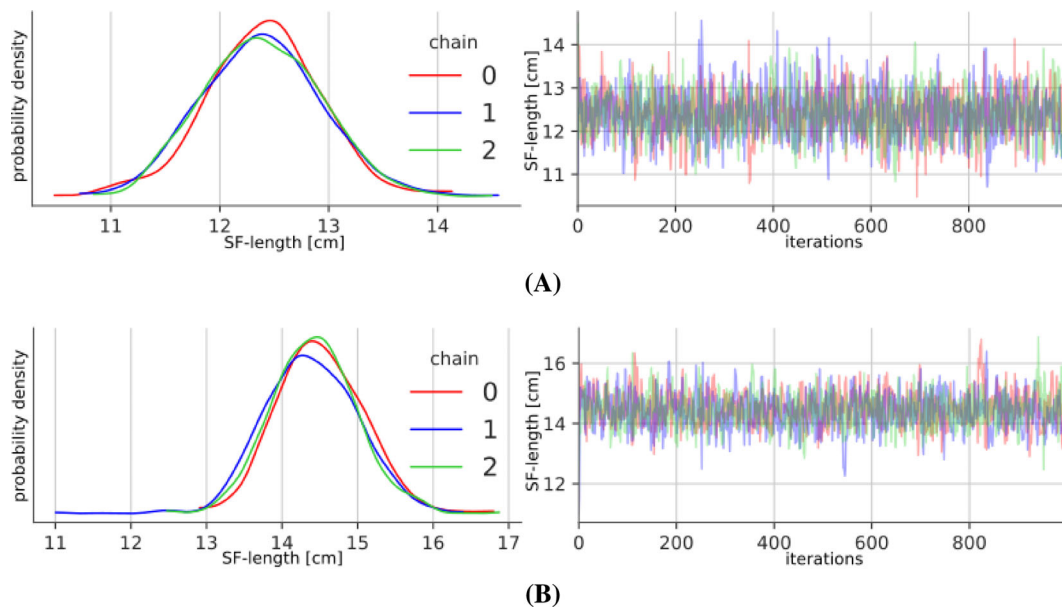
where  $r_t$  is the autocorrelation in the samples with lag  $t$  and  $N$  is the length of the Markov chain. The ESS is computed for each parameter separately and therefore we label a single chain for a single parameter of MCMC as  $\Theta_i$  with  $i \in \{1, 2\}$ , such that  $\Theta = [\Theta_1, \Theta_2]$  and  $\Theta^n = [\Theta_1^n, \Theta_2^n]$ . The autocorrelation can be empirically approximated with the natural estimator of the autocovariance function [31] given as

$$\widehat{\text{cov}}_t = \frac{1}{N} \sum_{n=1}^{N-t} [\Theta_i^n - \hat{\mu}_N] [\Theta_i^{n+t} - \hat{\mu}_N], \quad (29)$$

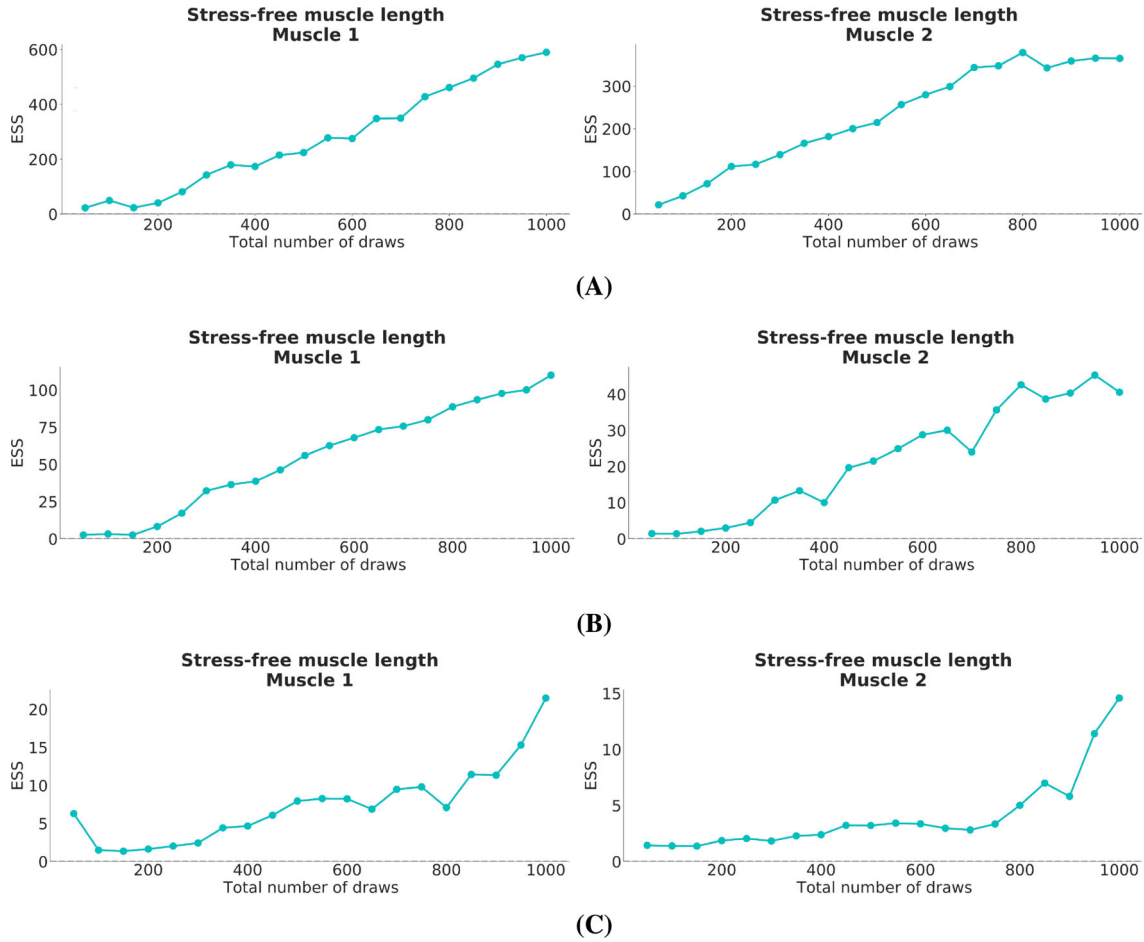




**FIGURE 11** Bayesian inversion experiments: scatter plot of the samples for the stress-free length (SF-length) of the muscles generated by the Hamiltonian Monte Carlo (HMC) sampling method, comparison of expected input value approximation based on the sampling results (yellow) to the exact input value (green) and to the initial start sample (turquoise) with the associated standard deviations (red and purple areas).



**FIGURE 12** Bayesian inversion experiments: comparison of the results from the Hamiltonian Monte Carlo sampling method for the stress-free length (SF-length) of the muscles for different start samples with [14.0, 14.0] (red), [11.0, 11.0] (blue), [14.5, 12.5] (green), leading to three different Markov chains. The results are displayed as probability density function (left) and trace plot (right) for muscle 1 (A) and muscle 2 (B).



**FIGURE 13** Bayesian inversion experiments: evolution of bulk effective sample size (ESS) for a run of 1000 iterations of the Hamiltonian Monte Carlo sampling method ca. 50% (A), the nonimproved Markov Chain Monte Carlo method in the best case ca. 10% (B) and in the worst case ca. 2% (C) of the total number of samples generated.

with the empirical expectation value

$$\hat{\mu}_N = \frac{1}{N} \sum_{n=0}^N \Theta_i^n, \quad (30)$$

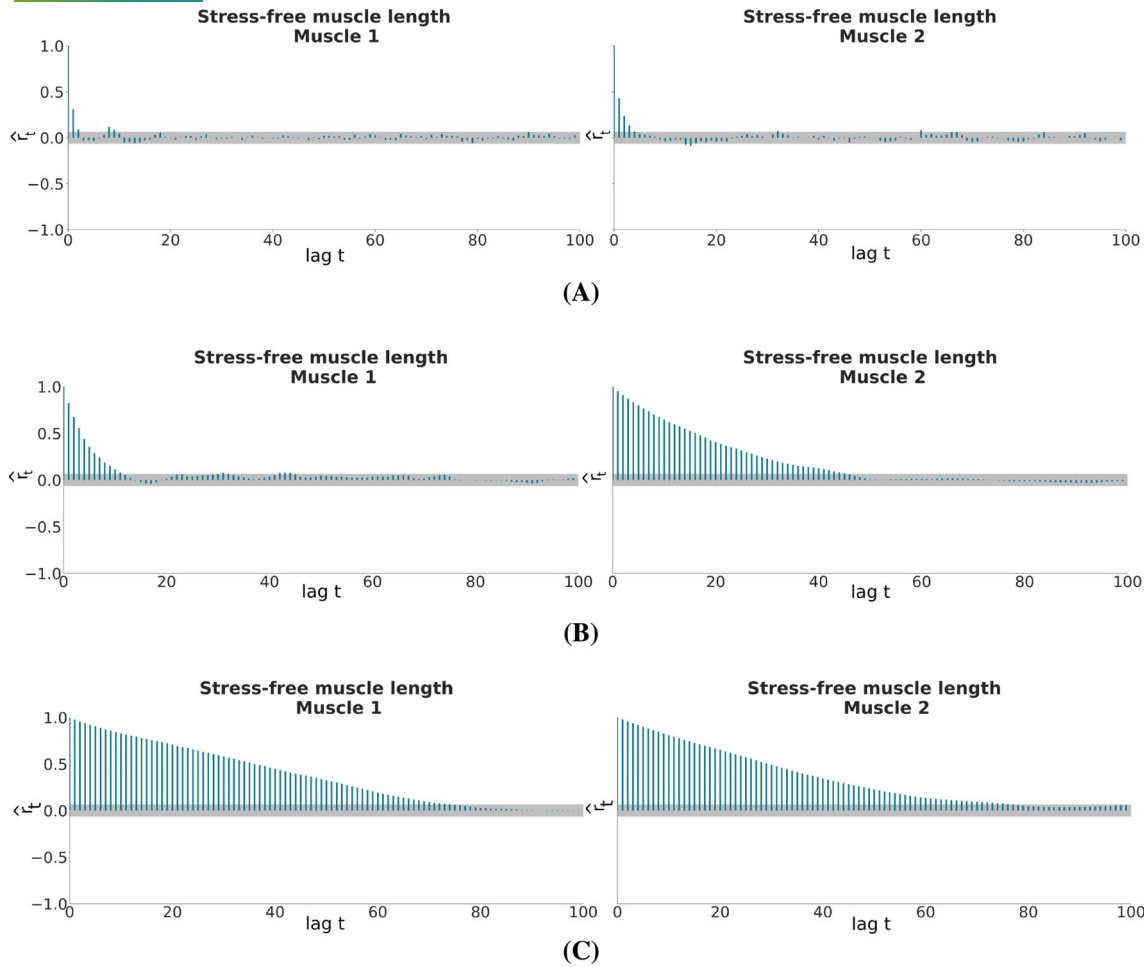
such that the autocorrelation with lag  $t$  is

$$\hat{r}_t = \frac{\widehat{\text{COV}}_t}{\widehat{\text{COV}}_0}. \quad (31)$$

However, in Figure 13 we display the ‘bulk-ESS’, a specialized ESS. The bulk-ESS checks especially how good the center of the distribution is resolved and focuses on a better resolution of point estimates, which we are interested in for the stress-free muscle lengths. It is directly derived from the ESS (28) by using a rank normalization, further details are described in [99]. In addition, Figure 14 shows the autocorrelation within the chains for different lags up to 100 iterations.

### 7.2.3 | Observations and discussion

In Figures 10A and 11, we observe that the generated samples for different true stress-free muscle lengths cluster around a mean value that depends on the true value although they have the same prior distributions as initial guess and the same starting sample.



**FIGURE 14** Bayesian inversion experiments: autocorrelation  $\hat{r}_t$  between the samples for different lags  $t$  of the Markov Chain for a run of 1000 iterations of the Hamiltonian Monte Carlo sampling method (A), nonimproved Markov Chain Monte Carlo method in the best case (B) and in the worst case (C) for different start samples.

The distributions approximated by the histograms in Figure 10B,D take a similar shape as a Gaussian distribution and have an expected value of 12.385 cm for muscle 1 and 14.518 cm for muscle 2 with standard deviations of 0.532 cm and 0.612 cm, respectively. Therefore the expected values resulting from the HMC algorithm are very close to the exact values of 12.5 cm for muscle 1 and 14.5 cm for muscle 2, shown in Figure 11.

When we compare the results for the three different start samples  $[14.0, 14.0]$ ,  $[11.0, 11.0]$ , and  $[14.5, 12.5]$  in multiple runs, the resulting distributions, shown in Figure 12, seem to converge to the same distribution for each muscle independent of the starting sample. To analyze the convergence more rigorously, we calculate the ‘estimated potential scale reduction’  $\hat{R}$  [99].  $\hat{R}$  measures the similarity of different MCMC chains by the comparison of the between-sequence variance  $B$  and the within-sequence variance  $W$  for  $M$  independently sampled sequences respectively chains of length  $N$ . Like the ESS,  $\hat{R}$  is calculated separately for each single parameter.  $\hat{R}$  for a single parameter is defined as

$$\hat{R} = \sqrt{\frac{\widehat{\text{var}}}{W}} \text{ with } \widehat{\text{var}} = \frac{N-1}{N} W + \frac{1}{N} B, \quad (32)$$

where the between-sequence variance is defined as

$$B = \frac{N}{M-1} \sum_{k=1}^M (\hat{\mu}_N^k - \hat{\mu}_{MN})^2, \text{ with } \hat{\mu}_N^k = \frac{1}{N} \sum_{n=1}^N \Theta_i^{n,k} \text{ and } \hat{\mu}_{MN} = \frac{1}{M} \sum_{k=1}^M \hat{\mu}_N^k \quad (33)$$

with  $\Theta_i^{n,k}$  denoting sample  $n$  of parameter  $i$  in sequence  $k$  and the within-sequence variance  $W$

$$W = \frac{1}{M} \sum_{k=1}^M (\hat{\sigma}_N^k)^2, \text{ with } (\hat{\sigma}_N^k)^2 = \frac{1}{N-1} \sum_{n=1}^N (\Theta_i^{n,k} - \hat{\mu}_N^k)^2. \quad (34)$$

$\hat{\text{var}}$  overestimates the variance for overdispersed starting points and for finite  $N$ , whereas  $W$  underestimates the variance of  $\Theta_i$ .  $W$  and  $\hat{\text{var}}$  converge to the true variance of the sequences from opposite directions, such that the estimated potential scale reduction  $\hat{R}$  approaches one. For the three different chains  $\hat{R}$  is 1.00 for muscle 1 and 1.01 for muscle 2 implying convergence of the chains for each dimension separately.

Autocorrelation within a chain is another issue to address, because it increases the uncertainty of the estimation of posterior quantities of interest. In Figure 13, we observe that for the MCMC method described in Section 6.2.1 the effective sample size in the best case of the runs for different start samples is close to 10 % of the total number of samples generated. Even in the worst case of these three runs, the effective sample size is just below 2 % compared to 50 % for the HMC method from Section 6.2.2. This is a direct consequence of the autocorrelation within the chains shown in Figure 14. The autocorrelation of the samples of the HMC method already drops under a negligible threshold at a lag of 5 samples compared to the nonimproved MCMC, whereas in the worst case the autocorrelation dropped under a negligible threshold only after around a lag of 75 samples.

In conclusion, we have shown, that the HMC method provides good results for predicting the distribution of the stress-free muscle length, if we know the range of motion of the muscles in a Hill-type muscle model. Additionally, we have seen that the improved HMC method generates samples with notably less autocorrelation than the standard MCMC method and therefore we obtain more information for the desired distribution with less samples. Finally, by comparing different samples we could confirm convergence of the chains relative to each other by calculating the estimated potential scale reduction. However, the estimated potential scale reduction and effective sample sizes are only initial indications of convergence of the MCMC method itself. This is due to the fact that they especially depend on the standard deviation of each chain but do not take into account the exact solution of the optimization problem [99].

## 8 | CONCLUSION AND OUTLOOK

We have presented first steps toward a comprehensive simulation, parameter inversion and optimization tool for coupled electrophysiological muscle-tendon-muscle systems as they are for example relevant in novel AMI surgery techniques. The AMI technique aims to improve the surgery outcome in terms of control and sensory pathways of the remaining parts of an amputated limb by re-establishing the physiological agonist-antagonist connections between muscles via a tendon.

We are approaching our goal from two sides: On the one hand, we develop a highly detailed simulation environment for full three-dimensional forward simulations including muscle activation via motor neurons and sensory feedback for a single muscle and between agonist and antagonist. On the other hand, we develop a Bayesian parameter estimation approach that delivers estimations of muscle prestretch, that is, the difference between the length of a muscle in a relaxed position in a physiological setting, that is, fixed by tendons, and the completely relaxed muscle length when detaching the muscle from all connections. This prestretch is known to be an important parameter to achieve an optimal range of motion and, thus, a valuable parameter in situations such as surgeries where this prestretch is artificially reconstructed.

Ultimately, we intend to replace the simplified Hill-type muscle models currently used in our Bayesian approach as a forward operator by more sophisticated surrogate models. Here, we envision (patient-)tailored data-based surrogates that are generated based on our full three-dimensional muscle models including all activation and sensory feedback components.

Currently, we simulate highly simplified muscle geometries with our full three-dimensional model presented in Section 3.2 and do not yet include the sensory feedback. However, we have shown that our partitioned simulation approach can yield a stable solver and delivers qualitatively plausible results for the two-muscle-on-tendon system if numerical thresholds are chosen carefully. A cadaver study, which is currently underway, will provide us with the necessary geometric parameters to generate more realistic muscle geometries. Experiments, which assess the reflex activity in AMI patients, which are also currently underway, will help us to complete and validate the sensory feedback model. Since sensory feedback is not only provided by muscle spindles but also other sensory organs, for example, Golgi-tendon organs, they can be included in the model [52]. With these enhancements, we expect to be able to realistically simulate three-dimensional deformations in more realistic muscle geometries. This will be important since changes to the geometry and position of the residual muscles during AMI surgery are expected to affect the contractile behavior of the muscles as well as the sensory feedback [44], since sensory organs, in particular muscle spindles, are distributed across the muscle and deliver location dependent feedback to the neural system [4, 86].

In order to increase the numerical efficiency of the coupled simulation, we will further optimize the quasi-Newton coupling methods and their numerical parameter configuration alongside with extending the coupling from the current

mechanical coupling between each muscle and the connecting tendon to the coupling between the two muscles via the sensory feedback.

In terms of the Bayesian approach, we have shown the potential of the method with Hamiltonian Monte-Carlo sampling for stochastic parameter inversion and simplified Hill-type muscle models. Ultimately, Bayesian inference will be replaced by Bayesian optimization to find the prestretch that maximizes the range of motion instead of only identifying the correct prestretch for a given range of motion. Bayesian inference might still be used to find the optimal parameters for the sensory feedback loops. Data for this inference will become available from the clinical reflex studies that are currently being conducted.

By including a new surrogate generated from our full muscle model we intend to improve the HMC algorithm through more elaborate initial conditions. We envision to reach faster execution times for each forward problem computation leading to increasing especially the capped number of Leap-frog steps in the NUTS for more accurate and informative results due to an improvement of each single sample in HMC. The prestretch optimization should eventually also take the sensory feedback into consideration as a second optimization criterion in addition to the range of motion of the AMI.

With respect to the AMI application, we envision to provide guidance on the optimal prestretch for different conditions determined by, for example, residual muscle length or tissue state. Especially the revision procedures, that is, creating an AMI after a previous conventional amputation, still pose major challenges [44]. Providing guidance to the treatment of revision cases would make the procedure accessible to many more patients.

In conventional amputations, little attention was given to tendons even knowing that many proprioceptive sensory organs, for example, Golgi-tendon organs, are located at the musculo-tendinous junctions [52]. The bradytrophy nature of the tendons and the risk of infection might be one explanation [69]. Using the extended model, we can explore the potential role of Golgi-tendon organs for AMI patients. With regard to the control of myocontrol prostheses [45], predictions of the EMG signals of the residual muscles can provide valuable insights. Also, this is only possible with a three-dimensional muscle model [57]. The full three-dimensional muscle model can potentially also be used as an education tool for surgeons, orthopedic technicians or physiotherapists. Further, it can be applied to other applications, for example, to improve the outcome of tendon transfers in hand surgery [12, 29].

## ACKNOWLEDGEMENTS


We thank the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) for supporting this work through EXC2075 in Germany's Excellence Strategy and in SPP 2311 (390740016 and 465243391). We acknowledge the support by the Stuttgart Centre for Simulation Science (SimTech). Finally, we also thank the preCICE team for their help in the configuration of the coupling with preCICE. Open access funding enabled and organized by Projekt DEAL. This study does not have any conflicts to disclose.

## DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in DARUS at <http://doi.org/10.18419/darus-4031>.<sup>106</sup>

## ORCID

Robin Lautenschlager  <https://orcid.org/0009-0004-9897-6614>

Laura Schmid  <https://orcid.org/0000-0002-1781-8429>

Oliver Röhrle  <https://orcid.org/0000-0002-1934-6525>

## REFERENCES

- [1] B. J. Alder and T. E. Wainwright, Studies in molecular dynamics. I. General method, *J. Comput. Phys.* **31** (1959), no. 2, 459–466.
- [2] F. C. Anderson and M. G. Pandy, Static and dynamic optimization solutions for gait are practically equivalent, *J. Biomech.* **34** (2001), no. 2, 153–161.
- [3] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, Automatic differentiation in machine learning: A survey, *J. Mach. Learn. Res.* **18** (2018), no. 153, 1–43.
- [4] M. Bergenheim, E. Ribot-Ciscar, and J.-P. Roll, Proprioceptive population coding of two-dimensional limb movements in humans: I. Muscle spindle feedback during spatially oriented movements, *Exp. Brain Res.* **134** (2000), no. 3, 301–310.
- [5] M. Betancourt, The convergence of Markov chain Monte Carlo methods: From the metropolis method to Hamiltonian Monte Carlo, *Ann. Phys.* **531** (2017), 1700214.
- [6] M. Betancourt. A conceptual introduction to Hamiltonian Monte Carlo. 2018.
- [7] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*, Vol 4, Springer, New York, NY, 2006.
- [8] M. Blümel, C. Guschlbauer, S. Daun-Gruhn, S. L. Hooper, and A. Büschges, Hill-type muscle model parameters determined from experiments on single muscles show large animal-to-animal variation, *Biol. Cybern.* **106** (2012), 559–571.



- [9] M. F. Bobbert, P. A. Huijing, and G. J. van Ingen Schenau, A model of the human triceps surae muscle-tendon complex applied to jumping, *J. Biomech.* **19** (1986), no. 11, 887–898.
- [10] M. Böl and S. Reese, Micromechanical modelling of skeletal muscles based on the finite element method, *Comput. Methods Biomech. Biomed. Engin.* **11** (2008), no. 5, 489–504.
- [11] C. P. Bradley, N. Emamy, T. Ertl, D. Göddeke, A. Hessenthaler, T. Klotz, A. Krämer, M. Krone, B. Maier, M. Mehl, T. Rau, and O. Röhrle, Enabling detailed, biophysics-based skeletal muscle models on HPC systems, *Front. Physiol.* **9** (2018), 816.
- [12] P. W. Brand, *Clinical mechanics of the hand, hand rehabilitation in occupational therapy*, Routledge (2012), 183–184.
- [13] S. Brooks, A. Gelman, G. Jones, and X. L. Meng, *Handbook of Markov chain Monte Carlo*, Chapman and Hall/CRC, New York, 2011.
- [14] A. H. Caillet, A. T. M. Phillips, C. Carty, D. Farina, and L. Modenese, Hill-type computational models of muscle-tendon actuators: a systematic review, *bioRxiv* (2022). <https://doi.org/10.1101/2022.10.14.512218>.
- [15] T. A. Carniel and E. A. Francello, A transversely isotropic coupled hyperelastic model for the mechanical behavior of tendons, *J. Biomech.* **54** (2017), 49–57.
- [16] M. J. Carty, D149. The Ewing amputation: Lessons learned from the first 31 cases, *Plast. Reconstr. Surg. Glob. Open* **11** (2023), no. 4 Suppl, 131.
- [17] S. Chib and E. Greenberg, Understanding the metropolis-Hastings algorithm, *Am. Stat.* **49** (1995), no. 4, 327–335.
- [18] G. Chourdakis, K. Davis, B. Rodenberg, M. Schulte, F. Simonis, B. Uekermann, G. Abrams, H. J. Bungartz, L. Cheung Yau, I. Desai, K. Eder, R. Hertrich, F. Lindner, A. Rusch, D. Sashko, D. Schneider, A. Totounferoush, D. Volland, P. Vollmer, and O. Z. Koseomur, preCICE v2: A sustainable and user-friendly coupling library [version 2; peer review: 2 approved], *Open Res. Eur.* **2** (2022), no. 51, 51.
- [19] R. R. Cisi and A. F. Kohn, Simulation system of spinal cord motor nuclei and associated nerves and muscles, in a web-based architecture, *J. Comput. Neurosci.* **25** (2008), no. 3, 520–542.
- [20] C. Crone, H. Hultborn, B. Jespersen, and J. Nielsen, Reciprocal IA inhibition between ankle flexors and extensors in man, *J. Physiol.* **389** (1987), no. 1, 163–185.
- [21] A. A. Cuellar, C. M. Lloyd, P. F. Nielsen, D. P. Bullivant, D. P. Nickerson, and P. J. Hunter, An overview of CellML 1.1, a biological model description language, *J. Comput. Sci.* **79** (2003), no. 12, 740–747.
- [22] S. Delp and S. Blemker, Three-dimensional representation of complex muscle architectures and geometries, *Ann. Biomed. Eng.* **33** (2005), 661–673.
- [23] T. J. M. Dick, A. A. Biewener, and J. M. Wakeling, Comparison of human gastrocnemius forces predicted by Hill-type muscle models and estimated from ultrasound images, *J. Exp. Biol.* **220** (2017), no. 9, 1643–1653.
- [24] J. Dideriksen, F. Negro, and D. Farina, The optimal neural strategy for a stable motor task requires a compromise between level of muscle cocontraction and synaptic gain of afferent feedback, *J. Neurophysiol.* **114** (2015), no. 3, 1895–1911.
- [25] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth, Hybrid Monte Carlo, *Phys. Lett. B* **195** (1987), no. 2, 216–222.
- [26] J. C. Eccles, Conduction and synaptic transmission in the nervous system, *Annu. Rev. Physiol.* **10** (1948), no. 1, 93–116.
- [27] L. A. Elias, R. N. Watanabe, and A. F. Kohn, Spinal mechanisms may provide a combination of intermittent and continuous control of human posture: predictions from a biologically based neuromusculoskeletal model, *PLoS Comput. Biol.* **10** (2014), no. 11, e1003944.
- [28] D. Frenkel, B. Smit, and M. A. Ratner, *Understanding molecular simulation: From algorithms to applications*, Vol 2, Academic Press San Diego, San Diego, 1996.
- [29] J. Fridén and R. L. Lieber, Tendon transfer surgery: clinical implications of experimental studies, *Clin. Orthop. Relat. Res.* **403** (2002), S163–S170.
- [30] J. H. Geertzen, S. M. van der Schans, P. C. Jutte, J. Kraeima, E. Otten, and R. Dekker, Myodesis or myoplasty in trans-femoral amputations. What is the best option? An explorative study, *Med. Hypotheses* **124** (2019), 7–12.
- [31] C. J. Geyer, *Introduction to Markov chain Monte Carlo*.
- [32] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, Eds., *Markov chain Monte Carlo in practice*, Chapman & Hall, London, 1996.
- [33] A. Gordon, A. F. Huxley, and F. Julian, The variation in isometric tension with sarcomere length in vertebrate muscle fibres, *J. Physiol.* **184** (1966), no. 1, 170–192.
- [34] E. J. Green, A. O. Finley, and W. E. Strawderman, *Introduction to Bayesian Methods in Ecology and Natural Resources*, Cham, Springer, Nature Switzerland AG, Cham, 2020.
- [35] B. Greitemann, L. Brückner, M. Schäfer, and R. Baumgartner, Amputation und prothesenversorgung, *Z. Orthop. Unfall.* **154** (2016), 329.
- [36] M. Günther, O. Röhrle, D. F. B. Haeufle, and S. Schmitt, Spreading out muscle mass within a hill-type model: A computer simulation study, *Comput. Math. Methods Med.* **2012** (2012), 1–13.
- [37] D. F. B. Haeufle, M. Günther, A. Bayer, and S. Schmitt, Hill-type muscle model with serial damping and eccentric force-velocity relation, *J. Biomech.* **47** (2014), 1531–1536.
- [38] D. Hawkins and M. Bey, A comprehensive approach for studying muscle-tendon mechanics, *J. Biomech. Eng.* **116** (1994), no. 1, 51–55.
- [39] T. Heidlauf, T. Klotz, C. Rode, E. Altan, C. Bleiler, T. Siebert, and O. Röhrle, A multi-scale continuum model of skeletal muscle mechanics predicting force enhancement based on actin–titin interaction, *Biomech. Model. Mechanobiol.* **15** (2016), 1423–1437.
- [40] T. Heidlauf and O. Röhrle, Modeling the chemoelectromechanical behavior of skeletal muscle using the parallel open-source software library opencmis, *Comput. Math. Methods Med.* **2013** (2013), 517287.
- [41] T. Heidlauf and O. Röhrle, A multiscale chemo-electro-mechanical skeletal muscle model to analyze muscle contraction and force generation for different muscle fiber arrangements, *Front. Physiol.* **5** (2014), 498.
- [42] L. Held and D. S. Bové, *Likelihood and Bayesian inference*, Statistics for Biology and Health. Springer, Berlin, Heidelberg, 2020.
- [43] B. Hernández-Gascón, J. Grasa, B. Calvo, and J. F. Rodríguez, A 3d electro-mechanical continuum model for simulating skeletal muscle contraction, *J. Theor. Biol.* **335** (2013), 108–118.

- [44] H. Herr and M. Carty, The agonist-antagonist myoneural interface, *Tech. Orthop.* **36** (2021), 337–344.
- [45] H. M. Herr, T. R. Clites, S. Srinivasan, S. G. Talbot, G. A. Dumanian, P. S. Cederna, and M. J. Carty, Reinventing extremity amputation in the era of functional limb restoration, *Ann. Surg.* **273** (2021), no. 2, 269–279.
- [46] A. V. Hill, The heat of shortening and the dynamic constants of muscle, *Proc. Royal Soc. Lon. B Biol. Sci.* **126** (1938), no. 843, 136–195.
- [47] A. Hodgkin and A. Huxley, A quantitative description of membrane current and its application to conduction and excitation in nerve, *J. Physiol.* **117** (1952), no. 4, 500–544.
- [48] M. D. Hoffman and A. Gelman, The no-u-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo, *J. Mach. Learn. Res.* **15** (2014), no. 1, 1593–1623.
- [49] P. H. W. Hoffmann, A hitchhiker's guide to automatic differentiation, *Numer. Algorithms* **72** (2015), no. 3, 775–811.
- [50] H. Ishwaran, Applications of hybrid Monte Carlo to Bayesian generalized linear models: Quasicomplete separation and neural networks, *J. Comput. Graph. Stat.* **8** (1999), no. 4, 779–799.
- [51] R. S. Iyengar, M. V. Pithapuram, A. K. Singh, and M. Raghavan, Curated model development using neuroid: A web-based neuromotor integration and design platform, *Front. Neuroinform.* **13** (2019), 56.
- [52] L. Jami, Golgi tendon organs in mammalian skeletal muscle: functional properties and central actions, *Physiol. Rev.* **72** (1992), no. 3, 623–666.
- [53] S.-S. Jin, H. Ju, and H.-J. Jung, Adaptive Markov chain Monte Carlo algorithms for Bayesian inference: recent advances and comparative study, *Struct. Infrastruct. Eng.* **15** (2019), no. 11, 1548–1565.
- [54] T. Johansson, P. Meier, and R. Blickhan, A finite-element model for the mechanical analysis of skeletal muscles, *J. Theor. Biol.* **206** (2000), no. 1, 131–149.
- [55] E. R. Kandel, J. H. Schwartz, T. M. Jessell, S. Siegelbaum, A. J. Hudspeth, and S. Mack, *Principles of neural science*, 5th ed., McGraw-Hill, New York, New York, 2013.
- [56] M. Kapardi, M. V. Pithapuram, Y. M. Rangayyan, R. S. Iyengar, A. K. Singh, S. Sripada, and M. Raghavan, In-silico neuro musculoskeletal model reproduces the movement types obtained by spinal micro stimulation, *Comput. Methods Prog. Biomed.* **220** (2022), 106804.
- [57] T. Klotz, L. Gizzi, U. Ş. Yavuz, and O. Röhrle, Modelling the electrical activity of skeletal muscle tissue using a multi-domain approach, *Biomech. Model. Mechanobiol.* **19** (2020), no. 1, 335–349.
- [58] L. Kudina, Reflex effects of muscle afferents on antagonist studied on single firing motor units in man, *Electroencephalogr. Clin. Neurophysiol.* **50** (1980), no. 3-4, 214–221.
- [59] F. Lindner, M. Mehl, S. Klaudius, and U. Benjamin, *A comparison of various quasi-newton schemes for partitioned fluid-structure interaction*, CIMNE, Barcelona, 2015, 477–488.
- [60] J. S. Liu and J. S. Liu, *Monte Carlo strategies in scientific computing*, Vol **75**, Springer, New York, NY, 2001.
- [61] V. G. Macefield and T. P. Knellwolf, Functional properties of human muscle spindles, *J. Neurophysiol.* **120** (2018), no. 2, 452–467.
- [62] B. R. MacIntosh, P. F. Gardiner, and A. J. McComas, Skeletal muscle: Form and function, *Hum. Kinet.* (2006).
- [63] D. J. MacKay, *Information theory, inference and learning algorithms*, Cambridge university press, Cambridge, 2003.
- [64] B. Maier, *Scalable biophysical simulations of the neuromuscular system*, Ph.D. Thesis, University of Stuttgart, Stuttgart, 2021.
- [65] B. Maier and M. Schulte, Mesh generation and multi-scale simulation of a contracting muscle–tendon complex, *J. Comput. Sci.* **59** (2022), 101559.
- [66] J. A. C. Martins, M. P. M. Pato, and E. B. Pires, A finite element model of skeletal muscles, *Virtual Phys. Prototyp.* **1** (2006), no. 3, 159–170.
- [67] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, Equation of state calculations by fast computing machines, *J. Comput. Phys.* **21** (1953), no. 6, 1087–1092.
- [68] M. P. Mileusnic, I. E. Brown, N. Lan, and G. E. Loeb, Mathematical models of proprioceptors. I. Control and transduction in the muscle spindle, *J. Neurophysiol.* **96** (2006), no. 4, 1772–1788.
- [69] S. A. Müller, A. Todorov, P. E. Heisterbach, I. Martin, and M. Majewski, Tendon healing: an overview of physiology, biology, and pathology of tendon healing and systematic review of state of the art in tendon bioengineering, *Knee Surg. Sports Traumatol. Arthrosc.* **23** (2015), 2097–2105.
- [70] R. M. Neal, *Probabilistic inference using Markov chain Monte Carlo methods*, Department of Computer Science, University of Toronto Toronto, ON, Canada, 1993.
- [71] R. M. Neal, *Bayesian learning for neural networks*, Lecture Notes in Statistics, Vol **118**, Springer-Verlag, New York, NY, 1996.
- [72] R. M. Neal, *MCMC using Hamiltonian dynamics*, CRC Press, New York, NY, 2011. <https://doi.org/10.1201/b10905-7>.
- [73] F. Negro and D. Farina, Decorrelation of cortical inputs and motoneuron output, *J. Neurophysiol.* **106** (2011), no. 5, 2688–2697.
- [74] M. N. Öüztörel and R. B. Stein, Analysis of a model for antagonistic muscles, *Biol. Cybern.* **45** (1982), 177–186.
- [75] S. A. Ovidia and M. Askari, *Upper extremity amputations and prosthetics, Seminars in plastic surgery*, Vol **29**, Thieme Medical Publishers, New York, NY, 2015, 55–61.
- [76] M. G. Pandy, F. E. Zajac, E. Sim, and W. S. Levine, An optimal control model for maximum-height human jumping, *J. Biomech.* **23** (1990), no. 12, 1185–1198.
- [77] J. Rasmussen, M. Damsgaard, and M. Voigt, Muscle recruitment by the min/max criterion—A comparative numerical study, *J. Biomech.* **34** (2001), no. 3, 409–415.
- [78] M. V. Razumova, A. E. Bukatina, and K. B. Campbell, Stiffness-distortion sarcomere model for muscle simulation, *J. Appl. Physiol.* **87** (1999), no. 5, 1861–1876.
- [79] E. Ribot-Ciscar and J.-P. Roll, Ago-antagonist muscle spindle inputs contribute together to joint movement coding in man, *Brain Res.* **791** (1998), no. 1-2, 167–176.

- [80] E. Rodríguez and G. Gesnouin, Effective mass of an oscillating spring, *Phys. Teach.* **45** (2007), 100.
- [81] O. Röhrle, J. Davidson, and A. Pullan, A physiologically based, multi-scale model of skeletal muscle structure and function, *Front. Physiol.* **3** (2012), 358.
- [82] O. Röhrle, M. Sprenger, and S. Schmitt, A two-muscle, continuum-mechanical forward simulation of the upper limb, *Biomech. Model. Mechanobiol.* **16** (2017), no. 3, 743–762.
- [83] O. Röhrle, U. Ş. Yavuz, and T. Klotz, Multiscale modeling of the neuromuscular system: coupling neurophysiology and skeletal muscle mechanics, *Wiley Interdiscip. Rev. Syst. Biol. Med.* **11** (2019), no. 6, e1457.
- [84] F. Romero and F. J. Alonso, A comparison among different hill-type contraction dynamics formulations for muscle force estimation, *Mech. Sci.* **7** (2016), no. 1, 19–29.
- [85] J. Rosen, M. B. Fuchs, and M. Arcan, Performances of hill-type and neural network muscle models—Toward a myosignal-based exoskeleton, *Comput. Biomed. Res.* **32** (1999), no. 5, 415–439.
- [86] A. Rowlerson, F. Mascarello, D. Barker, and H. Saed, Muscle-spindle distribution in relation to the fibre-type composition of masseter in mammals, *J. Anat.* **161** (1988), 37–60.
- [87] M. Sartori, J. W. Fernandez, L. Modenese, C. P. Carty, L. A. Barber, K. Oberhofer, J. Zhang, G. G. Handsfield, N. S. Stott, T. F. Besier, D. Farina, and D. G. Lloyd, Toward modeling locomotion using electromyography-informed 3d models: Application to cerebral palsy, *Wiley Interdiscip. Rev. Syst. Biol. Med.* **9** (2017), no. 2, e1368.
- [88] M. N. Schmidt, “Function factorization using warped Gaussian processes,” *Proceedings of the 26th Annual International Conference on Machine Learning, ICML’09*, Association for Computing Machinery, New York, NY, USA, 2009, pp. 921–928. <https://doi.org/10.1145/1553374.1553492>.
- [89] J. G. Scott and L. M. Mendell, Individual EPSPs produced by single triceps surae ia afferent fibers in homonymous and heteronymous motoneurons, *J. Neurophysiol.* **39** (1976), no. 4, 679–692.
- [90] T. Shu, S. S. Huang, C. Shallal, and H. M. Herr, Restoration of bilateral motor coordination from preserved agonist-antagonist coupling in amputation musculature, *J. Neuroeng. Rehabil.* **18** (2021), no. 1, 1–17.
- [91] H. Song, E. A. Israel, S. Gutierrez-Arango, A. C. Teng, S. S. Srinivasan, L. E. Freed, and H. M. Herr, Agonist-antagonist muscle strain in the residual limb preserves motor control and perception after amputation, *Commun. Med.* **2** (2022), no. 1, 97.
- [92] J. S. Speagle, A conceptual introduction to Markov chain Monte Carlo methods. 2019.
- [93] M. Sreenivasa, K. Ayusawa, and Y. Nakamura, Modeling and identification of a realistic spiking neural network and musculoskeletal model of the human arm, and an application to the stretch reflex, *IEEE Trans. Neural Syst. Rehabil. Eng.* **24** (2015), no. 5, 591–602.
- [94] S. S. Srinivasan, S. Gutierrez-Arango, A. C. E. Teng, E. Israel, H. Song, Z. K. Bailey, M. J. Carty, L. E. Freed, and H. M. Herr, Neural interfacing architecture enables enhanced motor control and residual limb functionality postamputation, *Proc. Natl. Acad. Sci.* **118** (2021), no. 9, e201955118.
- [95] S. S. Srinivasan, G. Tuckute, J. Zou, S. Gutierrez-Arango, H. Song, R. L. Barry, and H. M. Herr, Agonist-antagonist myoneural interface amputation preserves proprioceptive sensorimotor neurophysiology in lower limbs, *Sci. Transl. Med.* **12** (2020), no. 573, eabc5926.
- [96] W. W. Teka, K. C. Hamade, W. H. Barnett, T. Kim, S. N. Markin, I. A. Rybak, and Y. I. Molkov, From the motor cortex to the movement and back again, *PLoS One* **12** (2017), no. 6, e0179288.
- [97] J.-i. Ueda and Y. Sadamoto, A measurement of the effective mass of coil springs, *J. Phys. Soc. Jpn.* **66** (1997), no. 2, 367–368.
- [98] A. van Soest and M. Bobbert, The contribution of muscle properties in the control of explosive movements, *Biol. Cybern.* **69** (1993), no. 3, 195–204.
- [99] A. Vehtari, A. Gelman, D. Simpson, B. Carpenter, and P. C. Bürkner, Rank-normalization, folding, and localization: An improved  $\hat{R}$  assessing convergence of MCMC (with discussion), *Bayesian Anal.* **16** (2021), no. 2, 667–718.
- [100] V. L. Volk, L. D. Hamilton, D. R. Hume, K. B. Shelburne, and C. K. Fitzpatrick, Integration of neural architecture within a finite element framework for improved neuromusculoskeletal modeling, *Sci. Rep.* **11** (2021), no. 1, 22983.
- [101] U. von Toussaint, Bayesian inference in physics, *Rev. Mod. Phys.* **83** (2011), 943–999.
- [102] N. Walter, V. Alt, and M. Rupp, Lower limb amputation rates in Germany, *Medicina* **58** (2022), 1–101.
- [103] Z. Wang, J. Bardsley, A. Solonen, T. Cui, and Y. Marzouk, Bayesian inverse problems with priors: A randomize-then-optimize approach, *SIAM J. Sci. Comput.* **39** (2016), S140–S166.
- [104] D. Watt, E. K. Stauffer, A. Taylor, R. M. Reinking, and D. G. Stuart, Analysis of muscle receptor connections by spike-triggered averaging. 1. Spindle primary and tendon organ afferents, *J. Neurophysiol.* **39** (1976), no. 6, 1375–1392.
- [105] C. Zhang, *Scalable Hamiltonian Monte Carlo via surrogate methods*, Ph.D. Thesis, University of California, Irvine, 2016.
- [106] C. Homs Pons, and R. Lautenschlager, Replication Data for: Coupled Simulations and Parameter Inversion for Neural System and Electrophysiological Muscle Models [Data set]. DaRUS. (2024). <https://doi.org/10.18419/DARUS-4031>

**How to cite this article:** C. Homs-Pons, R. Lautenschlager, L. Schmid, J. Ernst, D. Göddeke, O. Röhrle, and M. Schulte, *Coupled simulations and parameter inversion for neural system and electrophysiological muscle models*, *GAMM-Mitteilungen*. **47** (2024), e202370009. <https://doi.org/10.1002/gamm.202370009>

## APPENDIX

In this Appendix, we present all parameters used to configure the simulation and inversion runs in Section 7. Note that not all parameters have been introduced in this paper, but are required as input for the respective simulation and Bayesian inversion software. We use the respective naming and notation of these tools.

### A.1 Parameters for the full muscle model simulations in Section 7.1

The material parameters for the three-dimensional electrophysiological muscle and tendon model in Tables A1–A3 have been used as input/configuration values for all simulation results presented in Section 7.1. We used OpenDiHu (commit 6ee259e) for the muscle and tendon model. We refer to the OpenDiHu documentation<sup>5</sup> for the exact definition of the material parameters.

**TABLE A1** Material parameters for muscle fibers (Equation 7).

$A_m$	$C_m$	$\sigma_{\text{eff}}$
500.0 cm <sup>-1</sup>	0.58 $\mu\text{F}/\text{cm}^2$	3.828 mA/cm

**TABLE A2** Material parameters for nonlinear tendon.

c	ca	ct	cat	ctt	mu	k1	k2
9.98 N/cm <sup>2</sup>	14.02	14.7	9.64	11.24	3.76 N/cm <sup>2</sup>	42.217e3 N/cm <sup>2</sup>	411.360e3 N/cm <sup>2</sup>

**TABLE A3** Material parameters for Mooney–Rivlin nonlinear muscle model.

c1	c2	b	d
3.176e-10 N/cm <sup>2</sup>	1.813 N/cm <sup>2</sup>	1.075e-2 N/cm <sup>2</sup>	9.1733

The parameters for mechanical coupling between each of the muscles and the tendon in the two-muscle-one-tendon scenario in Section 7.1.3 have been used for all simulations there in combination with OpenDiHu (commit 6ee259e) and preCICE v2.5.

### A.2 Parameters for Bayesian inference for the simplified Hill-type muscle model simulations in Section 7.2

The material parameters for the simplified Hill-type muscle model in Tables A6 and A7 [66, 84] and the numerical parameters in Table A8 have been used as input/configuration values for all simulation results presented in Section 7.2. The numerical parameters in Table A9 have been used for multiple runs of the nonimproved Markov Chain Monte Carlo method described in Section 6.2.1 for a comparison to the results of the Hamiltonian Monte Carlo method presented in Section 7.2.

**TABLE A4** Two-muscle-one-tendon simulations: Numerical parameters used in the mechanical tendon-muscle coupling via preCICE.

Time-window-size	Max-iterations	Displacement rel-convergence	Velocity rel-convergence	Traction abs-convergence
0.01	25	1e-3	1e-2	1e-1

**TABLE A5** Two-muscle-one-tendon simulations: Numerical parameters of the quasi-Newton scheme accelerating implicit mechanical coupling between the muscles and the tendon in each time step.

Preconditioner	Filter type	Filter limit	Initial-relaxation	Max-used-iterations	Time-windows-reused
residual-sum	QR2	1e-1	0.1	40	15

<sup>5</sup><https://opendihu.readthedocs.io/en/latest/>.

**TABLE A6** Material parameters for the passive muscle element of the simplified Hill-type muscle model.

$k_{PE}$	$\epsilon_0^M$	$\tau_c$
3.0	0.5	0.1 s

**TABLE A7** Material parameters for the active muscle element of the simplified Hill-type muscle model.

$\gamma$	$k_{CE1}$	$k_{CE2}$	$f_v^{\max}$	$\tau_{rise}$	$\tau_{fall}$	$\alpha_{min}$
0.45	0.25	0.06	1.6	$20.0 * 1e - 3$	$200.0 * 1e - 3$	

**TABLE A8** Two-muscle-one-tendon simulation: Numerical parameters for the solution of the forward problem for the Bayesian inversion.

Simulation parameters					
$T_{start}$	$T_{end}$	Activation muscle 1	Activation muscle 2	ODE-solver	Stepsize-controller
0.0	6.0	0.0-1.0	2.0-3.0	Tsit5	PIDController(rtol= $1e - 5$ , atol= $1e - 5$ )

**TABLE A9** Two-muscle-one-tendon simulation: Numerical parameters for the simulation of the nonimproved Markov Chain Monte Carlo method.

MCMC method parameters			
$\xi_j$ for $j \in \{1, 2, 3, 4\}$	$v_i$ for $i \in \{1, 2\}$	Total iterations	Burn-in iterations
$\mathcal{N}(0.0, 1.0)$	$\mathcal{N}(0.0, 0.1)$	1000	500
Start sample 1 (cm)	Start sample 2 (cm)	Start sample 3 (cm)	Start sample 4 (cm)
[15.5, 15.5]	[13.95, 13.95]	[12.4, 12.4]	[10.85, 10.85]