Numerical Identification of Motor Units in Muscle Tissue from High Resolution EMG Data

Von der Universität Bayreuth
zur Erlangung des Grades eines
Doktors der Naturwissenschaften (Dr. rer. nat.)
genehmigte Abhandlung

von

Tobias Andreas Sproll

aus Engen

1. Gutachter Prof. Dr. Anton Schiela
2. Gutachter Dr. Martin Weiser

Tag der Einreichung: 26.07.2023
Tag des Kolloquiums: 11.01.2024

Abstrakt (deutsch)

Im menschlichen Körper sind motorische Einheiten die kleinste kontrollierbare Einheit eines Muskels. Wenn wir uns bewegen und eine motorische Einheit aktiv ist, erzeugt diese Aktivität ein elektrisches Potential, welches auf der Haut gemessen werden kann. In der medizinischen Behandlung und Forschung kann es nun notwendig sein, bestimmte motorische Einheiten in Muskeln anhand von Messungen auf der Haut zu identifizieren. Jüngste Fortschritte in der Oberflächenelektromyographie ermöglichen es, Informationen über einzelne motorische Einheiten aus diesen Messungen zu gewinnen. Im ersten Teil dieser Arbeit stellen wir ein mathematische Modell vor, welches die numerische Identifizierung von motorischen Einheit aus einer Oberflächenelektromyographiemessung ermöglicht. Basierend auf einem elektrostatischen Modell leiten wir ein adjungierter Ansatz her der die effizienten Simulation einer Oberflächenelektromyographiemessung ermöglicht. Dieses Modell benutzen wir um ein nichtlineares Optimierungsproblem zur Identifizierung von motorischen Einheiten zu definieren. Wir zeigen dann, dass dieses Optimierungsproblem mindestens eine Lösung besitzt. Des Weiteren leiten wir die Optimalitätsbedingungen erster Ordnung her welche die Basis eines Algorithmus zur Lösung des Problems bilden.

Im zweiten Teil der Arbeit stellen wir einen Optimierungsalgorithmus vor, der die spezifische mathematische Struktur dieses Problems ausnutzt. Dieser Algorithmus basiert auf einem Newton Verfahren im Funktionenraum mit ungenauen Funktionsauswertungen durch adaptive numerische Quadratur. Wir präsentieren ein globales Konvergenzresultat für diese Methode und beschreiben detailliert seine algorithmische Implementierung. Abschließend studieren wir verschiedene numerische Beispiele welche die praktische Performance des Algorithmus demonstrieren.

Abstract (english)

In the human body, motor units are the smallest controllable unit of a muscle. When we move and a motor unit is active, this activity generates an electrical potential that can be measured on the skin. In medical treatment and research, it may now be necessary to identify specific motor units in muscles based on measurements on the skin. Recent advances in surface electromyography make it possible to obtain information about individual motor units from these measurements.

In the first part of this paper, we present a mathematical model that allows the numerical identification of a motor unit from a surface electromyography measurement. Based on an electrostatic model, we derive an adjoint approach allowing the efficient simulation of a surface electromyography measurement. We use this model to define a nonlinear optimization problem for identifying motor units. We then show that this optimization problem has at least one solution. Furthermore, we derive the first-order optimality conditions which form the basis of an algorithm to solve the problem.

In the second part of the paper, we present an optimization algorithm that exploits the specific mathematical structure of this problem. This algorithm is based on a Newton method in function space with inexact function evaluations by adaptive numerical quadrature. We present a global convergence result for this method and describe in detail its algorithmic implementation. Finally, we study various numerical examples that demonstrate the practical performance of the algorithm.

Acknowledgments

First, I would like to thank my supervisor Prof. Dr. Anton Schiela, who brought my attention to the topic during my studies at the TU Berlin and supported my research since then. His knowledge, motivation, and joyful nature helped to remove any obstacles in the research process and made writing this thesis possible.

Furthermore, I would like to thank Dr. Martin Weiser for reviewing this thesis as the second surveyor. I also thank Prof. Dr. Lars Grüne and Prof. Thomas Kriecherbauer, who were a part of the examination committee.

Next, I would like to thank Prof. Dr. Madeleine Lowery and her team, who supported the development of the model by providing insight into the exciting field of biomedical engineering. I also acknowledge Prof. Dr. Bernd Lapatki, who suggested the topic to my supervisor Prof. Dr. Anton Schiela.

I thank all the members of the Chair of Applied Mathematics for a pleasant and stimulating working environment: Dr. Matthias Stöcklein, Dr. Julian Ortiz, Dr. Manuel Schaller, Dr. Simon Pirkelmann, Dr. Georg Müller, Dr. Michael Baumann, Dr. Marleen Stieler, Dr. Arthur Fleig, Dr. Bastian Pötzl, Lisa Krügel, Frederik Köhne, Kilian Pioch, Mario Sperl, and Jonas Schießl. Special thanks go to Dr. Robert Bayer, who helped me countless times to install new software for this thesis and the lectures, and to Sigrid Kinder, who supported the whole Chair with her incredible administrative skills.

Last but not least, I would like to thank my parents, who supported me my whole life and encouraged me to pursue my goals. Without you, this accomplishment would not have been possible.

Contents

Chapter 1

Introduction

In the human body, muscles are responsible for movements. These muscles consist of many muscle fibers organized in so-called motor units. A motor unit is, thereby, the smallest controllable unit of the muscle. When using a muscle, the nervous system activates one or more of these motor units. This activation causes electrical signals, so-called action potentials, to propagate along the muscle fibers. These propagating action potentials create a spatially and temporally changing potential field. We can measure this potential with electrodes placed on the skin above the muscle.

A fundamental question in medical research and diagnosis is: what is the bioelectric source responsible for a specific measured potential on the skin? To answer this question, we need to find a way to identify these sources from the given measurements. Such identification of bioelectric activity from surface measurement is required in many fields of medicine, e.g., in measuring brain activity (EEG) or cardiac activity (ECG). Correspondingly, a lot of work has been performed to develop tools for computational assistance, see, e.g., [28] and references therein for EEG. In general terms, refinements of classical Tychonov regularization techniques are applied to the spatial problem.

The corresponding technique for measuring action potentials in muscles is called electromyography (EMG). It can be used for research (which motor unit is responsible for which movement?) or in pre-operative planning (where is the location of important nerves, which should not be harmed in operations?). Similar techniques as described above have also been applied to electromyography measurements [42, 57, 58]. These techniques yield a smooth, distributed reconstruction of sources, which is appropriate in those applications where the sources are smoothly distributed within the tissue. These approaches solve mainly spatial problems, not taking into account the spatio-temporal structure of the problem.

However, bioelectrical sources in motor units are known to have special structures. They consist of a characteristic action potential that is concentrated along a line in space and moves along a fiber. Approaches that directly attempt to process the spatio-temporal

information and exploit the high temporal resolution of surface EMG are less common. A notable exception is [46]. The authors consider a regularized least-squares approach for fitting the EMG signal by a linear combination of a moderate number of analytically predefined and prelocated waveforms. This enables fast computations in real-time with modest accuracy.

Our work aims to establish a mathematically sound approach to the EMG problem that, similarly to [46], considers the special structure of moving action potentials in muscles. Similar to [46], we use a least-squares tracking type functional for the identification. However, we introduce refined modeling approaches to simulate a surface EMG measurement from a given source. We also represent the source to be identified more flexibly via a curve that can be chosen freely inside the muscle tissue.

We can model, using a quasi-static approach, cf. [51], the potential Φ, generated from the moving action potentials, for each time instant t as the solution of a Poisson equation of the form

$$\int_{\overline{\Omega}} (\sigma(x)\nabla\Phi(x,t)) \cdot \nabla v(x)\,\mathrm{d}x + \int_{\partial\Omega_S} \mu\Phi(s,t)v(s)\,\mathrm{d}s = \int_{\overline{\Omega}} v(x)\,\mathrm{d}\rho(t) \quad \forall v \in C^\infty(\overline{\Omega}) \tag{1.1}$$

In this setting, the time-dependent source density $\rho(t)$, given through a moving action potential, is spatially concentrated on the motor unit and thus modeled as a Radon line measure. Solutions to this problem in the sense of Stampacchia can be found in $W^{1,p'}(\Omega)$ where $p' > d/(d-1)$, $\Omega \subset \mathbb{R}^d$ (cf., e.g., [53] and a discussion concerning uniqueness can be found in [47]). However, the direct numerical solution of (1.1) for all t in the time interval of interest would incur a high computational cost, i.e., for each time instant an elliptic PDE on a three-dimensional domain would have to be solved. That would render numerical approaches to the identification problem too costly. We overcome this difficulty by an adjoint approach reducing the simulation of a single measurement via (1.1) to evaluating a line integral.

Based on this simulation model, we establish a least-squares type tracking problem to identify the motor units from a surface EMG measurement. The subject of our identification problem is the trajectory of the moving action potential, represented by a parameterized curve. Unlike the least-squares approach in [46], our problem is formulated in an infinite-dimensional function space setting, which makes the analysis of the problem much more involved. Those kinds of infinite-dimensional optimization problems emerge in many application-related problems. Therefore, the prototype of this problem is studied very well, cf., e.g., [56, 63, 34], and we will employ techniques of analysis established in the field of research to show the existence of solutions and first-order optimality conditions. This lays the groundwork for an optimization-based numerical approach to solve our identification problem.

To solve the identification problem numerically, we rely on well-known algorithmic approaches, cf., e.g., [48, 6]. However, due to the special structure of our problem, some algorithmic adjustments are necessary. In particular, we use an adaptive quadrature

rule to evaluate the objective function changing the evaluation method from step to step. Thus, we interpret the evaluation as an inexact evaluation of continuous quantities rather than an exact evaluation of discrete quantities. Inexact function evaluations are not uncommon in nonlinear optimization, and the topic is studied in different settings, cf., e.g., [33, 38, 10, 13, 15, 62].

1.1 Outline

The goal of this thesis is to derive an optimization problem suitable for the identification of a motor unit from surface EMG measurements. Furthermore, we present a numerical algorithm that can solve the optimization problem. Finally, we validate the theoretical considerations numerically. The rest of this chapter shows an outline of this thesis.

Chapter 2 - Modeling of surface EMG Measurements This chapter explains the modeling aspect of the optimization problem. We start by describing the moving action potentials. Afterward, we discuss a biophysical model in form of a partial differential equation (PDE), describing the transmission of potentials in the human body. Moreover, we discuss this PDE from a mathematical point of view, i.e., its solvability. Finally, we introduce an adaptive approach for simulating a surface EMG measurement which we also analyze numerically.

Chapter 3 - An Optimization Problem to Identify a Motor The third chapter introduces the optimization problem used for the identification. We then conduct a thorough analysis of the problem to show the existence of solutions. Moreover, we define and analyze the augmented Lagrangian function and derive first-order optimality conditions for the optimization problem.

Chapter 4 - A Numerical Identification Algorithm This chapter presents the augmented Lagrangian method used to solve the optimization problem from the previous chapter. Thereby, we modify the inner Newton method such that it considers the specific problem structure. Furthermore, we study the convergence of the Newton method requiring two assumptions to be met. Therefore, we discuss these assumptions in detail before we prove a global convergence result. Additionally, we shortly discuss the local convergence of the Newton method.

Chapter 5 - A Practical Optimization Algorithm The fifth chapter treats the implementation details. Therefore, we first discuss the finite element discretization of the required quantities. Then, we explain the implementation of the adaptive quadrature algorithm and how we can incorporate the requirements needed for global convergence. Furthermore, we discuss the computation of the Lagrangian function and its derivatives. Afterward, we talk about the implementation of the Newton method. Thereby, we split the discussion into two parts. First, we discuss the computation of search directions and secondly the line search. We close the discussion of the implementation details by

explaining the augmented Lagrangian method, i.e., how we compute the update of the Lagrange multiplier.

Chapter 6 - Numerical Examples The sixth chapter studies some numerical examples. Therefore, we discretize the problem with finite elements as described in the previous sector. We then solve the optimization problem and study different examples which are interesting from a mathematical and a user point of view. Thereby the first couple of examples use a cylindrical domain representing some limb. For this geometry, we conduct a performance analysis and study the influence of the impulse response functions, two algorithmic parameters, geometric properties, and the position of the electrodes. Furthermore, we study an example where we identify a motor unit that consists of multiple muscle fibers. Last but not least, we study the identification in a hand geometry created from MRI data.

Chapter 7 - Conclusion The last chapter of this work summarizes the results and provides possibilities for future research.

Chapter 2

Modeling of Surface EMG Measurements

In the first part of this work, we derive a mathematical model that simulates a surface EMG measurement. This model builds the foundation for the optimization problem, which we will use to identify a motor unit from a surface EMG measurement. Before discussing the model, we shortly describe the principle of a surface EMG measurement.

When we use a muscle, our brain sends a signal that activates the motor unit responsible for the movement. A motor unit is, therefore, the smallest controllable unit of a muscle. When a motor unit is activated, two electrical signals, the so-called action potentials, propagate from the neuromuscular junction in the middle of the motor unit toward both ends. This propagation of electrical sources produces an electric potential. Since human tissue behaves like a volume conductor, we can measure this potential at the skin. Our model, therefore, consists of two parts. First, we model the activation of the motor unit. In this section, we model the motor unit and the current source density ρ_i, describing the moving action potentials. Secondly, we model the transmission of the generated potential through human tissue. For this purpose, we model a domain $\Omega \in \mathbb{R}^3$ representing some part of the human body. We then use Maxwell's equations to derive a PDE describing the transmission of the electric potential in this domain. As we will see, using this PDE to simulate a surface EMG measurement would be inefficient and associated with high computational costs. Thus, we introduce an adjoint approach for efficiently simulating a surface EMG measurement. We then close this chapter with a short numerical example, discussing some aspects of the numerical simulation. Parts of this chapter has been published in [64].

2.1 Modeling Motor Unit Activity

Before we can model the activation of a motor unit, we must first define what a motor unit is and how we can describe it mathematically. Motor units are the smallest controllable

units in a muscle. In general, more than one motor unit is activated. But these motor units are distinguishable using decomposition techniques, cf. [37, 23, 35]. Therefore, we assume that only one motor unit is active. Most muscles consist of several hundred motor units, and each motor unit consists of up to 2000 muscle fibers itself, see [20, 26]. Obviously, modeling all these muscle fibers on a microscopic level would exceed the computational effort. Therefore, we model the muscle at the macroscopic level as a domain Ω_M and select only one motor unit on the microscopic level. In the next section, we will discuss the properties of Ω_M in more detail.

In general, all muscle fibers in a motor unit are activated simultaneously. Furthermore, we can consider biological tissue as a linear volume conductor, and consequently, we can apply the principle of superposition, see [43, 45]. Since the muscle fibers of one motor unit lie in the same cylindrical region of the muscle, see [20, 26], these observations make it possible to consider the motor unit as single-fibered. Such a single-fibered motor unit represents the center of mass of the actual motor unit, i.e., the area of the motor unit with the most muscle fibers. Figure 2.1 shows a cylindrical motor unit with 100 muscle fibers.

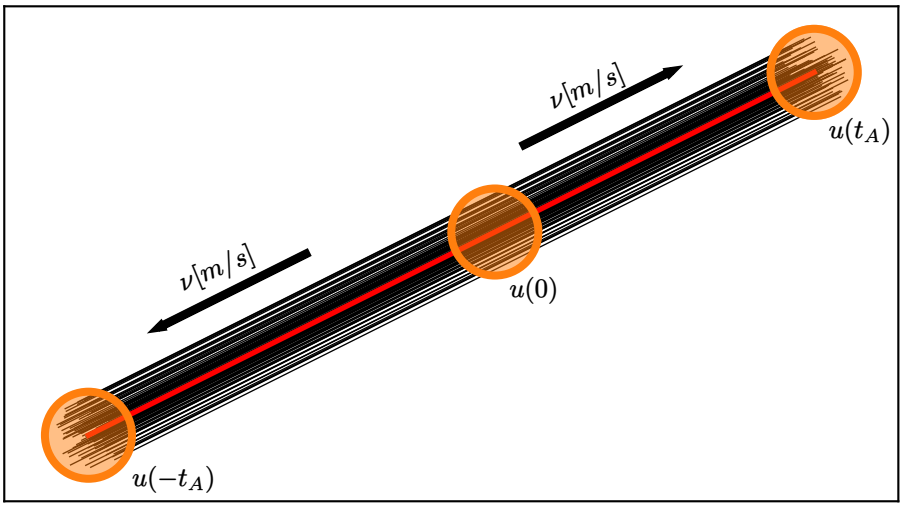


Figure 2.1: Sketch of a motor unit with 100 muscle fibers (black) and single-fibered representative (red)

Since the radius of such a single-fibered motor unit is very small, we will model our motor unit as (the trace of) a fixed curve u in Ω_M. To prove that the optimal problem (3.3) has at least one solution, we need that this curve u to satisfy some regularity requirements. Thus, we demand $u \in H^2(I, \Omega_M)$, where $H^2(I, \Omega_M)$ is the space of vector-valued functions $v : \mathbb{R} \supset I \mapsto \Omega_M$ such that each component is in the Sobolev space $H^2(I)$, cf. [1, 8]. Since $H^2(I)$ is continuously embedded in $C^1(I)$, we can, in this section, treat the curve u as if it is a regular C^1 curve. When we define the moving action

potentials, we choose a parameterization for u, determining the length of the interval I.

When a motor unit is activated, two action potentials propagate from the neuromuscular junction toward both ends of the motor unit. Those two action potentials extend spatially along that motor unit but also move along the motor unit during the activation period. The action potential takes a characteristic shape, sketched in Figure 2.2, which we map onto a specific time-varying segment of the curve describing the motor unit to obtain a line measure $\rho_l(t)$ at time t. This line measure will later define the current source density responsible for the measured electric potential. Further difficulties arise towards the ends of the fiber. Here the principle of conservation of charges is violated, and we must add additional source terms.

Biomedical modeling of action potentials starts with the following function

$$i_m(z) := \begin{cases} -c(\sigma_{in}, r, n_F)\exp(az)\left(6az + 6(az)^2 + (az)^3\right) & \text{if } z \leq 0 \\ 0 & \text{else,} \end{cases} \tag{2.1}$$

in terms of a reference parameter $z \in \mathbb{R}$. Here $a > 0$ is a scaling factor that determines the spatial extension of the signal, and $c(\sigma_{in}, r, n_F)$ is a constant depending on the intracellular conductivity σ_{in}, the radius of the motor unit r, and the number of muscle fibers n_F. We refer to [43, 52, 3] for a more detailed description of the action potential. The antiderivative of i_m is

$$I_m(z) := \begin{cases} -c(\sigma_{in}, r, n_F)\exp(az)\left(3(az)^2 + (az)^3\right)\frac{1}{a} & \text{if } z \leq 0 \\ 0 & \text{else} \end{cases}$$

and thus

$$\int_{\mathbb{R}} i_m(z)\ \mathrm{d}z = I_m(0) - \lim_{z\to-\infty} I_m(z) = 0,$$

which corresponds to the principle of conservation of charge in the body.

Up to now, the action potential is defined as a function on $\mathbb{R}$, so the next step is to define a pull-pack of i_m onto the given curve $u(I) \subset \Omega_M$. A common assumption in biomedical modeling is that the velocity ν with which the action potential propagates along the fiber is constant, see [43]. Since the curve u represents the trajectory of the two propagating action potentials, we choose the parameterization of the curve u such that it matches with the propagation velocity ν of the signal, in other words, $|\dot{u}(\tau)| \equiv \nu$. That means we can identify each point on the curve u with some $z \in \mathbb{R}$ via the arc length

$$z(\tau) = \int_0^\tau |\dot{u}(\xi)|\,\mathrm{d}\xi = \int_0^\tau \nu\,\mathrm{d}\xi = \nu\tau.$$

Therefore, $z(0) = 0$ corresponds to the neuromuscular junction $u(0)$, and we can identify points that are on the "right" side of the neuromuscular junction with some $z \in \mathbb{R}^+$ and

points on the "left" side with some $z \in \mathbb{R}^-$. With t_A, we denote the time one action potential needs to propagate from the neuromuscular junction to the fiber end. Thus, $z(-t_A)$ and $z(t_A)$ correspond to the "left" resp. "right" end of the motor unit. Consequently, we choose $I = [-t_A, t_A]$ and denote with $u(-t_A)$ and $u(t_A)$ the corresponding ends of the motor unit.

Next, we assume that the activation of the motor unit happens at some time $t_0 > 0$. Then, to model the action potential propagating from the neuromuscular junction toward the "right" end of the fiber, we shift the origin of the action potential $i_m(z)$ by $\nu \cdot (t_0 - t)$ and set

$$\tilde{\rho}(u(\tau), t) = \begin{cases} i_m\left(z(\tau) + \nu \cdot (t_0 - t)\right)\right) & \text{if } z(\tau) > 0 \\ 0 & \text{else.} \end{cases}$$

To model the second action potential that propagates in the opposite direction, we mirror the signal at point zero, which is equal to adding a minus sign before $z(\tau)$ and changing $z(\tau) > 0$ to $z(\tau) < 0$. By combining both action potentials, we get the line-measure

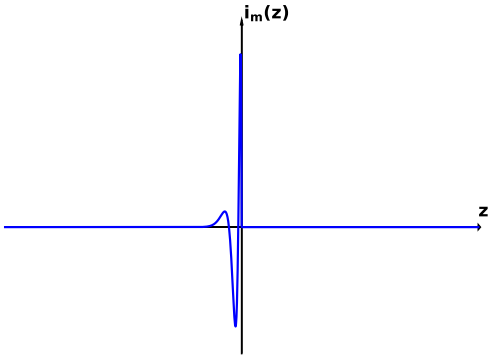
$$\tilde{\rho}_l(u(\tau), t) := \rho_l(\tau, t) := \begin{cases} i_m\left(z(\tau) + \nu \cdot (t_0 - t)\right)\right) & \text{if } z < 0, \\ i_m\left(-z(\tau) + \nu \cdot (t_0 - t)\right)\right) & \text{if } z > 0 \end{cases} = i_m\left(\nu \cdot (|\tau| + t_0 - t)\right) \tag{2.2}$$

as a source term.

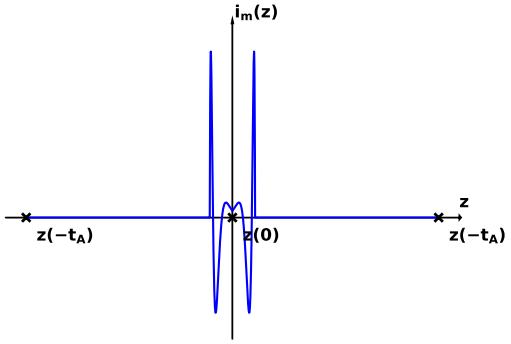
Conservation of charge and end effects As we have observed, our model i_m of the action potential respects the conservation of charge, i.e. its total integral over $\mathbb{R}$ vanishes. However, our definition of the measurement $y(t)$ (cf. 2.35, below) involves only an integral over a bounded subset of $\mathbb{R}$, and the corresponding total charge is given by (taking into account the substitution of variables formula):

$$\begin{aligned} \rho_{total}(t) &= \int_{-t_A}^{0} i_m\left(\nu \cdot (-\tau + t_0 - t)\right) \nu \, \mathrm{d}\tau + \int_{0}^{t_A} i_m\left(\nu \cdot (\tau + t_0 - t)\right) \nu \, \mathrm{d}\tau \\ &= -I_m(\nu \cdot (-\tau + t_0 - t))\Big|_{-t_A}^{0} + I_m(\nu \cdot (\tau + t_0 - t))\Big|_{0}^{t_A} \end{aligned} \tag{2.3}$$

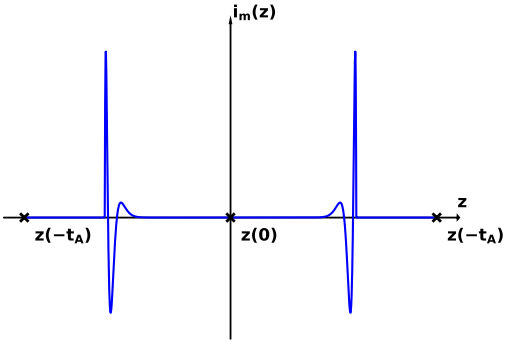
which is, in general, non-zero. This truncation of the integral can be observed in Figure 2.2 when comparing the prototype of the action potential (see Figure 2.2a) with the two propagating action potentials in the Figures 2.2b and 2.2d. In the beginning, the tails of the action potentials are not present on the motor unit (see Figure 2.2b). After some time, the first part of the action potentials is no longer on the motor unit (see Figure 2.2d). These truncations need to be compensated for. Otherwise, the principle of conservation of charge would be violated and yield characteristic artifacts in simulations,



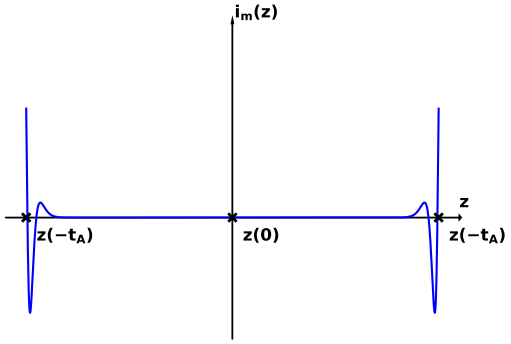
(a) Prototype of the action potential (2.1)



(b) Action potential truncated at the tail



(c) Action potential with negligible truncation



(d) Action potential truncated at the head

Figure 2.2: Different versions of the action potential

so-called *end effects*. We can observe those end effects when we discuss the simulation at the end of this chapter.

The representation of (2.3) by boundary terms at $\tau = -t_A, 0, t_A$ already suggests how to construct an appropriate compensation. We will add point (Dirac) measures at $u(0)$, $u(t_A)$, and $u(-t_A)$, scaled by the negatives of the corresponding boundary terms to the measure $\rho(t)$. This can also be interpreted physiologically: at the ends of the fibers, transitional charge imbalances are compensated by small charge displacements in the close vicinity of the end plates. A similar approach can be found in [26].

Figure 2.2b shows that the first truncation of the action potential is at the neuromuscular junction. In (2.3), this corresponds to the boundary term at $\tau = 0$. Therefore we get

$$\tilde{\rho}_s(u(0), t) := \rho_s(0, t) := 2I_m(\nu \cdot (t_0 - t)), \tag{2.4}$$

which is a Dirac measure at the neuromuscular junction $u(0)$. Figure 2.2c shows that after some time, the support of the action potential lies completely on the motor unit.

Therefore, the charge imbalances from the truncation at the neuromuscular junction tend exponentially to zero.

When the action potentials arrive at the ends $u(-t_A)$ and $u(t_A)$ of the motor unit, see action potentials in Figure 2.2d, they are again truncated. Thus we have to compensate the boundary terms at $\tau = -t_A$ and $\tau = t_A$ in (2.3) by Dirac measures at $u(-t_A)$ and $u(t_A)$ as follows:

$$\tilde{\rho}_s(u(-t_A), t) = \tilde{\rho}_s(u(t_A), t) := \rho_s(t_A, t) := -I_m(\nu \cdot (t_A + t_0 - t)). \tag{2.5}$$

Observe that these charges are 0, if $t \leq t_A + t_0$, since $I_m(z) = 0$ for $z \geq 0$. Our above-described compensation technique can reduce these end effects, as shown in the last part of this chapter. The current source density ρ_i is then the sum of the line measure $\tilde{\rho}_l(u(\tau), t)$ and the stationary source term

$$\tilde{\rho}_s(t) = \tilde{\rho}_s(u(0), t) + \tilde{\rho}_s(u(t_A), t) + \tilde{\rho}_s(u(-t_A), t). \tag{2.6}$$

The next step in deriving a model for simulating surface EMG measurements is to derive a model for the potential transmission in human tissue. Thus, this is the focus of the next section.

2.2 Transmission of Electric Potentials in Human Tissue

In this section, we study the behavior of electromagnetic fields in human tissue. Therefore, we derive a model that connects the charge density of the moving action potential ρ_i with the electric potential Φ. This model consists of two parts. The first part describes the transmission of the potential in the domain, and the second part deals with the behavior at the boundary. Due to the moving action potential, the model is potentially time-dependent. But there are some common assumptions when modeling a bioelectric system. Those assumptions transform the time-dependent model into a quasi-static model. We additionally study what happens when the potential transmissions from one tissue type into another. Finally, we transform the obtained PDE from its strong form into the weak form needed for the following mathematical analysis. Before we can derive this model, we must first discuss the mathematical description of human tissue.

Modeling Human Tissue

First, we note that it is sufficient to model only those parts of the body we are interested in. This is reasonable since we do not gather relevant information from parts far away. For example, we do not gain additional information from the left arm when we are interested in a motor unit in the right arm. We choose an open and bounded domain $\Omega \subset \mathbb{R}^3$ with Lipschitz boundary $\partial\Omega$ to represent such a body part. This means $\partial\Omega$ is locally the graph of a Lipschitz continuous function, see [18, 56]. Since the human body consists of different tissue types, we split the domain Ω into several subdomains Ω_i, such

that

$$\begin{aligned}\Omega &= \cup \Omega_i \\ \partial\Omega_{ij} &= \Omega_i \cap \Omega_j \quad \text{if } i \neq j,\end{aligned}$$

where $\partial\Omega_{ij}$ is the shared boundary between two tissue types. For the sake of simplicity, we will restrict our model to muscle, fat, and bone tissue. The muscle tissue we denote with Ω_M, the fat tissue with Ω_F, and the bone tissue with Ω_B. To denote the boundary between two subdomains, we use $\partial\Omega_{MF}$, $\partial\Omega_{MB}$, and $\partial\Omega_{FB}$. The order of the indices thereby depends on the direction of the outer normal vector. If, e.g., the vector points from muscle to fat, we denote it with $\partial\Omega_{MF}$. Consequently, we use $\partial\Omega_{FM}$ if the normal vector points from fat to muscle. Since we model only a part of a human body, we get an artificial boundary truncating the body, which we call $\partial\Omega_A$. Last but not least, skin tissue bounds the rest of our domain, and we label it with $\partial\Omega_S$. Thus,

$$\begin{aligned}\partial\Omega &= \partial\Omega_S \cup \partial\Omega_A \\ \emptyset &= \partial\Omega_S \cap \partial\Omega_A.\end{aligned}$$

In the human body, each tissue type has three different electromagnetic properties. Those properties are the conductivity σ, the permittivity ϵ, and the permeability μ. First, we note human tissue is not magnetic. Thus, the permeability is given by the permeability of a vacuum μ_0. We also know that the upper-frequency limit in human tissue is around $1kHz$, see [43, 51, 45]. We can assume that the tissue is purely resistive in this frequency range. Thus, the properties are independent of the exact frequency, see [43, 54]. All tissue types, excluding muscle tissue, are isotropic, [3, 43, 54]. Therefore, conductivity and permittivity are the same in each direction. Contrary, muscle tissue is anisotropic, i.e., the conductivity and permittivity are higher in the direction of the muscle fibers. Therefore, both properties depend on the geometry. Since we have an inverse problem, the orientation of the muscle fibers is a priory unknown. But for simple geometries, we can assume that muscle fibers are straight or, at least, only slightly curved. Therefore we can define, for example, the x-axis as the axial direction and rotate the geometry such that the motor unit is approximately parallel to the x-axis. Thus we can represent both properties with a rank two tensor of the form

$$\sigma_M = \begin{pmatrix} \sigma_{M,axial} & 0 & 0 \\ 0 & \sigma_{M,radial} & 0 \\ 0 & 0 & \sigma_{M,radial} \end{pmatrix},$$

see [43, 3]. To be consistent, we write the conductivity and permittivity as a scaled Identity matrix if the tissue is isotropic, e.g.,

$$\sigma_F = \begin{pmatrix} \sigma_F & 0 & 0 \\ 0 & \sigma_F & 0 \\ 0 & 0 & \sigma_F \end{pmatrix},$$

Table 2.1 shows conductivity and permittivity values for different tissue types. Note that we have chosen the average of resistive and conductive skin, and the permittivity is given relatively to the permittivity in a vacuum, which is $\epsilon_0 = 8.8 \cdot 10^{-12} \frac{As}{Vm}$. The conductivity is positive and constant for all tissue types, and thus the two estimates

$$\sigma_{\max} := \max\{\|\sigma_M\|_\infty, |\sigma_F|, |\sigma_B|\} < \infty \tag{2.7}$$

$$\sigma_{\min} := \min\{\lambda_{\min}(\sigma_M), |\sigma_F|, |\sigma_B|\} > 0. \tag{2.8}$$

are valid. This shows that the conductivity is in $L^\infty(\Omega)$ and elliptic, which we will need later for the ellipticity and boundedness of the bilinear form (2.30).

tissue type	fat	bone	muscle (axial)	muscle (radial)	skin
$\sigma[S/m]$	$4.0 \cdot 10^{-2}$	$2.0 \cdot 10^{-2}$	$4.0 \cdot 10^{-1}$	$9.0 \cdot 10^{-2}$	0.5
$\epsilon[rel.]$	$1.5 \cdot 10^5$	$5.85 \cdot 10^3$	$2.0 \cdot 10^7$	$4.4 \cdot 10^6$	$5.5 \cdot 10^4$

Table 2.1: Conductivity σ and permittivity ϵ for different tissue types at $100Hz$, cf. [59]

Next, we derive the physical model that describes the transmission of the electric potential through human tissue.

Using Maxwell's Equations to Describe the Potential Transmission

In general, Maxwell's equations describe Electromagnetic phenomena, see [36, 52]. Since the impressed current density $\mathbf{J}_i$ is harmonic with frequency ω, the induced electric field E is also harmonic, see [52]. Therefore, we use the time-harmonic Maxwell's equations given by:

$$0 = i\omega \boldsymbol{B} + \operatorname{curl} \boldsymbol{E} \tag{2.9}$$

$$0 = \operatorname{div} \boldsymbol{B} \tag{2.10}$$

$$-\boldsymbol{J} = i\omega \boldsymbol{D} - \operatorname{curl} \boldsymbol{H} \tag{2.11}$$

$$\rho = \operatorname{div} \boldsymbol{D} \tag{2.12}$$

where $\boldsymbol{E}$ is the electric field intensity, $\boldsymbol{H}$ is the magnetic field intensity, $\boldsymbol{D}$ is the electric flux density, $\boldsymbol{B}$ is the magnetic flux density, ρ is the electric charge density, and $\boldsymbol{J}$ is the electric source density, see [36, 32].

Equation (2.9) is called Faraday's law of Induction. It states that a temporal changing magnetic flux density causes a changing electric field intensity. Since, in general, there is no magnetic material in human tissue, see [51], and we further assume that there is no external magnetic field, we can neglect this equation.

Equation (2.10) is Gauss's Magnetic Law. It says that the magnetic flux is source-free. We can also neglect this equation.

Equation (2.11) is called Ampere's law. Similar to Faraday's law of Induction, it states that the flow of electric currents causes a spatially changing magnetic field.

Last but not least, equation (2.12) is Gauss's Electric Law. It describes how electric charges influence the resulting static electric field. More detailed, it says that charges are the endpoints of field lines.

Analyzing those equations, we notice that they do not cover material properties. We can add the material properties by inserting the so-called constitutive equations, see [36]. Similarly to [43, 51], we assume that human tissue is a linear material. But in contrast, we allow that the material is inhomogeneous and anisotropic. With these assumptions, the constitutive equations are

$$\boldsymbol{D} = \epsilon \boldsymbol{E} \tag{2.13}$$

$$\boldsymbol{B} = \mu \boldsymbol{H} \tag{2.14}$$

where $\epsilon : \mathbb{R}^3 \mapsto \mathbb{R}^{3\times 3}$ is the dielectric tensor and $\mu : \mathbb{R}^3 \mapsto \mathbb{R}^{3\times 3}$ is the permeability tensor. Inserting (2.13) and (2.14) into and (2.11) we get

$$-\boldsymbol{J} = \epsilon i\omega \boldsymbol{E} - \operatorname{curl} \mu^{-1} \boldsymbol{B} \tag{2.15}$$

Since the divergence of the curl is always zero, we can apply the divergence to (2.15) such that

$$-\operatorname{div} \boldsymbol{J} = \operatorname{div} \epsilon i\omega \boldsymbol{E}. \tag{2.16}$$

Next, we assume that human tissue behaves as a volume conductor, i.e., the material is conductive, see [43]. Thus the electric source density $\boldsymbol{J}$ can be written as the sum of a conducting current field $\sigma \boldsymbol{E}$ and an impressed current field $\mathbf{J}_i$, i.e.,

$$\boldsymbol{J} = \sigma \boldsymbol{E} + \boldsymbol{J}_i, \tag{2.17}$$

where $\sigma : \mathbb{R}^3 \mapsto \mathbb{R}^{3\times 3}$ is the conductivity tensor, see [51, Eq. 27]. The impressed current field $\mathbf{J}_i$ is, thereby, connected to the current source density ρ_i via

$$\operatorname{div} \mathbf{J}_i = -\rho_i, \tag{2.18}$$

see [51]. Inserting (2.17) and (2.18) into (2.16) leads to

$$\operatorname{div} (\sigma + i\omega\epsilon) \boldsymbol{E} = -\operatorname{div} \boldsymbol{J}_i = \rho_i. \tag{2.19}$$

Next, we introduce the magnetic vector potential A and the scalar potential Φ. Therefore, we first notice that according to the magnetic law of Gauss (2.10)

$$\operatorname{div} \boldsymbol{B} = 0.$$

Thus, we can conclude that there exists a magnetic vector potential $\boldsymbol{A} : \mathbb{R}^3 \mapsto \mathbb{R}^3$ with

$$\boldsymbol{B} = \operatorname{curl} \boldsymbol{A}.$$

Inserting this equation in Faraday's law (2.9), we get

$$0 = \operatorname{curl}\left(E + i\omega \boldsymbol{A}\right).$$

Similarly, we can now conclude that there exists an electric scalar potential $\Phi : \mathbb{R}^3 \mapsto \mathbb{R}$ with

$$E = -\nabla\Phi - i\omega \boldsymbol{A}. \tag{2.20}$$

Inserting (2.20) into (2.19) leads then to

$$-\operatorname{div}\left((\sigma + i\omega\epsilon)\left(\nabla\Phi + i\omega \boldsymbol{A}\right)\right) = \rho_i. \tag{2.21}$$

In most cases, solving this hyperbolic PDE would be too costly. But when modeling bio-electric phenomena, it is common to assume that the process is quasi-static, see [45, 51, 44]. The quasi-static assumption states that the transmission of the electric field is much faster than the propagation of action potentials generating it. Thus the electric field behaves at each point in time as if it is static. Additionally, the quasi-static assumptions state that we can neglect inductive and capacitive effects. Before we apply those assumptions to (2.21), we briefly discuss their consequences and under which circumstances they are applicable.

The Quasi-Static Approximation For the sake of simplicity, we assume a homogeneous domain Ω while studying the quasi-static approximation. If the domain is not homogeneous, we can consider (2.21) in its weak form. The weak form allows a discussion of the quasi-static approximation for each subdomain individually as long as the transmission of the potential is continuous at the boundary between two subdomains. Therefore, we will add this condition to our list of boundary conditions in the next section. The following study of quasi-stationary is a summary of the examination done by Plonsey and Heppner, see [51].

To derive the quasi-static approximation, we assume that Φ and $\boldsymbol{A}$ satisfy the Lorentz gauge

$$\operatorname{div} \boldsymbol{A} = -(\sigma + i\omega\epsilon)\mu\Phi, \tag{2.22}$$

see [51, 29]. Using the Lorentz gauge decouples the computation of Φ and $\boldsymbol{A}$ such that they are the solution of the Helmholtz equations

$$\begin{aligned} \Delta\Phi + k^2\Phi &= \frac{-\rho_i}{\sigma + i\omega\epsilon} \\ \Delta\boldsymbol{A} + k^2\boldsymbol{A} &= -\mu\boldsymbol{J}_i, \end{aligned}$$

where $k^2 = -i\omega\mu(\sigma + i\omega\epsilon)$ is the complex wave number, see [51, 29]. Fundamental solutions to this type of problem are given by

$$A(x) = \frac{\mu}{4\pi} \int_\Omega \frac{J_s(x') \exp(-ikR)}{R} \, \mathrm{d}x' \tag{2.23}$$

$$\Phi(x) = \frac{1}{4\pi(\sigma + i\omega\epsilon)} \int_\Omega \frac{\rho_i(x') \exp(-ikR)}{R} \, \mathrm{d}x'. \tag{2.24}$$

Here, $R := \|x - x'\|_2$ is the euclidean distance between a field point x (belongs to $\boldsymbol{A}$ resp. Φ) and a source point x' (belongs to $\mathbf{J}_i$ resp. ρ_i). The maximal distance depends on the actual body, but for simplicity, we assume $R_{max} \leq 0.5m$.

tissue type	fat	bone	muscle (axial)	muscle (radial)	skin
k	$0.004(1-i)$	$0.015 - 0.011i$	$0.015 - 0.011i$	$0.007 - 0.005i$	$0.014 - 0.014i$
$\exp(-ikR_{max})$	1	$0.999 - 0.001i$	$0.999 - 0.001i$	1	$0.99 - 0.007i$
$\lvert kR_{max}\rvert^2$	$8 \cdot 10^{-6}$	$3.9 \cdot 10^{-6}$	$8.7 \cdot 10^{-5}$	$1.9 \cdot 10^{-5}$	$9.8 \cdot 10^{-5}$
$\frac{\omega\epsilon}{\sigma}$	0.021	0.016	0.373	0.27	$6 \cdot 10^{-4}$

Table 2.2: Properties for the quasi-static assumptions for different tissue types

The first assumption we want to verify is that we can neglect propagation effects, i.e., the electric field E behaves as if it is static. Studying (2.23) and (2.24), we notice that $\exp(-ik\|x - x'\|)$ corresponds to the time the potential needs to react to changes in the source. We get $\exp(-ikR) \approx 1$, if $kR_{max} \ll 1$ implying that the potentials (2.23) and (2.24) are equivalent to their static counterparts, cf., e.g., [29, Eq. 2.29 and Eq. 5.65]. Table 2.2 shows the computed phase delays for all tissue types used. The data shows that $\exp(-ikR) \approx 1$ for all tissues with a maximal error of 0.01 such that we can assume that both fields behave as if they are static. Consequently, we can solve (2.21) for each time step separately.

Secondly, we verify that we can neglect inductive effects, i.e., $E = -\nabla\Phi$. The term $i\omega\boldsymbol{A}$ in (2.20) describes the magnetic induction, i.e., the part of the electric field induced from the vector potential. Plonsey and Heppner showed, using (2.24), (2.23) and (2.22), that $|\omega\boldsymbol{A}/\nabla\Phi| = |kR|^2$, see [51, Eq. 13]. Thus, we can neglect the magnetic induction if $|kR|^2 \ll 1$, which is similar to the condition we used to show that the fields are static, i.e., for static fields, we can neglect the magnetic induction. The computed values in Table 2.2 underlay this assumption. Hence, $E = -\nabla\Phi$ and (2.21) reduces to the complex Poisson problem

$$-\operatorname{div}((\sigma + i\omega\epsilon)\nabla\Phi) = \rho_i.$$

The last assumption we want to verify is that we can neglect capacitive effects. As we will see, this assumption is also the weakest. The complex dielectric constant $\sigma + i\omega\epsilon$ describes the capacitive effects of the material. This constant is real, i.e. $\sigma + i\omega\epsilon \approx \sigma$, if

$\omega\epsilon/\sigma \ll 1$. As the computed values in Table 2.2 show, this condition is only satisfied for fat, bone, and skin tissue. The quotient is rather large for both types of muscle tissue, implying that neglecting capacitive effects is not necessarily possible for muscle tissue. However, it's well known that the conductivity and permittivity of muscle tissue depend on the frequency, and the quotient varies in a wide range, see [55, 43]. Nevertheless, [55, Table II] shows that the effect on the measured potential is rather small for the material properties we use, see Table 2.1. Therefore, we still assume that we can neglect capacitive effects for muscle tissue, i.e., $\sigma \approx \sigma + i\omega\epsilon$.

Using the quasi-static approximation, (2.21) transform to

$$-\operatorname{div}(\sigma(x)\nabla\Phi(x,t)) = \rho_i(x,t) \quad \forall t,. \tag{2.25}$$

This equation finally models the transmission of the, from the action potential ρ_i, induced electric potential Φ through human tissue. As usual, when using a PDE in optimization, we want to transform (2.25) into its weak form. Therefore we also need some boundary conditions, which we discuss in the next section.

Boundary Conditions

Previously, when we modeled human tissue, we described three different boundary types. Those are skin that separates the tissue from the surrounding air, an artificial boundary truncating the body, and boundaries between different tissues. For the first two cases, we start with general "Robin" boundary conditions, see [56, Section 2.3.2]. For our problem, they are given by

$$\partial_\nu\Phi(s,t) - \mu(\Phi(s,t) - \Phi_0) = g(s),$$

where ∂_ν is a normal derivative, $\Phi_0 \in \mathbb{R}$ is the potential on the outside of the domain, $g(s) \in L^\infty(\partial\Omega)$ is a external source density, μ is the scaled trace operator, and with s we denote the boundary variable. In our setting, the normal derivative ∂_ν is the directional derivative into the normal outward direction ν, which is orthogonal to the boundary $\partial\Omega$. The trace operator is a linear continuous map $\mu : H^1(\Omega) \mapsto L^2(\partial\Omega)$ such that for all continuous functions $(\mu y)(s) = y(s)$ a.e. on $\partial\Omega$. For more details, we refer to [2, 1]. To incorporate material properties, we scale the trace operator with $\frac{\sigma_i}{\delta s}$, where δs is the thickness of the boundary layer and σ_i is the conductivity of the boundary tissue. For simplicity, we use the same letter for the scaled trace operator, i.e., $\mu := \frac{\sigma_i}{\delta s}\mu$.

Robin boundary conditions state that the outgoing flow of the potential given by $\partial_\nu\Phi(s)$ is proportional to the difference between the potential at the boundary and some external potential.

First, we consider the boundary to be skin. In this case, we determine that the electrical potential in the air is zero. Furthermore, we assume that no electrical sources are present in the air. That means $\Phi_0 = 0$ and $g(s) \equiv 0$, such that we get

$$\partial_\nu\Phi(s,t) = -\mu\Phi(s,t) \quad \text{at } \partial\Omega_0 \tag{2.26}$$

as a boundary condition.

In the case of an artificial boundary, we suppose that no relevant sources are in the "infinite" volume conductor, such that $g(x) = 0$. Next, we assume that the potential is zero if we are far enough away from the current source density ρ_i, implying $\Phi_0 = 0$ and $\delta s \to \infty$. Consequently, we get homogeneous Neumann boundary conditions

$$\partial_\nu \Phi(s,t) = 0 \quad \text{at } \partial\Omega_\infty. \tag{2.27}$$

Finally, we want the potential transmission to be continuous at the boundary between different tissue types. Therefore, we use the so-called transmission conditions

$$\sigma_i \partial_\nu \Phi(s,t) = \sigma_j \partial_\nu \Phi(s,t) \quad \text{at } \partial\Omega_{ij}. \tag{2.28}$$

Deriving the Weak Form

Combining the results that we have derived so far, we get

$$\begin{aligned} -\text{div}\sigma\nabla\Phi(x,t) &= \rho_i(x,t) && \text{in } \Omega \\ \partial_\nu \Phi(x,t) &= -\mu\Phi(x,t) && \text{at } \partial\Omega_S \\ \partial_\nu \Phi(x,t) &= 0 && \text{at } \partial\Omega_A \\ \sigma_i \partial_\nu \Phi(x,t) &= \sigma_j \partial_\nu \Phi(x,t) && \text{at } \partial\Omega_{ij}. \end{aligned} \tag{2.29}$$

Since there are no time derivatives, this is a spatial PDE. Therefore, we must solve it for each temporal snapshot $t \in [0,T]$. For fixed t, (2.29) is the so-called strong form of an elliptic PDE. The solution of such a PDE has to be at least two times differentiable. In general, this is a strong assumption, and there must not necessarily exist a solution to the problem. To overcome this problem, mathematicians have defined the weak derivative, Sobolev spaces, and the weak or variation form of a PDE. A detailed theory about those topics can be found, e.g., in [1, 56, 25].

We apply formal integration by parts to the first equation of (2.29) to derive the weak form. We must thereby consider that the conductivity σ depends on the tissue type and that the transmission of the potential Φ is continuous at the boundary. To study the influence of these two aspects, we exemplarily divide the integration domain Ω into two subdomains Ω_1 and Ω_2. Since the transmission condition does not affect the right side of the equation, we analyze only the left side of the problem. First, we multiply with a testfunction $v \in H^1(\Omega)$ and integrate over Ω. Since the subdomains are disjoint, we can then split the integral such that we get

$$-\int_\Omega \text{div}(\sigma\nabla\Phi)\text{v}\,\text{d}x = -\int_{\Omega_1} \text{div}(\sigma_1\nabla\Phi)\text{v}\,\text{d}x - \int_{\Omega_2} \text{div}(\sigma_2\nabla\Phi)\text{v}\,\text{d}x.$$

Applying integration by parts onto both integrals leads to

$$\ldots = \int_{\Omega_1} \sigma_1\nabla\Phi\nabla\text{v}\,\text{d}x - \int_{\partial\Omega_1} \sigma_1\partial_\nu\Phi\text{v}\,\text{d}s + \int_{\Omega_2} \sigma_2\nabla\Phi\nabla\text{v}\,\text{d}x - \int_{\partial\Omega_2} \sigma_2\partial_\nu\Phi\text{v}\,\text{d}s.$$

Then, we split the boundaries $\partial\Omega_1$ and $\partial\Omega_2$ into the shared part $\partial\Omega_{12}$ resp. $\partial\Omega_{21}$ and the remaining boundary parts, denoted with $\partial\Omega_1 \backslash \partial\Omega_{12}$ and $\partial\Omega_2 \backslash \partial\Omega_{21}$. Using those splits, we get

$$\begin{aligned} \ldots = &\int_{\Omega_1} \sigma_1 \nabla\Phi\nabla\mathrm{v}\,\mathrm{d}x - \int_{\partial\Omega_1\backslash\partial\Omega_{12}} \sigma_1 \partial_\nu\Phi\mathrm{v}\,\mathrm{d}s - \int_{\partial\Omega_{12}} \sigma_1\partial_\nu\Phi\mathrm{v}\,\mathrm{d}s \\ &+ \int_{\Omega_2} \sigma_1 \nabla\Phi\nabla\mathrm{v}\,\mathrm{d}x - \int_{\partial\Omega_2\backslash\partial\Omega_{21}} \sigma_2 \partial_\nu\Phi\mathrm{v}\,\mathrm{d}s - \int_{\partial\Omega_{21}} \sigma_2\partial_\nu\Phi\mathrm{v}\,\mathrm{d}s. \end{aligned}$$

The transmission condition (2.28) implies now that the integrals over $\partial\Omega_{12}$ and $\partial\Omega_{21}$ are the same, but since the orientation of the outer normal vector is different they have a different sign. Thus, they cancel each other out, and we can compute

$$\begin{aligned} \ldots &= \int_{\Omega_1} \sigma_1 \nabla\Phi\nabla\mathrm{v}\,\mathrm{d}x - \int_{\partial\Omega_1\backslash\partial\Omega_{12}} \sigma_1 \partial_\nu\Phi\mathrm{v}\,\mathrm{d}s + \int_{\Omega_2} \sigma_1 \nabla\Phi\nabla\mathrm{v}\,\mathrm{d}x - \int_{\partial\Omega_2\backslash\partial\Omega_{21}} \sigma_2 \partial_\nu\Phi\mathrm{v}\,\mathrm{d}s \\ &= \int_{\Omega} \sigma \nabla\Phi\nabla\mathrm{v}\,\mathrm{d}x - \int_{\partial\Omega} \sigma\partial_\nu\Phi\mathrm{v}\,\mathrm{d}s. \end{aligned}$$

Applying the same method to (2.29) and inserting the remaining boundary conditions (2.27) and (2.26) yield the following bilinear form

$$\begin{aligned} a &: H^1(\Omega) \times H^1(\Omega) \to \mathbb{R} \\ a(\Phi, v) &:= \int_{\Omega} (\sigma(x)\nabla\Phi(x)) \cdot \nabla v(x)\,\mathrm{d}x + \int_{\partial\Omega_S} \mu\Phi(s)v(s)\,\mathrm{d}s. \end{aligned} \tag{2.30}$$

Due to (2.7), (2.8), and the presence of Robin boundary conditions with $\mu > 0$ on $\partial\Omega_S$, this bilinear form is $H^1(\Omega)$-elliptic by a generalized Poincare inequality, cf., e.g., [56, Lemma 2.5]. Thus, by the Lax-Milgram theorem, we obtain a continuously invertible linear operator

$$\begin{aligned} A &: H^1(\Omega) \to H^1(\Omega)^* \\ (A\Phi)(v) &:= a(\Phi, v). \end{aligned}$$

Since $H^1(\Omega)$ is reflexive, we may identify $H^1(\Omega)^{**} \cong H^1(\Omega)$ and also consider the adjoint operator $A^* : H^1(\Omega) \to H^1(\Omega)^*$ of A as $(A^*v)(\Phi) = a(\Phi, v) = (A\Phi)(v)$.

In Section 2.1, we modeled the current source density ρ_i at a time-instant t with a line measure $\rho_l(t) \in \mathcal{M}(\overline{\Omega})$. Here $\mathcal{M}(\overline{\Omega})$ is the Banach space of Radon measures on $\overline{\Omega}$, which is isomorphic to the dual $C(\overline{\Omega})^*$ of the space of continuous functions by the well known Riesz representation theorem, see [8, Theorem 5.5]. Thus, we may introduce the following weak form:

$$a(\Phi(t), v) = \int_{\overline{\Omega}} v d\rho_i(t) \quad \forall v \in C^\infty(\overline{\Omega}). \tag{2.31}$$

Since $H^1(\Omega) \not\hookrightarrow C(\overline{\Omega})$ in our 3D-setting, we cannot write (2.31) as an operator equation $A\Phi(t) = \rho_i(t)$ in $H^1(\Omega)$, and thus the Lax-Milgram theorem cannot be applied directly. Nevertheless, by an approach due to Stampacchia, see [53], solvability of (2.31) with $\Phi(t) \in W^{1,p'}(\Omega)$ for some $p' < 3/2$ can be established.

In this approach, the bilinear form (2.30) is redefined on different spaces as

$$a_p : W^{1,p'}(\Omega) \times W^{1,p}(\Omega) \to \mathbb{R}$$

with $1/p + 1/p' = 1$ for $p > 3$, implying that $W^{1,p}(\Omega) \hookrightarrow C(\overline{\Omega})$. This gives rise to the following restricted pre-dual problem for some $l \in W^{1,p'}(\Omega)^* \hookrightarrow H^1(\Omega)^*$:

$$\text{find } \psi \in W^{1,p}(\Omega) : a_p(v, \psi) = l(v) \quad \forall v \in H^1(\Omega).$$

By Lax-Milgram, this problem has a solution $\psi \in H^1(\Omega)$, and it is a question of regularity theory if ψ is an element of $W^{1,p}(\Omega)$. If this is true for all $l \in W^{1,p'}(\Omega)^*$, which is known as "maximal regularity", then the pre-dual operator

$$\begin{aligned} {}^*A_p : W^{1,p}(\Omega) &\to W^{1,p'}(\Omega)^* \\ ({}^*A_p\psi)(v) &= a_p(v, \psi) \end{aligned}$$

is an isomorphism by the open mapping theorem.

Remark 1. *More generally, it can be shown that ψ is an element of $H^1(\Omega) \cap C(\overline{\Omega})$ if $l \in W^{1,p'}(\Omega)^*$ for $p' > d/(d-1)$, $\Omega \subset \mathbb{R}^d$, cf. e.g. [31]. Then, with some additional technical effort, we can still show the solvability of (2.31), but an additional criterion is required to single out a unique solution. A detailed discussion can be found in [47].*

For simplicity, we thus impose the following assumption:

Assumption 2.2.1. *The domain Ω and its subdomains Ω_j are sufficiently regular, such that the operator ${}^*A_p : W^{1,p}(\Omega) \to W^{1,p'}(\Omega)^*$ is an isomorphism for some $p > 3$.*

Under this assumption and since Sobolev spaces are reflexive, we can conclude that the adjoint $A_p := ({}^*A_p)^*$

$$\begin{aligned} A_p : W^{1,p'}(\Omega) &\to W^{1,p}(\Omega)^* \\ (A_p\phi)(w) &:= a_p(\phi, w) \end{aligned}$$

is an isomorphism, since adjoints of isomorphisms in normed spaces are also isomorphisms. Due to the continuous and dense embedding $W^{1,p}(\Omega) \hookrightarrow C(\overline{\Omega})$ we can use the corresponding adjoint embedding $C(\overline{\Omega})^* \hookrightarrow W^{1,p}(\Omega)^*$ to regard the charge $\rho(t)$ as an element of $W^{1,p}(\Omega)^*$ for each t, and we obtain unique solvability of the operator equation:

$$A_p\Phi(t) = \rho(t)$$

Hence, a unique electric potential $\Phi(t) \in W^{1,p'}(\Omega)$ that satisfies (2.31) exists for each $\rho(t)$. Since all spaces are reflexive, we can identify the adjoint and pre-adjoint operators, i.e., $A_p^* = {}^*A_p$.

2.3 Simulating Measurements with an Adjoint Approach

With the above-derived model, the potential $\Phi(t)$ can, in principle, be computed in the whole domain for every t if $\rho(t)$ is given. However, the computational effort to do so with finite elements is too large, given that we are only interested in a certain number of measurements $y_i(t)$ at the boundary of Ω. Thus, we develop a more efficient adjoint approach to compute a desired measurement $y(t) \in \mathbb{R}$ from given $\rho(t)$.

In our setting, the potential is measured with small circular electrodes on the skin as follows

$$y(t) := B(\Phi(t)) = \frac{1}{|D|} \int_D \Phi(s,t)\, \mathrm{d}s,$$

where $D \subset \partial\Omega_S$ is the area of the electrode. The trace theorem, cf., e.g. [56, Theorem 2.1], implies that B is well defined as an element of $W^{1,p'}(\Omega)^*$.

Let $\rho(t) \in W^{1,p}(\Omega)^*$ and denote by $\Phi(t) \in W^{1,p'}(\Omega)$ the solution of

$$(A_p \Phi(t))(v) = \rho(t)(v) \quad \forall v \in W^{1,p}(\Omega).$$

Now consider the solution $\omega \in W^{1,p}(\Omega)$ of the *adjoint* problem:

$$(A_p^* \omega)(\phi) = B(\phi) \quad \forall \phi \in W^{1,p'}(\Omega). \tag{2.32}$$

which corresponds to the following Poisson problem in its weak form:

$$\int_\Omega \sigma(x) \nabla\phi(x) \cdot \nabla\omega(x)\, \mathrm{d}x + \int_{\partial\Omega_S} \mu\phi(s)\omega(s)\, \mathrm{d}s = \frac{1}{|D|} \int_D \phi(s,t)\, \mathrm{d}s \quad \forall \phi \in W^{1,p'}(\Omega). \tag{2.33}$$

Then we compute easily

$$y(t) = B(\Phi(t)) = (A_p^* \omega)(\Phi(t)) = a_p(\Phi(t), \omega) = (A_p \Phi(t))(\omega) = \rho(t)(\omega).$$

That means we can compute the potential at an electrode efficiently by evaluating

$$y(t) = \int_\Omega \omega(x)\, \mathrm{d}\rho_i(t). \tag{2.34}$$

where $\omega \in W^{1,p}(\Omega) \hookrightarrow C(\overline{\Omega})$ is the previously computed solution of the adjoint problem (2.32). We can now insert the previously derived definition of $\rho_i = \rho_l + \rho_s$ into (2.34). Additionally, we stress the dependence of the measurement y on u by including u as an

argument. Using (2.2) and (2.6) we get

$$\begin{aligned} y(u,t) &= \int_{\Omega} \omega(x) d(\tilde{\rho}_l + \tilde{\rho}_s)(t) \\ &= \int_{-t_A}^{t_A} \omega(u(\tau))\tilde{\rho}_l(u(\tau),t)|\dot{u}(\tau)| \,\mathrm{d}\tau + \sum_{\tau\in\{-t_A,0,t_A\}} \omega(u(\tau))\tilde{\rho}_s(u(\tau),t) \\ &= \int_{-t_A}^{t_A} \omega(u(\tau))\nu\tilde{\rho}_l(u(\tau),t) \,\mathrm{d}\tau + \sum_{\tau\in\{-t_A,0,t_A\}} \omega(u(\tau))\tilde{\rho}_s(u(\tau),t) \\ &= \int_{-t_A}^{t_A} \omega(u(\tau))\nu\rho_l(\tau,t) \,\mathrm{d}\tau + \sum_{\tau\in I_B} \omega(u(\tau))\rho_s(\tau,t) \end{aligned} \tag{2.35}$$

with $I_B := \{-t_A, 0, t_A\}$.

Now the computation of $y(u,t)$ requires just the evaluation of this line integral, which is much cheaper than computing the solution of an elliptic equation. Clearly, $y(u,t)$ also depends on the solution ω of (2.32) and thus on the domain D of the electrode used for the measurement. If, for $i = 1 \dots n_E$, electrodes D_i are considered, we denote the corresponding measurements by $y_i(u,t)$

For our forward problem (2.31), we may assume that the moving charge is completely contained in the muscular subdomain Ω_M, which is disjoint with the domains of measurement $D_i \subset \partial\Omega_s$. We can thus invoke regularity results to obtain more smoothness of the restriction $\omega|_{\Omega_M}$. This is useful to render sensitivities of $y(t)$ concerning perturbations of the support of $\rho_i(t)$ well defined, which in turn is needed for the later optimal control problem (3.3).

Lemma 2.3.1. *The solution ω of (2.32) is in $C^\infty(\Omega_M) \cap W^{1,p}(\Omega)$.*

Proof. By (2.33) ω is the solution of an elliptic equation subject to inhomogenous Robin boundary conditions on D but without interior source terms. Let us consider the restriction $\omega|_{\Omega_M}$ to Ω_M. We observe that $\omega|_{\Omega_M}$ satisfies a homogenous Laplace equation in the weak form on Ω_M subject to Dirichlet boundary conditions on $\partial\Omega_M$, given simply by the condition that $\omega|_{\partial\Omega_M}$ is the trace of ω on $\partial\Omega_M$. Moreover, the coefficient σ is constant on Ω_M. Such problems, however, are known to be C^∞ regular in the interior, cf., e.g., [25, Cor. 8.11]. □

We close this section with a short discussion about simulating a surface EMG measurement.

Numerical Simulation of a surface EMG Measurement To illustrate the properties of the forward problem, we perform the simulation of a surface EMG measurement using the previously established model. We consider a measurement for a single-fibered

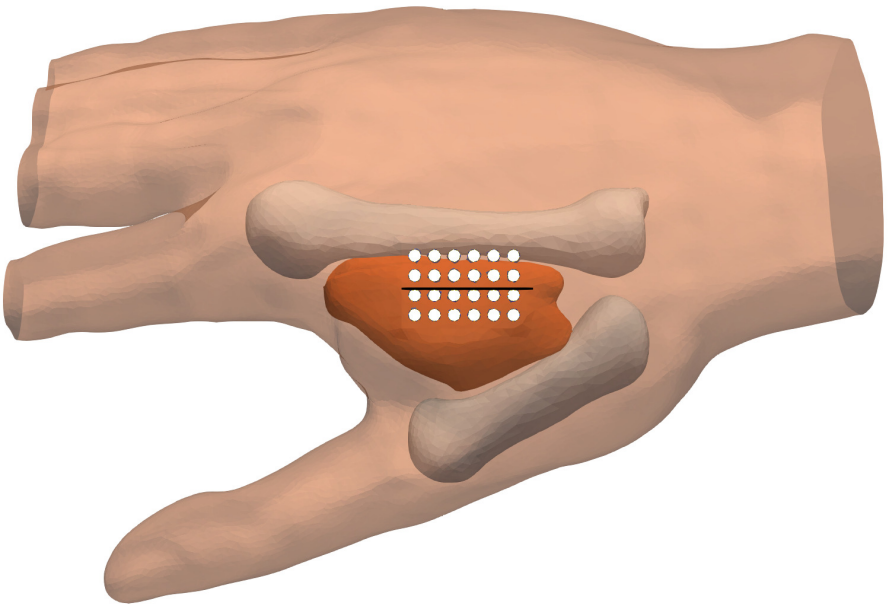


Figure 2.3: Geometrical setup for a numerical simulation: white dots: electrodes, dark brown: FDI muscle, light brown: bones and remaining tissue, black line: motor unit.

motor unit in the first dorsal interosseous (FDI) muscle of the right hand. The maximal extension of the hand is approximately 15 cm from the wrist to the fingers, 10 cm from the little finger to the thumb, and between 2 and 5 cm from the back of the hand to the front of the hand. At this point, we would like to thank the authors of [49] for sharing the STL files of their MRI measurements. Since we discuss the implementation details extensively in the chapters 5 and 6, we only summarize the key aspects of the implementation in this section.

For simplicity, our model contains only the FDI muscle and the first two metacarpal bones. The rest of the domain was modeled as fat tissue. Figure 2.3 shows the geometrical model and a grid of 24 circular electrodes (white dots) placed above the FDI muscle. The location of the motor unit is depicted by the black straight line that crosses the electrode grid horizontally. Its depth below the electrode grid is 4 mm. The motor unit is represented by a piecewise cubic Hermite polynomial on 20 subintervals, which also allows the representation of curved motor units with high accuracy.

To incorporate the electrode grid into the STL files, we used the CAD software Blender [7]. To generate a mesh from the STL data, we used gmsh [24]. We performed all the following computations in C++, where we used the toolbox Dune [5] for all mesh-related operations and the finite element toolbox Kaskade7 [27] to compute the finite element discretization of the adjoint problem.

The numerical computation of the adjoint solutions is done by a finite element method on a triangulation $\mathcal{T}$ of Ω consisting of 414195 tetrahedra. On $\mathcal{T}$ we used continuous

piecewise quadratic ansatz functions to discretize $W^{1,p}(\Omega)$ and $W^{1,p'}(\Omega)$ by

$$W_h := \{w \in C(\Omega, \mathbb{R}) : w|_K \in P_2(K)\ \forall K \in \mathcal{T}\}.$$

A Galerkin method applied to the adjoint problem (2.32) leads to the discrete problem

$$\begin{aligned} &\text{find } \omega_h \in W_h \text{ s.t.} \\ &\quad (A_p^* \omega_h)(\phi) = B(\phi) \quad \forall \phi \in W_h. \end{aligned}$$

After finite element discretization, we end up with a large sparse linear system of equations, which we must solve. We used the preconditioned conjugated gradient method from the linear algebra library Eigen [22] to solve this linear system. Since no grid hierarchy is available, we used a standard incomplete Cholesky decomposition, see [41], as a preconditioner.

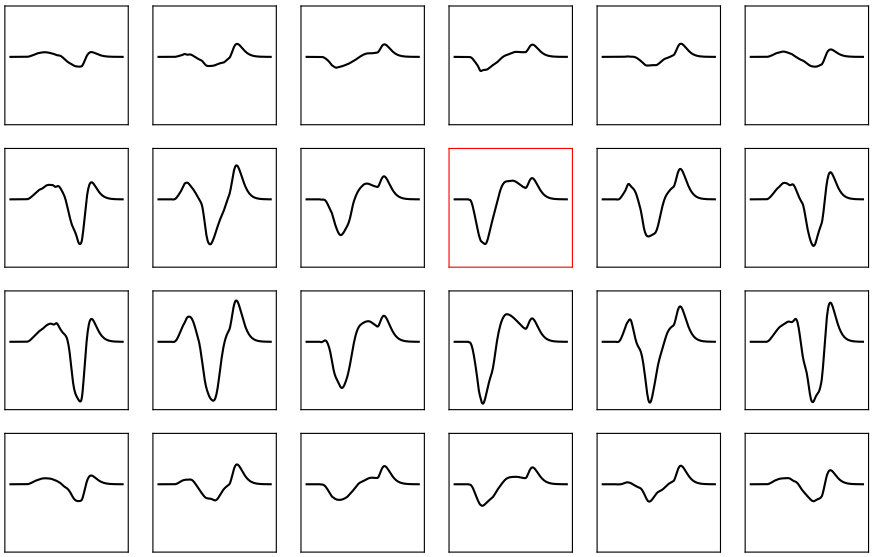
To evaluate the line integrals (2.35), we perform a numerical quadrature along the motor unit, i.e., the trajectory of u. As seen in Figure 2.2, the action potential is only nonzero on a small part of the trajectory but shows large oscillations there. Thus, a standard piecewise quadrature rule on uniform intervals would be inefficient. We, therefore, use an adaptive quadrature algorithm. Since this algorithm influences the convergence of the later introduced optimization algorithm, cf. Section 4.2, we provide a detailed discussion of the adaptive quadrature algorithm in Section 5.1.

Figure 2.4a visualizes the simulated time-dependent signal on all 24 electrodes. Depending on their location, the electrodes yield different measurements. For example, the strength of the measured signal depends on the distance between the electrodes and the source.

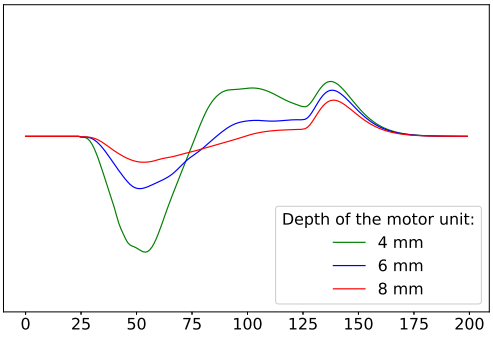
The identification of the depth of a source from boundary measurements is often difficult. We thus perform a variation in depth of the motor unit and compare the simulation result. Figure 2.4b shows the simulated measurement of one electrode (marked in red in Figure 2.4a) for motor units with different depths. The simulated measurements show that most parts of the simulated potential decrease very quickly if we increase the depth. But due to the concentrated stationary sources, the potential decreases much slower at the end of the measurement. That is a well-known effect when modeling monopolar signals, see [21, 26]. This property can affect numerical identification, which is why we will study the effect in a separate example in chapter 6.

As discussed next, our adjoint approach significantly reduces the required numerical effort per simulated measurement, at least for high temporal resolution. For example, the computation of 200 time steps by a direct approach would require the solution of 200 PDEs of the form (1.1). By an adjoint approach, we only have to solve the PDE (2.32) 24 times, which is the number of electrodes used. In this example, the numerical effort concerning the solution of PDEs is reduced by a factor of 8.

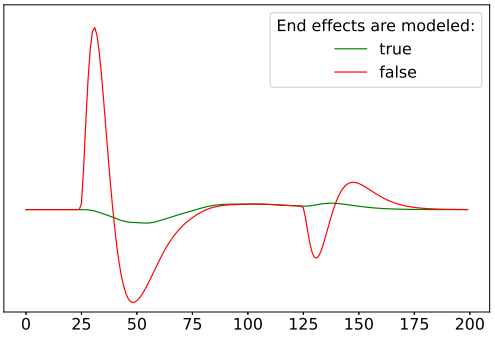
Once the weighing functions are computed, the numerical cost of the quadrature formulas needed for simulation with our adjoint approach is almost negligible. In the



(a) Simulated measurement on a 6x4 electrode grid. The geometric configuration corresponds to the one, shown in Figure 2.3. The curves show the temporal behavior of the measured potential.



(b) Impact of the motor unit depth on the simulated measurement



(c) Comparison of a surface EMG simulation with (green) and without (red) correction of end effects

Figure 2.4: Simulated EMG measurement for a single fibered motor unit for $0.2s$

above-described example, we observed the following computational times: on a standard workstation, the single-threaded computation of one PDE solution by a cg iteration requires about 7 seconds (assembly of the problem data and setup of the preconditioner not included), while the simulation of all 24 measurements with the adjoint approach required 0.3 seconds for all 200 time steps in total, and thus around 1.5 milliseconds per time step.

Certainly, these results depend on the spatial and temporal resolution of the problem. Furthermore, parallelization is possible in both the direct and the adjoint approaches. But the numbers give a clear impression of the advantages of our adjoint approach, already for a single simulation. If multiple simulations with the same geometry have to be performed, e.g., inside an optimization algorithm, the weighing functions only have to be computed once. Then the computational savings of the adjoint approach, compared to a direct simulation, are even more pronounced.

Chapter 3

An Optimization Problem to Identify a Motor Unit

This chapter discusses the optimization problem and was partially published in [64, 65]. It is an inverse problem to the forward problem described in Section 2.3 in the following sense: up to now, the motor unit was modeled as the trace of a given curve u, and we derived a model for the simulation of the measurements $y_i(u,t)$ at n_E electrodes via (2.35). From now on, we assume that measurements $y_{m,i}(t)$ are available and we are looking for a curve u, such that the corresponding simulated values $y_i(u,t)$ and the measurements $y_{m,i}(t)$ fit well, i.e. that the difference

$$z_i(u,t) := y_i(u,t) - y_{m,i}(t)$$

becomes small for all t. Collecting everything in the vectors $y(u,t), y_m(t), z(u,t) \in \mathbb{R}^{n_E}$ and using the standard Euclidean norm $\|\cdot\|_{n_E}$ on $\mathbb{R}^{n_E}$, this leads to the following least-squares type tracking term

$$J_1(u) = \frac{1}{2}\int_0^T \|z(u,t)\|_{n_E}^2 \,\mathrm{d}t = \frac{1}{2}\int_0^T \|y(u,t) - y_m(t)\|_{n_E}^2 \,\mathrm{d}t, \tag{3.1}$$

where the components of $y(u,t)$ are defined via (2.35).

We can make a rough guess of the location of the motor unit u by inspecting the given measurement y_m and the subdomains of Ω, cf., e.g., Figure 2.4a or 6.2. Thus, we can choose a reference trajectory u_{ref} (e.g., a piecewise linear curve that connects the estimated location of the neuromuscular junction and the end-plates) a priori. Furthermore, motor units are smooth in healthy tissue. These two aspects justify the following regularization term:

$$J_2(u) := \int_{-1}^{1} \frac{\alpha_1}{2}\|u(\tau) - u_{ref}(\tau)\|_2^2 + \frac{\alpha_2}{2}\|\ddot{u}(\tau)\|_2^2 \,\mathrm{d}\tau, \tag{3.2}$$

where $\|\cdot\|_2$ is the standard Euclidean norm on $\mathbb{R}^3$ (consequently $\langle\cdot,\cdot\rangle_2$ is the standard Euclidean scalar product on $\mathbb{R}^3$). This term also yields the necessary compactness to show the existence of optimal solutions.

Finally, we add a constraint ensuring that the signal passes the motor unit with constant speed $\nu > 0$, as assumed in section 2.1. Therefore, we define the constraint function

$$\begin{aligned} G : H^2(-t_A, t_A, \mathbb{R}^3) &\mapsto H^1(-t_A, t_A, \mathbb{R}) \\ [G(u)](\tau) &:= \|\dot{u}(\tau)\|_2^2 - \nu^2, \end{aligned}$$

and demand that $[G(u)](\tau) = 0$ for almost every $\tau \in [-t_A, t_A]$. We also demand that the solution is located in the muscle tissue $\overline{\Omega}_M$. Combining those two constraints, we get the following admissible set

$$U_{ad} := \{v \in H^2(-t_A, t_A, \mathbb{R}^3) | \, v(\tau) \in \overline{\Omega}_M,\ G(v)(\tau) = 0, \text{ for a. e. } \tau \in [-t_A, t_A]\}.$$

This definition of the admissible set U_{ad} is reasonable, since $H^2(-t_A, t_A, \mathbb{R}^3)$ is embedded in $C^1(-t_A, t_A, \mathbb{R}^3)$. Collecting everything, we get the optimization problem:

$$\min_{u \in U_{ad}} J(u) := J_1(u) + J_2(u). \tag{3.3}$$

Alternatively, we can write this problem as an unconstrained problem by adding an indicator function such that

$$\min_{u \in H^2(-t_A, t_A, \mathbb{R}^3)} F(u) := J(u) + \iota_{U_{ad}}(u). \tag{3.4}$$

Remark 2. *Some straightforward extensions of this identification problem are conceivable: for example, we may include the speed ν and the scaling parameter a in the set of variables we want to identify. For simplicity of presentation, we assume these parameters to be given.*

The rest of this chapter discusses the existence of solutions and derives first-order optimality conditions. Furthermore, we introduce the Lagrangian function and show some properties needed when numerically solving the problem.

3.1 Analysis of the Optimization Problem

The goal of this section is to show that the optimization problem (3.3) has at least one solution. Therefore, we first discuss the required properties and secondly, we prove the existence result.

Properties of the Objective Function

This section lays the foundation for the existence proof in the next section and consists of three auxiliary results. First, we show that the admissible set U_{ad} is weakly closed and

that the equality constraint $G(u)$ is well-defined and twice differentiable. The second result shows that the function J is twice differentiable and weakly lower continuous. Note that J and G do not need to be twice differentiable. But we want to use a Newton-like method to solve the optimization problem numerically, requiring the second derivative. Finally, we use those two results to show that objective function F is radially unbounded and weakly lower semi-continuous.

Lemma 3.1.1. *The admissible set U_{ad} is weakly closed, and the equality constraint G is well defined and twice Fréchet differentiable.*

Proof. To show that U_{ad} is weakly closed, it is sufficient to show that the sets

$$U_1 = \{v \in H^2(-t_A, t_A, \mathbb{R}^3) | \, v(\tau) \in \overline{\Omega}_M \text{ for a.e. } \tau \in [-t_A, t_A]\}$$

and

$$U_2 = \{v \in H^2(-t_A, t_A, \mathbb{R}^3) | \, G(v)(\tau) = 0 \text{ for a.e. } \tau \in [-t_A, t_A]\}$$

are weakly closed. The admissible set U_{ad} is then weakly closed as an intersection of finitely many weakly closed sets.

Let $u : [-t_A, t_A] \mapsto \overline{\Omega}_M$ be a regular curve. We know, from [1, Theorem 6.2], that there exists a compact embedding $E : H^2(-t_A, t_A, \mathbb{R}^3) \mapsto C^1(-t_A, t_A, \mathbb{R}^3)$ implying that there exists a $v \in H^2(-t_A, t_A, \mathbb{R}^3)$ such that $Ev = u$, cf., e.g., [1, Section 4.2]. Additionally, the embedding implies the existence of a continuous representative $Eu \in C^1(-t_A, t_A, \mathbb{R}^3)$ with $Eu = u$ and $\|E\dot{u}\|_{0,\infty} \leq c_E \|u\|_{2,2} < \infty$. Thus U_1 is well-defined and not empty.

Let now $\{u_k\} \subset U_1$ be a weakly convergent sequence with limit u. Since the embedding E is compact there exist a subsequence u_{k_l} such that $Eu_{k_l} \to E_1 u$ in $C^1(-t_A, t_A, \mathbb{R})$. Furthermore, there exist a another subsequence $Eu_{k_{l_i}}$ that converges pointwise to Eu for all $\tau \in [-t_A, t_A]$. We can conclude from $Eu_{k_{l_i}}(\tau) \in \overline{\Omega}_M$ and $\overline{\Omega}_M$ closed that $Eu(\tau) \in \overline{\Omega}_M$ and thus also $u(\tau)$. This shows that U_1 is weakly closed.

Using Hölder's (H.) and the Cauchy-Schwartz inequality (C.S.), we can compute

$$\begin{aligned}
\|\|\dot{u}\|_2^2\|_{1,2}^2 = &\int_{t_A}^{t_A} \|\dot{u}\|_2^4 + 4\langle \ddot{u}(\tau), \dot{u}(\tau)\rangle_2^2 \, \mathrm{d}\tau \overset{C.S.}{\leq} 4 \int_{t_A}^{t_A} \|\dot{u}(\tau)\|_2^4 + \|\ddot{u}(\tau)\|_2^2 \|\dot{u}(\tau)\|_2^2 \, \mathrm{d}\tau \\
= &4 \int_{t_A}^{t_A} \|\dot{u}(\tau)\|_2^2 \left(\|\dot{u}(\tau)\|_2^2 + \|\ddot{u}(\tau)\|_2^2 \right) \, \mathrm{d}\tau \overset{H.}{\leq} 4 \|\|\dot{u}\|_2^2\|_{0,\infty} \int_{t_A}^{t_A} \|\dot{u}(\tau)\|_2^2 + \|\ddot{u}(\tau)\|_2^2 \, \mathrm{d}\tau \\
= &4 \Big(\sup_{\tau \in [-t_A, t_A]} |\sum_{i=0}^{2} \dot{u}_i^2(\tau)| \Big) \|\dot{u}\|_{1,2}^2 \leq 12 \|\dot{u}\|_{0,\infty}^2 \|\dot{u}\|_{1,2}^2 < \infty
\end{aligned}$$

showing that $\|\dot{u}\|_2^2 \in H^1(-t_A, t_A, \mathbb{R})$. Thus,

$$\begin{aligned}
G : &H^2(-t_A, t_A, \mathbb{R}^3) \mapsto H^1(-t_A, t_A, \mathbb{R}) \\
&[G(u)](\tau) = \|\dot{u}(\tau)\|_2^2 - \nu^2
\end{aligned}$$

is well-defined.

It is well known that, as continuous bilinear form, G is twice Fréchet differentiable, see [19]. The derivatives of G are then

$$\begin{aligned} &G' : H^2(-t_A, t_A, \mathbb{R}^3) \mapsto \mathcal{L}(H^2(-t_A, t_A, \mathbb{R}^3), H^1(-t_A, t_A, \mathbb{R})) \\ &[G'(u)(v)](\tau) = \langle \dot{u}(\tau), \dot{v}(\tau) \rangle_2 \end{aligned}$$

and

$$\begin{aligned} &G'' : H^2(-t_A, t_A, \mathbb{R}^3) \mapsto \mathcal{L}\left(H^2(-t_A, t_A, \mathbb{R}^3), \mathcal{L}(H^2(-t_A, t_A, \mathbb{R}^3), H^1(-t_A, t_A, \mathbb{R}))\right) \\ &[G''(u)(v, w)](\tau) = \langle \dot{v}(\tau), \dot{w}(\tau) \rangle_2. \end{aligned}$$

Let now $\{u_k\} \subset U_2$ be a weakly convergent sequence with limit u. As above there exist a subsequence u_{k_l} such that $Eu_{k_l} \to Eu$ in $C^1(-t_A, t_A, \mathbb{R}^3)$. Since G is differentiable, it is also continuous, implying

$$0 = G(Eu_{k_l}) \to G(Eu).$$

Thus, U_2 is weakly closed, concluding the proof. □

Remark 3. *The H^2-regularization term J_2 is essential for the weak closedness of U_2. It is not hard to construct a zig-zagging sequence of trajectories u_n with $|\dot{u}_n| = \nu$ a.e. (and thus bounded in $W^{1,\infty}$), such that the weak limit $\bar{u}$ violates $|\dot{\bar{u}}| = \nu$. From a computational point of view, such a regularization term does not impose severe difficulties since u can be easily discretized in an H^2 conformal fashion by a piecewise polynomial spline which is globally in C^1.*

Lemma 3.1.2. *The function $J : H^2(-t_A, t_A, \mathbb{R}^3) \supset U_{ad} \mapsto \mathbb{R}$ is continuous and weakly lower semi-continuous. If $u(\tau) \in \Omega_M$ for all $\tau \in [-t_A, t_A]$, then J is twice Fréchet differentiable at u.*

Proof. First, we note that, as a sum, J is continuous, Fréchet differentiable and weakly lower semi-continuous, if J_1 and J_2 are continuous, Fréchet differentiable and weakly lower semi-continuous.

To show the three properties for J_1, we define, for fixed t and k, the mapping

$$\begin{aligned} \psi_k : \Omega_M \times [-t_A, t_A] &\mapsto \mathbb{R} \\ \psi_k(x, \tau) &:= \nu \rho_l(\tau, t) \omega_k(x). \end{aligned}$$

with derivative

$$\begin{aligned} \psi_k' : \Omega_M \times [-t_A, t_A] &\mapsto \mathcal{L}(\mathbb{R}^3, \mathbb{R}) \\ \psi_k'(x, \tau)(v) &:= \nu \rho_l(\tau, t) \omega_k'(x)(v) \end{aligned}$$

and

$$\begin{aligned}\psi_k'' &: \Omega_M \times [-t_A, t_A] \mapsto \mathcal{L}(\mathbb{R}^3; \mathcal{L}(\mathbb{R}^3; \mathbb{R}))\\ \psi_k''(x,\tau)(v,w) &:= \nu\rho_l(t,\tau)\omega_k''(x)(v,w).\end{aligned}$$

We know, from Lemma 2.3.1, that $\omega_k \in C^\infty(\Omega_M)$ and thus also $\omega_k', \omega_k'' \in C^\infty(\Omega_M)$. Furthermore, ρ_l is continuous as a composition of continuous functions, cf. (2.2) and (2.1). Therefore, $\psi_k(\cdot,\tau)$, $\psi_k'(\cdot,\tau)$ and $\psi_k''(\cdot,\tau)$ are continuous for all $\tau \in [-t_A, t_A]$ and ψ_k is twice Fréchet differentiable, see [60, Page 192]. The Theorems 6.3 and 6.7 in [4] imply that the corresponding superposition operator is well-defined from $C(-t_A, t_A, \mathbb{R}^3)$ into itself and twice Fréchet differentiable. Furthermore, Theorem 8.8 in [8] implies the existence of a continuous compact embedding $E : H^2(-t_A, t_A, \mathbb{R}^3) \mapsto C(-t_A, t_A, \mathbb{R}^3)$ with $Eu = u$. Therefore, the superposition operator

$$\begin{aligned}\Psi_k &: H^2(-t_A, t_A, \mathbb{R}^3) \supset U_{ad} \mapsto C(-t_A, t_A, \mathbb{R}^3)\\ \Psi_k(u)(\tau) &:= \psi_k(Eu(\tau), \tau)\end{aligned}$$

is well-defined and twice Fréchet differentiable, see [4, Theorem 6.3 and 6.7]. The Fréchet derivatives are then given by

$$\begin{aligned}\Psi_k' &: U_{ad} \mapsto \mathcal{L}\left(H^2(-t_A, t_A, \mathbb{R}^3); C(-t_A, t_A, \mathbb{R})\right)\\ \left[\Psi_k'(u)(v)\right](\tau) &= \psi_k'(Eu(\tau), \tau)(Ev(\tau))\end{aligned}$$

and

$$\begin{aligned}\Psi_k'' &: U_{ad} \mapsto \mathcal{L}\left(H^2(-1,1,\mathbb{R}^3); \mathcal{L}\left(H^2(-t_A, t_A, \mathbb{R}^3); C(-t_A, t_A, \mathbb{R})\right)\right)\\ \left[\Psi_k''(u)(v,w)\right](\tau) &= \psi_k''(Eu(\tau), \tau)(Ev(\tau), Ew(\tau)).\end{aligned}$$

The correction terms $\omega_k(u(\cdot))\rho_s(\cdot,t)$ are, with the same argumentation, well-defined and Fréchet differentiable. Thus, for all t,

$$\begin{aligned}y &: H^2(-t_A, t_A, \mathbb{R}^3) \times [0,T] \mapsto \mathbb{R}\\ y_k(u,t) &= \int_{-t_A}^{t_A} \omega_k(u(\tau))\nu\rho_l(\tau,t)\,\mathrm{d}\tau + \sum_{\tau \in I_B} \omega_k(u(\tau))\rho_s(\tau,t)\end{aligned}$$

is well-defined and twice Fréchet differentiable with derivatives

$$\begin{aligned}y_k' &: H^2(-t_A, t_A, \mathbb{R}^3) \mapsto \mathcal{L}\left(H^2(-t_A, t_A, \mathbb{R}^3); \mathbb{R}\right)\\ y_k'(u,t)(v) &= \int_{-t_A}^{t_A} \left[\omega_k'(u)(v)\right](\tau)\nu\rho_l(\tau,t)\,\mathrm{d}\tau + \sum_{\tau \in I_B} \left[\omega_k'(u)(v)\right](\tau)\rho_s(\tau,t)\end{aligned}$$

and

$$y_k'' : H^2(-t_A, t_A, \mathbb{R}^3) \mapsto \mathcal{L}\left(H^2(-t_A, t_A, \mathbb{R}^3), \mathcal{L}\left(H^2(-t_A, t_A, \mathbb{R}^3); \mathbb{R}\right)\right)$$

$$y_k''(u,t)(v,w) = \int_{-t_A}^{t_A} \left[\omega_k''(u)(v,w)\right](\tau)\nu\rho_l(\tau,t)\,\mathrm{d}\tau + \sum_{\tau \in I_B} \left[\omega_k''(u)(v,w)\right](\tau)\rho_s(\tau,t)$$

Defining the component-wise derivatives

$$\left[\omega'(u)(v)\right](\tau) := \begin{bmatrix} \left[\omega_1'(u)(v)\right](\tau) \\ \vdots \\ \left[\omega_{n_E}'(u)(v)\right](\tau) \end{bmatrix}$$

and

$$\left[\omega''(u)(v,w)\right](\tau) := \begin{bmatrix} \left[\omega_1''(u)(v,w)\right](\tau) \\ \vdots \\ \left[\omega_{n_E}''(u)(v,w)\right](\tau) \end{bmatrix}$$

implies, using the chain and sum rule, that

$$J_1 : H^2(-t_A, t_A, \mathbb{R}^3) \mapsto \mathbb{R}$$

$$J_1(u) = \frac{1}{2}\int_0^T \|z(u,t)\|_{n_E}^2\,\mathrm{d}t = \frac{1}{2}\int_0^T \|y(u,t) - y_m(t)\|_{n_E}^2\,\mathrm{d}t$$

is twice Fréchet differentiable with derivatives

$$J_1' : H^2(-t_A, t_A, \mathbb{R}^3) \mapsto \mathcal{L}(H^2(-t_A, t_A, \mathbb{R}^3), \mathbb{R})$$

$$J_1'(u)(v) = \int_0^T \langle z(u,t), z'(u,t)(v)\rangle_{n_E}\,\mathrm{d}t = \int_0^T \langle y(u,t) - y_m(t), y'(u,t)(v)\rangle_{n_E}\,\mathrm{d}t$$

and

$$J_1'' : H^2(-t_A, t_A, \mathbb{R}^3) \mapsto \mathcal{L}\left(H^2(-t_A, t_A, \mathbb{R}^3), \mathcal{L}(H^2(-t_A, t_A, \mathbb{R}^3), \mathbb{R})\right)$$

$$\begin{aligned} J_1''(u)(v,w) &= \int_0^T \langle z(u,t), z''(u,t)(v,w)\rangle_{n_E} + \langle z'(u,t)(v), z'(u,t)(w)\rangle_{n_E}\,\mathrm{d}t \\ &= \int_0^T \langle y(u,t) - y_m(t), y''(u,t)(v,w)\rangle_{n_E} + \langle y'(u,t)(v), y'(u,t)(w)\rangle_{n_E}\,\mathrm{d}t. \end{aligned}$$

Since $J_1 : C(-t_A, t_A, \mathbb{R}^3) \mapsto \mathbb{R}$ is differentiable, it is also continuous and thus lower semi-continuous in $C(-t_A, t_A, \mathbb{R}^3)$. Let $\{u_k\} \subset U_{ad}$ be a weak converging sequence with

limit u. Since the embedding E is compact, there exists a subsequence u_{k_l} such that $Eu_{k_l} \to Eu$ in $C(-t_A, t_A, \mathbb{R}^3)$. Thus,

$$\lim_{Eu_{k_l} \to Eu} J_1(Eu_{k_l}) \geq J_1(Eu)$$

showing that J_1 is weakly lower semi-continuous in $H^2(-t_A, t_A, \mathbb{R}^3)$.

J_2 is a convex quadratic bilinear form, and it is well known that they are twice Fréchet differentiable from L_2 into itself and weakly lower semi-continuous, see [19]. The derivatives of J_2 are given through

$$\begin{aligned} J_2' &: H^2(-t_A, t_A, \mathbb{R}^3) \mapsto \mathcal{L}(H^2(-t_A, t_A, \mathbb{R}^3), \mathbb{R}) \\ J_2''(u)(v) &= \int_{-t_A}^{t_A} \alpha_1 \langle u(\tau) - u_{\text{ref}}(\tau), v(\tau) \rangle_2 + \alpha_2 \langle \ddot{u}(\tau), \ddot{v}(\tau) \rangle_2 \, \mathrm{d}\tau \end{aligned}$$

and

$$\begin{aligned} J_2'' &: H^2(-t_A, t_A, \mathbb{R}^3) \mapsto \mathcal{L}\left(H^2(-t_A, t_A, \mathbb{R}^3), \mathcal{L}(H^2(-t_A, t_A, \mathbb{R}^3), \mathbb{R})\right) \\ J_2''(u)(v, w) &= \int_{-t_A}^{t_A} \alpha_1 \langle v(\tau), w(\tau) \rangle_2 \alpha_2 + \langle \ddot{v}(\tau), \ddot{w}(\tau) \rangle_2 \, \mathrm{d}\tau \end{aligned}$$

The sum rule indicates now that J is continuous and Fréchet differentiable, and since the lim inf is super-additive, J is weakly lower semi-continuous. □

Collecting the derivatives from the previous proof, we get the derivatives

$$\begin{aligned} J' &: H^2(-t_A, t_A, \mathbb{R}^3) \mapsto \mathcal{L}(H^2(-t_A, t_A, \mathbb{R}^3), \mathbb{R}) \\ J'(u)(v) &= \int_0^T \langle z(u,t), z'(u,t)(v) \rangle_{n_E} \, \mathrm{d}t + \int_{-t_A}^{t_A} \alpha_1 \langle u(\tau) - u_{\text{ref}}(\tau), v(\tau) \rangle_2 + \alpha_2 \langle \ddot{u}(\tau), \ddot{v}(\tau) \rangle_2 \, \mathrm{d}\tau \end{aligned}$$

and

$$\begin{aligned} J'' &: H^2(-t_A, t_A, \mathbb{R}^3) \mapsto \mathcal{L}\left(H^2(-t_A, t_A, \mathbb{R}^3), \mathcal{L}(H^2(-t_A, t_A, \mathbb{R}^3), \mathbb{R})\right) \\ J''(u)v &= \int_0^T \langle z(u,t), z''(u,t)(v,w) \rangle_{n_E} + \langle z'(u,t)(v), z'(u,t)(w) \rangle_{n_E} \, \mathrm{d}t \\ &\quad + \int_{-t_A}^{t_A} \alpha_1 \langle v(\tau), w(\tau) \rangle_2 \alpha_2 + \langle \ddot{v}(\tau), \ddot{w}(\tau) \rangle_2 \, \mathrm{d}\tau \end{aligned}$$

Remark 4. *Recall that the derivatives $\omega'(u)$ and $\omega''(u)$, required for computing $y'(u,t)$ resp. $y''(u,t)$, are well-defined by Lemma 2.3.1, since u is assumed to be contained in the muscular tissue Ω_M. However, if we approximate ω by a finite element function, $\omega'(u)$ and $\omega''(u)$ are only piecewise continuous.*

Lemma 3.1.3. *The objective function F is weakly lower semi-continuous and radially unbounded.*

Proof. We already know from Lemma 3.1.2 that J is weakly lower semi-continuous. It remains to show that the indicator function $\iota_{U_{ad}}$ is lower semi-continuous.
A function $f : X \mapsto \mathbb{R}$ is weakly lower semi-continuous if the level sets $N_\alpha f = \{x \in X | f(x) \leq \alpha\}$ are weakly closed for all $\alpha \in \mathbb{R}$, see [17, Theorem 7.4.11]. For the indicator function, the level sets are given through

$$N_\alpha \iota_{U_{ad}}(u) = \begin{cases} U_{ad} & \text{if } \alpha > 0 \\ \emptyset & \text{else.} \end{cases}$$

Lemma 3.1.1 states that U_{ad} is weakly closed. Furthermore, the empty set is always weakly closed. Thus we can conclude that the indicator function is weakly lower semi-continuous.

The objective function F is called radially unbounded (or weakly coercive) if

$$\lim_{\|u\|_{2,2} \to \infty} F(u) = +\infty,$$

see [61, Def. 25.16]. Obviously, $F(u) > 0$ and for $\|u\|_{2,2} \to \infty$ either J_2 or $\iota_{U_{ad}}$ goes to infinity and therefore F is radially unbounded. □

Existence of Solutions

Using the three auxiliary results from the previous section, we can prove the existence of optimal solutions by standard techniques:

Theorem 3.1.4. *The optimization problem*

$$\min_{u \in H^2(-t_A, t_A, \mathbb{R}^3)} F(u) := J(u) + \iota_{U_{ad}}(u).$$

has at least one solution $u_* \in H^2(-t_A, t_A, \mathbb{R}^3)$.

Proof. Let $\{u_n\}$ be a minimizing sequence. Since F is radially unbounded, we can conclude that u_n is bounded, see Lemma 3.1.3. Since $H^2(-t_A, t_A, \mathbb{R}^3)$ is reflexive, there exist a weakly convergent subsequence u_{n_k} with limit u_*, see [8, Theorem 3.18]. By Lemma 3.1.3 F is weakly lower semi-continuous and thus

$$\inf F \leq F(u_*) \leq \liminf_{k \to \infty} F(u_{n_k}) = \inf F,$$

which shows that the limit point u_* is a minimizer of F. □

In the next chapter, we will develop an algorithmic approach to solve the optimization problem (3.3) numerically. The foundation of this approach will be an augmented Lagrangian method, requiring the Lagrangian Function and the corresponding first-order optimality conditions. Thus, the following section analyzes these two aspects.

3.2 Analysis of the Lagrangian Function

This section defines and analyzes the augmented Lagrangian function required for the algorithmic approach presented in the next chapter. Therefore, we first derive first-order optimality conditions for the problem. And secondly, we define the augmented Lagrangian function and study some properties required for a global convergence result.

First-Order Optimality conditions

Deriving first-order optimality conditions is complicated by the geometric constraint $u \subset \overline{\Omega}_M$, which we imposed to assert the existence of solutions. However, from a practical point of view, we can expect that the optimal solution u_* is contained in Ω_M without the need to enforce this as a constraint because the measured values originate from a true signal, emitted from some trajectory $u \subset \Omega_M$. In addition, the part J_2 of the objective function penalizes the departure of u from the reference trajectory u_{ref}, which reasonably will be chosen to lie in Ω_M. Therefore, for simplicity, we will drop the geometric constraint from now and assume that the optimal solution u_* lies in the muscle domain Ω_M. Thus, we can rewrite the optimal control problem 3.3 as a pure equality-constrained problem:

$$\begin{aligned} &\min_{u \in H^2(-t_A, t_A, \mathbb{R}^3)} J(u) \\ &\text{s.t.} \quad G(u)(\tau) = 0 \quad \text{for a. e. } \tau \in [-t_A, t_A]. \end{aligned} \tag{3.5}$$

As usual, we eliminate the equality constraint with the help of a Lagrange multiplier, which leads to the following result:

Theorem 3.2.1. *Let u_* be a local minimizer of (3.5) that lies in Ω_M. Then there exist a Lagrange multiplier $\tilde{\lambda} \in H^1(-t_A, t_A, \mathbb{R})^*$, such that*

$$\begin{aligned} 0 &= J'(u_*) + G'(u_*)^* \tilde{\lambda} \\ 0 &= G(u_*)(\tau) \quad \textit{for a. e. } \tau \in [-t_A, t_A] \\ \tilde{\lambda} &\in H^1(-t_A, t_A, \mathbb{R})^* \end{aligned} \tag{3.6}$$

Proof. First, we recall that $\tilde{\lambda} \in H^1(-t_A, t_A, \mathbb{R})^*$ is called Lagrange multiplier, if

$$\tilde{\lambda} \in K^+ \tag{3.7}$$

$$\tilde{\lambda}\,(G(u_*)) = 0 \tag{3.8}$$

$$J'(u_*) - G'(u_*)^* \tilde{\lambda} \in C(u_*)^+ \tag{3.9}$$

is fulfilled, see [63, Eq. 1.1]. Here K is a convex closed cone such that $G(u) \in K$ and K^+ is the dual cone of K. Since we have pure equality constraints, $K = \{0\} \subset H^1(-t_A, t_A, \mathbb{R})$ and from the definition of the dual cone, we can conclude that $K^+ = H^1(-t_A, t_A, \mathbb{R})^*$, see [56, 63]. Thus, (3.7) is equivalent to $\tilde{\lambda} \in H^1(-t_A, t_A, \mathbb{R})^*$.

Next, $[G(u_*)](\tau) = 0$ implies that condition (3.8) is always fulfilled and thus we can replace it by $[G(u_*)](\tau) = 0$. Furthermore, $C(u_*)$ is the conical hull of $H^2(-t_A, t_A, \mathbb{R}^3)$, which is $H^2(-t_A, t_A, \mathbb{R}^3)$ implying that the dual cone $C(u_*)^+ = \{0\}$. Thus, condition (3.9) simplifies to

$$J'(u_*) - G'(u_*)^* \tilde{\lambda} = 0.$$

Since we already know from Lemma 3.1.2 that J and G are Fréchet differentiable, we can conclude from [56, Theorem 6.3] that a Lagrange multiplier $\tilde{\lambda}$ exists, if G satisfies the regularity condition of Zowe and Kurcyusz, which is given through

$$G'(u_*)C(u_*) + K(-G(u_*)) = H^1(-t_A, t_A, \mathbb{R}).$$

Here $K(x)$ is the conical hull of K at x, cf., e.g., [56, Page 330]. We can conclude from $K = \{0\}$ and $G(u_*) = 0$ that $K(G(u_*)) = \{0\}$ and since $C(u_*) = H^2(-t_A, t_A, \mathbb{R}^3)$, this condition is equivalent to $G'(u_*)$ surjective. To show that G' is surjective, we choose for arbitrary $w \in H^1(-t_A, t_A, \mathbb{R})$

$$v(\tau) = \int_{-t_A}^{\tau} \frac{\dot{u}_*(\xi) w(\xi)}{\nu^2} \, \mathrm{d}\xi,$$

which is in $H^2(-t_A, t_A, \mathbb{R}^3)$ and has the derivative

$$\dot{v}(\tau) = \frac{\dot{u}_*(\tau) w(\tau)}{\nu^2}.$$

Therefore,

$$[G'(u_*)(v)](\tau) = \langle \dot{u}_*(\tau), \frac{\dot{u}_*(\tau) w(\tau)}{\nu^2} \rangle_2 = w(\tau) \frac{\langle \dot{u}_*(\tau), \dot{u}_*(\tau) \rangle_2}{\nu^2} = w(\tau)$$

and since $w \in H^1(-t_A, t_A, \mathbb{R})$ was arbitrary, $G'(u_*)$ is surjective. □

We can conclude from the definition of an adjoint operator that $G'(u_*)^* \tilde{\lambda} = \tilde{\lambda}(G'(u_*))$ where the Lagrange multiplier $\tilde{\lambda} \in H^1(-t_a, t_a, \mathbb{R})^*$ is a linear function. Furthermore, we know from the Riesz-Fréchet representation theorem, see [8, Theorem 5.5], that there exists a $\lambda \in H^1(-1, 1, \mathbb{R})$ such that

$$\tilde{\lambda}(v) = \langle \lambda, v \rangle_{1,2} \quad \forall v \in H^1(-1, 1, \mathbb{R}).$$

Therefore, we can write, for all $\delta u \in H^2(-t_A, t_A, \mathbb{R}^3)$, the KKT condition (3.6) as

$$\begin{aligned} 0 &= \int_0^T \langle y(u,t) - y_m(t), y'(u,t)(\delta u(t))\rangle_{n_E} \,\mathrm{d}t + \langle \lambda, \langle \dot{u}, \delta\dot{u}\rangle_2\rangle_{1,2} \\ &\quad + \int_{-t_A}^{t_A} \alpha_1 \langle u(\tau) - u_{\mathrm{ref}}(\tau), \delta u(\tau)\rangle_2 + \alpha_2 \langle \ddot{u}(\tau), \delta\ddot{u}(\tau)\rangle_2 \,\mathrm{d}\tau \\ 0 &= \|\dot{u}(\tau)\|_2^2 - \nu^2 \quad \text{for a.e. } \tau \in [-t_A, t_A] \\ \lambda &\in H^1(-t_A, t_A, \mathbb{R}). \end{aligned} \tag{3.10}$$

The augmented Lagrangian Function

Previously, we eliminated the equality constraints by adding an indicator function $\iota_{U_{ad}}$, cf. (3.4). Obviously, the indicator function $\iota_{U_{ad}}$ is not twice Fréchet differentiable, restricting the selection of optimization algorithm. Alternatively, we can eliminate the equality constraints with a quadratic penalty term $\frac{\mu}{2}\langle G(u), G(u)\rangle_{1,2}$ and solve a sequence of sub-problems with increasing μ. This approach creates a sequence of solutions u_k converging to u_* for $\mu \to \infty$, see [48, Theorem 17.1]. Since the quadratic penalty term is smooth, we can use standard techniques from unconstrained optimization to solve the sub-problems. But it is well known that the second derivative of the penalty term is, for increasing μ, ill-conditioned, cf., e.g., [48, Page 505 ff.]. One possibility to reduce the ill-conditioning is to incorporate the Lagrange multiplier into the penalty term, i.e., by using

$$\begin{aligned} &J_3 : H^2(-t_A, t_A, \mathbb{R}^3) \times H^1(-1, 1, \mathbb{R}) \times \mathbb{R}^+ \mapsto \mathbb{R} \\ &J_3(u, \xi) = \langle \lambda + \frac{\mu}{2} G(u), G(u)\rangle_{1,2} \\ &= \int_{-t_A}^{t_A} \left(\lambda + \mu(\|\dot{u}(\tau)\|_2^2 - \nu^2)\right)\left(\|\dot{u}(\tau)\|_2^2 - \nu^2\right) + \left(\dot{\lambda} + \mu\langle \dot{u}(\tau), \ddot{u}(\tau)\rangle_2\right)\langle \dot{u}(\tau), \ddot{u}(\tau)\rangle_2 \,\mathrm{d}\tau, \end{aligned} \tag{3.11}$$

as penalty term, where $\xi := (\lambda, \mu) \in H^1(-1, 1, \mathbb{R}) \times \mathbb{R}^+$ is a parameter combining the Lagrange multiplier λ and the penalty parameter μ. Adding the penalty term (3.11) to the objective function J results in the so-called augmented Lagrangian Function $\mathcal{L}(u, \xi)$, and we can replace the optimization problem (3.4) with

$$\min_{u \in H^2(-1,1,\mathbb{R}^3)} \mathcal{L}(u, \xi) := J_1(u) + J_2(u) + J_3(u). \tag{3.12}$$

When discussing the algorithm used to solve (3.12), see Chapter 4, we require that $\mathcal{L}$ is twice Fréchet differentiable and that the first derivative satisfies a Lipschitz condition. Thus, we finish this section by showing these properties.

Lemma 3.2.2. *Let $u(\tau) \in \Omega_M$ for all $\tau \in [-t_A, t_A]$, then the Lagrangian function $\mathcal{L}(u, \xi)$ is, on Ω_M, twice Fréchet differentiable in u.*

Proof. We have already proved that $J_1(u)$, $J_2(u)$, and $G(u)$ are twice Fréchet differentiable, cf. Lemma 3.1.2 and Lemma 3.1.1. Since G is twice Fréchet differentiable, the chain and product rule imply that J_3 is also twice Fréchet differentiable with derivatives

$$\begin{aligned}
J_3' &: H^2(-t_A, t_A, \mathbb{R}^3) \times H^1(-1, 1, \mathbb{R}) \times \mathbb{R}^+ \mapsto \mathcal{L}(H^2(-t_A, t_A, \mathbb{R}^3), \mathbb{R}) \\
J_3'(u, \xi)(v) &= \langle \lambda + \mu G(u), G'(u)(v) \rangle_{1,2} \\
&= \int_{-t_A}^{t_A} \left(\lambda + \mu(\langle \dot{u}(\tau), \dot{u}(\tau) \rangle_2 - \nu^2) \right) \langle \dot{u}(\tau), \dot{v}(\tau) \rangle_2 \\
&\quad + \left(\dot{\lambda} + 2\mu \langle \ddot{u}(\tau), \dot{u}(\tau) \rangle_2 \right) \left(\langle \ddot{u}(\tau), \dot{v}(\tau) \rangle_2 + \langle \dot{u}(\tau), \ddot{v}(\tau) \rangle_2 \right) \mathrm{d}\tau
\end{aligned}$$

and

$$\begin{aligned}
J_3'' &: H^2(-t_A, t_A, \mathbb{R}^3) \times H^1(-1, 1, \mathbb{R}) \times \mathbb{R}^+ \\
&\quad \mapsto \mathcal{L}\left(H^2(-t_A, t_A, \mathbb{R}^3), \mathcal{L}(H^2(-t_A, t_A, \mathbb{R}^3), \mathbb{R}) \right) \\
J_3''(u, \xi)(v, w) &= \langle \lambda + \mu G(u), G''(u)(v, w) \rangle_{1,2} + \mu \langle G'(u)(v), G'(u)(w) \rangle_{1,2} \\
&= \int_{-t_A}^{t_A} \left(\lambda + \mu(\langle \dot{u}(\tau), \dot{u}(\tau) \rangle_2 - \nu^2) \right) \langle \dot{v}(\tau), \dot{w}(\tau) \rangle_2 \\
&\quad + \left(\dot{\lambda} + 2\mu \langle \ddot{u}(\tau), \dot{u}(\tau) \rangle_2 \right) \left(\langle \ddot{v}(\tau), \dot{w}(\tau) \rangle_2 + \langle \dot{v}(\tau), \ddot{w}(\tau) \rangle_2 \right) \\
&\quad + \mu \langle \dot{u}(\tau), \dot{v}(\tau) \rangle_2 \langle \dot{u}(\tau), \dot{w}(\tau) \rangle_2 \\
&\quad + \mu \left(\langle \ddot{u}(\tau), \dot{v}(\tau) \rangle_2 + \langle \dot{u}(\tau), \ddot{v}(\tau) \rangle_2 \right) \left(\langle \ddot{u}(\tau), \dot{w}(\tau) \rangle_2 + \langle \dot{u}(\tau), \ddot{w}(\tau) \rangle_2 \right) \mathrm{d}\tau
\end{aligned}$$

Finally, the sum rule implies that the augmented Lagrangian function is twice Fréchet differentiable. □

Lemma 3.2.3. *For all $u, w := u + \beta \delta u \in U_{ad}$ with $\beta < 1$ there exist a constant $L > 0$ such that the derivative of the Lagrangian function $\mathcal{L}$ satisfies the following Lipschitz condition:*

$$\left| \left[\mathcal{L}'(u, \xi) - \mathcal{L}'(w, \xi) \right] (\delta u) \right| \leq \beta L \|\delta u\|_{2,2}^2 \quad \forall u,\ w \in U_{ad}, \beta < 1. \tag{3.13}$$

Proof. First, we note that it is, using the triangle inequality, sufficient to show that J_1, J_2, and J_3 satisfy the Lipschitz condition.

The action potential i_m and its antiderivative I_m are continuous and thus bounded, i.e., there exists a constant c_I such that $|i_m(z)| < c_I$ and $|I_m(z)| < c_I$ for all $z \in \mathbb{R}$. Since the current source density ρ_l is just a shifted projection of i_m onto the motor uni u, we can conclude that also $\rho_l(\tau, t) < c_I$ for all $\tau \in [-t_A, t_A]$ and all t. Furthermore, we can conclude from (2.4) and (2.5) that $\rho_s(\cdot) < 2c_I$. We may assume, since the source density is bounded, that the measurements are bounded, i.e., there exists a constant c_M such that $\|y_m(t)\|_{n_E} < c_M$ for all t.

Lemma 2.3.1 states that $\omega_k \in C^\infty(\bar{\Omega}_M) \cap H^1(\Omega)$, implying that ω_k, ω_k', and ω_k'' are bounded on $\bar{\Omega}_M$, i.e., there exist a constant $M_\omega > 0$ such that

$$\begin{aligned} \|\omega(\underline{u})\|_{n_E} &\leq M_\omega && \forall \underline{u} \in \bar{\Omega} \\ \|\omega'(\underline{u})(\underline{v})\|_{n_E} &\leq M_\omega \|\underline{v}\|_2 && \forall \underline{u} \in \bar{\Omega} \\ \|\omega''(\underline{u})(\underline{v}, \underline{w})\|_{n_E} &\leq M_\omega \|\underline{v}\|_2 \|\underline{w}\|_2 && \forall \underline{u} \in \bar{\Omega}. \end{aligned}$$

Thus, we can conclude that ω_k and ω_k' are Lipschitz continuous, more precisely there exists a constant $L_\omega > 0$ such that

$$\begin{aligned} \|\omega(\tilde{u}) - \omega(\tilde{w})\|_{n_E} &\leq L_\omega \|\tilde{u} - \tilde{w}\|_2 && \forall \tilde{u}, \tilde{w} \in \bar{\Omega}_M. \\ \| \left[\omega'(\tilde{u}) - \omega'(\tilde{w})\right](\tilde{v})\|_{n_E} &\leq L_\omega \|\tilde{u} - \tilde{w}\|_2 \|\tilde{v}\|_2 && \forall \tilde{u}, \tilde{w} \in \bar{\Omega}_M. \end{aligned}$$

Note that we use the same constants M_ω and L_ω for the functions and the derivatives. In general, these can be different constants, but since their magnitude is irrelevant to the proof, we simplify by using the same constant. We can, using these bounds, compute

$$\begin{aligned} \|y(u,t)\|_{n_E} &= \| \int_{-t_A}^{t_A} \omega(u(\tau)) \nu \rho_l(\tau, t) \, \mathrm{d}\tau + \sum_{\tau \in I_B} \omega(u(\tau)) \rho_s(\tau, t) \|_{n_E} \\ &\leq \nu c_I \int_{-t_A}^{t_A} \|\omega(u(\tau))\|_{n_E} \, \mathrm{d}\tau + 2c_I \sum_{\tau \in I_B} \|\omega(u(\tau))\|_{n_E} \\ &\leq (2t_A \nu + 6) c_I M_\omega =: M_y \end{aligned}$$

and

$$\begin{aligned} &\|y(w,t) - y(u,t)\|_{n_E} \\ =&\| \int_{-t_A}^{t_A} \nu \rho_l(\tau, t) \left(\omega(w(\tau)) - \omega(u(\tau))\right) \, \mathrm{d}\tau + \sum_{\tau \in I_B} \left(\omega(w(\tau)) - \omega(u(\tau)\right) \rho_s(\tau, t) \|_{n_E} \\ \leq& \nu c_I \int_{-t_A}^{t_A} \|\omega(w(\tau)) - \omega(u(\tau))\|_{n_E} \, \mathrm{d}\tau + 2c_I \sum_{\tau \in I_B} \|\omega(w(\tau)) - \omega(u(\tau)\|_{n_E} \\ \leq& \nu c_I L_\omega \int_{-t_A}^{t_A} \|w(\tau) - u(\tau)\|_2 \, \mathrm{d}\tau + 2c_I L_\omega \sum_{\tau \in I_B} \|w(\tau) - u(\tau)\|_2 \\ \leq& (\nu + 6) c_I L_\omega \|w - u\|_{0,2} \end{aligned}$$

showing that $y(u,t)$ is, for all t, bounded and satisfies the Lipschitz condition with constant $L_y := (\nu + 6) c_I L_\omega$.

Since the simulation $y(u,t)$ and the measurements $y_m(t)$ are bounded, we can conclude that the difference $z(u,t) := y(u,t) - y_m(t)$ is also bounded, i.e.

$$\|z(u,t)\|_{n_E} \leq \|y(u,t)\|_{n_E} + \|y_m(t)\|_{n_E} \leq M_y + c_M = M_z.$$

Using the same bounds as above, we can similarly compute

$$\begin{aligned}\|y'(u)(\delta u)\|_{n_E} &= \| \int_{-t_A}^{t_A} \left[\omega'(u)(\delta u)\right](\tau)\nu\rho_l(\tau,t)\,\mathrm{d}\tau + \sum_{\tau\in I_B}\left[\omega'(u)(\delta u)\right](\tau)\rho_s(\tau,t)\|_{n_E} \\ &\leq \nu c_I M_\omega \int_{-t_A}^{t_A} \|\delta u(\tau)\|_2\,\mathrm{d}\tau + 2M_\omega c_I \sum_{\tau\in I_B}\|\delta u(\tau)\|_2 \\ &\leq \underbrace{(\nu+4)c_I M_\omega}_{:=M_{dy}}\|\delta u\|_{0,2} = M_{dy}\|\delta u\|_{0,2}\end{aligned}$$

and

$$\begin{aligned}&\|\left(y'(w,t)-y'(u,t)\right)(\delta u)\|_{n_E} \\ =&\| \int_{-t_A}^{t_A} \nu\rho_l(\tau,t)\left[\left(\omega'(w)-\omega'(u)\right)(\delta u)\right](\tau)\,\mathrm{d}\tau + \sum_{\tau\in I_B}\left[\left(\omega'(w)-\omega'(u)\right)(\delta u)\right](\tau)\rho_s(\tau,t)\|_{n_E} \\ \leq&\nu c_I L_\omega \int_{-t_A}^{t_A}\|w(\tau)-u(\tau)\|_2\|\delta u(\tau)\|_2\,\mathrm{d}\tau + 2c_I L_\omega\sum_{\tau\in I_B}\|w(\tau)-u(\tau)\|_2\|\delta u(\tau)\|_2 \\ \leq&(\nu+6)c_I L_\omega\beta\|\delta u\|_{0,2}^2\end{aligned}$$

showing that $y'(u,t)$ is, for all t, bounded and satisfies the Lipschitz condition with constant $L_{dy} := (\nu+6)c_I L_\omega\beta$.

Since $y(u,t)$ and $y'(u,t)$ are bounded and satisfying the Lipschitz condition, we can conclude from

$$\begin{aligned}&\left|\left[J_1'(w)-J_1'(u)\right](\delta u)\right| \\ =&\left|\int_0^T \langle y'(w,t)(\delta u), z(w,t)\rangle_{n_E} - \langle y'(u,t)(\delta u), z(u,t)\rangle_{n_E}\,\mathrm{d}t\right| \\ \leq&\int_0^T \left|\langle (y'(w,t)-y'(u,t))(\delta u), z(w,t)\rangle_{n_E} + \langle y'(u,t)(\delta u), y(w,t)-y(u,t)\rangle_{n_E}\right|\,\mathrm{d}t \\ \leq&\int_0^T \|(y'(w,t)-y'(u,t))(\delta u)\|_{n_E}\|z(w,t)\|_{n_E} + \|y'(u,t)(\delta u)\|_{n_E}\|y(w,t)-y(u,t)\|_{n_E}\,\mathrm{d}t \\ \leq&\left(M_z L_{dy} + M_{dy}L_y\right)T\|\delta u\|_{0,2}^2\end{aligned}$$

that $J_1'(t)$ satisfies the Lipschitz condition with constant $L_1 := \left(M_z L_{dy} + M_{dy}L_y\right)T$.

The simple computation

$$\begin{aligned}\left|\left[\tilde{J}_2'(w)-\tilde{J}_2'(u)\right](\delta u)\right| &\leq \frac{\alpha_1}{2}\int_{-t_A}^{t_A}|\langle w(\tau)-u_{ref}(\tau),\delta u(\tau)\rangle_2-\langle u(\tau)-u_{ref}(\tau),\delta u(\tau)\rangle_2|\,\mathrm{d}\tau\\ &\quad+\frac{\alpha_2}{2}\int_{-t_A}^{t_A}|\langle \ddot{w}(\tau),\delta\ddot{u}(\tau)\rangle_2-\langle \ddot{u}(\tau),\delta\ddot{u}(\tau)\rangle_2|\,\mathrm{d}\tau\\ &=\frac{\beta\alpha_1}{2}\int_{-t_A}^{t_A}\langle\delta u(\tau),\delta u(\tau)\rangle_2\,\mathrm{d}\tau+\frac{\beta\alpha_2}{2}\int_{-t_A}^{t_A}\langle\delta\ddot{u}(\tau),\delta\ddot{u}(\tau)\rangle_2\,\mathrm{d}\tau\\ &\leq \beta\max\{\alpha_1,\alpha_2\}\|\delta u\|_{2,2}^2\end{aligned}$$

shows that J_2 satisfies the Lipschitz condition with constant $L_2:=\max\{\alpha_1,\alpha_2\}$.

To show that J_3 satisfies the Lipschitz condition is more complicated. Thus we split the discussion into three parts.

We know that there exists a compact embedding $E:H^1(-t_A,t_A,\mathbb{R}^3)\mapsto C(-t_A,t_A,\mathbb{R}^3)$, see [1, Theorem 6.2], implying the existence of a representative $E\dot{u}\in C(-t_A,t_A,\mathbb{R}^3)$ with $E\dot{u}=\dot{u}$ and $\|E\dot{u}\|_{0,\infty}\leq c_E\|\dot{u}\|_{1,2}<\infty$, cf., e.g., [1, Section 4.2]. Using Hölder's inequality (H.), Young's inequality (Y.), and the Cauchy-Schwartz inequality (C.S.), we can compute for $v_1,v_2\in H^1(-t_A,t_A,\mathbb{R}^3)$:

$$\begin{aligned}\langle\|v_1\|_2^2,\|v_2\|_2^2\rangle_{0,2}&\overset{H.}{\leq}0.5\|\|v_1\|_2^2\|_{0,\infty}\int_{t_A}^{t_A}\|v_2(\tau)\|_2^2\,\mathrm{d}\tau+0.5\|\|v_2\|_2^2\|_{0,\infty}\int_{t_A}^{t_A}\|v_1(\tau)\|_2^2\,\mathrm{d}\tau\\ &\leq\left(\sup_{\tau\in[-t_A,t_A]}|\sum_{i=0}^{2}v_{1,i}^2(\tau)|\right)\|v_2\|_{0,2}^2+\left(\sup_{\tau\in[-t_A,t_A]}|\sum_{i=0}^{2}v_{2,i}^2(\tau)|\right)\|v_1\|_{0,2}^2\\ &\leq 3\|v_1\|_{0,\infty}^2\|v_2\|_{0,2}^2+3\|v_2\|_{0,\infty}^2\|v_1\|_{0,2}^2\end{aligned}$$

and

$$\begin{aligned}\langle\frac{d}{dt}\|v_1\|_2^2,\frac{d}{dt}\|v_2\|_2^2\rangle_{0,2}&=4\int_{-t_A}^{t_A}\langle\dot{v}_1(\tau),v_1(\tau)\rangle_2\langle\dot{v}_2(\tau),v_2(\tau)\rangle_2\,\mathrm{d}\tau\\ &\overset{C.S.}{\leq}4\int_{-t_A}^{t_A}\|\dot{v}_1(\tau)\|_2\|v_1(\tau)\|_2\|\dot{v}_2(\tau)\|_2\|v_2(\tau)\|_2\,\mathrm{d}\tau\\ &\overset{Y.}{\leq}2\int_{-t_A}^{t_A}\|\dot{v}_1(\tau)\|_2^2\|v_2(\tau)\|_2^2+\|v_1(\tau)\|_2^2\|\dot{v}_2(\tau)\|_2^2\,\mathrm{d}\tau\\ &\overset{H.}{\leq}6\|v_1\|_{0,\infty}^2\|\dot{v}_2\|_{0,2}^2+6\|v_2\|_{0,\infty}^2\|\dot{v}_1\|_{0,2}^2,\end{aligned}$$

implying

$$\langle \|v_1\|_2^2, \|v_2\|_2^2 \rangle_{0,2} \leq 6\|v_1\|_{0,\infty}^2 \|v_2\|_{1,2}^2 + 6\|v_2\|_{0,\infty}^2 \|v_1\|_{1,2}^2.$$

Using the last inequality and the embedding, we can easily compute

$$\| \left[G'(w) - G'(u)\right](\delta u)\|_{1,2}^2 = \|\beta G'(\delta u)(\delta u)\|_{1,2}^2 = \beta \| \|\delta \dot{u}\|_2^2 \|_{1,2}^2 \leq 12\beta \|\delta \dot{u}\|_{0,\infty}^2 \|\delta \dot{u}\|_{1,2}^2,$$

$$\begin{aligned}
\langle G'(u)(\delta u), G(w) - G(u) \rangle_{1,2} &= \langle \langle \dot{u}, \delta \dot{u} \rangle_2, \|\dot{w}\|_2^2 - \|\dot{u}\|_2^2 \rangle_{1,2} \\
&= \langle \langle \dot{u}, \delta \dot{u} \rangle_2, \beta \langle \dot{u}, \delta \dot{u} \rangle_2 + \beta^2 \langle \delta \dot{u}, \delta \dot{u} \rangle_2 \rangle_{1,2} \\
&\overset{\beta \leq 1}{\leq} \beta \left(\langle \langle \dot{u}, \delta \dot{u} \rangle_2, \langle \dot{u}, \delta \dot{u} \rangle_2 \rangle_{1,2} + \langle \langle \dot{u}, \delta \dot{u} \rangle_2, \langle \delta \dot{u}, \delta \dot{u} \rangle_2 \rangle_{1,2} \right) \\
&\overset{C.S.}{\leq} \beta \langle \|\dot{u}\|_2^2 + \|\dot{u}\|_2 \|\delta \dot{u}\|_2, \|\delta \dot{u}\|_2^2 \rangle_{1,2} \\
&\overset{Y.}{\leq} \beta \langle 2\|\dot{u}\|_2^2 + \|\delta \dot{u}\|_2, \|\delta \dot{u}\|_2^2 \rangle_{1,2} \\
&\leq 12\beta \left(\|\dot{u}\|_{0,\infty}^2 \|\delta \dot{u}_2\|_{1,2}^2 + \|\delta \dot{u}\|_{0,\infty}^2 \|\dot{u}\|_{1,2}^2 + \|\delta \dot{u}\|_{0,\infty}^2 \|\delta \dot{u}\|_{1,2}^2 \right) \\
&\leq 12\beta \left(2c_E \|\dot{u}\|_{1,2}^2 + \|\delta \dot{u}\|_{0,\infty}^2 \right) \|\delta \dot{u}_2\|_{1,2}^2
\end{aligned}$$

and

$$\begin{aligned}
|\langle G(w), \beta G'(\delta u)(\delta u) \rangle_{1,2}| &= |\langle \|\dot{u} + \beta \delta \dot{u}\|_2^2 - \nu^2, \beta \|\delta \dot{u}\|_2^2 \rangle_2 \rangle_{1,2}| \\
&\overset{\beta \leq 1}{\leq} \beta \left(|\langle \|\dot{u}\|_2^2 + \|\delta \dot{u}\|, \|\delta \dot{u}\|_2^2 \rangle_2| + \nu^2 \|\delta \dot{u}\|_{1,2}^2 \right) \\
&\leq 12\beta \left(2c_E \|\dot{u}\|_{1,2}^2 + \|\delta \dot{u}\|_{0,\infty}^2 \right) \|\delta \dot{u}_2\|_{1,2}^2.
\end{aligned}$$

Combining these results, we can compute

$$\begin{aligned}
&|J_3(w, \xi) - J_3(u, \xi)| = |\langle \lambda + \mu G(w), G'(w)(\delta u) \rangle_{1,2} - \langle \lambda + \mu G(u), G'(u)(\delta u) \rangle_{1,2}| \\
\leq &|\langle \lambda, \beta G'(\delta u)(\delta u) \rangle_{1,2}| + |\langle G(w), \beta G'(\delta u)(\delta u) \rangle_{1,2}| + |\mu \langle G(w) - G(u), G'(u)(\delta u) \rangle_{1,2}| \\
\overset{H}{\leq} &\|\lambda\|_{1,2}^2 \|G'(\delta u)(\delta u)\|_{1,2}^2 + 24\beta \left(2c_E \|\dot{u}\|_{1,2}^2 + \|\delta \dot{u}\|_{0,\infty}^2 \right) \|\delta \dot{u}_2\|_{1,2}^2 \\
&\leq 24\beta \left((2 + \|\lambda\|_{1,2}^2) c_E \|\dot{u}\|_{1,2}^2 + \|\delta \dot{u}\|_{0,\infty}^2 \right) \|\delta \dot{u}_2\|_{1,2}^2.
\end{aligned}$$

Thus, J_3 satisfies the condition with $L_4 = 24 \left((2 + \|\lambda\|_{1,2}^2) c_E \|\dot{u}\|_{1,2}^2 + \|\delta \dot{u}\|_{0,\infty}^2 \right)$.

Collecting all the interim results, we get

$$|\left[\mathcal{L}'(w, \xi) - \mathcal{L}'(u, \xi)\right](\delta u)| \leq \beta \underbrace{(L_1 + L_2 + L_3 + L_4)}_{=:L} \|\delta u\|_{2,2}^2$$

which concludes the proof. □

Chapter 4

A Numerical Identification Algorithm

This chapter discusses the augmented Lagrangian method we use to solve the optimal control problem numerically. Previously, we introduced the augmented Lagrangian function, which transforms the constrained problem into an unconstrained one. The augmented Lagrangian method now consists of an Newton method to solve the unconstrained problem and an update for the Lagrangian multiplier λ and the penalty parameter μ. To solve the problem numerically, we replace the integrals in the objective function with quadrature rules. As mentioned when discussing the numerical simulation, cf. Section 2.3, we use adaptive quadrature rules such that we can not rely on standard convergence results. Thus, we discuss a modified convergence result and its requirements. Parts of this chapter has been published in [65].

To distinguish between the iterates of the augmented Lagrangian method and the iterates of the Newton method, we use, from now on, $\tilde{u}_{\mathbf{k}}$ for the iterates of the augmented Lagrangian method and u_k for the iterates of the Newton method.

4.1 An Augmented Lagrangian Method

With the help of the previously defined augmented Lagrangian function $\mathcal{L}$, we can transform the unconstrained problem (3.3) into the unconstrained problem

$$\min_{u \in H^2(-t_A, t_A, \mathbb{R}^3)} \mathcal{L}(u, \xi) = J_1(u) + J_2(u) + J_3(u, \xi), \tag{4.1}$$

where J_1 is a tracking term, cf. (3.1), J_2 is a regularization term, cf. (3.2), and J_3 is a penalty term incorporating the Lagrange multiplier λ and a penalty parameter μ, cf. (3.11). Since we use fixed parameters λ and μ when discussing the Newton method, we can combine them into one penalty parameter $\xi := (\lambda, \mu)$. For increasing μ and with a suitable update strategy for λ, the solution of (4.1) converges to the solution of

the constrained problem (3.3), see [6, Proposition 4.2.2]. We can achieve this with the following algorithm:

Algorithm 1: Augmented Lagrangian Algorithm

Input: $\tilde{u}_0$, $\mu_0 = 1$, $0 < c_1 < 1$, $1 < c_2$, and $\lambda_0 \equiv 0$;
do
  $\tilde{u}_{k+1} \leftarrow$ Solve (4.1) with an Newton method;
  if $\|G(\tilde{u}_{k+1})\|_{1,2} < c_1 \|G(\tilde{u}_k)\|_{1,2}$ **then**
    update λ_{k+1} and set $\mu_{k+1} = \mu_k$;
  else
    $\mu_{k+1} = c_2 \mu_k$ and set $\lambda_{k+1} = \lambda_k$;
  end
while $\|\mathcal{L}'(\tilde{u}_{k+1}, \xi_k\|_{2,2^*} > \varepsilon_1$ *and* $\|G(\tilde{u}_{k+1})\|_{1,2} > \varepsilon_2$;
Output: $\tilde{u}_k$ and λ_k

Concerning the outer loop of the augmented Lagrangian method, it remains to define an update strategy for the Lagrangian multiplier λ_k. To define an update, we take a look at the following quadratic optimization problem

$$\min_{w \in H^1(-1,1,\mathbb{R}^3)} \frac{1}{2}\langle w, w\rangle_{1,2} + J'(\tilde{u}_k)(w) \quad \text{s.t. } G'(u_k)(w) = 0,$$

which has the following first-order optimality conditions

$$\begin{aligned} 0 &= \langle w, v\rangle_{1,2} + J'(\tilde{u}_k)(v) + \langle \lambda, G'(\tilde{u}_k)(v)\rangle_{1,2} \\ 0 &= G'(\tilde{u}_k)(w). \end{aligned} \tag{4.2}$$

Testing (4.2) with

$$v \in \ker G'(\tilde{u}_k)^\perp := \{v \in H^1(-1,1,\mathbb{R}^3) : \langle v, w\rangle_{1,2} = 0 \quad \forall w \in \ker G'(\tilde{u}_k)\}$$

transforms the first equation of (4.2) into

$$0 = J'(\tilde{u}_k)(v) + \langle \lambda, G'(\tilde{u}_k)(v)\rangle_{1,2} \quad \forall v \in \ker G'(\tilde{u}_k)^\perp.$$

If $\tilde{u}_k = u^\star$, then the solution $\bar{\lambda}$ of (4.2) is equal to the original Lagrangian multiplier $\lambda^\star$. Thus, we can update the Lagrangian multiplier by solving (4.2). Defining $\nabla_{H_1} J(\tilde{u}_k)$ by $\langle \nabla_{H_1} J(\tilde{u}_k), v\rangle_{1,2} = J'(\tilde{u}_k)v$ for all v we observe that $\bar{\lambda}$ is indeed the solution of the minimization problem

$$\min_{\lambda \in H^1} \|\nabla_{H_1} J(\tilde{u}_k) - G'(\tilde{u}_k)^* \lambda\|_{1,2}^2,$$

where $G'(\tilde{u}_k)^*$ is the Hilbert space adjoint of $G'(\tilde{u}_k)$.

Evaluation of Functionals with Adaptive Quadrature

While the outer loop of the augmented Lagrangian method is fairly standard, our particular problem structure motivates some algorithmic adjustments for the inner Newton

method solving (4.1). To apply the Newton method, we must compute the Lagrangian function $\mathcal{L}$ and its derivatives. To do this numerically, we replace the integrals in J_1, J_2, and J_3 with quadrature rules. We note that after finite element discretization, the integrands in J_2 and J_3 are piecewise polynomials of some fixed order. Thus, we can evaluate those integrals exactly with a piecewise quadrature rule of sufficient order.

It remains to compute the functional J_1, given by (3.1). We observe that the integrand of (3.1) requires the evaluation of $z(u,t)$, which in turn can only be computed by an evaluation of an integral, namely (2.35). This nested integral structure is the most challenging part of the computation.

For the outer integral of (3.1), we can use any piecewise quadrature rule as long as the segmentation of the time interval is fine enough. Furthermore, the measured values are, in general, discrete. Thus, it is suitable to choose the quadrature points of the outer integral such that they match the times at which we measured the potential. The inner integral in (2.35) is more difficult to compute. As shown in Section 2.1, the action potential is a propagating signal with only small support on the motor unit u and shows large oscillations in this area. Those oscillations make a standard piecewise quadrature rule on uniform intervals inefficient. Thus, an adaptive quadrature rule is appropriate to compute $z(u,t)$. Due to this adaptive procedure, which changes from step to step, the computed result cannot be interpreted as an exact evaluation of a discrete quantity, but as an inexact evaluation of a continuous quantity. Thus, a deliberate adjustment of accuracy requirements is necessary to avoid interference of the introduced errors with the global convergence of our algorithm. Inexact function evaluations are not uncommon in nonlinear optimization, and the topic is studied in different settings, cf., e.g., [33, 38, 9, 13, 15, 62]. Our approach resembles [9], where a Trust-Region algorithm is shown to converge globally under a condition, similar to (4.4), below. Using a sufficient error bound, cf. Lemma 4.1.1 below, we can ensure that our algorithm satisfies this condition, which we then use to modify a classical global convergence result for backtracking line search algorithms, cf., e.g., [48, Theorem 3.2].

We will discuss those two aspects in more detail when we later show the convergence of the Newton method. To indicate that J_1 and thus also $\mathcal{L}$ are computed inexactly with a quadrature rule, we use an index Q and write $J_{1,Q}$ respectively $\mathcal{L}_Q$. Since we can compute J_2 and J_3 exactly, we don't use an index for them and consequently we use $\mathcal{L}_Q = J_{1,Q} + J_2 + J_3$. The following lemma connects the accuracy of the computation of z with the relative error in the computation of differences of $\mathcal{L}$:

Lemma 4.1.1. *Let $0 < \varsigma < 1$ and assume that the following error bound for the difference $z(\cdot,t) = y(\cdot,t) - y_m(t)$ holds pointwise for every t and two trajectories u, v:*

$$\begin{aligned} &|\langle z(v,t) - z_Q(v,t), z(v,t) + z_Q(v,t)\rangle_n - \langle z(u,t) - z_Q(u,t), z(u,t) + z_Q(u,t)\rangle_n| \\ \leq &\frac{2\varsigma}{T}|\mathcal{L}_Q(v,\xi) - \mathcal{L}_Q(u,\xi)|. \end{aligned} \tag{4.3}$$

Then the Lagrangian function can be evaluated with the following accuracy estimate:

$$|\,(\mathcal{L}(v,\xi)-\mathcal{L}(u,\xi))-(\mathcal{L}_Q(v,\xi)-\mathcal{L}_Q(u,\xi))\,|\le\varsigma|\mathcal{L}_Q(v,\xi)-\mathcal{L}_Q(u,\xi)|. \tag{4.4}$$

Proof. As mentioned above, we can compute J_2 and J_3 exactly by choosing a quadrature rule of sufficient order. Therefore:

$$\begin{aligned}
&|(\mathcal{L}(v,\xi)-\mathcal{L}(u,\xi)-(\mathcal{L}_Q(v,\xi)-\mathcal{L}_Q(u,\xi))\,|=|\,(J_1(v)-J_1(u))-(J_{1,Q}(v)-J_{1,Q}(u))\,|\\
&\le\frac{1}{2}\int_0^T\left|\left(\|z(v,t)\|_{n_E}^2-\|z_Q(v,t)\|_{n_E}^2\right)-\left(\|z(u,t)\|_{n_E}^2-\|z_Q(u,t)\|_{n_E}^2\right)\right|\,\mathrm{d}t\\
&=\frac{1}{2}\int_0^T|\langle(z-z_Q)(v,t),(z+z_Q)(v,t)\rangle_n-\langle(z-z_Q)(u,t),(z+z_Q)(u,t)\rangle_n|\,\mathrm{d}t\\
&\overset{(4.3)}{\le}\varsigma|\mathcal{L}_Q(v,\xi)-\mathcal{L}_Q(u,\xi)|.
\end{aligned}$$

□

Since using adaptive quadrature has the largest effect on the inner Newton method, we study it now in detail.

4.2 Newton Line-Search with Inexact Evaluations

By an Newton method, we attempt to find a stationary point of (3.12). We will employ a Hessian modification strategy to ensure that only directions of sufficient descent are computed.

To compute a step δu_k of the Newton method, we minimize the following quadratic model of $\mathcal{L}_Q(u_k,\xi)$:

$$\min_{\delta u_k\in H^2(-1,1,\Omega)}\mathcal{L}_Q'(u_k,\xi)(\delta u)+H_Q(u_k;\Lambda)(\delta u_k,\delta u_k)$$

where H_Q is a modified hessian with respect to some $\Lambda\ge 0$:

$$H_Q(u_k;\Lambda)(v,v):=\mathcal{L}_Q''(u_k,\xi)(v,v)+\Lambda\langle v,v\rangle_{2,2}.$$

This is equivalent to solving the linear equation:

$$0=\mathcal{L}_Q'(u_k,\xi)(v)+H_Q(u_k;\Lambda)(\delta u_k,v)\quad\forall v\in H^2(-1,1,\Omega). \tag{4.5}$$

For our considerations, we require the following dual norms for linear and bilinear forms on $H^2(-1,1,\Omega)$:

$$\begin{aligned}
\|\ell\|_{2,2^*}&:=\sup_{\|v\|_{2,2}=1}|\ell(v)|\\
\|b\|_{2,2^*}&:=\sup_{\|v\|_{2,2}=1,\|w\|_{2,2}=1}|b(v,w)|
\end{aligned}$$

Lemma 4.2.1. *Assume that for some $0 < \gamma$ we have the estimate*

$$\gamma \|v\|_{2,2}^2 \leq H_Q(v;\Lambda)(v,v). \tag{4.6}$$

Then we obtain the following bound for δu_k:

$$-\mathcal{L}_Q'(u_k,\xi)(\delta u_k) \geq \gamma \|\delta u_k\|_{2,2}^2. \tag{4.7}$$

Proof.

$$\mathcal{L}_Q'(u_k,\xi)(\delta u_k) \overset{(4.5)}{=} -H_Q(u_k;\Lambda)(\delta u_k, \delta u_k) \leq -\gamma \|\delta u_k\|_{2,2}^2.$$

□

Using this lemma, we can devise a strategy for choosing Λ to enforce (4.7). Starting with $\Lambda_0 = 0$, we compute δu_k and test if (4.7) holds. If not, we use $\Lambda_1 = 0.1$ and use $\Lambda_{k+1} = 2\Lambda_k$ to create a increasing sequence of $\Lambda > 0$ until (4.7) is satisfied. Since $\|\mathcal{L}_Q''(u_k,\xi)\|_{2,2^*}$ is bounded, this loop will terminate at the latest if $\Lambda \geq \gamma + \|\mathcal{L}_Q''(u_k,\xi)\|_{2,2^*}$ and we obtain

$$\|H_Q(u_k,\Lambda)\|_{2,2^*} \leq \Lambda + \|\mathcal{L}_Q''(u_k,\xi)\|_{2,2^*}.$$

As usual, the iterate u_k is then updated via $u_{k+1} := u_k + \beta_k u_k$, where the damping factor β_k is computed by a standard back-tracking line search, which terminates if an Armijo condition of the form

$$\mathcal{L}_{Q_\beta}(u_k + \beta \delta u_k, \xi) \leq \mathcal{L}_{Q_\beta}(u_k,\xi) + \eta \mathcal{L}_{Q_\beta}'(u_k,\xi)(\beta \delta u_k) \quad \beta \in]0,1] \tag{4.8}$$

is fulfilled. Observe that (4.8) is evaluated by a possibly different quadrature rule Q_β than the quadrature rule Q, used to compute δu_k. Nevertheless, under certain conditions, we can show that the back-tracking line search terminates with some β that is above a certain lower bound:

Lemma 4.2.2. *Let $\eta \in]0,1[$, $\sigma > 0$ and assume that δu_k fulfills*

$$\mathcal{L}_{Q_\beta}'(u_k,\xi)(\delta u_k) \leq \sigma \mathcal{L}_Q'(u_k,\xi)(\delta u_k) \tag{4.9}$$

for all quadrature rules employed during the line-search back-tracking. Then there exists a $\bar{\beta}(\eta, L, \gamma, \sigma)$ such that for all $\beta < \bar{\beta}$ the Armijo condition (4.8) is fulfilled.

Proof. Since $\mathcal{L}_{Q_\beta}'(\cdot,\xi)$ satisifies a Lipschitz condition, see Lemma 3.2.3, we can compute

$$\begin{aligned}
&\mathcal{L}_{Q_\beta}(u_k + \beta\delta u_k,\xi) - \mathcal{L}_{Q_\beta}(u_k,\xi) - \mathcal{L}_{Q_\beta}'(u_k)(\beta\delta u_k) \\
&= \int_0^1 \left[\mathcal{L}_{Q_\beta}'(u_k + t\beta\delta u_k,\xi) - \mathcal{L}_{Q_\beta}'(u_k,\xi)\right](\beta\delta u_k)\,\mathrm{d}t \\
&\overset{(3.13)}{\leq} \int_0^1 tL\beta^2 \|\delta u_k\|_{2,2}^2\,\mathrm{d}t \leq \frac{L\beta^2}{2}\|\delta u_k\|_{2,2}^2.
\end{aligned}$$

Together with (4.7) and (4.9) this implies

$$\begin{aligned}
&\mathcal{L}_{Q_\beta}(u_k + \beta\delta u_k, \xi) - \mathcal{L}_{Q_\beta}(u_k, \xi) \leq \mathcal{L}'_{Q_\beta}(u_k)(\beta\delta u_k, \xi) + \frac{L\beta^2}{2}\|\delta u_k\|^2_{2,2} \\
\leq &\eta\mathcal{L}'_{Q_\beta}(u_k)(\beta\delta u_k, \xi) + (1-\eta)\sigma\mathcal{L}'_{Q}(u_k)(\beta\delta u_k, \xi) + \frac{L\beta^2}{2}\|\delta u_k\|^2_{2,2} \\
\leq &\eta\beta\mathcal{L}'_{Q_\beta}(u_k)(\delta u_k, \xi) + \beta\left(\frac{L\beta}{2} - (1-\eta)\sigma\gamma\right)\|\delta u_k\|^2_{2,2}.
\end{aligned}$$

The last term is negative if we choose $0 < \beta < \frac{2(1-\eta)\sigma\gamma}{L}$, and thus the result follows. □

Thus, a line-search algorithm (which we will elaborate on in Algorithm 6, below) can terminate with two different outcomes: either (4.8) holds eventually, or (4.9) is violated at some point. In the latter case, the step δu_k should be rejected and recomputed with tighter tolerance, because the predicted decrease $\mathcal{L}'_Q(u_k, \xi)\delta u_k$ may have been too optimistic for this direction. We can now connect these parts to the following conceptual algorithm:

Algorithm 2: Newton method with line search

Input: $u_0 = \tilde{u}_\mathbf{k}$, $tol_0 = 10^{-5}$, $\varepsilon_1 = 10^{-6}$;
do
  $\Lambda = 0$;
  do
    $\delta u_k \leftarrow$ solve (5) with Alg. 5 and tolerance tol_k ; `// see Sec. 5.3`
    increase Λ ; `// only if (4.6) is violated`
  while (4.6) *is violated*;
  $\beta_k \leftarrow$ apply line search Alg. 6, until (4.8) holds or (4.9) is violated ;
  `// see Sec. 5.3`
  if (4.9) *is violated* **then**
    tighten tol_k and set $u_{k+1} = u_k$;
  else
    $u_{k+1} = u_k + \beta_k\delta u_k$;
  end
while $\|\mathcal{L}'(u_{k+1}, \xi)\|_{2,2^*} \geq 0.9\varepsilon_1$;
Output: $\tilde{u}_{\mathbf{k}+1} = u_k$;

As usual, we want to know under which conditions Algorithm 2 converges globally. Thus, we will discuss this question in the next section.

4.3 A Global Convergence Result

To formulate a global convergence result that takes into account adaptive quadrature, let us introduce δv_k as the solution of the following equation using the exact derivative:

$$0 = \mathcal{L}'(u_k, \xi)(v) + H_Q(u_k; \Lambda)(\delta v_k, v) \quad \forall v \in H^2(-1, 1, \Omega), \tag{4.10}$$

satisfying the inequality

$$\|\mathcal{L}'(u_k,\xi)\|_{2,2^*} \leq \|H_Q(u_k;\Lambda)\|_{2,2^*}\|\delta v_k\|_{2,2} \tag{4.11}$$

We assume that the connection between δv_k and δu_k satisfies

$$\|\delta v_k\|_{2,2} \leq \Theta\|\delta u_k\|_{2,2}. \tag{4.12}$$

This can be fulfilled uniformly if the quadrature rule is kept sufficiently accurate. The main result of this section is then:

Theorem 4.3.1. *Let $0 < \gamma \leq \Gamma$, $\varsigma, \eta \in]0,1[$, $\sigma, \Theta > 0$ be fixed constants. Assume that numerical quadrature rules are used, such that in each step, the inequalities (4.4), (4.7), (4.9), and (4.12) are satisfied, and such that $\|H_Q(u_k;\Lambda)\|_{2,2^*} \leq \Gamma$.*

Then Algorithm 2 converges globally, i.e.,

$$\lim_{k\to\infty} \|\mathcal{L}'(u_k,\xi)\|_{2,2^*} = 0.$$

Proof. From Lemma 4.2.2 it follows that there exists a $\bar{\beta} > 0$ such that $\bar{\beta} < \beta_k \forall k$. Then we can compute::

$$\begin{aligned}
\mathcal{L}(u_{k+1},\xi) - \mathcal{L}(u_k,\xi) &= \mathcal{L}(u_{k+1},\xi) - \mathcal{L}(u_k,\xi) - (\mathcal{L}_{Q_{\beta_k}}(u_{k+1},\xi) - \mathcal{L}_{Q_{\beta_k}}(u_k,\xi)) \\
&\quad + (\mathcal{L}_{Q_{\beta_k}}(u_{k+1},\xi) - \mathcal{L}_{Q_{\beta_k}}(u_k,\xi)) \\
&\overset{(4.4)}{\leq} (1-\varsigma)\left(\mathcal{L}_{Q_{\beta_k}}(u_{k+1},\xi) - \mathcal{L}_{Q_{\beta_k}}(u_k,\xi)\right) \\
&\overset{(4.8)}{\leq} (1-\varsigma)\eta\beta_k\mathcal{L}'_{Q_{\beta_k}}(u_k,\xi)\delta u_k \\
&\overset{(4.9)}{\leq} (1-\varsigma)\eta\sigma\beta_k\mathcal{L}'_{Q}(u_k,\xi)\delta u_k \\
&\overset{(4.7)}{\leq} -(1-\varsigma)\eta\sigma\bar{\beta}\gamma\|\delta u_k\|_{2,2}^2 \leq 0.
\end{aligned}$$

Since G is continuous, see Lemma 3.1.1, and $u_k(\tau) \in U_{ad}$ for all k and all τ, it follows that $\|G(u_k)\|_{1,2}$ is bounded and thus $\mathcal{L}(u_k,\xi)$ is bounded from below, more precisely

$$\begin{aligned}
\mathcal{L}(u_k,\xi) &= \underbrace{J(u_k)}_{\geq 0} + \underbrace{\langle\lambda, G(u_k)\rangle_{1,2}}_{\geq -\|\lambda\|_{1,2}\|G(u_k)\|_{1,2}} + \underbrace{\frac{\mu}{2}\langle G(u_k), G(u_k)\rangle_{1,2}}_{\geq 0} \\
&\geq -\|\lambda\|_{1,2}\|G(u_k)\|_{1,2} =: \underline{\mathcal{L}}.
\end{aligned}$$

Therefore, $\sum_{k=0}^{\infty} \mathcal{L}(u_{k+1},\xi) - \mathcal{L}(u_k,\xi) \geq \underline{\mathcal{L}} - \mathcal{L}(u_0,\xi)$ implying $|\mathcal{L}(u_{k+1},\xi) - \mathcal{L}(u_k,\xi)| \to 0$ and we can conclude that $\|\delta u_k\|_{2,2^*} \to 0$. Finally (4.11) and (4.12) imply:

$$0 \leq \|\mathcal{L}'(u_k,\xi)\|_{2,2^*} \leq \Gamma\|\delta v_k\|_{2,2} \leq \Gamma\Theta\|\delta u_k\|_{2,2} \to 0.$$

□

The above-stated result guarantees only the global convergence, but that doesn't include the local convergence of the Newton method. Thus, we add a second error bound, which we explain next.

Local Convergence

It is well known that the local convergence rate of an SQP method depends on accurate search directions, e.g., the better the preconditioner approximates the Hessian of the objective function, the faster the SQP method converges locally. For our problem, the search direction additionally depends on the accuracy of the adaptive quadrature. Thus, we want that the difference $\delta e_k := \delta u_k - \delta v_k$ satisfies

$$\|\delta e_k\| = \|\delta u_k - \delta v_k\|_H \leq \epsilon \|\delta v_k\|_H \quad \forall k, \tag{4.13}$$

where $\|\cdot\|_H$ is the following energy norm:

$$\|v\|_H := H(u_k, \Lambda)(v, v) \quad \forall v \in H^2(-t_A, t_A, \Omega_M).$$

Note that, by the choice of Λ, cf. Lemma 4.2.1, H is positive definite such that $\|\cdot\|$ is in fact a norm. Furthermore, ϵ is a moderate tolerance, e.g., $\epsilon = 10^{-2}$. Unfortunately, we can not include this directly in the adaptive quadrature algorithm. But the following result gives us an applicable condition.

Lemma 4.3.2. *Assume that the error bound*

$$\left| \langle z(u_k, t), z'(u_k, t)(\delta e_k) \rangle_{n_E} - \langle z_Q(u_k, t), z_Q'(u_k, t)(\delta e_k) \rangle_{n_E} \right| \leq \frac{\epsilon^2}{T} \|\delta v_k\|_H^2 \tag{4.14}$$

is satisfied for all $t \in [0, T]$. Then $\|\delta e_k\| = \|\delta u_k - \delta v_k\|_H \leq \epsilon \|\delta v_k\|_H$.

Proof. With (4.5) and (4.10) we compute

$$\begin{aligned} \|\delta e_k\|_H^2 &= H(u_k; \Lambda)(\delta e_k, \delta e_k) = \left| \left[J_1'(u_k) - J_{1,Q}'(u_k) \right] (\delta e_k) \right| \\ &\leq \int_0^T \left| \langle z(u_k, t), z'(u_k, t)(\delta e_k) \rangle_{n_E} - \langle z_Q(u_k, t), z_Q'(u_k, t)(\delta e_k) \rangle_{n_E} \right| \, \mathrm{d}t \leq \epsilon^2 \|\delta v_k\|_H^2. \end{aligned}$$

Taking the square root on both sides provides the result. □

Remark 5. *If we assume that the energy norm $\|\cdot\|_H$ and H^2-norm are equivalent, i.e. there exist $0 < \gamma \leq \Gamma$ such that $\gamma \|v\|_{2,2} \leq \|v\|_H \leq \Gamma \|v\|_{2,2}$, then (4.13) implies (4.12).*

It is well known that, under certain conditions, Newton methods converge locally quadratic or superlinear, see [16]. However, we can not expect this for our problem structure. Although the theoretical result 2.3.1 states that $\omega_k \in C^\infty(\Omega_M) \cap W^{1,p}$, we must compute them numerically with finite elements and thus the discrete version $\omega_{k,h}$ satisfies this property only locally on each tetrahedron. Thus, small jumps occur in the first

derivative at the boundary between two adjacent tetrahedra, and the regularity requirements for neither quadratic nor superlinear convergence are satisfied. But under specific conditions, these discontinuities are no longer a factor. If we can identify the motor unit exactly and if there are no measure errors, i.e., no white noise, then the difference $z(u_k, t) = y(u_k, t) - y_m(t)$ goes to zero for $u_k \to u_*$. Even though this situation is practically impossible, we will see this theoretical consideration in one of our examples.

Now that we have proven that the inner Newton method converges globally and have derived a measure for fast local convergence, we discuss how we implemented the augmented Lagrangian method numerically.

Chapter 5

A Practical Optimization Algorithm

This section discusses the implementation of our algorithmic concept, described above. We thereby split the discussion into four parts. First, we discuss the discretization and the evaluation of the required quantities. The second part discusses the adaptive computation of the Lagrange function, i.e., how we incorporate the error bound into an adaptive quadrature algorithm. Furthermore, this part explains the computation of the Lagrange function $\mathcal{L}$ and its derivatives. Thirdly, we discuss the implementation of the Newton method. Therefore, we first explain how we transform the linear equations (4.5) and (4.10) into systems of linear equations and how we solve them. Secondly, we discuss the computation of the search directions, and thirdly the details of the line search. We then close this chapter by discussing the augmented Lagrangian method, i.e., the implementation of the Lagrange multiplier update. Parts of this chapter has been published in [65].

5.1 Discretization of the Problem

To solve the problem numerically, we must discretize and evaluate the required quantities. This includes the discretization of the motor unit u and the computation and evaluation of the impulse response functions $\omega(u)$. For this purpose, we use two different finite element approaches. Thus, we first discuss the usage of cubic Hermite ansatz functions to discretize the motor units u. And secondly, we discuss a standard finite element approach to discretize the impulse response functions ω_k. Since the evaluation of ω_k will be a bottleneck of our algorithm, we discuss in detail how we can use an efficient neighborhood search to evaluate ω_k.

Discretization of Signal Trajectories To discretize u, we first choose a segmentation $\mathcal{T}_{FE}$ of the interval $[-t_A, t_A]$. Therefore, we divide the interval into n subintervals

with length $h = 2t_a/n$. The segmentation $\mathcal{T}_{FE}$ contains then $n+1$ uniformly distributed segmentation points τ_j. We then employ cubic Hermite finite elements on $\mathcal{T}_{FE}$ to discretize $u \in H^2(-t_A, t_A, \Omega)$ in a conformal way, see [30, Section 8.6.2]. We thus obtain the following ansatz space:

$$V_h := \{v \in C^1(-t_A, t_A, \Omega) : v|_I \in P_3(I)\ \forall I \in \mathcal{T}_{FE}\} \subset H^2(-t_A, t_A, \Omega_M),$$

with basis $\{\varphi_0, ..., \varphi_m\}$ to be defined in more detail. The foundation of the basis functions φ_i are the following Hermite basis functions

$$\left.\begin{aligned} b_i(\tau) &= \frac{(h - |\tau - \tau_i|)^2(h + 2|\tau - \tau_i|)}{h^3} \\ b_{n+1+i}(\tau) &= \frac{(h - |\tau - \tau_i|)^2(\tau - \tau_i)}{h^2} \end{aligned}\right\} \quad \text{if } \tau \in [\tau_{i-1}, \tau_{i+1}]$$
$$b_{n+1+i}(\tau) = b_i(\tau) = 0 \qquad \text{if } \tau \notin [\tau_{i-1}, \tau_{i+1}] \Bigg\} \text{ for } 0 < i < n. \tag{5.1}$$

Now, we must consider that the image of functions $u \in V_h$ is three-dimensional. Therefore, we multiply b_i with the unit vectors $e_l \in \mathbb{R}^3$, and define

$$\varphi_{3i+l}(\tau) = e_l b_i(\tau) \quad \text{for } i = 0, ..., 2n+1 \text{ and } l = 0, 1, 2,$$

implying that $m = 6n + 5$. We can then write each $u(\tau) \in V_h$ as

$$u(\tau) := \sum_{i=0}^{m} \underline{u}_i \varphi_i(\tau). \tag{5.2}$$

As usual, $\underline{u} = (\underline{u}_0 \cdots \underline{u}_m)^T \in \mathbb{R}^m$ is then a discrete representative of $u \in V_h$. Similar to the well-known Lagrange basis functions, the Hermit basis functions have, for $0 \le i, j \le n$ and $i \neq j$, the following properties:

$$\begin{aligned} b_i(\tau_i) = 1, \qquad \dot{b}_i(\tau_i) = 0, \qquad b_i(\tau_j) = \dot{b}_i(\tau_j) = 0, \\ b_{n+1+i}(\tau_i) = 0, \quad \dot{b}_{n+1+i}(\tau_i) = 1, \quad b_{n+1+i}(\tau_j) = \dot{b}_{n+1+i}(\tau_j) = 0. \end{aligned}$$

Thus, we get for $j = 0, ..., n$:

$$u(\tau_j) = \sum_{l=0}^{2} \underline{u}_{3j+l} \varphi_{3j+l}(\tau_j) = \begin{pmatrix} \underline{u}_{3j} \\ \underline{u}_{3j+1} \\ \underline{u}_{3j+2} \end{pmatrix}$$

and

$$\dot{u}(\tau_j) = \sum_{l=3n+3}^{3n+5} \underline{u}_{3j+l} \varphi_{3j+l}(\tau_j) = \begin{pmatrix} \underline{u}_{3(n+1+j)} \\ \underline{u}_{3(n+1+j)+1} \\ \underline{u}_{3(n+1+j)+2} \end{pmatrix}.$$

Furthermore, the support of the basis functions consists of only two neighboring subintervals, cf. (5.1). Thus, we can efficiently compute

$$u(\tau) = \sum_{l=0}^{5} \underline{u}_{3i+l} \varphi_{3i+l}(\tau) + \sum_{l=l=3n+3}^{l=3n+8} \underline{u}_{3i+l} \varphi_{3i+l}(\tau)$$

for $\tau \in [\tau_i, \tau_{i+1}]$ and $0 \le i \le n-1$.

Discretization of Impulse Response Functions To compute J_1, we must evaluate the impulse response functions ω_k and their derivatives along the trajectory. These evaluations are well defined in the continuous setting since $\omega_k \in C^\infty(\Omega_M) \cap W^{1,p}(\Omega)$, see Lemma 2.3.1. The impulse response functions do not change during the optimization, so we can compute them a-priory, using finite elements on Ω. To this end, we use standard Lagrange elements on a triangulation $\mathcal{K}$ of Ω. On $\mathcal{K}$, we use continuous piecewise polynomial ansatz functions to discretize $W^{1,p}(\Omega)$ and $W^{1,p'}(\Omega)$ by

$$W_h := \{w_h \in C(\Omega, \mathbb{R}) : w_h|_K \in P_m(K) \ \forall K \in \mathcal{K}\}.$$

For the computation of $H_Q(u_k; \Lambda)$, we need the second derivatives of the $\omega_{k,h}$. Therefore, we must use at least polynomials of order $m = 2$ or alternatively $m = 3$ as ansatz functions.

Concerning the triangulation, there are two domains of interest where we want the mesh to be sufficiently fine. These two regions are the domain where the electrodes are located and the muscle tissue. Due to the circular shape of the electrodes and a possible curved skin, the mesh size is possibly already quite large after modeling the electrodes. Therefore, it can be more efficient to increase the polynomial order instead of refining the triangulation in the muscle tissue domain. Clearly, we have to find a good compromise between the mesh size and the used polynomial order. In the next chapter, we will discuss this in more detail when presenting a numerical example. Consequently, the derived mesh has, in general, no hierarchical structure.

We can then compute approximations $\omega_{k,h}$ for each ω_k with a standard Galerkin method, applied to the adjoint problem (2.30), see, e.g., [30, Chapter 8.2]. This leads to the following discrete problem

$$\begin{aligned} &\text{find } \omega_{k,h} \in W_h \text{ s.t.} \\ &\quad (A_p^* \omega_{k,h})(\phi_h) = B_k(\phi_h) \quad \forall \phi_h \in W_h \end{aligned}$$

with

$$(A_p^* \omega_{k,h})(\phi_h) = \int_\Omega (\sigma(x) \nabla \phi_h(x)) \cdot \nabla \omega_{k,h}(x) \, \mathrm{d}x + \int_{\partial\Omega_S} \mu \phi_h(s) \omega_{k,h}(s) \, \mathrm{d}s$$

and

$$B_k(\phi_h) = \frac{1}{|D_k|} \int_{D_k} \phi_h(s) \, \mathrm{d}s,$$

where $D_k \subset \partial\Omega_S$ is the domain of the k^{th} electrode. Let now $\{\phi_{h,0}, \ ..., \ \phi_{h,N}\}$ be a basis of W_h. As usual, when using a Galerkin method, we can represent $\omega_{k,h}$ by

$$\omega_{k,h}(x) = \sum_{i=0}^{N} \underline{\omega}_{k,i} \phi_{h,i}(x),$$

where $\underline{\omega}_k := (\underline{\omega}_{k,0} \dots \underline{\omega}_{k,0}) \in \mathbb{R}^N$ is the discrete representative of $\omega_{k,h} \in W_h$.

Using the linearity of A_p^*, we can compute

$$\begin{aligned}(A_p^* \omega_{k,h})(\phi_{h,j}) &= \int_\Omega (\sigma(x) \nabla \phi_{h,j}(x)) \cdot \sum_{i=0}^N \underline{\omega}_{k,i} \nabla \phi_{h,i}(x) \,\mathrm{d}x + \int_{\partial\Omega_S} \mu \phi_{h,j}(s) \sum_{i=0}^N \underline{\omega}_{k,i} \phi_{h,i}(s) \,\mathrm{d}s \\ &= \sum_{i=0}^N \underline{\omega}_{k,i} \underbrace{\left(\int_\Omega (\sigma(x) \nabla \phi_{h,j}(x)) \cdot \nabla \phi_{h,i}(x) \,\mathrm{d}x + \int_{\partial\Omega_S} \mu \phi_{h,j}(s) \phi_{h,i}(s) \,\mathrm{d}s \right)}_{(:= \underline{A}_p^*)_{ij}}.\end{aligned}$$

Furthermore, it is sufficient to test with all basis functions $\phi_{h,i}$ such that we end up with the following linear system of equations:

$$\underline{A}_p^* \underline{\omega}_k = \underline{B}_k \tag{5.3}$$

where $(\underline{B}_k)_j = B_k(\phi_{h,j})$. To solve (5.3), we use a standard preconditioned conjugate gradient method, see, e.g., [48, Chapter 5]. We used the version provided by Eigen, see [22], and since there is, in general, no grid hierarchy available, we use a standard incomplete Cholesky decomposition, see [41], as a preconditioner.

Evaluation of $\omega_{k,h}$ Since we use an adaptive quadrature to compute z, see below, we have to evaluate the finite element approximations $\omega_{k,h}$ of the impulse response functions at all quadrature points. For a given quadrature point $x_i = u(\tau_i)$, the evaluation of $\omega_{k,h}(u(\tau_i))$ is

$$\omega_{k,h}(u(\tau_i)) = \sum_{\mathrm{supp}(\phi_{h,i}) \subset K_i} \phi_{h,i}(x_i) \underline{\omega}_{k,i},$$

where K_i is the tetrahedron for which $x_i \in K_i$. In the absence of any additional information, identifying K_i for a given x_i is, in general, numerically expensive. In that case, a full search through all tetrahedra of the grid may be necessary to find K_i

To find K_i efficiently, we exploit that the quadrature points are ordered along the trajectory of u. Thus, we can use a neighborhood search: if $K_{i-1} \ni x_{i-1}$ is known, and x_i is the next quadrature point, we test all neighbors of K_{i-1} if they contain x_i. If that fails, we compute the distance between the center of the neighbors and the quadrature point x_i. We then choose the neighbor with the lowest distance to the quadrature point and test its neighbors. We repeat this procedure until we find the tetrahedron that contains x_i or a maximal number of tetrahedra has been tested. In the latter outcome, or if an initial inclusion $x_{i-1} \in K_{i-1}$ is not known, we fall back to a full search (or hierarchic search if possible) over the whole grid.

Using an adaptive quadrature to compute $z(u_k, t)$ for several time instances t requires the evaluation $\omega_{k,h}$ along the same trajectory. Thus, we stored the tetrahedron containing the point $u_k(-t_A)$ such that the initial inclusion is mostly known. Furthermore, this

has the advantage that in most cases a neighborhood search is sufficient to update the initial inclusion after updating u_k during the Newton iteration.

5.2 Adaptive Computation of the Lagrange Function

This section discusses the computation of $\mathcal{L}(u_k, \xi_k)$ at an iterate u_k. Due to the problem structure, we can split this computation into two parts. On the one hand, we have quantities that do not require adaptive quadrature, namely J_2 and J_3. On the other hand, we have J_1 depending on the computation of the simulated measurements $z(u_k, t)$ via adaptive quadrature of the integral (2.35), based on Gauß-Kronrod quadrature rules [39]. Thus, we first discuss the adaptive quadrature algorithm with the required modifications made to incorporate the error bounds. The second part of this section discusses the computation J_1, J_2, and J_3.

The Adaptive Quadrature Algorithm In section 4.2, we have defined the two error bounds (4.3) and (4.14) to ensure that the Newton converges both globally and locally. These two bounds need to be met by the computation of $z(u_k, t)$ via adaptive quadrature, which has to be performed for each electrode and for each time instance where we simulate a measurement. Thus, most of the computation time is spent during the quadrature algorithm. The algorithmic framework we present is thereby pretty standard, and similar versions can be found, e.g., in [12, 50].

The two error bounds require slightly different versions of our adaptive quadrature algorithm. However, their general structure stays the same. Thus, we will first explain the general algorithmic approach for an arbitrary global error bound $\mathbf{E}$ that has to satisfy a given global tolerance *tol* and then discuss the specific details when we talk about the computation of the step and the line search. The careful reader will notice that the two error bounds require different quantities, i.e., we have to evaluate $z(u_{k+1}, t)$ for (4.3) and $z'(u_k, t)(\delta u_k - \delta v_k)$ for (4.14). Mentioning this differentiation every time the algorithm evaluates $z(u_k, t)$, the description of the algorithm would become unnecessarily complicated. Thus, when writing that the algorithm computes $z(u_k, t)$, we imply that we also compute the corresponding second value.

Adaptive quadrature uses a partition of the domain of integration, which is gradually refined with the help of error indicators. Consider a given partition $\mathcal{T}$ of the integration interval $[-t_A, t_A]$, which consists of $|\mathcal{T}|$ sub-intervals. We start with computing the contribution to z on each sub-interval $I \in \mathcal{T}$, once with the Gauß rule (z_{G_I}) and once with the corresponding Kronrod rule (z_{K_I}). Simultaneously, we can evaluate local error estimates $\mathbf{E}_I$, which are the portions of the global error on the interval I. Using these quantities, we then compute $z_K(u,t) = \sum_{I\in\mathcal{T}} z_{K_I}(u,t)$, $z_G(u,t) = \sum_{I\in\mathcal{T}} z_{G_I}(u,t)$, and a global error estimate $\mathbf{E}$ (to be specified below). If $\mathbf{E} > tol$ is violated, we choose $\mathcal{T}_R \subset \mathcal{T}$ such that $\mathcal{T}_R$ is the smallest subset with

$$\sum_{I\in\mathcal{T}_R} \mathbf{E}_I \geq \chi \sum_{I\in\mathcal{T}} \mathbf{E}_I. \tag{5.4}$$

Here, χ is a parameter that shall ensure that $\mathcal{T}_R$ is not too small, i.e., such that the adaptive quadrature does not need too many iterations. To find the smallest subset $\mathcal{T}_R$, we sort the local error estimates $\mathbf{E}_I$ by value and choose those with the highest magnitude. In numerical practice, it turned out that it is a good strategy to choose χ relatively large, e.g., $\chi = 0.75$. However, frequently the errors are concentrated in only a few intervals. Thus, we choose a safety parameter ε_S to avoid the unnecessary refinement of too many intervals with very small contributions, i.e., an interval I is not refined if the corresponding local error estimate $\mathbf{E}_I < \varepsilon_S$. One possible choice is $\varepsilon_S = tol/|\mathcal{T}|$, which corresponds to the magnitude of an average error distribution.

Next, we refine all intervals in $\mathcal{T}_R$ and reevaluate the integral on this new partition. Since function evaluations are, in general, expensive, we store the local values $z_{G_I}(u,t)$, $z_{K_I}(u,t)$, and $\mathbf{E}_I$ and only recompute the local values on the refined intervals. Additionally to z_G and z_K, the algorithm delivers a new segmentation $\mathcal{T}$ of $[-t_A, t_A]$, which we can use to compute $z'(u_k,t)$ and $z''(u_k,t)$. This results in the following algorithm:

Algorithm 3: Adaptive Quadrature

Input: u_k, $\mathcal{T}_0$, tol ;
Variable Input: u_{k+1} or δe_k ; `// depends on the choice of` $\mathbf{E}$
Set $\mathcal{T}_R = \mathcal{T}_0$;
while $|\mathcal{T}_j| < maxIntervals$ **do**
  compute z_{K_I}, z_{G_I}, and $\mathbf{E}_I$ for all $I \in \mathcal{T}_R$;
  compute z_K, z_G and $\mathbf{E}$; `// possibilities for` $\mathbf{E}$`, see` (5.6) `and` (5.5)
  if $\mathbf{E} < tol$ **then**
    break ;
  else
    choose $\mathcal{T}_R \subset \mathcal{T}_j$ such that (5.4) is satisfied ;
    $\mathcal{T}_{j+1} \leftarrow$ refine all intervals in $\mathcal{T}_R$;
  end
end
Output: z_K, z_G, and $\mathcal{T}_j$;

Computation of $\mathcal{L}(u,\xi)$ The last quantity to compute is the Lagrange functional $\mathcal{L}(u,\xi)$ and its derivatives. If we inspect the terms in $\mathcal{L}$, we notice that J_2, J_3, and the regularization term of H are only compositions of parts of the $H^2(-t_A, t_A, \mathbb{R}^3)$ scalar product. The only exception is J_1, such that we treat the computation of J_1 separately.

First, we recall that u is, after finite element discretization, locally a polynomial of order three. Consequently, the first and second derivatives are then polynomials of order two respectively one. Using the finite element discretization of u, cf. (5.2), we can compute

$$\langle u(\tau), u(\tau)\rangle_2 = \langle \sum_{i=0}^{m} \underline{u}_i \varphi_i(\tau), \sum_{i=0}^{m} \underline{u}_i \varphi_i(\tau)\rangle_2 = \sum_{i=0}^{m}\sum_{j=0}^{m} (\underline{u}_i \underline{u}_j)\langle \varphi_i(\tau), \varphi_j(\tau)\rangle_2,$$

$$\langle \dot{u}(\tau), \dot{u}(\tau) \rangle_2 = \langle \sum_{i=0}^{m} \underline{u}_i \dot{\varphi}_i(\tau), \sum_{i=0}^{m} \underline{u}_i \dot{\varphi}_i(\tau) \rangle_2 = \sum_{i=0}^{m} \sum_{j=0}^{m} (\underline{u}_i \underline{u}_j) \langle \dot{\varphi}_i(\tau), \dot{\varphi}_j(\tau) \rangle_2,$$

and

$$\langle u(\tau), u(\tau) \rangle_2 = \langle \sum_{i=0}^{m} \underline{u}_i \ddot{\varphi}_i(\tau), \sum_{i=0}^{m} \underline{u}_i \ddot{\varphi}_i(\tau) \rangle_2 = \sum_{i=0}^{m} \sum_{j=0}^{m} (\underline{u}_i \underline{u}_j) \langle \ddot{\varphi}_i(\tau), \ddot{\varphi}_j(\tau) \rangle_2$$

showing that all integrands of J_2 and J_3 are polynomials of order lower than eight. Thus, we can compute them exactly with a piecewise Gauss quadrature rule of order five. Using $v, w \in V_h$, the same argumentation implies that $\langle u(\tau), v(\tau) \rangle_2$, $\langle v(\tau), w(\tau) \rangle_2$, $\langle \dot{u}(\tau), \dot{v}(\tau) \rangle_2$, $\langle \dot{v}(\tau), \dot{w}(\tau) \rangle_2$, $\langle \ddot{u}(\tau), \ddot{v}(\tau) \rangle_2$, and $\langle \ddot{v}(\tau), \ddot{w}(\tau) \rangle_2$ are also polynomials of order lower than eight. Thus, we can use the same Gauss rule to compute the derivatives of J_2 and J_3.

To evaluate J_1, we have to compute the outer integral which defines the tracking problem. Here, we can simply use the piecewise midpoint rule, since this only transforms the problem from a continuous into a discrete tracking problem. Furthermore, the measured values $y_m(t)$ are, in general, also discrete, this is naturally the best method if we choose the quadrature points to be equal with the times where we have measured the potentials y_m. Thus, this determines the times steps t_i for which we compute $z(u, t_i)$ with the above-described adaptive quadrature algorithm. Since we use a Galerkin approach, we must ensure that the segmentation $\mathcal{T}_{\delta v}$, generated by the adaptive quadrature, contains the segmentation points of $\mathcal{T}_{FE}$. Thus, we use $\mathcal{T}_{FE}$ as initial segmentation when using the adaptive quadrature. As before, we could compute z, z', and z'' simultaneously, but due to the adaptive quadrature, this would be costly concerning computational effort and memory storage. This has mainly two reasons: Firstly, we would have to evaluate $\omega'_{k,h}$ and $\omega''_{k,h}$ repeatedly. And secondly, the adaptive quadrature requires storing the values of the derivatives (since z is vector-valued, they are, in fact, matrices and tensors) for each interval in the segmentation. Therefore, we decided to compute solely $z(u, t_i)$ adaptively and use the final segmentation $\mathcal{T}_{\delta v}$ to compute z'_G, z'_K, and z''_G concerning the Galerkin approach. Although this approach requires an additional iteration over $\mathcal{T}_{\delta v}$, we think it is the better option.

Remark 6. *The careful reader may notice that the error bound (4.13) requires the evaluation of $z'(u_k, t_i)\delta e_k$. Thus using (4.13) contradicts the above-described justification, since we anyway must evaluate $\omega'_{k,h}$ and store z' in this case. But testing solely with δe_k requires only storing one additional vector instead of a matrix. And since we anyway need an additional loop to compute z'', the above-described approach is, in our opinion, the better alternative.*

Since the computation of $z(u, t_i)$ requires only read access to $\omega_{k,h}$, we can easily parallelize the piecewise midpoint rule. Therefore, we first choose an interval $[0, T]$ on which we want to simulate a measurement. We then choose n equidistantly distributed quadrature points t_i with $t_0 = dt/2$, $t_{i+1} = t_i + dt$, $t_n = T - dt/2$, and $dt = T/(n+1)$. Next, we must decide how many threads we want to use. The workstation we use for

the numerical examples in the next section has four CPU cores, such that we can use up to four threads for the computation. The algorithm creates then $nbr_{threads}$ threads, where each thread computes the following: First, the thread locks the global index i such that the other threads can not change the index. Then, we test if we still must compute $z(u, t_i)$ respectively if $i < n + 1$. When this is true, we set $t_m = t_i$, increase i, and unlock the index i so the other treads can continue. We then compute $z(u, t_m)$ adaptively with Algorithm 3. Finally, we lock the value of J_1 (to avoid adding values simultaneously to J_1), add $\langle z(u, t_m), z(u, t_m)\rangle_2$ to J_1, and unlock J_1 again. Contrarily, if $i \geq n + 1$ (all quadrature points were visited), we also unlock i and exit the thread. This lead then to the following algorithm:

Algorithm 4: Parallel implementation of a piecewise midpoint rule

Input: u_k, $\{t_0, ..., t_n\}$, $nbr_{threads} = 4$;
Set $J_1 = 0$ and $i = 0$;
create $nbr_{threads}$ threads computing:
while *True* **do**
  lock i;
  if $i < n + 1$ **then**
    set $m = i$, $i = i + 1$ and unlock i
  else
    unlock i and exit thread;
  end
  compute $z(u_k, t_m)$;
  lock J_1, $J_1 = J_1 + ds \cdot \langle z(u_k, t_m), z(u_k, t_m)\rangle_2$, unlock J_1;
end
Output: $J_1 = J_1(u_k)$;

5.3 Details of the Inexact SQP-Method

In section 4.2, we proved that the Newton method converges globally if the error bound (4.4) for the numerical quadrature is fulfilled and if in addition (4.7), (4.9), and (4.12) hold. Additionally, we added the error bound (4.13) to ensure that the Newton method converges locally at a reasonable speed. In this section, we devise algorithmic measures to enforce these error bounds by customizing the numerical quadrature.

The exact evaluation of (4.4) requires the computation of the exact values of the integrals, which are not available. Thus, we have to replace the exact quantities with estimates. A standard method to derive such estimates is to replace the integrals with quadrature rules of higher order. As already mentioned when describing the adaptive quadrature, cf. Section 5.2, we are using the well-known Gauß-Kronrod quadrature formula. These quadrature rules consist of a Gauß quadrature rule, denoted by an index G, and an extended quadrature rule of higher order (called Kronrod extension), denoted by the

index K. With their help, we define the quantities:

$$
\begin{aligned}
E(u,v,t) &:= |\langle (z_K - z_G)(v,t), (z_K + z_G)(v,t)\rangle_n - \langle (z_G - z_K)(u,t), (z_G + z_K)(u,t)\rangle_n| \\
\mathcal{E}(u,v,t) &:= |\langle z_K(u,t), z_K'(u,t)v\rangle_n - \langle z_G(u,t), z_G'(u,t)v\rangle_n| \\
tol_{E,k} &:= \frac{2\varsigma}{T}|\mathcal{L}_K(u_{k+1}, \xi_k) - \mathcal{L}_K(u_k, \xi_k)| \\
tol_{\mathcal{E},k} &:= \frac{\epsilon^2}{T}\|\delta v_k\|_H^2
\end{aligned}
$$

The error bounds (4.3) and (4.14) are then replaced by their computable counterparts:

$$E(u_k, u_{k+1}, t) \leq tol_{E,k} \quad \forall t \in [0,T] \tag{5.5}$$
$$\mathcal{E}(u_k, \delta e_k, t) \leq tol_{\mathcal{E},k} \quad \forall t \in [0,T] \tag{5.6}$$

with $\delta e_k := \delta u_k - \delta v_k$. To compute the directions δu_k and δv_k, we apply a Galerkin method to (4.5) and (4.10), which lead to the discrete problems:

$$H_G(u_k; \Lambda_k)(\delta u_k, w) + \mathcal{L}_G'(u_k, \xi_k)w = 0 \quad \forall w \in V_h. \tag{5.7}$$
$$H_G(u_k; \Lambda_k)(\delta v_k, w) + \mathcal{L}_K'(u_k, \xi_k)w = 0 \quad \forall w \in V_h. \tag{5.8}$$

Similar to the computation of the impulse response functions $\omega_{k,h}$, we can transform (5.7) and (5.8) into systems of linear equation by inserting the basis representation of δu_k. Therefore, we first note that it is sufficient to test (5.7) and (5.8) with all basis function $\varphi_i \in V_h$, cf. 5.1. Using $\delta u_k = \sum_{i=0}^{m} (\underline{\delta u_k})_i \varphi_i$, we can compute

$$H_G(u_k; \Lambda_k)\left(\sum_{i=0}^{m} (\underline{\delta u_k})_i \varphi_i, \varphi_j\right) = \sum_{i=0}^{m} (\underline{\delta u_k})_i H_G(u_k; \Lambda_k)(\varphi_i, \varphi_j) \quad j = 0, ..., m$$

such that we get the following systems of equations

$$\underline{H}\,\underline{\delta u_k} = -\underline{L} \text{ and } \underline{H}\,\underline{\delta v_k} = -\underline{\tilde{L}} \tag{5.9}$$

where the matrix entries $\underline{H}_{ij}$ are given through

$$\underline{H}_{ij} := H_G(u_k; \Lambda_k)(\varphi_i, \varphi_j).$$

Consequently, we get the right-hand sides

$$\underline{L}_i := \mathcal{L}_G'(u_k, \xi_k)(\varphi_i)$$

and

$$\underline{\tilde{L}}_i := \mathcal{L}_K'(u_k, \xi_k)(\varphi_i).$$

Since we use an adaptive quadrature rule, these two linear systems of equations are pretty small such that we can use a direct solver, i.e., a Cholesky decomposition of $\underline{H}$. And

since the matrix is the same for both systems, we must only compute the decomposition once. Next, we want to discuss how we incorporated the error bounds (5.5) and (5.6) into our algorithm. The first part where we can incorporate them is the computation of the direction δu_k. There we use (5.6) together with the adaptive quadrature to ensure that the algorithm converges quickly, locally, i.e., that (4.13) is satisfied. And secondly, we use (5.5) to enforce global convergences, cf. Theorem 4.3.1, when applying the line search.

Computation of Search Directions When computing the directions of descent δu_k and δv_k, defined by (5.7) and (5.8), respectively, we need to ensure that they satisfy condition (4.13) and thus also (4.12). To achieve this, we use the following correction loop, specified in Algorithm 5. First, we compute $\mathcal{L}'_G(u_k, \xi_k)$, $\mathcal{L}'_K(u_k, \xi_k)$ and $H_G(u_k; \Lambda)$ as described above, cf. Section 5.2. The J_1 part of $\mathcal{L}'$ is thereby computed adaptively such that a given error bound is satisfied. Second, we solve (5.7) and (5.8) using an inner feedback loop to ensure that $H_G(u_k; \Lambda)$ is positive definite, i.e., satisfies (4.7). After we have computed these candidates for δu_k and δv_k, we test if they satisfy (4.13). If this is not the case, we tighten the given error bound and restart the computation. Otherwise, Algorithm 5 terminates successfully and returns δu_k and δv_k. Due to the higher order quadrature rule, δv_k is more accurate than δu_k. Thus, we will use δv_k as input for the following line-search procedure.

Algorithm 5: Computation of Newton direction

Input: u_k, tol_{k-1} and $\mathcal{T}_0$; `// in general` $\mathcal{T}_0 = \mathcal{T}_{FE}$
compute J'_2, J'_3, J''_2 and J''_3 on $\mathcal{T}_{FE}$;
for *i=1,2,...* **do**
  if $i == 1$ **then**
    $z'_G, z'_K, z''_G, \mathcal{T}_i \leftarrow$ Alg. 3 (Input: u_k, $\mathcal{T}_0$, $tol_{E,k-1}$, (5.10)) ;
  else
    $z'_G, z'_K, z''_G, \mathcal{T}_i \leftarrow$ Alg. 3 (Input: u_k, δe_k, $\mathcal{T}_{i-1}$, $tol_{\mathcal{E},k}$, (5.6));
  end
  compute $J'_{1,G}, J'_{1,K}, J''_{1,G}, \mathcal{L}'_G$ and $\mathcal{L}'_K$;
  do
    choose Λ and compute H_G ; `// choice of` Λ, `see Sec.` 4.2
    $\delta u_k, \delta v_k \leftarrow$ solve (5.7) resp. (5.8) ;
  while (4.7) *is not satisfied*;
  if $\|\delta e_k\|_H \leq \epsilon \|\delta v_k\|_H$ **then**
    terminate "accuracy requirement (5.6) fulfilled";
  else
    update δe_k and $tol_{\mathcal{E},k}$;
  end
end
Output: δv_k and $\mathcal{T}_{\delta v_k} = \mathcal{T}_k$;

Let us now explain the features of Algorithm 5 in more detail. As described in Section 5.2, we can split the computation of $\mathcal{L}'_G(u_k, \xi_\mathtt{k})$, $\mathcal{L}'_K(u_k, \xi_\mathtt{k})$ and $H_G(u_k; \Lambda)$ into two parts. First, we can exactly compute $J'_2(u_k), J'_3(u_k, \xi_\mathtt{k}), J''_2(u_k)$ and $J''_3(u_k, \xi_\mathtt{k})$ on the given finite element segmentation $\mathcal{T}_{FE}$ with a Gauss quadrature rule of order five. These computations do not need adaptive quadrature and thus can be performed before the loop. Second, we compute $J'_{1,G}(u_k)$, $J'_{1,K}(u_k)$, and $J''_{1,G}(u_k)$ requiring the "adaptive" computation of $z'_G(u_k, t)$, $z'_K(u_k, t)$, and $z''_G(u_k, t)$ as described in the previous section.

When computing $z(u_k, t)$ adaptively with algorithm 3, we use, in general, the error criterion (5.6). This, however, is only possible for $i \geq 2$, since (5.6) requires an approximation of the difference $\delta u_k - \delta v_k$ which is only available after the first pass of the loop. Thus, in the first pass of the loop, we have to replace (5.6) by an alternative criterion. A reasonable choice would be using the error bound (5.5) used in the line search, cf. below. But this error bound depends on u_{k+1} and $tol_{E,k}$, which are unavailable. Thus, we drop the quantities requiring u_{k+1} and replace $tol_{E,k}$ with $tol_{E,k-1}$ resp. 10^{-5} if $k = 0$. These modifications lead to the following error bound:

$$|\langle (z_K - z_G)(u_k, t), (z_K + z_G)(u_k, t) \rangle_n| \leq tol_{E,k-1}. \tag{5.10}$$

Beginning with the second pass of the loop, we can use δu_k and δv_k from the previous step to apply the error bound (4.13) when adaptively computing $z(u_k, t)$.

After the computation of $z'_G(u_k, t)$, $z'_K(u_k, t)$, and $z''_G(u_k, t)$, we can compute $J'_{1,G}(u_k)$, $J''_{1,G}(u_k)$, and $J'_{1,K}(u_k)$, and thus $\mathcal{L}'_G(u_k, \xi_\mathtt{k})$, $\mathcal{L}'_K(u_k, \xi_\mathtt{k})$, and $H_G(u_k; \Lambda)$ as described in section 5.2, needed to define the linear systems (5.7) and (5.8).

When solving (5.7) and (5.8), we first choose $\Gamma = 0$ and compute $H_G(u_k; \Gamma)$. Then we solve the resulting systems of linear equations (5.9) with the help of a Cholesky decomposition and test if (4.7) is satisfied. If this test fails, we increase Γ, i.e., set $\Gamma = 0.1$ and multiply it with 2 after each failed test. Next, we update $H_G(u_k; \Gamma)$ and recompute δu_k and δv_k until they fulfill (4.7).

Finally, we test if (4.13) is satisfied. If this test fails, we update δe_k and $tol_{\mathcal{E},k}$ and go back to the computation of the derivatives of J_1. Otherwise, we continue with the line search which we are describing now.

Inexact Line Search When computing the damping factor β, we must ensure that the Lagrangian functional satisfies (4.4). Therefore, it is necessary to evaluate $z(u_k, t)$ and $z(u_{k+1}, t)$ adaptively using the error bound (5.5). This error bound depends on $tol_{E,k}$, which is not a priori known. Thus, we cannot use the error bound directly. Instead, we add a feedback loop to a standard backtracking line search algorithm, cf., e.g., [48, Algorithm 3.1], and use $tol_{E,k-1}$ as initial tolerance. As initial segmentation for the adaptive quadrature algorithm, we use the final segmentation $\mathcal{T}_{\delta v_k}$ of the step computation. This leads to the modified backtracking algorithm 6, which we describe now.

We start with $\beta = 1$ and set $u_{\text{try}} = u_k + \beta\delta v_k$. As describe in Section 5.2, we can compute $J_2(u_{\text{try}})$ and $J_3(u_{\text{try}}, \xi_{\mathtt{k}})$ without resorting to adaptive quadrature. Furthermore, we can reuse $J_2(u_k), J_3(u_k, \xi_{\mathtt{k}}), J_2'(u_k)$ and $J_3'(u_k, \xi_{\mathtt{k}})$ computed during the computation of δu_k. To obtain $J_{1,G}(u_{\text{try}})$, $J_{1,K}(u_{\text{try}})$, $J_{1,G}(u_k)$, and $J_{1,K}(u_k)$, we compute $z(u_k, t)$ and $z(u_{\text{try}}, t)$ with the adaptive quadrature algorithm 3. Thereby, we must ensure that both quantities satisfy the error bound (5.5). Thus, we slightly modify the adaptive quadrature algorithm such that it computes $z(u_k, t)$ and $z(u_{\text{try}}, t)$ simultaneously with the same Gauss-Kronrod rule. The modifications read as follow: Since $z(u_k, t)$ and $z(u_{\text{try}}, t)$ are independent, we can compute the local quantities $z_{G,I}(u_k, t)$, $z_{K,I}(u_k, t)$, $z_{G,I}(u_{\text{try}}, t)$, $z_{K,I}(u_{\text{try}}, t)$, and $E_I(u_k, u_{try}, t)$ separately while iterating over the segmentation $\mathcal{T}_j$ (starting with $\mathcal{T}_{\delta v_k}$).

After computing the global values as usual, we test if (5.5) is satisfied. If this is not the case, we choose a refinement partition $\mathcal{T}_R$ such that (5.4) is satisfied. As usual, we refine $\mathcal{T}_j$ and recompute $z_{G,I}(u_k, t), z_{K,I}(u_k, t)$, $z_{G,I}(u_{\text{try}}, t)$, and $z_{K,I}(u_{\text{try}}, t)$ on the refined intervals.

Algorithm 6: Computation of the damping factor β

Input: u_k, δv_k, $J_2(u_k)$, $J_3(u_k, \xi_{\mathtt{k}})$, $J_2'(u_k)\delta v_k$, $J_3'(u_k, \xi_{\mathtt{k}})\delta v_k$, $\mathcal{T}_{\delta v_k}$;
Parameter: $\varsigma = 0.9$, $\eta = 10^{-3}$, $\sigma = 0.01$, $\beta = 2$, $tol_{E,k} = tol_{E,k-1}$;
do
  do
    $\beta \leftarrow 0.5\beta$ and $u_{\text{try}} \leftarrow u_k + \beta\delta v_k$;
    compute $J_2(u_{\text{try}})$ and $J_3(u_{\text{try}}, \xi_{\mathtt{k}})$ on $\mathcal{T}_{FE}$;
    $z_G(u_k, t), z_K(u_k, t), z_G(u_{\text{try}}, t), z_K(u_{\text{try}}, t), \mathcal{T}_\beta \leftarrow$ Alg. 3
      (Input: u_k, u_{try}, $\mathcal{T}_{\delta v_k}$, $tol_{E,k}$, and error bound (5.5));
    compute $z_K'(u_k, t)\delta v_k$ on $\mathcal{T}_\beta$;
    compute $J_{1,G}(u_k), J_{1,K}(u_k)$, $J_{1,K}'(u_k)\delta v_k$, $J_{1,G}(u_{\text{try}})$, and $J_{1,K}(u_{\text{try}})$;
    compute $\mathcal{L}_G(u_k, \xi_{\mathtt{k}}), \mathcal{L}_K(u_k, \xi_{\mathtt{k}})$, $\mathcal{L}_K'(u_k, \xi_{\mathtt{k}})\delta v_k$, $\mathcal{L}_G(u_{\text{try}}, \xi_{\mathtt{k}})$ and $\mathcal{L}_K(u_{\text{try}}, \xi_{\mathtt{k}})$;
  while $\mathcal{L}_K(u_{\text{try}}, \xi_{\mathtt{k}}) - \mathcal{L}_K(u_k, \xi_{\mathtt{k}}) > \beta\eta\mathcal{L}_K'(u_k, \xi_{\mathtt{k}})\delta v_k$;
  if $\mathcal{L}_{Q_\beta}'(u_k, \xi_{\mathtt{k}})\delta v_k \geq \sigma\mathcal{L}_{Q_{\delta v_k}}'(u_k, \xi_{\mathtt{k}})\delta v_k$ **then**
    terminate "(4.9) violated";
  else
    $tol_{E,k} = \frac{2\varsigma}{T}|\mathcal{L}_K(u_{\text{try}}) - \mathcal{L}_K(u_k)|$;
  end
while $|(\mathcal{L}_K(u_{\text{try}}, \xi_{\mathtt{k}}) - \mathcal{L}_K(u_k, \xi_{\mathtt{k}})) - (\mathcal{L}_G(u_{\text{try}}, \xi_{\mathtt{k}}) - \mathcal{L}_G(u_k, \xi_{\mathtt{k}}))| \geq tol_{E,k}$;
terminate "(4.8) and estimated (4.4) fulfilled";
Output: β;

As before we can compute $z_K'(u_k, t)\delta v_k$ on the final segmentation $\mathcal{T}_\beta$. Next, we can compute $J_{1,G}(u_k), J_{1,K}(u_k)$, $J_{1,K}'(u_k)\delta v_k$, $J_{1,G}(u_{\text{try}})$, $J_{1,K}(u_{\text{try}})$, $\mathcal{L}_G(u_k, \xi_{\mathtt{k}}), \mathcal{L}_K(u_k, \xi_{\mathtt{k}})$, $\mathcal{L}_K'(u_k, \xi_{\mathtt{k}})\delta v_k$, $\mathcal{L}_G(u_{\text{try}}, \xi_{\mathtt{k}})$ and $\mathcal{L}_K(u_{\text{try}}, \xi_{\mathtt{k}})$ and test if the Armijo condition is satisfied.

If not, we multiply β by 0.5 and repeat the process until the Armijo condition is fulfilled. When we have found an acceptable β, we update $tol_{E,k}$ and test if (4.4) is still satisfied. If this is not the case, we restart the line search with the updated error bound. This final test ensures that (4.4) is satisfied at termination. The test seldom fails since we already used the adaptive quadrature to control this error.

The last section of this chapter treats the update of the Lagrange multiplier λ_k occasionally happening after a successful pass of the Newton method.

5.4 Details of the Lagrange Parameter Update

After a successful pass of the Newton method, we test if $\|G(u)\|_{2,2^*}$ was reduced enough, more precisely that the feasibility of the solution was increased, cf. 4.1. If this was the case we update the Lagrange parameter λ_k by solving

$$\begin{aligned} 0 &= \langle w, v\rangle_{1,2} + J'(\tilde{u}_k)(v) + \langle \lambda_{k+1}, G'(\tilde{u}_k)(v)\rangle_{1,2} \\ 0 &= G'(\tilde{u}_k)(w) \end{aligned} \qquad \forall v \in H^2(-t_A, t_A, \mathbb{R}^3),$$

cf. Section 4.1. Analyzing this equation, we note that the first equation is an integral equation while the second equation pointwise, i.e., $[G'(u)(w)](\tau) = 0$ for almost all $\tau \in [-t_A, t_A]$. But we can, using the fundamental lemma of calculus of variations, see, e.g., [11, Section 2.2], transform the second equation into an integral equation of the form

$$\langle G'(\tilde{u}_k)(w), \tilde{v}\rangle_{1,2} = 0 \quad \forall \tilde{v} \in H^1(-1, 1, \mathbb{R}).$$

Furthermore, it is well-known that it is numerically more stable to compute an update $\delta\lambda_k$ instead of the new iterate λ_{k+1}. Thus, we write $\lambda_{k+1} = \lambda_k + \delta\lambda_k$ and compute

$$\begin{aligned} 0 &= \langle w, v\rangle_{1,2} + J'(\tilde{u}_k)(v) + \langle \lambda_{k+1}, G'(\tilde{u}_k)(v)\rangle_{1,2} \\ &= \langle w, v\rangle_{1,2} + J'(\tilde{u}_k)(v) + \langle \lambda_k, G'(\tilde{u}_k)(v)\rangle_{1,2} + \langle \delta\lambda_k, G'(\tilde{u}_k)(v)\rangle_{1,2} \\ &= \langle w, v\rangle_{1,2} + \mathcal{L}'(\tilde{u}_k, (0, \lambda_k))(v) + \langle \delta\lambda_k, G'(\tilde{u}_k)(v)\rangle_{1,2}. \end{aligned}$$

To solve these integral equations numerically, we must discretize $\lambda_k, \delta\lambda_k \in H^1(-1, 1, \mathbb{R})$. Since λ_k is only in H^1, we could use standard Lagrange basis polynomials for this purpose. But since we already use Hermite basis polynomials for the discretization of the trajectory u, we use the same basis polynomials to discretize λ_k which makes the assembling of the matrix G, see below, much easier. Furthermore, using Hermite finite elements respects the continuity of the Lagrange multiplier. When discretizing λ_k we must consider that, in difference to u, the image of λ is in $\mathbb{R}$. Thus, we get the ansatz space:

$$\tilde{V}_h := \{v \in C^1(-t_A, t_A, \mathbb{R}) : v|_I \in P_3(I)\ \forall I \in \mathcal{T}_{FE}\} \subset H^2(-t_A, t_A, \Omega_M),$$

with basis $(b_0(\tau), ..., b_{2n+1}(\tau))$. Similar to the above-described finite element discretization, it is now sufficient to test with all basis functions $\varphi_j \in V_h$ and $b_j \in \tilde{V}_h$. Thus, we

must solve

$$\begin{aligned} 0 &= \langle w, \varphi_j \rangle_{1,2} + \mathcal{L}'(\tilde{u}_{\mathtt{k}}, (0, \lambda_{\mathtt{k}}))(\varphi_j) + \langle \delta\lambda_{\mathtt{k}}, G'(\tilde{u}_{\mathtt{k}})(\varphi_j) \rangle_{1,2} && \forall \varphi_j \in V_h \\ 0 &= \langle G'(\tilde{u}_{\mathtt{k}})(w), b_j \rangle_{1,2} && \forall b_j \in \tilde{V}_h. \end{aligned}$$

Using $\delta\lambda = \sum_{i=0}^{2n+1} \underline{\delta\lambda}_i b_i \in \tilde{V}_h$ and $w = \sum_{i=0}^{m} \underline{w}_i \varphi_i \in V_h$ we can compute

$$\begin{aligned} \langle w, \varphi_j \rangle_{1,2} &= \langle \sum_{i=0}^{m} \underline{w}_i b_i, \varphi_j \rangle_{1,2} = \sum_{i=0}^{m} \underline{w}_i \langle \varphi_i, \varphi_j \rangle_{1,2} \\ \langle \delta\lambda_{\mathtt{k}}, G'(\tilde{u}_{\mathtt{k}})(\varphi_j) \rangle_{1,2} &= \langle \sum_{i=0}^{2n+1} \underline{\delta\lambda}_i b_i, G'(\tilde{u}_{\mathtt{k}})(\varphi_j) \rangle_{1,2} = \sum_{i=0}^{2n+1} \underline{\delta\lambda}_i \langle b_i, G'(\tilde{u}_{\mathtt{k}})(\varphi_j) \rangle_{1,2} \\ \langle G'(\tilde{u}_{\mathtt{k}})(w), b_j \rangle_{1,2} &= \langle G'(\tilde{u}_{\mathtt{k}})(\sum_{i=0}^{m} \underline{w}_i \varphi_i), b_j \rangle_{1,2} = \sum_{i=0}^{m} \underline{w}_i \langle G'(\tilde{u}_{\mathtt{k}})(\varphi_i), b_j \rangle_{1,2}, \end{aligned}$$

such that we get the following linear system of equations:

$$\begin{bmatrix} \mathrm{M} & \mathrm{G}^T \\ \mathrm{G} & 0 \end{bmatrix} \begin{bmatrix} \underline{w} \\ \underline{\delta\lambda} \end{bmatrix} = \begin{bmatrix} \tilde{\mathrm{L}} \\ 0 \end{bmatrix}$$

with

$$\begin{array}{lll} \mathrm{M} \in \mathbb{R}^{m \times m}, & \mathrm{M}_{i,j} = \langle \varphi_i, \varphi_j \rangle_{1,2}, & i,j = 0, ..., m \\ \mathrm{G} \in \mathbb{R}^{2n+1 \times m}, & \mathrm{G}_{i,3j+l} = \langle b_i, G'(u_{k+1}) \varphi_j \rangle_{1,2}, & i = 0, ..., n+1;\ j = 0, ..., m \end{array}$$

and

$$\mathrm{L} \in \mathbb{R}^m, \qquad \mathrm{L}_i = -\mathcal{L}'(u_{k+1}, \xi_{\mathtt{k}})\varphi_i, \qquad i = 0, ..., m.$$

Similar to the step computation, the resulting linear system of equations has a moderate size. Therefore, we can use a direct solver to solve it. But contrarily, this system of linear equations is due to the local support of the basis functions ϕ_i sparse. Thus, we use a sparse LU decomposition based on the SuperLU package, see [14].

Computing the matrix entries M_{ij} and G_{ij} does not depend on the simulation y. Thus, we can compute them without resorting to the adaptive quadrature rule. As described in section 5.1 and 5.2, we can use a Gauss quadrature rule of order 5 to compute the matrix entries on each interval of the segmentation $\mathcal{T}_{FE}$. Furthermore, computing M_{ij} only depends on the basis functions φ_i, such that we can compute M once in the beginning and reuse a stored version.

The computation of the right-hand side L differs only in two aspects from the evaluation of $\underline{L}$ when computing the direction δu_k, cf. Section 5.3. First, we set $\mu = 0$. And secondly, the update of the Lagrange multiplier doesn't depend on any error bound. Thus, we can use a simple relative error criterion when "adaptively" computing $z'_K(\tilde{u}_{\mathtt{k}}, t)$ as described in Section 5.2.

Chapter 6

Numerical Examples

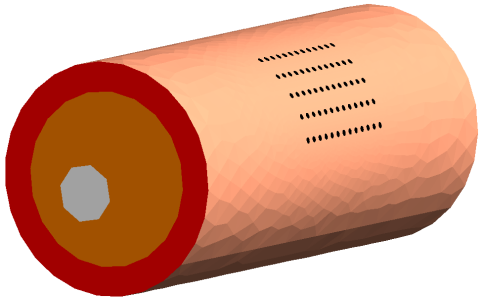
The final chapter studies seven numerical examples. The foundations for these examples are two different geometries. The first four examples use a cylindrical geometry representing a part of an upper arm and the last example uses a geometry representing a right hand containing the first dorsal interossei (FDI) muscle and the two neighboring bones. We use these geometries to verify that the optimization problem (3.3) combined with the augmented Lagrangian method, see Section 4.1, is capable of identifying a motor unit from a surface EMG measurement. The first example serves as a proof of concept. Furthermore, we conduct a performance analysis for this first numerical example. The second example studies the influence of the impulse response functions, i.e. the accuracy with which compute them. The third example studies the influence of the algorithmic parameters ε and α_2 on the identifying process. The fourth example studies the influence of anatomical circumstances, i.e. the fat thickness and the distance between the motor unit and the electrodes. The fifth example studies the influence of the electrode positions, i.e. the distance between two electrodes and their number. When we modeled a surface EMG measurement, we assumed for simplicity that motor units are singled-fibered, which is anatomically incorrect. Thus, the sixth example simulates a multi-fibered motor unit and shows that we can identify the center of mass of this motor unit. The last example visualizes the identification process in a more complicated geometry, namely the right hand. For this example, we also conduct a second performance analysis. Parts of this chapter has been published in [65].

6.1 Identification of a Motor Unit in a Cylindrical Domain

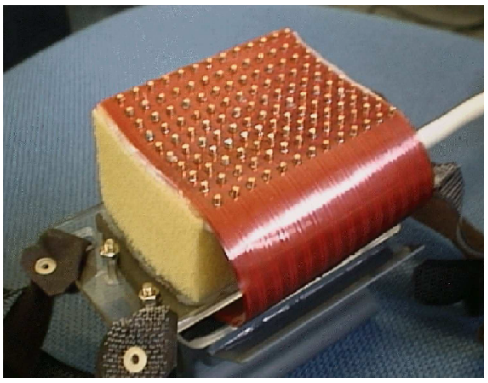
This first example visualizes the general capability of our approach for identifying a motor unit. Furthermore, we conduct a performance analysis of the augmented Lagrangian method. But before we can discuss the numerical results, we must describe the general setup of the example.

General Setup

For the first, and also for the following five, examples, we use a cylindrical domain representing a part of a limb, e.g., the middle part of an upper arm. This cylinder consists of three layers, where the inner layer represents a bone (white), the middle layer represents muscle tissue (orange), and the outer layer is fat tissue (red). For the setup of the electrodes, we oriented us on a real measuring device, see Figure 6.1b, containing 9×13 electrodes with a distance of $4mm$ between the electrode centers. But to keep the numerical effort at a reasonable level, we consider only a grid of $5 \times 13 = 65$ electrodes (black circles) on the skin (beige). Furthermore, we increased the distance between the electrodes to $8mm$ in the horizontal and vertical directions. The cylinder has a length of $18cm$ and a diameter of $8cm$. The bone has a diameter of $2cm$, the muscle tissue has a diameter of $7.5cm$, and the fat layer has a depth of $0.5cm$. As shown in Figure 6.1a, we shifted the bone and placed the electrodes on the opposite boundary.



(a) Cylindrical domain with 65 electrodes



(b) Measure foil with circular electrodes

Figure 6.1: Geometric setting and real measure foil

To create a FE conforming mesh from the geometry, we used gmsh, see [24]. Concerning the triangulation, there are two domains of interest where we want the mesh to be sufficiently fine. The first region is the domain where the electrodes are placed. Due to the circular shape of the electrodes and a possible curved skin, the mesh must be sufficiently fine in this region. Since the shape of the electrodes is defined during the mesh generation, this area is initially pretty fine and has no hierarchical structure. The second region is that part of the muscle tissue, where the motor unit is roughly known to be located. This area was chosen as a cylinder with a radius of $0.5cm$ around the axis $(x,\ 0.,\ 0.029)$. We refined the mesh inside this cylinder two resp. three times resulting in two different meshes containing 82581 resp. 181084 tetrahedra, which we will use later for comparison purposes.

For the approximation of a motor unit, a finite element discretization of $[-1, 1]$ that consists of 19 subintervals is sufficient. Therefore, the matrix representation of $H(u, \Lambda)$ is only a 120×120 matrix, such that we can use a direct solver, i.e., a Cholesky de-

composition, to solve (4.5), cf. Section 5.3. The matrix for computing the update of the Lagrange multiplier is similarly small, but, in contrast to $H(u, \Lambda)$, it is sparse, see Section 5.4.

We created synthetic measurement data, see Figure 6.2 by simulating measurements for a given reference motor unit. Analytically our reference motor unit is given through

$$u(t) = \begin{pmatrix} \cos(\pi/90)\sin(0.38t) - 0.02\sin(\pi/90) \\ 0.1\cos(0.38t) \\ \sin(\pi/90)\sin(0.38t) + 0.02\cos(\pi/90) \end{pmatrix} \quad \text{for } t \in [-1, 1].$$

Figure 6.3 (below) shows two views of the reference motor unit (cyan). The left picture visualizes the positions relative to the electrodes, and the right picture shows the change of depth respectively the distance to the electrodes. This setting combines two critical difficulties, which are usually encountered separately in practice: the change of depth (e.g., in facial muscles) and a curved motor unit (e.g., in the biceps). It is thus slightly more complicated than most practical use cases.

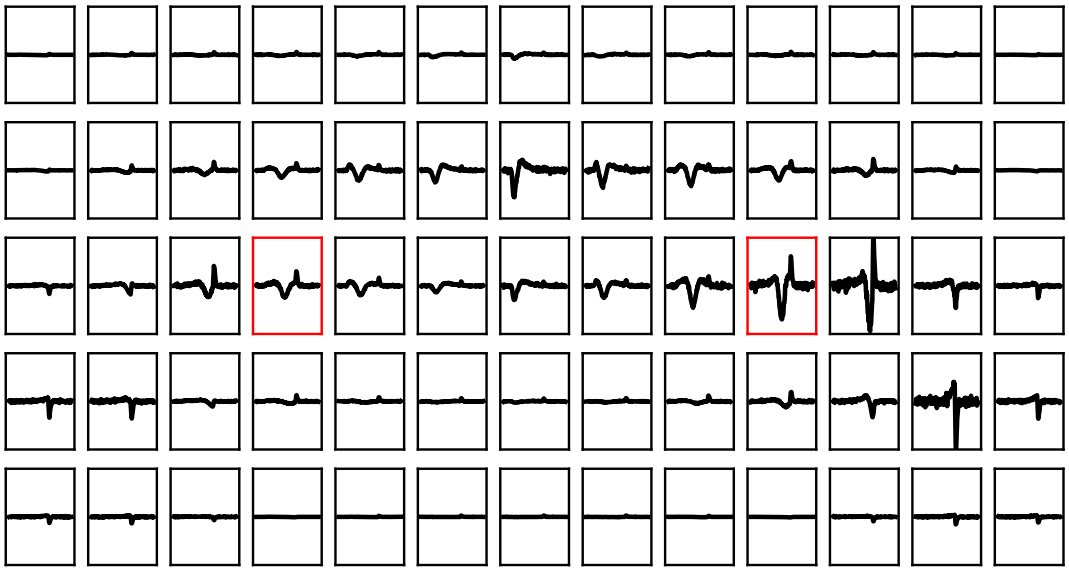


Figure 6.2: Simulated measurement for a reference motor unit at 65 electrodes

To simulate the measurements, we divided the time interval $[0, 2]ms$ into 200 equidistantly distributed measure points t_i and computed for each measure-point $y(u, t_i)$ with (2.35). We used the adaptive quadrature algorithm 3 with a standard relative error criterion and tolerance 10^{-6} to compute the integral. To create a realistic measurement, we added white noise with a data-to-noise ratio of 5%. Figure 6.2 shows the 65 simulated

measurements in a grid. The measurements contained in the red boxes are depicted in detail in Figure 6.4 (below) as reference measurements in the discussion of this example.

Looking at the grid of simulated measurements in Figure 6.2, we can already guess roughly the trajectory of the motor unit visually. Additionally, we notice that the measured signal gets stronger from left to right indicating that the distance between the motor unit and the electrodes is shorter on the right part of the measurement. Based on these observations we can select an initial guess for the solution. To challenge our algorithm, i.e. the globalization part, we ignored most of the visual information and chose a straight line with a constant depth. In Figure 6.3, the initial trajectory is visualized in red. We then used the presented algorithm to identify the reference motor unit from the simulated measurement. To solve the optimization problem we selected the parameters $\alpha_1 = 0.1$, $\alpha_2 = 0.01$, and $\varepsilon = 0.01$. Furthermore, we terminated the inner Newton method if $\|\mathcal{L}'(u_{k+1}, \xi)\|_{2,2^*} < 0.9 \cdot 10^{-6}$ and the augmented Lagrangian method if $\|G(u)\| < 10^{-6}$ or after eight steps. We will now discuss the result of this identification process.

Numerical Analysis

We start with a qualitative discussion of the identification, using impulse response functions computed with cubic ansatz polynomials on a grid that was refined three times in the area of the motor unit. Figure 6.3 compares the reference motor unit, shown in cyan, with the computed solution, shown in blue. The right picture shows that the changing depth is identified very well. As the left picture shows, the curved trajectory is also identified well with a slight deviation at the left side where the motor unit is further away from the electrodes.

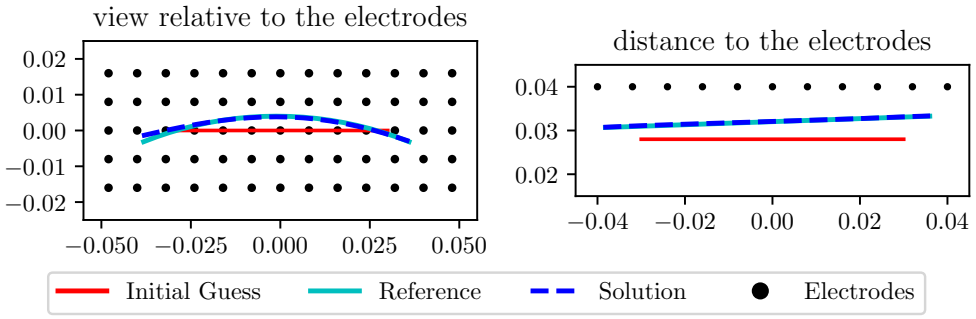


Figure 6.3: Comparison of the computed solution with the reference trajectory

Figure 6.4 compares the measurement produced by the reference trajectory with the measurement generated from the computed solution for two selected electrodes. We can see that both measurements fit well and mainly differ in the added noise. This observation underlines that the algorithm is suitable to identify a motor unit from a given sEMG measurement.

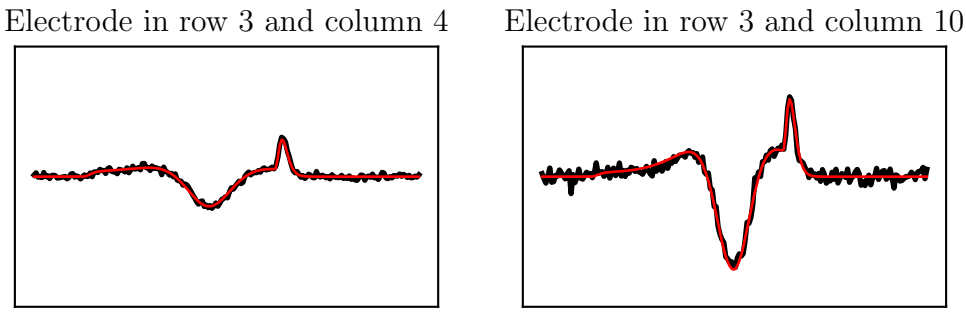


Figure 6.4: Comparison of simulated (black) and identified measurement (red).

Performance study Figure 6.5 shows that the algorithm needs 33 Newton steps distributed over 7 augmented Lagrangian steps. The first augmented Lagrangian step requires the most Newton steps, since it has the least accurate initial guess, and thus globalization is active. After this first pass of the Newton method, all other augmented steps require 2 to 5 Newton iterations, but we do not see superlinear convergence. We attribute for this observation to the non-smoothness of the discretized impulse response functions $\omega_{k,h}$. Therefore, small jumps in the first derivative, occur at the boundary between two adjacent tetrahedra. We will later observe the influence of the impulse response functions in more detail.

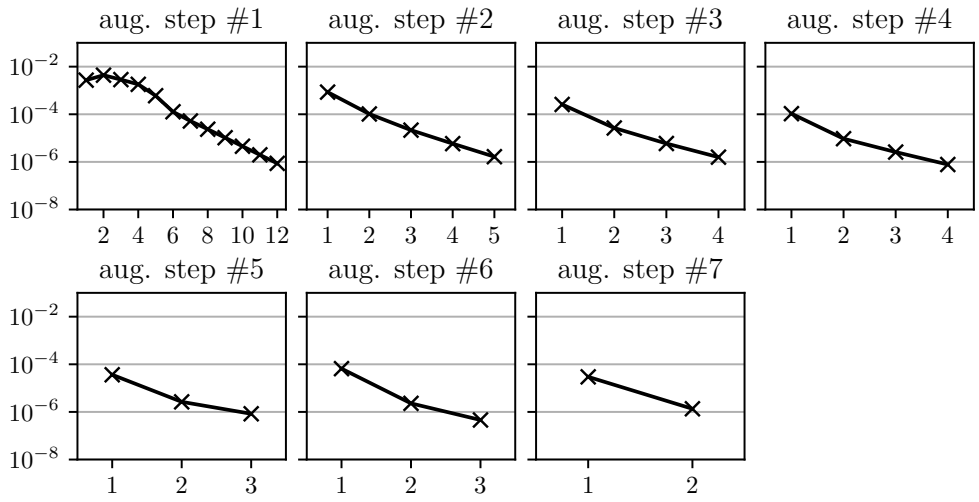


Figure 6.5: Energy norm of $\|\delta v_k\|_H$ during all augmented Lagrangian steps

Figure 6.6 visualizes the properties of the augmented Lagrangian method. We see that $\|G(u)\|_{2,2}$ converges linearly to zero. To achieve this, the algorithm alternates between increasing the penalty parameter μ and updating the Lagrange multiplier λ. In theory,

we would expect that $\|\mathcal{L}'\|_{2,2^*}$ also goes to zero. But achieving this goal requires the norm of the constraints to be much smaller and, thus, also a larger penalty parameter. In combination with the non-smooth derivatives of $\omega_{k,h}$ this led to numerical instabilities, such that we decide to use a softer termination criterion for $\|\mathcal{L}'\|_{2,2^*}$. From a practical point of view, the chosen tolerances are already pretty small, i.e., a tolerance of 10^{-6} corresponds to changes in the micrometer area. Furthermore, the nature of the constraint function implies that further decreasing $\|G(u)\|_{2,2}$ only changes the parameterization of the solution but not the trajectory. Thus, it is reasonable to use a moderate termination criterion.

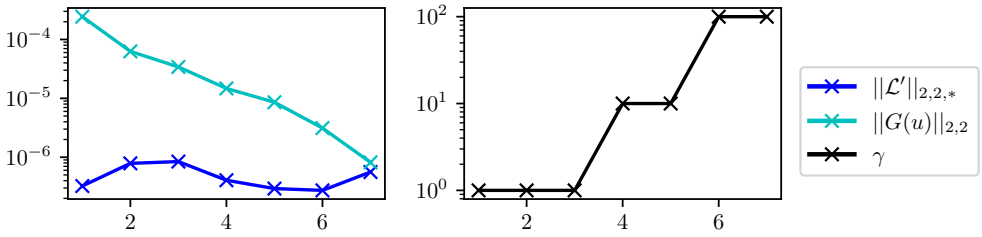


Figure 6.6: Key data of the augmented Lagrangian algorithm

Lastly, Figure 6.7 compares the tolerance, given to the adaptive quadrature, with the size of the resulting segmentation of the integration domain. For this example, the segmentation after the line search is the same as after computing δv_k. Thus, the plot shows only the final number of points in the segmentation. Furthermore, plotting the data from all measure points t_i would be messy. For the sake of brevity, we visualize the average and maximal number of points among all used segmentations in each step and concentrate our discussion on the first pass of the inner Newton method. Since both tolerances depend on differences that get very small during the algorithm, the adaptive quadrature algorithm refines the segmentation more for the last steps of the algorithm. Next, we notice that the average number of grid points is much lower than the maximal. This observation indicates that the adaptive quadrature needs only a fine segmentation for a few measure points and underlines the theoretical considerations in Section 5.2 and justifies using adaptive quadrature.

From this first example, we can conclude, that the optimization problem (3.3) combined with Algorithm 1 can identify a motor unit from a surface EMG measurement with reasonable effort. Furthermore, it justifies using an adaptive quadrature. The next example studies now the influence of the impulse response functions.

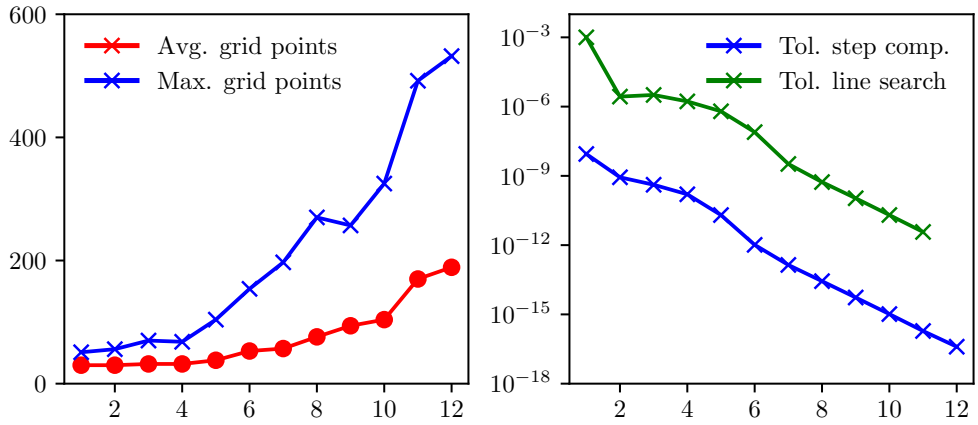


Figure 6.7: Comparison of the average and maximal grid size (left) after using adaptive quadrature with different tolerances (right)

6.2 Influence of Impulse Response Functions

This example discusses the influence of the discretization of the impulse response functions $\omega_{k,h}$ by finite elements. Discretization introduces small discontinuities of the derivatives $\omega'_{k,h}$ at the facets of the triangulation, which may affect the performance of our algorithm. To assess the situation, we solve the above problem using two different grid resolutions and, on each grid, finite elements of degrees two and three.

In our numerical experience, the inner Newton method behaves, after the first augmented Lagrangian step, similarly for all impulse response functions and only differs in the first augmented Lagrangian step. We attribute this behavior to the observation that the trajectory is already identified quite well. Thus, the remaining augmented Lagrangian steps mainly optimize the parameterization, which needs 2 to 7 Newton steps per augmented Lagrangian step. Figure 6.8 shows the convergence rates for the first pass of the inner Newton method. We notice that the algorithm requires the most iterations using the coarse grid (2 refinements) and polynomials of order 2. For the other sets of impulse functions, the convergence rate is approximately the same.

The size and order of the finite element space also influence the computational times for each step. Since the usage of the adaptive quadrature rule influences the computation times, the effect of the choice of the impulse response function is not that clear when comparing the average times needed to compute a Newton step. Therefore, we start by comparing the times the algorithm needs to simulate a measurement, see Table 6.1. Here, we notice that increasing the polynomial degree from two to three approximately doubles the computation time. Since changing the polynomial degree from two to three, increases the degrees of freedom per tetrahedron from 10 to 20, we could expect this behavior. Refining the grid increases the computation times by only about 50%. The observed behavior is similar when comparing the average times needed to compute a Newton step

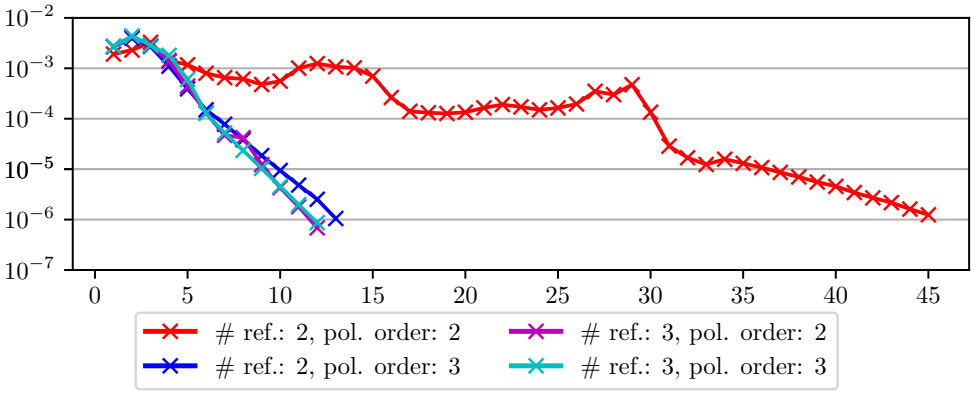


Figure 6.8: Comparison of $\|\delta v_k\|_H$ for different impulse response functions

considering that the adaptive quadrature produces different fine segmentations.

Including the times needed to compute the impulse response functions in our discussion, we observe that increasing the polynomial degree increases the computation time by a factor of 7, whereas refining the domain increases the computation time only by a factor of 3.5. Thus, if the optimization problem is solved only once, the computation of the impulse response function is possibly too time-consuming when using polynomials of degree 3. But if the optimization problem is solved multiple times, e.g., when identifying different motor units in one muscle, then polynomials of higher degrees are the better choice. In addition, higher-degree finite elements may yield more accurate identification results.

ref./ degree	Newton steps	avg. points in $[-1, 1]$	avg. time simulation [s]	avg. time Newton step [s]	total time [s]	time to compute $\omega_{k,h}$ [s]
2/2	69	88.7	1.33	4.47	299	158.72
2/3	35	84.1	2.54	8.85	285	1135.22
3/2	40	134	1.81	6.54	246	557.04
3/3	33	78.4	2.69	8.15	249	4664.88

Table 6.1: Comparing results for different impulse response functions $\omega_{k,h}$

6.3 Influence of Algorithmic Parameters

This section analyzes the influence of two algorithmic parameters on the identification. The first parameter we want to analyze is ε influencing the accuracy with which we compute δv_k. Secondly, we study the influence of the regularity parameter α_2 on the identified trajectory. For these examples, we use the same setup as in the first example but change the parameters accordingly.

Influence of ε To study the influence of the direction δv_k we solve the problem four times with $\epsilon = 0.5, 0.1, 0.01$ and 0.001. The solutions are, for all four parameters, visually not distinguishable, i.e. we get the same trajectory as in the first example, see Figure 6.3. But the convergence rate and computation times change significantly. Since the behavior of the inner Newton method is similar for all augmented steps, we only visualize the first pass of the inner Newton method in Figure 6.9.

First, we notice that the algorithm has the worst convergence rate for $\varepsilon = 0.5$. But for the other parameters, it has almost no effect. The reason for this observation becomes quite obvious when taking into account the error bound used for the adaptive quadrature. While for $\varepsilon = 0.5$, the step δv_k is accepted almost immediately, meaning that only the simplified error bound (5.10) is used, all other choices of ε require at least one additional computation of δv_k with the error bound (5.6).

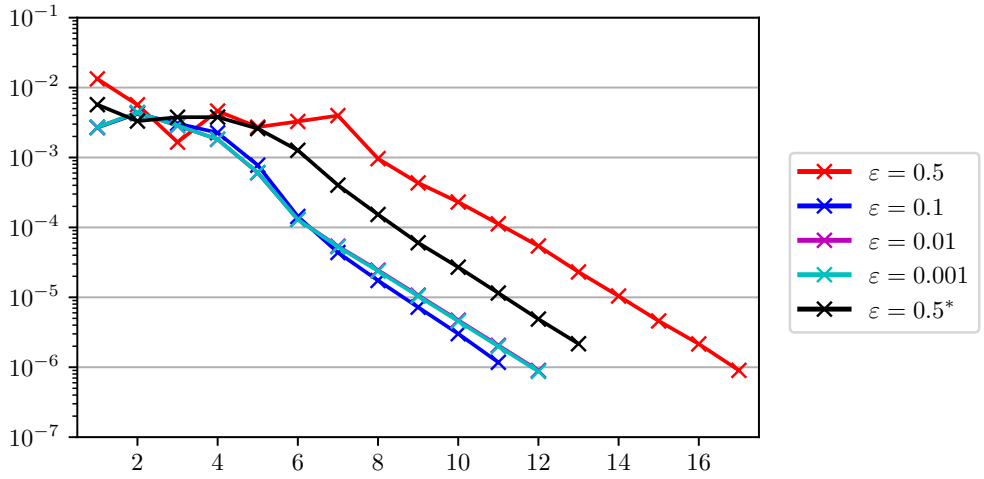


Figure 6.9: Energy Norm of δv for different tolerances ε (*: forced usage of (5.6))

From this observation we may conclude that the choice of the parameter is secondary and the important part is using the advanced error bound. To verify this observation, we solved the problem once more with $\varepsilon = 0.5$, but this time we automatically rejected the first version of the directions and always computed δv_k and δu_k with the more advanced error bound (Remember that we can not apply (5.6) directly, cf. Section 5.3). Using this approach resulted in a convergence rate that was still worse than with $\varepsilon = 0.1$ but better than simply using $\varepsilon = 0.5$ without forcing the usage of the error bound (5.6). Thus, for this example, $\varepsilon = 0.1$ seems to be the optimal choice. Additionally, using $\varepsilon = 0.1$ provides the fastest total computation time. As Table 6.2 shows, the time required to compute δv_k is slightly bigger than for $\varepsilon = 0.5$, but since it needs lesser steps the total computation time is much slower. Using smaller ε increases the computation times significantly but we do not gain any advantage regarding the convergence raid. Thus, we can conclude that is unnecessary to choose a small parameter ε.

ϵ	steps	avg. time Newton step [s]	Total Time [s]
0.5	44	2.5	160.1
0.1	33	3.3	127.7
0.01	33	5.4	195.0
0.001	33	8.9	302.2
0.5*	40	3.0	164.1

Table 6.2: Computation times for different tolerances ε (*: forced usage of (5.10))

Influence of α_2 The second parameter we want to analyze in this section is the regularization parameter α_2 controlling the smoothness of u. Therefore, we solve the problem three times with $\alpha_2 = 0.1, 0.01$ and 0.001. Figure 6.10 compares the identified motor units for the different choices of α_2.

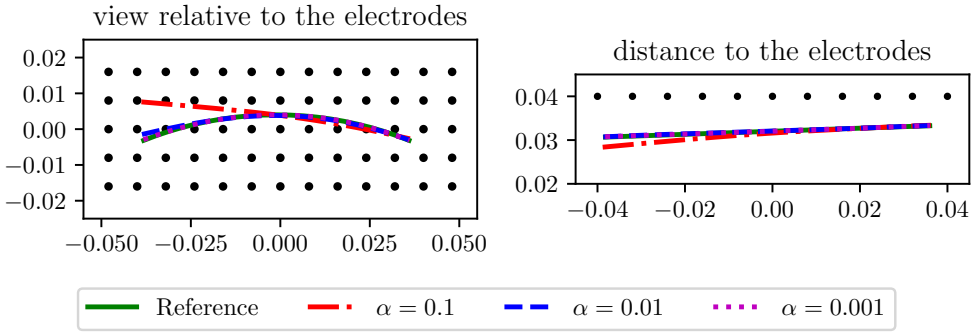


Figure 6.10: Solution for different regularization parameters α_2

The first observation we make is that the identification of the motor unit does not work very well for $\alpha_2 = 0.1$. For this choice, the solution matches the reference trajectory only on the right part of the measurements, i.e. in the region where the motor unit is closer to the electrodes. Additionally, the algorithm has problems identifying the correct depth on the left part of the measurements. As seen in the first example, the identification of the motor unit works pretty well for $\alpha_2 = 0.01$ and we see that it is almost perfect for $\alpha_2 = 0.001$.

The left picture in Figure 6.11 shows that the choice of α_2 has only a minimal influence on the convergence rate of the first pass of the inner Newton method. But in contrast, the right picture shows that the augmented Lagrangian has, for $\alpha_2 = 0.001$, not reached the termination criteria $\|G(u)\|_{2,2} \leq 1e-6$ after eight steps. The reason for this observation is, that the second derivative becomes indefinite if the penalty parameter μ is too large, which causes numerical instabilities for the later augmented Lagrangian steps. Thus, we can conclude that, for this example, α_2 is too small to compensate for the increasing penalty parameter μ.

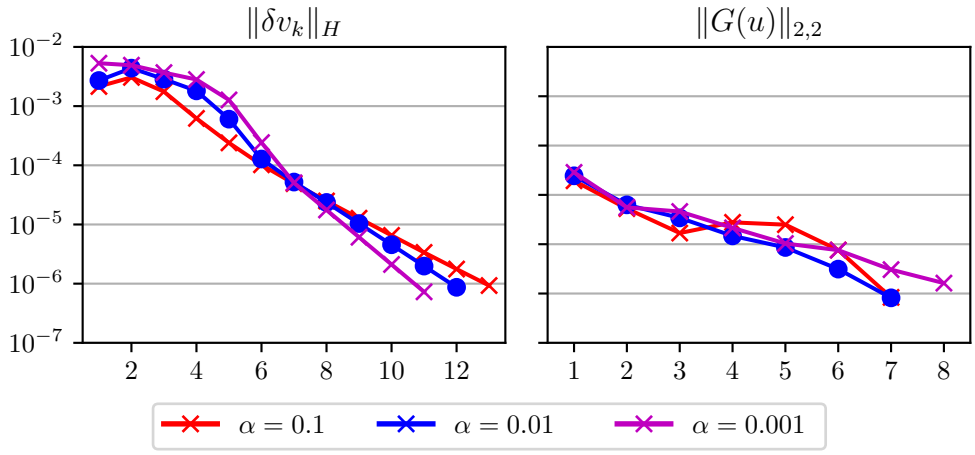


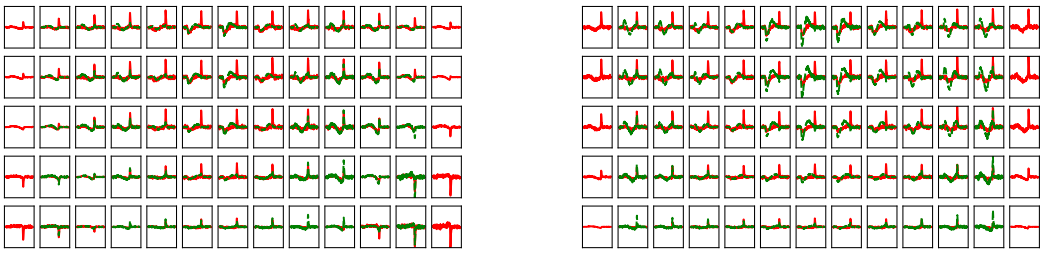
Figure 6.11: $\|\delta v_k\|_H$ and $\|G(u)\|_{2,2}$ for different regularization parameters α_2

This example showed that the regularity parameter α_2 has a non-neglectable influence on the Newton method. If it is too big we get a straight line and if it is too small the second derivative is no longer positive definite leading to numerical instabilities. Thus we have to choose this parameter with care or use an update strategy for α_2. We could, for example, start with $\alpha = 0.001$ and after the algorithm has finished, we could restart with $\alpha_2 = 0.001$ and use the previously computed solution as u_{ref}. In this case, we can increase α_1 without fearing that the globalization fails, since we are already close to the solution.

6.4 Influence of the Anatomy

In Section 2.3 we have observed that increasing the distance between the motor unit and the electrodes adds an additional effect to the measurement, the so-called end-effects. The reason for these effects is that potential generated from the point sources, introduces to ensure that the conservation of charges is not violated, gets less damped than the potential generated from the moving action potentials. Due to the material properties of fat, this effect is even more pronounced if the fat layer is thicker. Thus, this example studies the effect of the fat layer, and the distance between the motor unit and the electrodes. Therefore, we increased, in comparison to the first example, the distance between the motor unit and the electrodes from $10mm$ to $20mm$. Then we simulated two measurements. For the first measurement, see red measurement in Figure 6.12a, we used the same domain as in the first example. When we compare this measurement with the one from the first example, see Figure 6.2, we can clearly see some spikes at the end of the measurements which are the end-effects. But it is still possible to visually identify the trajectory of the motor unit and also the changing depth is observable. Thus in this case, the end-effects do not influence the identification and we can see in Figure 6.13 that the computed solution matches with the reference trajectory as well as in the first

example where no end-effects were observable.



(a) Simulated measurements for a $5mm$ fat layer, without (red) and with double differential filter (green)

(b) Simulated measurements for a $10mm$ fat layer without (red) and with double differential filter (green)

Figure 6.12: Comparison of Measurements

For the second measurement, see red measurement in 6.12b, we used the same geometry but we increased the thickness of the fat layer from $5mm$ to $10mm$. For this domain, the observation is completely different. We notice that this time the end effects dominate the measurements and neither the trajectory nor the changing depths are clearly observable. As a result, the identified motor unit does not match the reference trajectory as nearly as well as before, see Figure 6.13.

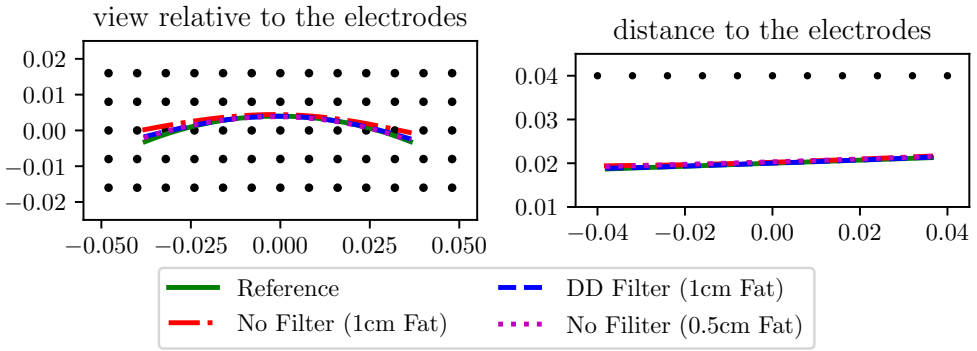


Figure 6.13: Solution for different thick fat layers and with using a DD filter

To overcome this problem, the authors of [21] introduced different detection systems to filter out the end effects generated from stationary sources. To decide which filter to use, we shortly recall that muscle tissue is anisotropic, i.e. the conductivity is higher in the axial direction. Thus, the generated potential propagate faster in the axial direction, and consequently the influence of the stationary sources on the measurement is different regarding the two space directions x (axial) and y (radial). Thus, the simulation results in [21, Section 4.3.3] show that the double differential detection system filters out the

non-propagating parts the best. This detection system uses three electrodes that are neighbors concerning the axial direction and generates a new measurement from them. Therefore, it multiplies the measurement from the middle electrode by two and subtracts the two other measurements. Using this filter reduces the electrode grid from 5×13 to 5×11, but it still covers the whole motor unit in the axial direction. Comparing the red (without filter) and green measurements (with filter) in Figure 6.12, we notice that the double detection system filters out the end effects almost perfectly. Since the axial direction is, in our example, the x-axis, we attribute the observation of smaller end effects to simulating a measurement for a curved motor unit. For the domain with the $10mm$ thick fat layer, the algorithm could identify the motor unit pretty when using the double differential filter, see Figure 6.13. For the domain with the $5mm$ thick fat layer, we could not detect a viable difference between the solution, which is why we decided to plot only one of them. This observation is not surprising, since the identification works already pretty well without the double differential detection system.

6.5 Comparing Different Electrode Setups

Our next example analyzes the influence of the electrode positions. The foundation for the different electrode configurations builds the 5 by 13 electrode grid used for the first example. As we have seen in the first example the identification works pretty well with this setup. Since the computational effort for precomputing the impulse response functions is proportional to the number of electrodes, the question that now arises is if it is possible to reduce the number of electrodes. Therefore, we solve the optimization problem four times with different electrode girds and compare the solutions to the one from the first example.

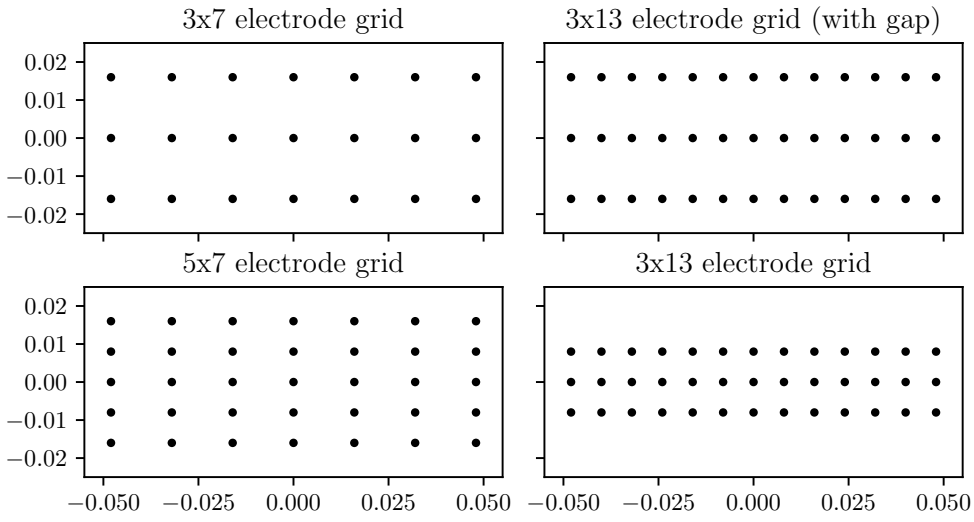


Figure 6.14: Different electrode setups

Before we compare the computed solutions, we shortly describe the different electrode setups visualized in Figure 6.14. The first configuration (top left) uses only every second electrode, i.e., the distance between the electrodes is doubled from $8mm$ to $16mm$, but the same area is covered. Thus, this setup contains 3×7 electrodes. The second configuration (bottom left) uses every second row in the electrode grid, resulting in 5×7 electrodes. For this setup, the distance between two electrodes is $8mm$ in the y-direction and $16mm$ in the x-direction. The third example (top right) uses then every second column such that we get 3×13 electrodes. Here, the distances between two electrodes are swapped, i.e., the distance is $8mm$ in the x-direction and $16mm$ in the y-direction. The fourth and final configuration (bottom right) uses the three middle rows of the electrode grid resulting also in 3×13 electrodes. As in the first example the distance between two electrodes is $8mm$ in all spacial directions.

The first trajectories we want to compare to the solution computed with the "full" electrode grid, are the solutions computed with the first and second configurations described above. Using the first setup (3×7 electrodes) to compute a solution, results in a trajectory that is close to the reference trajectory but the algorithm can not correctly identify the curvature, cf. red trajectory in Figure 6.15. And also the changing depth is identified slightly worse than in the first example, but still in an acceptable error margin. Comparing the solution (magenta trajectory) of the second configuration (5×7 electrodes) with the solution of the first example (blue trajectory), we notice that the identification works pretty well on the right side of the measurements where the trajectory is closer to the electrodes, but on the left side it has difficulties to identify the curvature. Furthermore, the identification of the depth is as good as in the first example.

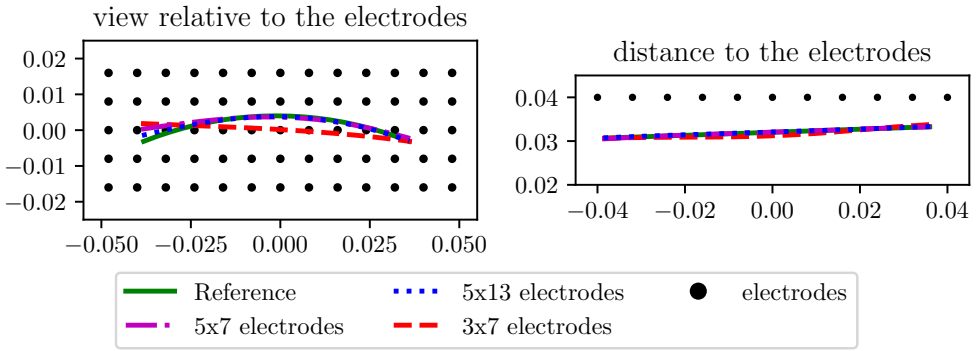


Figure 6.15: Solutions for the first and second electrode setup

Next, we compare the solutions of the third and fourth setups to the solution of the first example (blue trajectory). The trajectory identified with the third configuration (3×13 electrodes with a gap) is similar to the solution of the first electrode configuration, but at least on the right side of the measurements, the curvature is identified a little bit better, cf. red trajectory in Figure 6.16. Using the fourth configuration (3×13 electrodes

without a gap)results in a solution (magenta trajectory) that is similar to the solution of the first example.

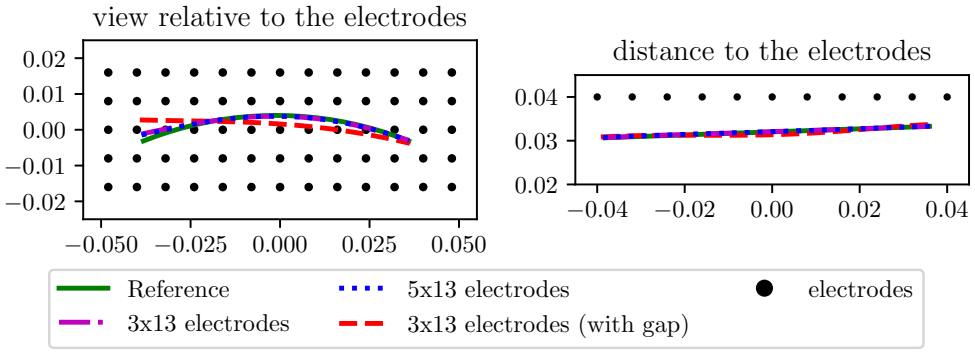


Figure 6.16: Solutions for the third and fourth electrode setup

Analyzing these four examples, we can conclude that not all electrodes from the initial electrode grid are required for a satisfying identification but the position is important. So far we can say that the top and bottom rows of electrodes are not necessary for a good identification. Comparing the second, third, and fourth examples, we may conclude that it is maybe possible to use every second electrode in the three middle rows, cf. Figure 6.17. Thus, we computed a fifth solution where we used this setup. Figure 6.17 shows that the identified trajectory is quite good but not as good as in the first example or as for the fourth electrode configuration. But we notice that there is the possibility to reduce the distance between the electrodes and still cover the whole trajectory.

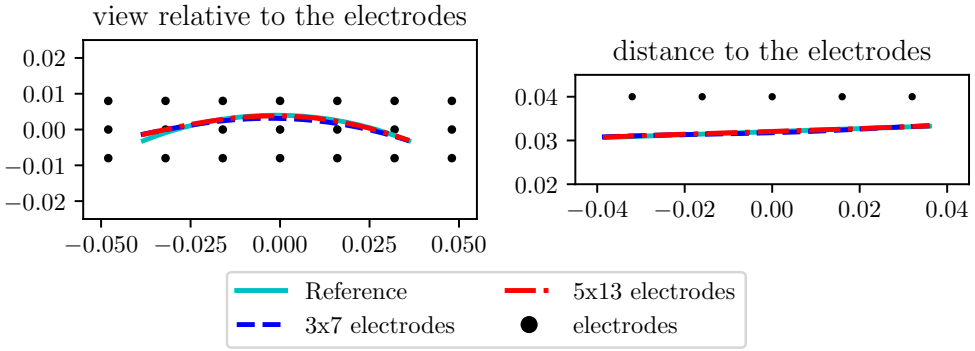


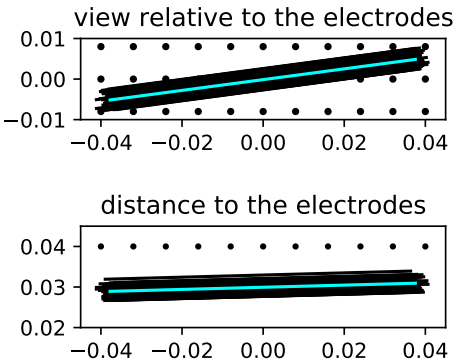
Figure 6.17: Solution for the fifth electrode setup

Since using 49 electrodes provided the same solution as using 65 electrodes, we can conclude that is possible to identify the motor unit with much less electrodes than are in the concrete measure device, namely 117 (see Figure 6.1b). Reducing the number of electrodes to 21 provided also a satisfying solution and there is also the possibility

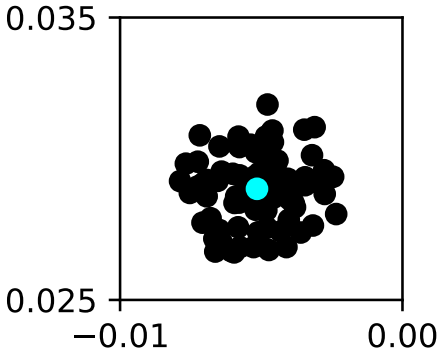
to optimize the distance between the electrodes. Unfortunately, the reduction of the number of electrodes has only a minor effect on the computation time. On the one hand, it indeed reduces the time required to compute a Newton step from $8.15s$ (65 electrodes) to $4.6s$ (49 electrodes) respective $4.2s$ (21 electrodes), but on the other hand, the required number of Newton steps increased. Thus the total computation time is for all three setups approximately the same and lies between 240 and $260s$. Nevertheless, it can still be a good idea to reduce the number of electrodes, since computing the impulse response functions is numerically expensive, i.e. if we use higher-order finite elements.

6.6 Identifying a Multi-Fiber Motor Unit

When we modeled simulating a surface EMG measurement, we mentioned that a motor unit can consist of several hundred motor units. But for simplification, we used the tissue properties to apply the superposition principle such that we can represent a motor unit by a single-fibered motor unit, see Section 2.1. This example shall now verify that this is a valid assumption and that we can identify the center of mass of a multi-fiber motor unit with our approach. Therefore, we selected randomly 100 muscle fibers and simulated for each one a surface EMG measurement. To get a realistic distribution of muscle fibers, we oriented us on the description of a motor unit given in [26]. The x-coordinate of the neuromuscular junction is thereby Gaussian distributed with a mean value of 0 and a standard deviation of $0.0012m$. The y and z-coordinates are then uniformly distributed in a circle with center $(0., 0.03)m$ and a radius of $0.003m$. The speed of the moving action potentials is chosen Gaussian distributed with a mean value of $3.8m/s$ and a standard deviation of $0.02m/s$. Additionally, we rotated the motor unit around the point $(0., 0., 0.)m$. Figure 6.18a shows the distribution of the 100 muscle fibers (black) and the center of mass (cyan) in the axial direction, once relative to the electrodes and once in the xz-plane visualizing the depth. Furthermore, Figure 6.18b visualizes a slice through the motor unit and shows the distribution of the neuromuscular junctions.



(a) Views of the motor unit in axial direction



(b) Slice through the motor unit

Figure 6.18: Multi-Fiber motor unit (black) and its center of mass (cyan)

Since we can apply the superposition principle, we get the simulated measurement of the motor unit as the sum of all measurements. To compensate for the fact that we simulated a measurement for more than one muscle fiber, we multiplied the source term ρ_i by the number of muscle fibers. We then solved the optimization problem with the same parameters as in the first example. As Figure 6.19 shows, the trajectory of the center of mass of the multi-fiber motor unit and the identified trajectory match perfectly. Thus, we can conclude that our assumption to model a single-fibered motor unit is justified.

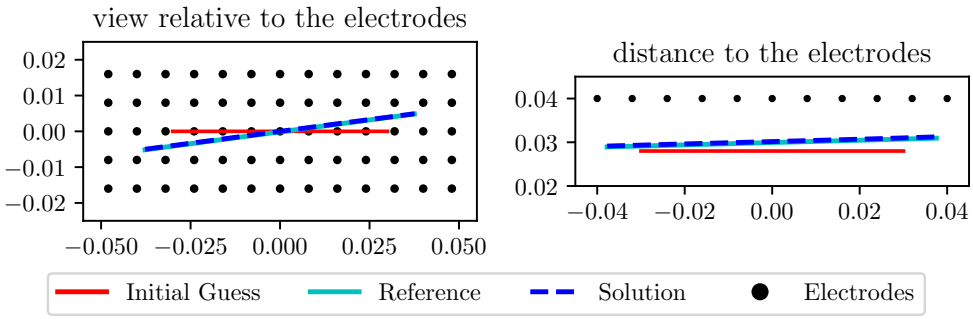


Figure 6.19: Solution for identifying the center of mass of a multi-fiber motor unit

Last but not least, we want to analyze if using a multi-fiber motor unit as a reference has any influence on the convergence rate. When comparing the convergence rates in Figure 6.20 with the convergence rates from the first example (Figure 6.5), we notice a similar behavior, such that we can conclude that using multiple muscle fibers does not increase the difficulty of the problem in a notable fashion.

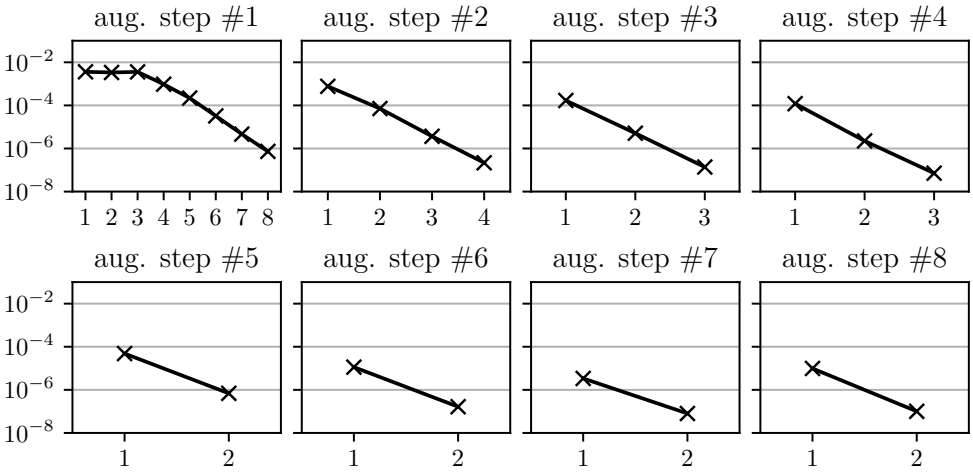
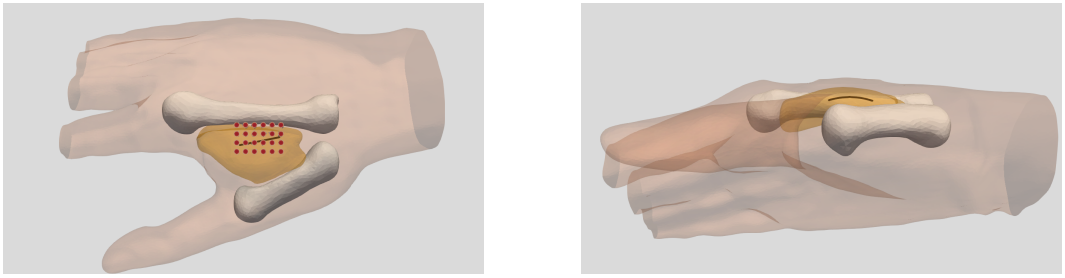


Figure 6.20: Energy Norm of δv for identifying a multi-fiber motor unit

6.7 Identifying a Motor Unit in the FDI Muscle

The last example of this chapter studies the identification of a motor unit in more complex geometry, namely the geometry of the right hand which we already used for the simulation in Section 2.3. But for this example, we choose a trajectory that approximates the shape of the FDI Muscle instead of the straight line used for the first simulation example in Section 2.3. Figure 6.21a shows a view of the hand where we look at the back of the hand and it shows the trajectory (black) relative to the electrodes (red). The second picture, see Figure 6.21b, shows the hand from a perspective where we look at the thumb. In this picture, we can see the depth of the motor unit, i.e. the distance to the electrodes. Furthermore, we can clearly see that the trajectory follows the curvature of the FDI muscle (shown in orange).



(a) Look at the back of the hand

(b) Look at the thumb

Figure 6.21: Two different perspectives of the hand geometry

Similar to the first and sixth examples, the motor unit is identified pretty well. As Figure 6.22 shows, the computed trajectory (blue) matches nearly perfectly with the reference trajectory (cyan) if comparing them relative to the electrodes. Furthermore, we also notice only a small deviation comparing the distance to the electrodes.

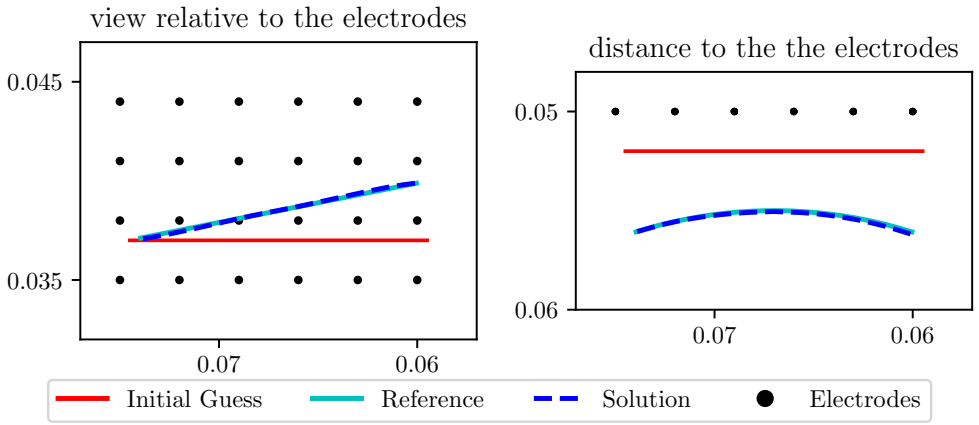


Figure 6.22: Solution for identifying a motor unit in a hand

Looking at the convergence rates in Figure 6.23, we see a similar behavior as for the first example. As before we can only observe fast local convergence for the first pass of the Newton method. From the second pass on, the Newton method requires then, similar to the previous examples, only three to five steps.

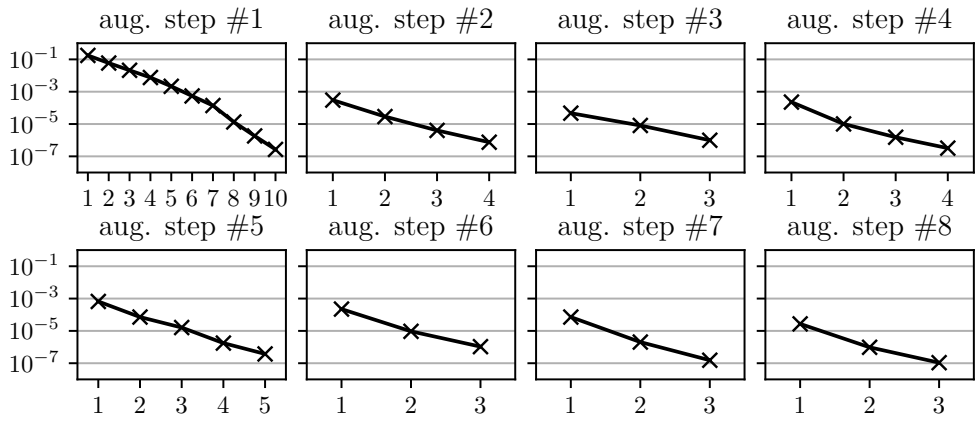


Figure 6.23: Energy Norm of δv for identifying a motor unit in a hand

When comparing the behavior of the augmented Lagrangian algorithm in this setting with the one from the first example, we notice that the norm of the constraint function converges slower to zero, cf. Figures 6.24 and 6.6. Consequently, the penalty parameter is increased faster, i.e. the algorithm updates the Lagrangian parameter first and increases then the penalty parameter three times. Afterward, the last four augmented steps update only the Lagrange multiplier. But the required amount of augmented steps is similar underlying the observation that the trajectory is identified in the first step of the augmented Lagrangian algorithm and the following steps only update the parametrization such that it matches the constraints.

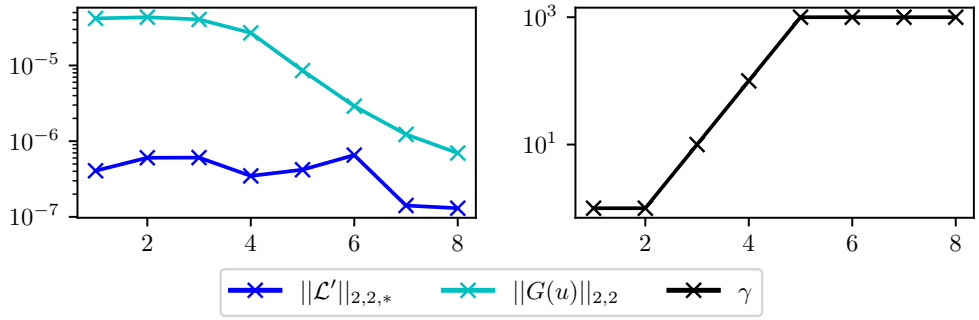


Figure 6.24: Key data of the augmented Lagrangian algorithm in the hand setting

Comparing the Figures 6.25 and 6.7, we notice a similar behavior of the adaptive qua-

drature. This observation underlines the conclusion from the first example that using an adaptive quadrature is justified.

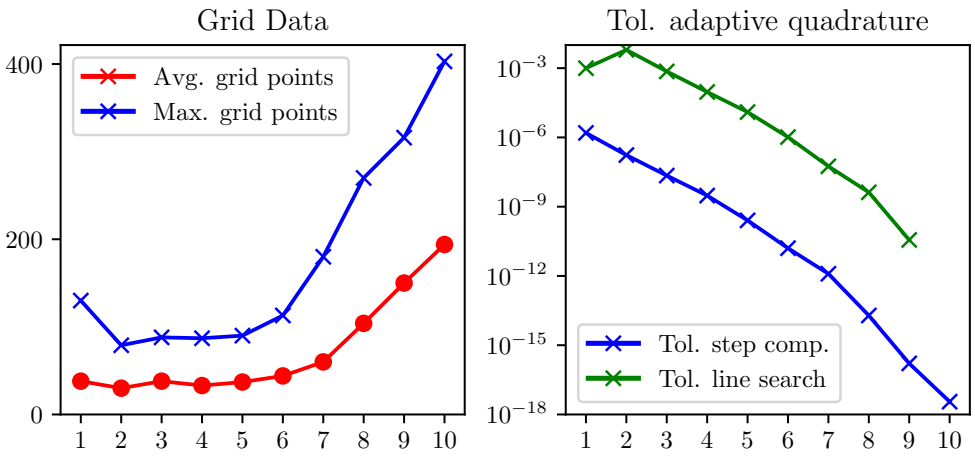


Figure 6.25: Comparison of the average and maximal grid size after using adaptive quadrature

Exact Identification When we discussed the local convergence in Section 4.3, we mentioned that the influence of the impulse response functions is neglectable if the motor unit is identified exactly. Thus, to close this chapter, we solved the hand problem for a straight motor unit and a reference measurement without white noise. As Figure 6.26 shows, we can then observe, for this setting, quadratic convergence for the last three steps, verifying the theoretical consideration.

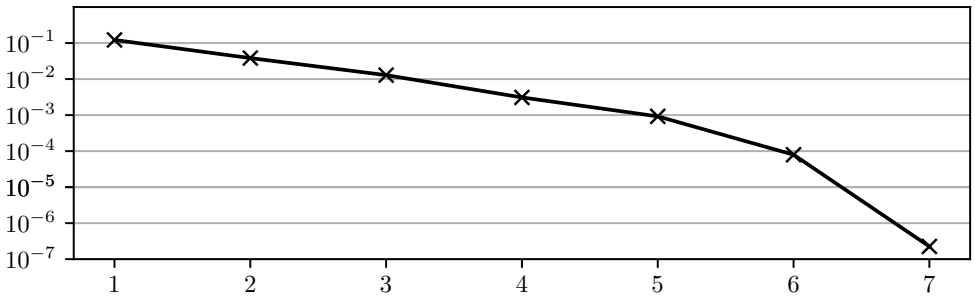


Figure 6.26: Energy Norm of δv when the motor unit is exactly identified

Chapter 7

Conclusion

This thesis derived an optimization problem to identify a motor unit in muscle tissue from a surface EMG measurement. Therefore, we first modeled the activation of motor units, i.e., the propagation of action potentials along the motor units. The motor unit was thereby modeled as a curve $u \in H^2(-t_A, t_A, \Omega_M)$ representing the trajectory of the action potentials. Next, we derived a quasi-static model for the transmission of the electric potential in human tissue. Using these two models, we could simulate a surface EMG measurement by solving a PDE for each time step in a selected measure interval, which is numerically expensive. A decisive step was then using an adjoint approach to enable an efficient simulation of the measured signal by evaluating a simple line integral. Inserting this forward model into a tracking-type optimization problem allows the numerical identification of a motor unit from a surface EMG measurement. We showed that at least one solution exists for this optimization problem. Furthermore, we derived first-order optimality conditions (3.10) serving as a basis for a computational approach.

Our optimization algorithm is based on an augmented Lagrangian method. We then used a Newton method to solve the resulting sequence of unconstrained problems. Furthermore, we used adaptive quadrature to efficiently compute the line integral (2.35). Thus, we interpreted the evaluation as an inexact evaluation of quantities requiring algorithmic measurements to ensure global convergence. Consequently, we had to incorporate these algorithmic measurements into the Newton method.

From the examples, we can conclude that our optimization algorithm is capable of identifying motor units from surface EMG measurements in an efficient way, at least in the context of synthetically generated measurements. The examples also showed that a couple of factors exist that influence the identification in different ways. First, there are algorithmic parameters that we can control, e.g., the regularity parameter α_2 or the accuracy parameter ε regulating the accuracy of the search directions. On the other hand, anatomical and experimental aspects exist that we can control only to some degree or that are uncontrollable. We can, for example, vary the position of the electrodes

on the skin. But their possible locations are predefined by the anatomy of the patient. Furthermore, the fourth example showed that the tissue properties, i.e., the thickness of the fat layer, influence the measurement and thus also on the identification. This is an example for an aspect we can not control, but we presented a possibility to deal with this aspect.

Although we covered the identification in detail, a few open questions still exist. As the first example has shown, the quality of the impulse response functions has a non-negligible influence on the convergence of the Newton method. Thus, their influence should be examined in more detail, and the computation of higher-order functions should be accelerated, e.g., by using a hierarchical basis in the polynomial degree with a suitable preconditioner (cf., e.g., [40]). Also, testing the algorithm with actual measurements should be the objective of further studies. When dealing with actual measurements, we must also consider the tissue properties. Although their values are given in the literature, they can vary depending on the frequency of the action potentials. Thus, the simulated measurement contains uncertainties that should be considered when identifying a motor unit.

Another uncertainty is the velocity ν with which the action potential propagates along the motor unit. For our examples, we assumed that we know this velocity, but in practice, it will be unknown. Thus, box constraints would be more appropriate than equality constraints. Another possibility would be to add the velocity to the variables we want to identify. Both options complicate the optimality problem requiring further analysis.

Bibliography

[1] Robert A. Adams and John J. F. Fournier. *Sobolev spaces*, volume 140 of *Pure and applied mathematics*. Acad. Press, Amsterdam, 2. edition, 2003.

[2] Walter Alt. *Nichtlineare Optimierung: Eine Einführung in Theorie, Verfahren und Anwendungen*. Studium Aufbaukurs Mathematik. Vieweg + Teubner, Wiesbaden, 2., überarbeitete und erweiterte auflage edition, 2011.

[3] S. Andreassen and A. Rosenfalck. Relationship of intracellular and extracellular action potentials of skeletal muscle fibers. *Critical Reviews in Bioengineering*, 6(4):267–306, 1981.

[4] Jürgen Appell and Petr P. Zabrejko. *Nonlinear superposition operators*, volume 95 of *Cambridge tracts in mathematics*. Cambridge Univ. Press, Cambridge, 1990.

[5] Peter Bastian, Markus Blatt, Andreas Dedner, Nils-Arne Dreier, Christian Engwer, René Fritze, Carsten Gräser, Christoph Grüninger, Dominic Kempf, Robert Klöfkorn, Mario Ohlberger, and Oliver Sander. The Dune framework: Basic concepts and recent developments. *Computers & Mathematics with Applications*, 81:75–112, 2021.

[6] Dimitri P. Bertsekas. *Nonlinear programming*. Athena Scientific, Belmont, Mass., 2. ed., 3. print edition, 2008.

[7] Blender Online Community. Blender - a 3D modelling and rendering package. `http://www.blender.org`, 2021.

[8] Haim Brezis. *Functional Analysis, Sobolev Spaces and Partial Differential Equations*. Universitext. Springer Science+Business Media LLC, New York, NY, 2010.

[9] Richard G. Carter. Numerical optimization in hilbert space using inexact function and gradient evaluations. *Technical report 89-45, ICASE, Langley*, 1989.

[10] Richard G. Carter. On the global convergence of trust region algorithms using inexact gradient information. *SIAM Journal on Numerical Analysis*, 28(1):251–265, 1991.

[11] A. Chenciner, S. S. Chern, B. Eckmann, P. de La Harpe, F. Hirzebruch, N. Hitchin, L. Hörmander, M.-A. Knus, A. Kupiainen, G. Lebeau, M. Ratner, D. Serre, Y. G. Sinai, N. J. A. Sloane, J. Tits, B. Totaro, A. Vershik, M. Waldschmidt, M. Berger, J. Coates, S. R. S. Varadhan, Mariano Giaquinta, and Stefan Hildebrandt. *Calculus of Variations I*, volume 310. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[12] Elise de Doncker. An adaptive extrapolation algorithm for automatic integration. *ACM SIGNUM Newsletter*, 13(2):12–18, 1978.

[13] Ron S. Dembo and Trond Steihaug. Truncated-newton algorithms for large-scale unconstrained optimization. *Mathematical programming*, 26(2):190–212, 1983.

[14] James W. Demmel, Stanley C. Eisenstat, John R. Gilbert, Xiaoye S. Li, and Joseph W. H. Liu. A supernodal approach to sparse partial pivoting. *SIAM J. Matrix Analysis and Applications*, 20(3):720–755, 1999.

[15] Peter Deuflhard. Global inexact newton methods for very large scale nonlinear problems. *IMPACT of Computing in Science and Engineering*, 3(4):366–393, 1991.

[16] Peter Deuflhard. *Newton Methods for Nonlinear Problems: Affine Invariance and Adaptive Algorithms*, volume 35 of *SpringerLink Bücher*. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2011.

[17] Jacques Dixmier. *General Topology*. Undergraduate Texts in Mathematics. Springer, New York, NY, 1984.

[18] M. Dobrowolski. *Angewandte Funktionalanalysis: Funktionalanalysis, Sobolev-Räume und elliptische Differentialgleichungen*. Springer, Berlin, 2., korrigierte und überarb. aufl. edition, 2010.

[19] Ivar Ekeland and Roger Temam. *Convex analysis and variational problems*, volume 28 of *Classics in applied mathematics*. SIAM, Philadelphia, unabridged, corr. republ edition, 1999.

[20] R. M. Enoka. Morphological features and activation patterns of motor units. *Journal of clinical neurophysiology : official publication of the American Electroencephalographic Society*, 12(6):538–559, 1995.

[21] Dario Farina, Corrado Cescon, and Roberto Merletti. Influence of anatomical, physical, and detection-system parameters on surface EMG. *Biological Cybernetics*, 86(6):445–456, 2002.

[22] Gaël Guennebaud, Benoît Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010.

[23] Marco Gazzoni, Dario Farina, and Roberto Merletti. A new method for the extraction and classification of single motor unit action potentials from surface EMG signals. *Journal of Neuroscience Methods*, 136(2):165 – 177, 2004.

[24] Christophe Geuzaine and Jean-François Remacle. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009.

[25] David Gilbarg and Neil S. Trudinger. *Elliptic partial differential equations of second order*. Classics in mathematics. Springer, Berlin, reprint of the 1998 ed. edition, 2001.

[26] T.H.J.M. Gootzen, D. F. Stegeman, and A. van Oosterom. Finite limb dimensions and finite muscle length in a model for the generation of electromyographic signals. *Electroencephalography and Clinical Neurophysiology/Evoked Potentials Section*, 81(2):152–162, 1991.

[27] Sebastian Götschel, Martin Weiser, and Anton Schiela. Solving Optimal Control Problems with the Kaskade7 Finite Element Toolbox. In *Advances in Dune*, pages 101–112. Springer, Berlin, 2012.

[28] Roberta Grech, Tracey Cassar, Joseph Muscat, Kenneth P. Camilleri, Simon G. Fabri, Michalis Zervakis, Petros Xanthopoulos, Vangelis Sakkalis, and Bart Vanrumste. Review on solving the inverse problem in EEG source analysis. *Journal of neuroengineering and rehabilitation*, 5:25, 2008.

[29] David J. Griffiths. *Introduction to electrodynamics*. Always learning. Pearson, Boston, 4. ed., international ed. edition, 2013.

[30] Wolfgang Hackbusch. *Elliptic Differential Equations*, volume 18. Springer Berlin Heidelberg, Berlin, Heidelberg, 2017.

[31] R. Haller-Dintelmann, C. Meyer, J. Rehberg, and A. Schiela. Hölder Continuity and Optimal Control for Nonsmooth Elliptic Problems. *Appl Math Optim*, 60(3):397–428, 2009.

[32] George W. Hanson and Alexander B. Yakovlev. *Operator theory for electromagnetics: An introduction*. Springer, New York, NY, 2002.

[33] Matthias Heinkenschloss and Luis N. Vicente. Analysis of inexact trust-region sqp algorithms. *SIAM Journal on Optimization*, 12(2):283–302, 2002.

[34] Michael Hinze, Rene Pinnau, Michael Ulbrich, and Stefan Ulbrich. *Optimization with PDE constraints*, volume 23 of *Mathematical modelling*. Springer, Dordrecht, 2009.

[35] A. Holobar, M. A. Minetto, and D. Farina. Accurate identification of motor unit discharge patterns from high-density surface emg and validation with a novel signal-based performance metric. *Journal of Neural Engineering*, 11(1):016008, 2014.

[36] Andreas Kirsch and Frank Hettlich. *The mathematical theory of time-harmonic Maxwell's equations: Expansion-, integral-, and variational methods*, volume 190 of *Applied Mathematical Sciences*. Springer, Cham, 2015.

[37] Bert U. Kleine, Johannes P. van Dijk, Bernd G. Lapatki, Machiel J. Zwarts, and Dick F. Stegeman. Using two-dimensional spatial information in decomposition of surface EMG signals. *Journal of Electromyography and Kinesiology*, 17(5):535–548, 2007.

[38] D. P. Kouri, M. Heinkenschloss, D. Ridzal, and B. G. van Bloemen Waanders. Inexact objective function evaluations in a trust-region algorithm for pde-constrained optimization under uncertainty. *SIAM Journal on Scientific Computing*, 36(6):A3011–A3029, 2014.

[39] Aleksandr S. Kronrod. *Nodes and Weights of Quadrature Formulas.* Consultants Bureau Enterprises, 1965.

[40] Sabine Le Borne. Hierarchical preconditioners for high-order FEM. In *Domain decomposition methods in science and engineering XXII*, volume 104 of *Lect. Notes Comput. Sci. Eng.*, pages 559–566. Springer, Cham, 2016.

[41] C J Lin and J J More. Incomplete Cholesky factorizations with limited memory. *SIAM Journal on Scientific Computing*, 21, Sep 1999.

[42] Yang Liu, Yong Ning, Sheng Li, Ping Zhou, William Z. Rymer, and Yingchun Zhang. Three-Dimensional Innervation Zone Imaging from Multi-Channel Surface EMG Recordings. *International Journal of Neural Systems*, 25(6):1550024, 2015.

[43] M. M. Lowery. EMG Modeling and Simulation: 8. In *Surface Electromyography : Physiology, Engineering, and Applications*, pages 210–246. John Wiley & Sons, Ltd, 2016.

[44] Madeleine M. Lowery, Nikolay S. Stoykov, Allen Taflove, and Todd A. Kuiken. A multiple-layer finite-element model of the surface EMG signal. *IEEE Transactions on Biomedical Engineering*, 49(5):446–454, 2002.

[45] Jaakko Malmivuo and Robert Plonsey. *Bioelectromagnetism: Principles and applications of bioelectric and biomagnetic fields.* Oxford Univ. Press, New York, NY, 1995.

[46] Luca Mesin. Real time identification of active regions in muscles from high density surface electromyogram. *Computers in Biology and Medicine*, 56:37–50, 2015.

[47] Christian Meyer, Lucia Panizzi, and Anton Schiela. Uniqueness Criteria for the Adjoint Equation in State-Constrained Elliptic Optimal Control. *Numerical Functional Analysis and Optimization*, 32(9):983–1007, 2011.

[48] Jorge Nocedal and Stephen J. Wright. *Numerical optimization.* Springer series in operations research. Springer, New York, NY, 1999.

[49] Diego Pereira Botelho, Kathleen Curran, and Madeleine M. Lowery. Anatomically accurate model of EMG during index finger flexion and abduction derived from diffusion tensor imaging. *PLOS Computational Biology*, 15(8):e1007267, 2019.

[50] R. Piessens. *Quadpack: A Subroutine Package for Automatic Integration*, volume v.1 of *Springer Series in Computational Mathematics Ser.* Springer Berlin / Heidelberg, Berlin, Heidelberg, 1983.

[51] R. Plonsey and D. B. Heppner. Considerations of quasi-stationarity in electrophysiological systems. *The Bulletin of Mathematical Biophysics*, 29(4):657–664, 1967.

[52] P. Rosenfalck. Intra- and extracellular potential fields of active nerve and muscle fibres. A physico-mathematical analysis of different models. *Acta Physiologica Scandinavica. Supplementum*, 321:1–168, 1969.

[53] Guido Stampacchia. Le problème de Dirichlet pour les équations elliptiques du second ordre à coefficients discontinus. *Annales de l'institut Fourier*, 15(1):189–257, 1965.

[54] Dick F. Stegeman, Joleen H. Blok, Hermie J. Hermens, and Karin Roeleveld. Surface EMG models: properties and applications. *Journal of Electromyography and Kinesiology*, 10(5):313–326, 2000.

[55] Nikolay S. Stoykov, Madeleine M. Lowery, Allen Taflove, and Todd A. Kuiken. Frequency- and time-domain fem models of emg: capacitive effects and aspects of dispersion. *IEEE transactions on bio-medical engineering*, 49(8):763–772, 2002.

[56] Fredi Tröltzsch. *Optimal control of partial differential equations: Theory, methods, and applications*, volume 112 of *Graduate studies in mathematics Applied mathematics.* American Math. Soc, Providence, RI, 2010.

[57] Kees van den Doel, Uri M. Ascher, and Dinesh K. Pai. Computed myography: three-dimensional reconstruction of motor functions from surface EMG data. *Inverse Problems*, 24(6):065010, 2008.

[58] Kees van den Doel, Uri M. Ascher, and Dinesh K. Pai. Source localization in electromyography using the inverse potential problem. *Inverse Problems*, 27(2):025008, 2011.

[59] J. P. van Dijk, M. M. Lowery, B. G. Lapatki, and D. F. Stegeman. Evidence of Potential Averaging over the Finite Surface of a Bioelectric Surface Electrode. *Annals of Biomedical Engineering*, 37(6):1141–1151, June 2009.

[60] Eberhard Zeidler. *Nonlinear Functional Analysis and its Applications: III: Variational Methods and Optimization.* Springer New York, New York, NY and s.l., 1985.

[61] Eberhard Zeidler. *Nonlinear Functional Analysis and its Applications: II/B: Nonlinear Monotone Operators.* Springer New York, New York, NY and s.l., 1990.

[62] J. Carsten Ziems and Stefan Ulbrich. Adaptive multilevel inexact sqp methods for pde-constrained optimization. *SIAM Journal on Optimization*, 21(1):1–40, 2011.

[63] J. Zowe and S. Kurcyusz. Regularity and stability for the mathematical programming problem in Banach spaces. *Appl Math Optim*, 5(1):49–62, 1979.

Own Publications

[64] Tobias Sproll and Anton Schiela. An adjoint approach to identification in electromyography: Modeling and first order optimality conditions. *Inverse Problems*, 2021.

[65] Tobias Sproll and Anton Schiela. Numerical solution of an identification problem in electromyography, 2023. `https://epub.uni-bayreuth.de/id/eprint/7068`.

[66] Tobias Sproll, Anton Schiela, and Madeleine Lowery. Numerical identification of motor units using an optimal control approach. *IFAC-PapersOnLine*, 51(2):174–179, 2018. 9th Vienna International Conference on Mathematical Modelling.

List of Figures

List of Algorithms

List of Tables

Eidesstattliche Versicherung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderene als die von mir angegebenen Quellen und Hilfsmittel verwendet habe.

Weiterhin erkläre ich, dass ich die Hilfe von gewerblichen Promotionsberatern bzw. -vermittlern oder ähnlichen Dienstleistern weder bisher in Anspruch genommen habe, noch künftig in Anspruch nehmen werde.

Zusätzlich erkläre ich hiermit, dass ich keinerlei frühere Promotionsversuche unternommen habe.

Bayreuth, den