TU WIEN

MASTER THESIS

Automated Calculation of Optimal Adjustment Parameters for Myoelectric Hand Prostheses

Author:
Sigrid GERGER
Matr. No.: 01225874

Supervisor:
Ao.Univ.Prof. Dipl.-Ing. Dr. techn. Günther RAIDL

A thesis submitted in fulfillment of the requirements for the degree of Master of Science

in the

Algorithms and Complexity Group
Institute of Logic and Computation

January 8, 2019

Declaration of Authorship

I, Sigrid GERGER, declare that this thesis titled, "Automated Calculation of Optimal Adjustment Parameters for Myoelectric Hand Prostheses" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

"Thanks to my legs, that they walk on my floor,
Thanks to my lungs, they breath air,
Thanks to my mind, it is thinking my thoughts,
Thanks to my wounds, that they heal,
Thanks to my heart, it is pumping my blood,
Thanks to my soul, that it cares."

Sue

Abstract

Automated Calculation of Optimal Adjustment Parameters for Myoelectric Hand Prostheses

by Sigrid GERGER

The adjustment of myoelectric arm prostheses is a very sensitive topic, as improper choice of values for settable parameters causes dysfunction of the device and frustration and rejection of the user. Due to the complexity of the system and the rare occurrence of myoelectric treatments, the state of the art approach of manual prosthesis adjustment leads to frequent failure, which is the reason why it was sensible to come up with a way of supporting orthopedic technicians by developing a reliably aiding algorithmic tool with a convenient user interface for data collection and further parameter calculation. Therefore a prototype software has been set up during a six-months research project completed at *Otto Bock Holding GmbH & Co. KG* and refined in the following years at *TU Wien*, which guides both user and orthopedic technician through a process of data recording by giving a scheduled instruction on movement generations and furthermore calculates required parameters based on the user's prevailing abilities.
This thesis presents these newly developed ideas to do so by firstly introducing basic knowledge about prosthetic treatment and adjustment from its anatomical and technical point of view in Chapter 1 and 2. Furthermore, as developed approach is based on ideas formulated in terms of mathematical optimization, an overview on the most important definitions and techniques of latter is given in Chapter 3. After providing these basics for further understanding, Chapter 4 presents a mathematically precise description of given prosthetic system and its functionality components. Chapter 5 finally describes the idea of making use of mathematical optimization for a problem solution by formulating given task as optimization problem, capable of being solved algorithmically.
By realizing a first, naive, enumerative implementation derived in Section 5.3.1.2 and enhancing it towards a generalized formulation, being very flexible and easily expandable to more complex systems, additionally turning capable of being solved by commercially available optimization solvers (Section 5.3.1.6), these two approaches could be analyzed and compared with each other and the former, manual approach. Section 6.4 registers, that both the user feedback and the numerical outcome showed great success of both algorithmic versions, while the mathematical interpretation expectedly indicated a significantly better result of the generalized, advanced implementation compared to the naive, enumerative method.
A summary of the previously presented findings can be found in conclusive Chapter 7, which also postulates encouraging reasons and motivating arguments about why this topic deserves further attention due to both the technical and the humanitarian side.

Keywords: Myoelectric prosthetic treatment; Michelangelo hand prosthesis; Prosthetic parameter adjustment; Algorithmic calculation of settable parameters; Mathematical optimization; Constraint linearization

Automatisierte Berechnung optimaler Einstellungsparameter myoelektrischer Handprothesen

Die Einstellung myoelektrischer Handprothesen ist ein sehr sensibles Thema, da eine falsche Wahl der einzustellenden Parameter zu Fehlfunktionen der Prothese und dementsprechend zu Frustration und Ablehnung der Anwender führt. Aufgrund der Komplexität des Systems und des seltenen Auftretens myoelektrischer Versorgungen kommt es im Zuge der gängigen Methode der manuellen Protheseneinstellung regelmäßig zu Fehlern, weswegen es sinnvoll wurde, eine Möglichkeit der verlässlichen, algorithmischen, benutzerfreundlichen Unterstützung für Orthopädietechniker zu entwickeln, um relevante Daten sammeln und in weiterer Folge passende Parameter berechnen zu können. Darum wurde im Rahmen eines sechs Monate dauernden Projekts bei *Otto Bock Holding GmbH & Co. KG* eine experimentelle Software entwickelt, die Orthopädietechniker und Prothesenanwender gemeinsam durch einen Prozess der gezielten Datenaufnahme führt und in weiterer Folge notwendige Parameter aufgrund der zuvor eruierten Fähigkeiten des Anwenders errechnet.
Diese Diplomarbeit präsentiert eine Lösung des beschriebenen Problems, indem zuerst die Grundlagen der Prothesenversorgung und -einstellung aus anatomischer und technischer Sicht in Kapitel 1 bzw. Kapitel 2 erörtert werden. Weiters, da die hier entwickelte Herangehensweise auf der Idee der mathematischen Optimierung basiert, werden in Kapitel 3 grundlegende Begriffe und Techniken dieses Gebiets vorgestellt. Nachdem genanntes Basiswissen für das weitere Verständnis zur Verfügung gestellt wurde, präsentiert Kapitel 4 eine mathematisch präzise Formulierung des vorliegenden Handprothesensystems und deren Funktionalitäten. Kapitel 5 beschreibt schlussendlich die Idee, wie mathematische Optimierung als Werkzeug eingesetzt werden kann, um vorliegende Problemstellung algorithmisch zu lösen.
Indem eine Lösung zuerst als naives Aufzählungsverfahren implementiert und in weiterer Folge zu einer verallgemeinerten, flexiblen und leicht erweiterbaren Form umformuliert wurde, die zusätzlich die Benützung von kommerziellen Lösungsprogrammen möglich macht, konnten genannte Ansätze miteinander und mit der ursprünglichen, manuellen Variante verglichen werden.
Kapitel 6.4 hält fest, dass sowohl das Feedback der Anwender als auch die numerischen Daten große Erfolge beider algorithmischer Versuche aufzeigen, während aus der Interpretation der numerischen Ergebnisse sogar eine signifikante Verbesserung des verallgemeinerten, flexibleren Ansatzes im Vergleich zu ursprünglichem, aufzählenden Verfahren zu folgern ist.
Kapitel 7 bietet eine Zusammenfassung der zuvor erörterten Resultate und liefert weiters Motivation und Gründe für die Wichtigkeit der weiteren Forschung auf diesem Gebiet in der Zukunft.

Schlüsselwörter: Myoelektrische Prothesenversorgung; Michelangelo Prothese; Einstellung von myoelektrischen Handprothesen; Algorithmische Berechnung von Parametern; Mathematische Optimierung; Constraint-Linearisierung

Acknowledgements

Foremost, my sincere thanks goes to Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Günther Raidl, who undertook the supervision of this thesis, despite my unconventional scheduling of the whole master study. With incredible patience and overwhelming interest he supported me in giving my ideas a mathematical form and making this project one last achievement within my studies of technical mathematics, which I worked through the last years with great passion and fascination.

Furthermore, I would like to thank Sebastian Amsüss, who used to be my mentor during my internship at *Ottobock*. From the first day on he taught me the amazing topic of myoelectric prostheses and their technical aspects, passed on his interest and creativity to me and was always there when I was in need for any programming advice.

Special thanks goes to Gernot Lampl, who let me watch his magnificent work at *Orthopädietechnik Schmied*, let me help building prostheses with my own hands and introduced me to many fascinating people and their life stories, who inflamed my desire to continue working in the medical field.

Heartfelt thanks go to my beloved fellow students, whom I spent uncountable hours with. What began as study group, ended up in true and beautiful friendship I will be grateful for my entire life.

Last, but definitely not least, I am profoundly grateful to my parents and my family Roswitha, Katrin and Julia, who were always there for me, helped me through the pressure I used to put on myself and taught me to start trusting in who I am and what I can. I want to thank Selina for helping me out of so many difficult situations by her unbelievably wise and great advises, and finally true gratefulness to my friends in Greece, who taught me the δ.

Contents

List of Figures

List of Tables

Dedicated to my parents, my sister and Julia, who put more in my mind than they should, and more in my heart than they think.

Part I

Introduction

"Die Hand ist offenbar nicht nur ein Werkzeug, sondern viele. Sie ist gewissermaßen ein Werkzeug vor allen Werkzeugen. Demjenigen Wesen, das die meisten Fertigkeiten sich anzueignen vermag, hat die Natur also die Hand als das Organ verliehen, das von allen am vielfältigsten verwendbar ist."

- Aristotle (384 - 322 BC)

Health – one of the most fundamental needs of every human being and an essential base in order to make an addition of passion, pleasure, interest and productivity to an individual life possible. If this rooting requirement is not met, it is hardly manageable to keep up a satisfactory way of existence. As people tend to forget about the importance of their physical wellbeing within society's demanding and tiring daily routine, a sudden interruption of wholesomeness, even if it is just of slight extend, is often experienced very intensively and acts as an effective reminder of the importance of appreciation and maintenance of a healthy physical state.

When considering even the easiest activity, not even recognized as actual effort in everyday life, even small impairments of the body can turn out to make latter to a hardly viable or even impossible task. Imagining actions such as getting dressed, opening the water tap or scratching an itching spot, these movements are totally natural and usually of no big conscious concern. These simple movements however turn out to become an unbelievably huge obstacle for someone who is suffering from restrictions of his body's physiological capacity.

An example for one of these severe impacts in a human being's life is the dysfunction or the total loss of a limb. Such an experience does not only change a lot of mechanical and physiological circumstances, but furthermore has a tremendously dramatic impact on the concerned person's mental state and his environment. Often followed by deep desperation and depression, it is also not rarely accompanied by temporary or chronic, actual or phantom limb pain and in many cases, due to the complexity of the neuronal system changed and damaged in these instances, not satisfactorily treatable.

As in the past, after an accident or illness and subsequent necessity of surgical removal of an extremity, it was – and nowadays still is – the commonly considered opinion and practice to surgically preserve as much of the body part as possible, researchers and surgeons also came up with new approaches, concerning amputation techniques which take also account of advantages towards a more harsh removal of remained corporal matter. Both, appearing very beneficial in a lot of cases, as they enable opportunities of treatment the conventional, preserving way of surgery does not offer, but also very disputed concerning the ethical aspect, the newly developed ideas arose high interest in more investigation within the topic of amputation and further treatment.

But not only the initial step of surgical rescue, preservation respectively removal of an insured limb has great influence on the further course of an impacted person. The loss of extremity can be followed up by a series of further smaller or bigger interventions and often entails the necessity of a long and demanding period of physiotherapy and training. As one very promising treatment for upper limb amputees is the one of the myoelectric prosthesis, which's functionality is controlled by the conscious contraction of certain remained muscle areas of the user and resulting myo-signal patterns measured by appropriately placed electrodes, a sophisticated and professional therapy approach must be offered to the user and a sufficient level of motivation towards the desire to put effort into the training must prevail within the concerned person in order to provide him access to this promising tool.

Once latter mentioned circumstances are granted and guarantee an auspicious path towards an enhancement of the user's quality of life, there is still a long list of obstacles to appear potentially. Despite a well-completed physiotherapy and a technically and mechanically perfectly working and suiting myoelectric prosthesis, the final, very sensitive issue of the right setting of parameters for an individual user within the prosthesis internal control unit is often the reason of total failure of the whole treatment. As the settable parameters, defining the way of reaction of the prosthesis regarding the actions of the user, are mostly dependent on latter's muscle performance and thus totally individual and differing from person to person, it is a complex and tough task to determine suitable values from a given data set containing the required information about the user's capability and manner of generating activating movement patterns. In spite of an actually perfectly working device, the wrong choice of adjustment can make the usage of a myoelectric prosthesis impossible. Functionality such as the opening, closing or rotation of the prosthetic hand can become an exhausting or impossible task, degenerating into frustration and the loss of motivation to keep on the training and usage of the treatment. This furthermore does not only lead to a waste of available and enhancing opportunities and thus a decrease of livability, but also to a potential loss of strength and capacity of remaining body parts.

These emergences of failure unfortunately are of no rare character however, as the state-of-the-art method of parameter adjustment in prosthetic treatment is the manual approach of an orthopedic technician, usually with the aid of a graphical user interface provided by the product's fabricator, enabling the technician to visually analyze collected user data and putting his resulting perceptions into the decision of values. This again is very often accompanied by mistakes and failures, on the one hand caused by the complexity of the topic itself, on the other hand due to a widely common lack of experience of orthopedic technicians, which is indeed highly understandable, when the rare occurrence of myoelectric treatment is taken into account.

So the importance of consideration of this topic, arisen from the fact that it has so much impact on an affected person's capability of overcoming a tremendous loss such as the one of the amputation of a part of his own body, encouraged the desire and will to find more reliable methods for the determination of settable parameters in prosthesis adjustment. By receding from the current practice of orthopedic technicians making decisions about the adjustment based on their visual estimate and experience without granting them any more than the basic support of an educational tutorial beforehand, but by rather enhancing old manners by the idea of searching for automated, computer-aided ways using reliable, efficient and verified algorithms for analysis and calculations in order to support the orthopedic technician's process of consideration, ideas towards a more stable way of prosthesis adjustment were gathered and worked on.

This motivated the content of this thesis, based on a six month lasting research work at *Ottobock*[1], which was intended to find alternative, software-grounded ways of myo-prosthesis adjustment, test their potentialities and weaknesses and draw conclusions about the reasonableness of putting effort in further investigation and development towards alternative ways of amputee treatment. Within a working period of half a year, data collection and analysis has been carried out, algorithmic approaches have been come up with and tested for functionality, reliability and efficiency and have resulted in a prototype software, providing a graphical user interface as a visually supporting and executing tool to orthopedic technicians, on the one hand using physiotherapeutical associations, espoused by written instructions and pictures in order to present a guided path of data collection to the user and his treating orthopedic technician, on the other hand making use of the concept of mathematical optimization in order to achieve the postulated goals of proper parameter determination. Furthermore, developed prototype software and its outcome was compared with the current manual adjustment of a series of users in order to draw conclusions about the quality of the developed alternative approach of parameter calculation. By both analyzing the resulted numerical data and gathering user feedback by setting up questionnaires, the opportunities and vices of the newly developed ideas were observed and lead to an interesting insight into the potential of these new considerations.

Aiding as documentational material for carried out research, as well as continuation of the mathematical aspect of the prototype software, which's sophistication was widely neglected due to the initially intended purpose of this project to be of prototype nature, this thesis introduces the topic of amputation by providing the basic terms and knowledge about anatomical aspects in Section 1.1 leading to actual surgical matters in Section 1.2. After a brief overview of several amputation techniques, this section is continued by the explanation of how upper limb treatment can be categorized and controlled.
Chapter 2 aims to deliver more detailed information about the ideas of the usage and control of myoelectric prostheses, as *Ottobock*'s *Michelangelo* prosthesis, which is the one considered within this framework, belongs to the category of latter. By firstly providing understanding of the concept of myoelectric control by giving a specific example, it continues with the final path towards the topic of prosthesis adjustment, its current methods and arising difficulties (Section 2.1). As completion of this introductory chapter, the research project, which is presented in this thesis, is reviewed and the resulted prototype, called *Guided APS-Software*[2], is pictured (Section 2.2). The thesis continues with a discourse, explaining the basic concepts of mathematical optimization in Section 3, showing different types of optimization problems as well as the most common ways to solve them. Furthermore, Section 3.4 gives an insight into the opportunities of reformulating a given optimization problem towards required ways in order to grant a basic knowledge about the techniques which are later used in order to achieve the set goal for this thesis.
Knowledge about given *Michelangelo* prosthesis[3] in mathematical terms is provided in Section 4 as necessary foundation in order to understand following approach to set up an appropriate optimization program for the given task (Section 5).
Starting with non-linear ideas in Section 5.3.1.1, several generalizing considerations lead to a final linear mixed-integer program of the form, previously postulated as

[1] *Otto Bock Holding GmbH & Co. KG*, `https://www.ottobock.com/de/`

[2] *APS*=Automated Prosthesis Setup

[3] prosthetic treatment for transradial amputees, `http://www.ottobock.de/prothetik/produkte-a-bis-z/armprothetik/michelangelo-hand/`

goal formulation for this work (Section 5.3.1.6).
To its end, Section 6 presents the experiment of testing the results of the Guided APS-Software, set up during the six-month project at *Ottobock*, versus the outcome of the approach of linearization, developed as continuation of latter within the framework of this thesis. It gives ideas about the appearance of the optimization program, which was conceived for a chosen prosthesis adjustment program and grants insight into both approaches' implementations (Section 6.3).
A conclusive summary can be read in Section 7, where virtues and vices of old and new considerations are briefly recapitulated and an idea about the meaning of this thesis' revealings and outcomes is given.

Chapter 1

Basics

To lead to the main topic of myoelectric prosthesis adjustment, this chapter introduces the fundamental concepts of prosthesis treatment, beginning from passive, cosmetic prostheses, followed by the more complex concepts of active prostheses such as body-powered prostheses and externally powered myoelectric prostheses. To understand the significance of the different prosthetic systems, a brief overview of basic anatomical and surgical terms will be given in advance and the different forms of amputations and their characteristics in treatment limitations and possiblities are pointed up.

1.1 Anatomy

> *"Die Anatomie [...] zerlegt die Organismen in ihre [...] Bestandteile, untersucht ihre äußeren, sinnlich wahrnehmbaren Eigenschaften und ihre innere Structur [sic], und lernt aus dem Todten, was das Lebendige war. Sie zerstört mit den Händen einen vollendeten Bau, um ihn im Geiste wieder aufzuführen, und den Menschen gleichsam nachzuerschaffen. Eine herrlichere Aufgabe kann sich der menschliche Geist nicht stellen."*[1]

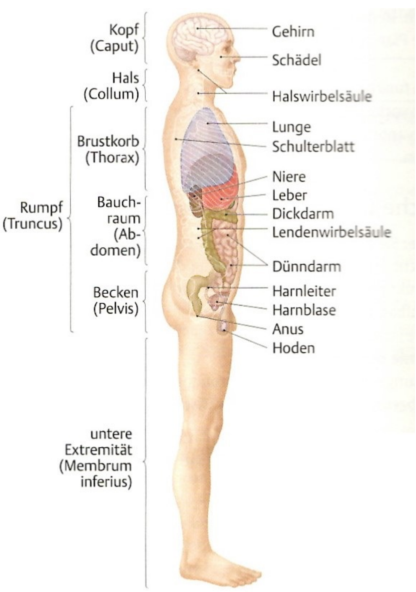

FIGURE 1.1: Rough segmentation of the body [2]

The human body can be roughly divided into head (**Caput**), neck (**Collum**), trunk (**Truncus**), upper limb (**Membrum superius**) and lower limb (**Membrum inferius**) (Figure 1.1).
The upper limb is connected to the trunk by the pectoral girdle (**Cingulum membri superioris**), which itself consists of the collarbone (**Clavicula**) and the blade bone (**Scapula**). The free part of the upper limb (**Pars libera membris superioris**) is segmented into upper arm (**Brachium**), elbow (**Cubitus**), underarm (**Antebrachium**) and hand (**Manus**) (Figure 1.4) [2].

Muscle activity is activated by changes in voltage of the cell membrane of their connected neurons (**action potentials**) [31]. Myoelectric prostheses make use of these voltage changes by measuring EMG signals via surface electrodes placed near well-controllable muscles. Therefore, when it

[1] Joseph Hyrtl, 1811-1894, Anatomist

comes to understanding surgical strategies and following decisions in prosthetic treatment, it is important to comprehend the neuronal and muscular physiology of the upper limb. Figure 1.2 and 1.3 give an overview of the neurons of the upper limb respectively their corresponding muscles.

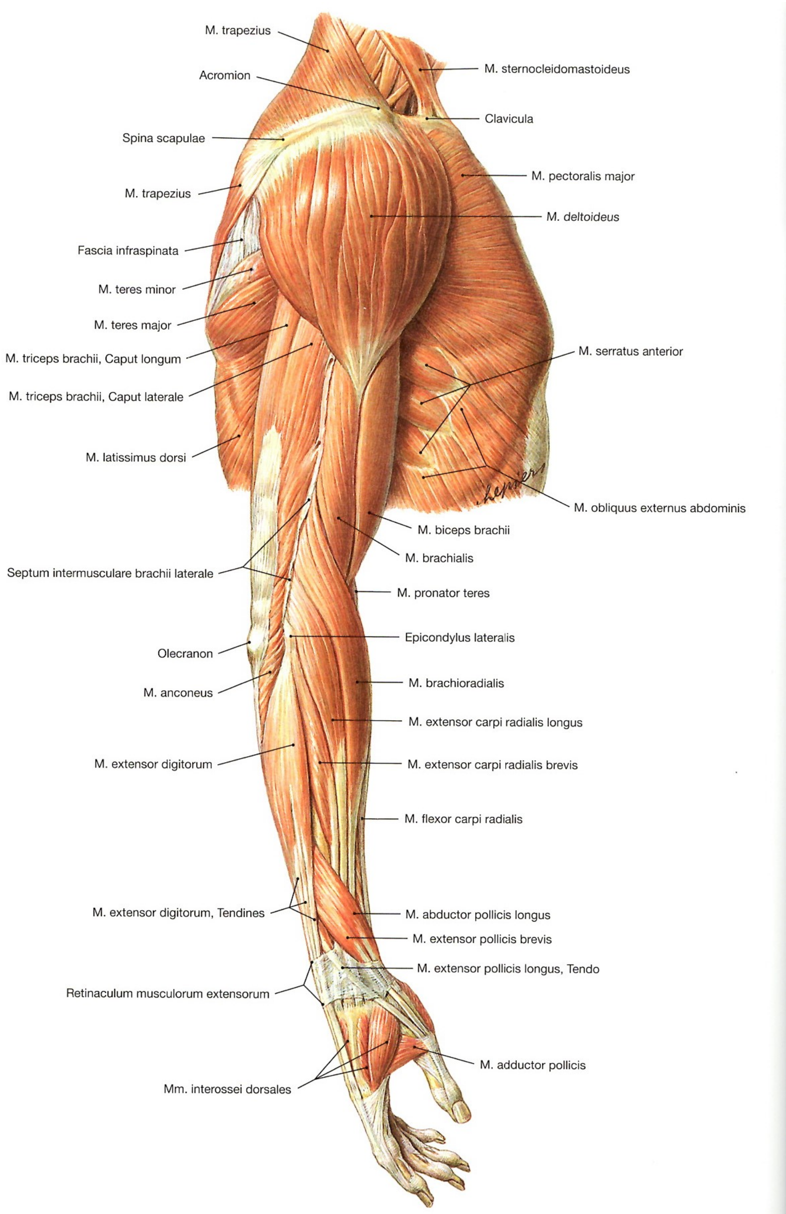

FIGURE 1.2: Upper limb muscles [30]

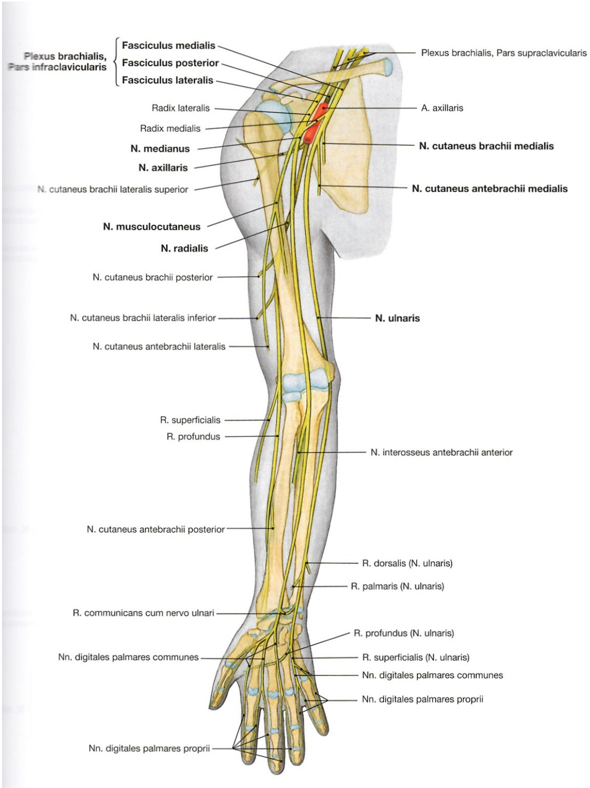

FIGURE 1.3: Upper limb neurons [30]

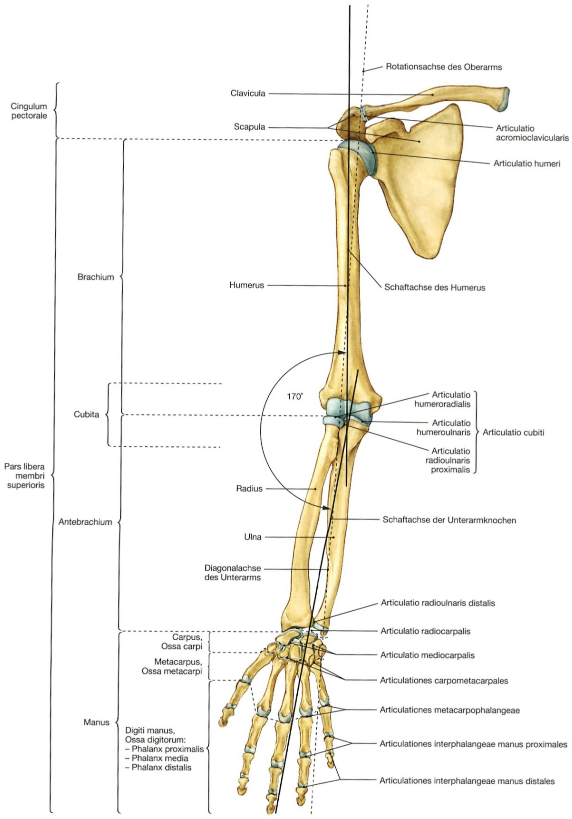

FIGURE 1.4: Upper limb bones [30]

1.1.1 Main Axes, Directions and Locations of the Body Parts

In order to be able to talk about body parts and their locations in an unambiguous way, it is sensible to define certain terms of location and direction. The orientation of the body is divided into three main axes: The **vertical axis** (Figure 1.5 (1)), which proceeds lengthways from vertex to sole (craniocaudal), the **sagittal axis** (Figure 1.5 (3)), proceeding from back to forth through the back and the front body plane and the **transversal axis** Figure 1.5 (2)), which runs from left to right, connecting symmetric parts of the two sides of the body. The main axes define four main planes: the **median plane** (Figure 1.5 (III)) and its parallel shifted **sagital plane**, the **frontal plane** (Figure 1.5 (I)) and the **transversal plane** (Figure 1.5 (II)) ([1]).

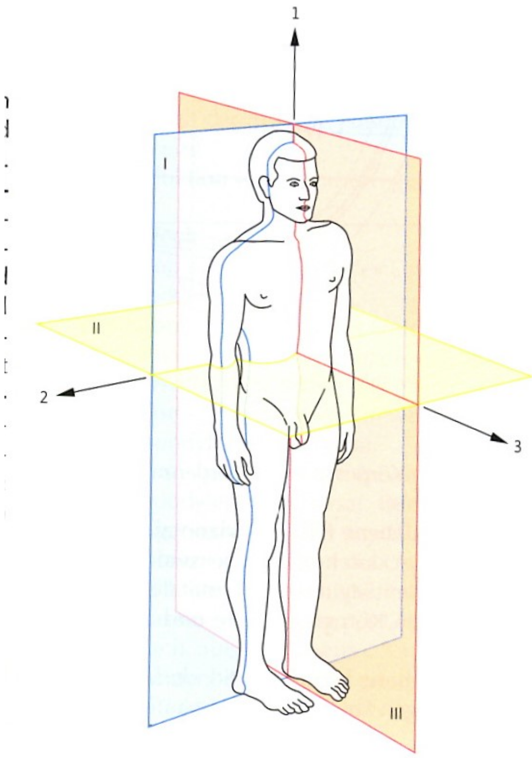

FIGURE 1.5: Main axes [1]

The terms of directions are defined independently from the spacial location of the body, so a clear way of speaking is possible [1]. Directions leading trunk-wards are termed **proximal**, while those leading limb-wards are called **distal**. Analogously, directions towards and away from the median plane are called **medial** and **lateral**. The **cranial** direction leads towards the head, the direction leading to the rump is termed **caudal** (Figure 1.6).

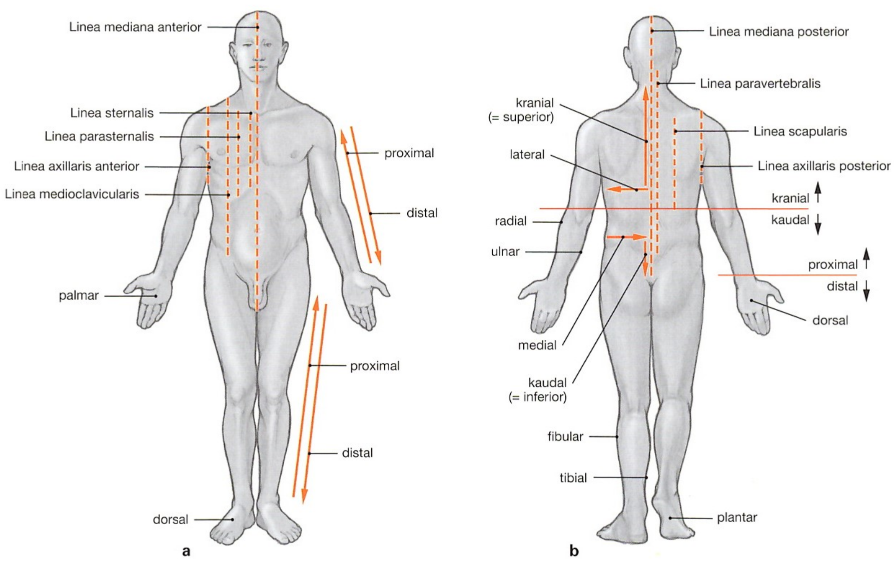

FIGURE 1.6: Directions and locations of the body parts [30]

1.1.2 Terms of Body Movements

When it comes to using a myoelectric hand prosthesis, four important body movements occur due to the EMG-based control system. **Extension** is the dilation of a body part, e.g. the arm, **flexion** its inflection, the two directions of **rotation** are called **supination** and **pronation** [30].

1.2 Upper Limb Amputation

> *"**Am | pu | ta | tiōn** [lat.], die kunftgemäße Ablösung einzelner Körperteile mittels chirurg. Instrumente. [...] Die A. muss unternommen werden, wenn das Leben durch ein örtliches Leiden gefährdet ist, das sich nur durch die Wegnahme des kranken Teils beseitigen lässt. [...] Die Entscheidung darüber, ob ein Körperteil geopfert werden soll, muss auch nach sozialen Gesichtspunkten getroffen werden."*
>
> \- Der große Brockhaus, Leipzig 1928

Despite great progressions in surgery and prosthetic within the last years, an amputation of the upper limb at any height is still a dramatic encroachment in human health and denotes a great loss of physical integrity [3]. Due to the lack of reliable statistics it is not possible to postulate accurate numbers of amuptation cases, but in fact there are four main issues leading to amputations: Cancer, infection, lymphatic circulatory disorders and the traumatic loss of a limb caused by an accident [34]. According to [3] the latter makes up $80 - 90\%$ of all cases.
If replantation of a lost extremity is no option, and beside certain other exceptional situations such as certain cancer cases, the aim is to set a required amputation as distal as possible, since a larger lever arm promotes an enhanced use of prostheses due to better muscular conditions [23].

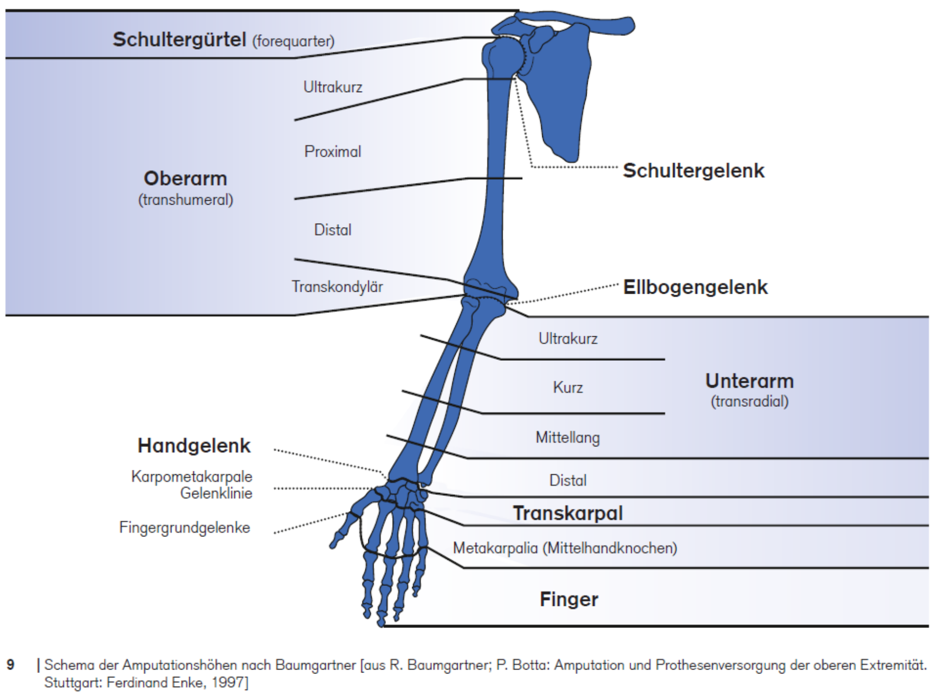

FIGURE 1.7: Heights of amputations [3]

Figure 1.7 shows the classification of amputation hights, divided into amputation heights along the antebrachium (**transradial**) respectively the brachium (**transhumeral**).

1.2.1 Transradial Amputations

Transradial amputations are the most common amputations performed and also the most beneficial ones, since in distal transradial amputations the length of remaining extremity engendering full shoulder and elbow function and a long lever arm provides good conditions for prosthesis adjustment and an optically pleasing result [36].

1.2.2 Elbow Disarticulations

Elbow disarticulations still provide satisfying conditions for prosthetic treatments due to a well-remained ability of suspension, supination and pronation. However, compared with transradial amputations, disadvantages like cosmetic issues related to an inequality in length of the prosthetically treated and the healthy side of the body [36], or problems concerning the stump padding [34] occur.

1.2.3 Transhumeral Amputations

Transhumeral amputations signify the most restricting loss of upper limb function and impact on prosthesis treatment complications. A functional prosthesis in case of a more distal level of amputation can be considered, while prosthetic treatment after amputation above the diaphyseal brachium is very problematic and mostly leaves only options of cosmetic prostheses, often combined with special constructions to fixate the device [34].

1.2.4 Specialties

Amputations are rare and always very individual surgeries. In the following, to give an idea of the contrasting variety of treatment options and their differences in practicability, strongly depending on prevailing circumstances, two extraordinary and very diverging surgery methods should shortly be introduced.

1.2.4.1 Krukenberg Plastic

FIGURE 1.8: Double sided Krukenberg treatment [23]

The so called *Krukenberg-Plastic* is a surgical technique developed by Hermann Krukenberg[2] during World War I. Within this procedure, ulna and radius are separated and covered with skin graft, such that not only a very intuitive way of grasping is provided by the surgically constructed grasping forceps, but also tactile sensation is preserved, which is of inestimable value especially for blind upper limb amputees.

[2] 1863-1935, German doctor and orthopedist

Due to the aesthetic aspect however, this kind of treatment nowadays is only considered as surgical option for amputees in developing countries and war zones [23]. In [13], an example of 15 Krukenberg procedures is given, which were done by the International Committee of the Red Cross on 11 single or double hand amputees in the civil war region of Sierra Leone in West Africa. The impressive results of this surgical mission show that the Krukenberg treatment gives back essential skills such as dressing or feeding oneself, making a survival for people living under dangerous circumstances possible.

1.2.4.2 Targeted Muscle Reinnervation

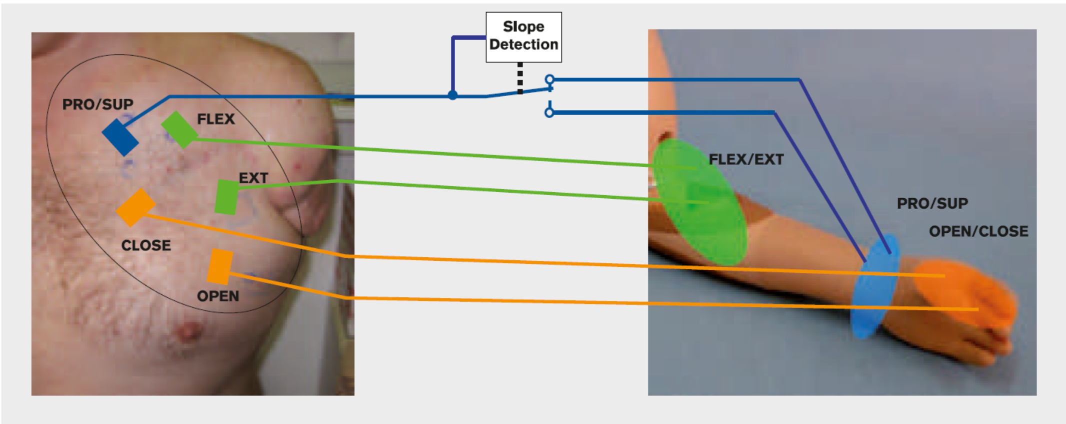

FIGURE 1.9: Prosthesis control scheme after glenohumeral TMR [23]

In contrast to the indeed practical but obviously very pragmatic and old-fashioned Krukenberg method, a very innovative surgical option called *Targeted Muscle Reinnervation* has appeared recently. Performed in cases of transhumeral and glenohumeral[3] amputation levels, this surgery technique makes use of residual nerves, formally connected to muscles of the now amputated arm, transferring them into muscle regions of the remaining limb in order to enable them to generate signals for myoprosthesis control [32]. Figure 1.10 shows typically involved nerve fivers and their target muscles [7]. In this way, well working signal spots for up to six surface electrodes can be created, opening a whole new range of opportunities for myoelectric prosthetics [36].

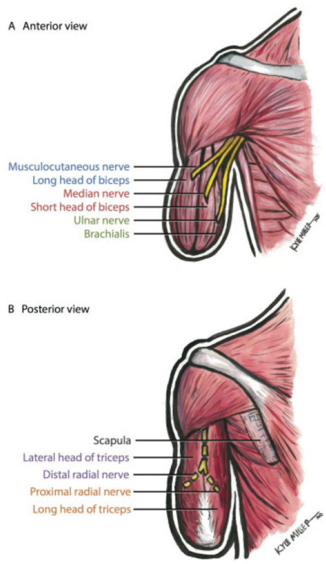

FIGURE 1.10: Surgical Plan for transhumeral TMR – nerves and related target muscles are color-coded. [7]

In glenohumeral amputation, regions of the Musculus Pectoralis Major (Figure 1.2) are used for reinnervation and provide a scheme of prosthesis control like shown in Figure 1.9, making a variety of functionality components such as opening and closing of a prosthetic hand, flexion and extension of an elbow joint and pronation respectively

[3] Shoulder Disarticulation

supination of a rotational joint possible. The newly inner-
vated muscles provide very intuitive prosthesis control to the user and at some point even leaves the option of activating several functionality components synchronously [23].

1.3 Prosthetic Treatment

In [5], available prosthetic treatments are categorized in terms of their capability of function and the way of controlling latter (Figure 1.11).

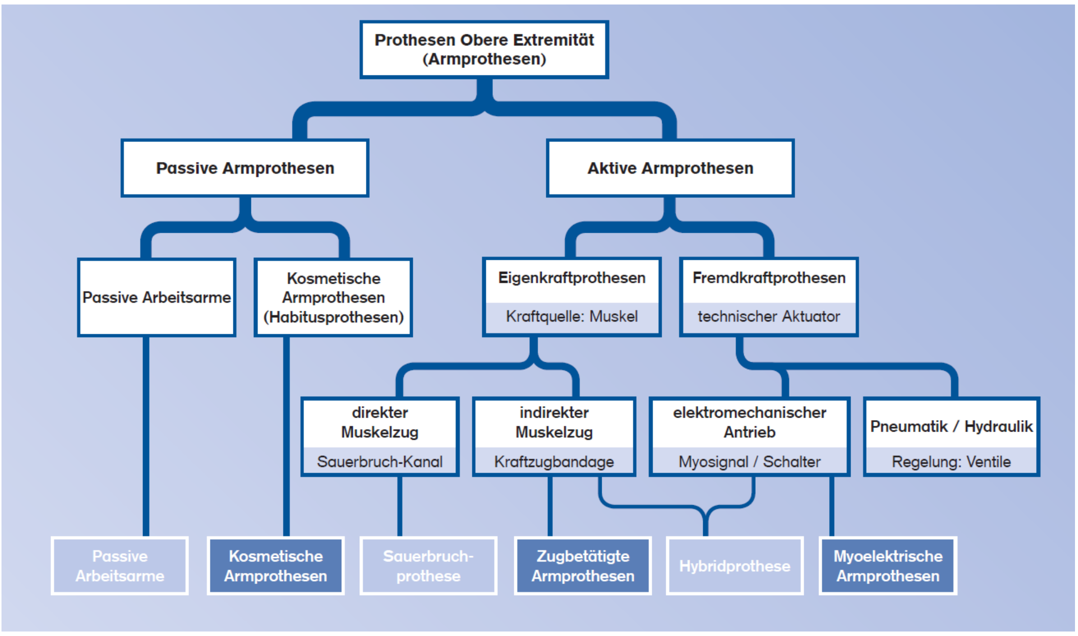

FIGURE 1.11: Upper limb prostheses classification [5]

1.3.1 Passive Prostheses

The term *passive* should be taken unbiased, since it only emphasizes the fact that those kind of prostheses are not able to generate any movement, neither by the help of a person's working parts of the body (1.3.2.1) nor by any mechatronic device (1.3.2.2) [23]. These kinds of prostheses can be used in any level of amputation, however they mostly occur in **minor amputations**[4] and transhumeral amputations where a more advanced treatment is not realizable, and either serve as cosmetic device only, or as support to grasp or hold objects or compensate unbalanced weight of the two sides of the body [5].

1.3.2 Active Prostheses

Unlike passive upper limb prostheses, active prosthetic treatments support the user by being able to generate movements such as grasping and rotating with artifical hand and rotation joint or extention and flexion in case of an existent artificial elbow joint [23]. Depending on the source of power used for generating these movements, active prostheses are subdivided into two further categories, body-powered and externally powered prostheses [5].

[4]Amputation of phalanx, finger or parts of the hand [34].

1.3.2.1 Body-Powered Prostheses

Movements of body-powered prostheses are realized by cable constructions between the harness and the movable prosthetic component (Figure 1.12). Movements generated by the user such as glenohumeral flexion or scapular protraction produce tension within the cable system, resulting in prosthesis function, e.g. hand opening or closing or elbow flexion [36]. A big advantage of body-powered prostheses is the independence of any external power source, which also implies a positive effect on the device's weight.

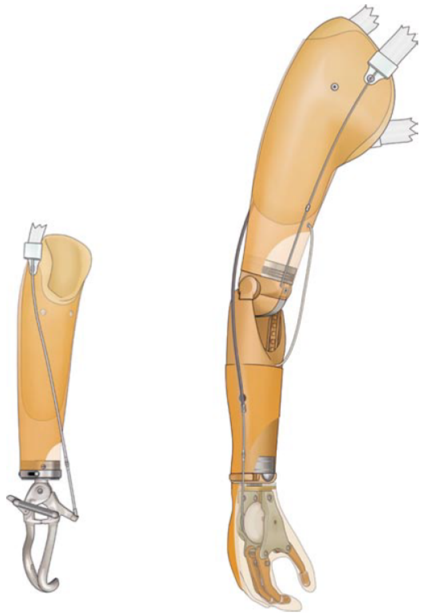

FIGURE 1.12: Body-powered prostheses – cable construction [5]

However, challenging, training-intense movements [5] and long transmission paths reduce the convenience of this sort of active prostheses, leading to the desire of externally powered or hybrid systems whenever possible [34].

1.3.2.2 Externally Powered Prostheses

This kind of prosthesis consists of at least one motorized component, such as a hand being capable of opening and closing or a rotation joint, driven by a battery [36] the user can recharge though certain access spots on the socket whenever needed.
Of course, a way of information flow between user and prosthesis is needed in order to tell the device, which movements are intended to being performed. The most common method to achieve this sort of conscious prosthesis control nowadays is to make use of surface electrodes (Figure 1.13), which measure changes in neuronal membran voltage caused by muscle contractions and relaxations (Figure 1.14, Section 1.1) and transmit them to the prosthesis internal control unit for further processing [23].

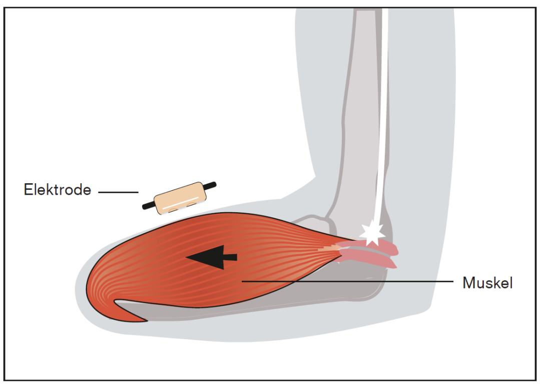

FIGURE 1.13: Surface electrode placed on myo-signal providing muscle [23]

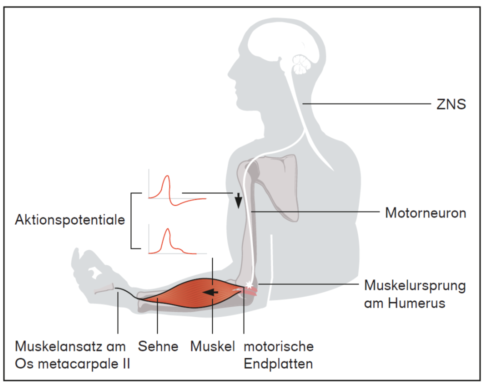

FIGURE 1.14: Nervous conduction of musculus flexor carpi radialis [17]

Figure 1.15 shows the measurement of two electrodes placed on two different muscle region spots of the user's limb being in state of relaxation. As long as the signals stay under a certain, internally-defined threshold, no action of the prosthesis is triggered. In the moment of contraction of one muscle, the measured EMG signal changes to a higher level (Figure 1.16) and generates corresponding activities in case of threshold overshoot (Chapter 4).

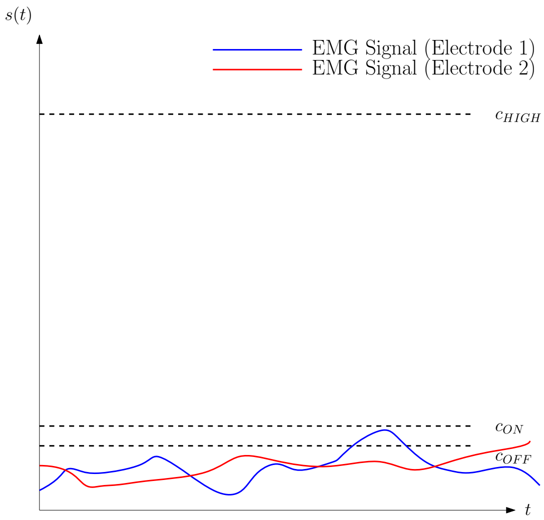

FIGURE 1.15: Measured EMG signal of a relaxed limb – signals below a certain threshold are ignored by the internal control unit and not transmitted for further processing or movement generation. No arising prosthesis activity.

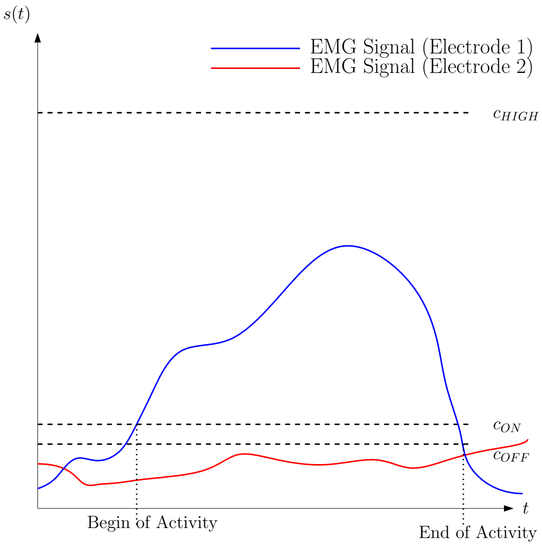

FIGURE 1.16: Measured EMG signal of a moving limb – muscle contractions lead to changes in measured EMG signal and initialize corresponding prosthesis movements. Movement stops as signal undershoots threshold again.

For good prosthesis control, it is necessary to position the electrodes on well-controllable remained muscle areas on the user's residual limb. Hereby, a proper placement of the electrodes within the manufacturing of the prosthesis socket is a demanding task for the orthopaedic technician and essential to make a use of the device possible at all. Figure 1.17 shows the significant influence of the electrode position on the quality of EMG signal within one single muscle region.

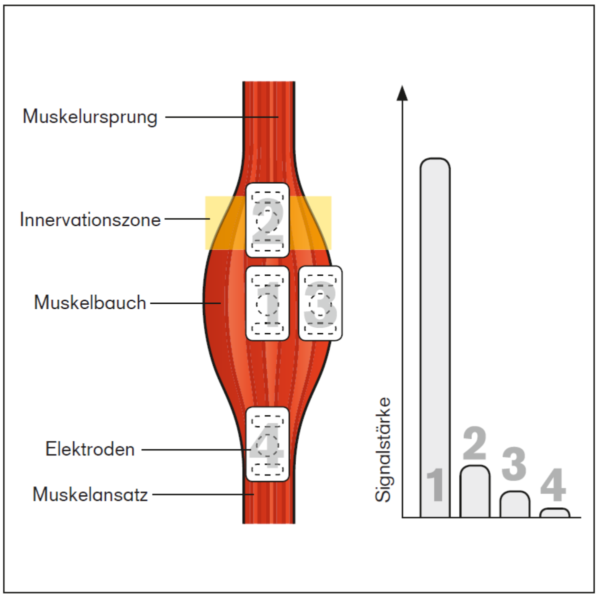

FIGURE 1.17: Effect of different electode positions [23]

A sensible number of electrodes depends strongly on the condition of the user's limb, the standard number for nowadays' myoelectric[5] prostheses such as the *Michelangelo* hand or the *Myobock* hand by *Ottobock* use two electrodes for a maximum of comfort and functionality.

[5]mỹs, gr., muscular

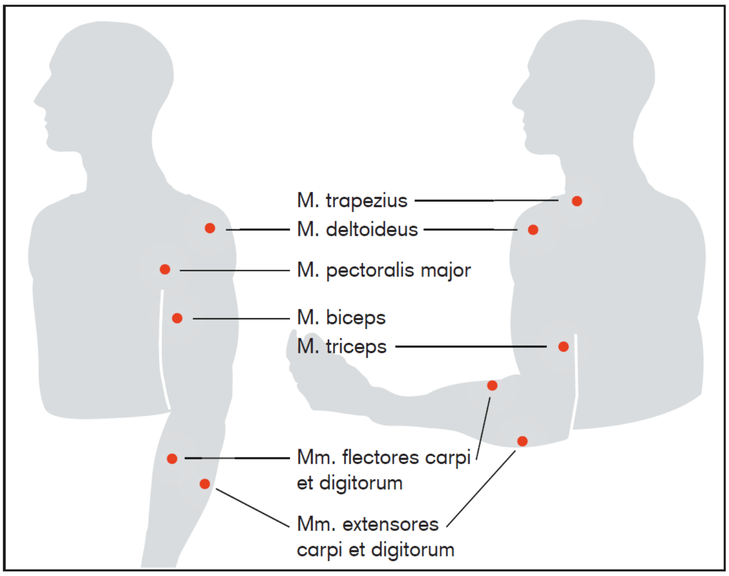

FIGURE 1.18: Commonly used muscle groups for electrode positioning within myoelectric prosthesis treatment [9]

The muscles used for a two-electrode treatment must provide well separated, strong enough EMG signals and are often chosen as antagonistic muscles, since synchronous contraction respectively relaxation is their natural behavior [23]. Figure 1.18 gives an overview of the most commonly used muscle regions for myoelectric prosthesis control. Once the electrodes are placed properly, the aim of the user is to train his or her residual muscle regions in order to be able to consciously generate specific contraction respectively relaxation patterns, which are recognized by the prosthesis' internal control unit as certain commands, leading to intended movements of the motorized components (Chapter 2).

Chapter 2

Parameter Adjustment

As shortly introduced in Section 1.3.2.2, the idea of functionality control of myoelectric hand prostheses is to consciously generate EMG signal patterns by contracting respectively relaxing muscle regions, measured by surface electrodes, which previously must have had been appropriately placed in the socket.
In order to tell the prosthesis' internal control unit, which shapes of EMG signals should be recognized as commands for certain prosthesis activities, several parameters have to be set and the different patterns have to be associated with desired actions.

FIGURE 2.1: *Ottobock*'s *Michelangelo* hand

Example: In case of a myoelectric treatment, consisting of a prosthetic hand, which is capable of performing hand opening and closing in two different modes of grasping for instance, a common way of closing the hand would be to generate muscle contractions, which produce EMG signals in the shape of Figure 1.16 (Section 4.4.2.2). The signal of electrode 1, overshooting the internally defined threshold c_{ON} is then connected to the functionality of hand-closing and initializes this certain kind of activity, as long as the measured EMG signal is strong enough. In order to stop the movement, the user has to relax the observed muscle region such that the signal falls below a certain threshold – c_{OFF} – again. If, on the other side, hand opening is

intended to be performed, the user has to contract the antagonistic muscle in order to bring measured EMG signal of electrode 2 to an appropriate level. A parameter, influencing this functionality component of the device, would for example be the so called *amplify factor* (Section 4.2). This factor is applied to the user's measured, raw EMG signal, in case his or her signals are too weak or too strong to appropriately generate required EMG signal patterns for a proper or comfortable use of the device. The challenge is to find an amplify factor, which perfectly supports the user in generating required EMG signal patterns, such that an easy, comfortable control of the device is possible.
The two different grasps (Section 4.4.1.1) as second functionality component in this example also need to be connected to EMG signal patterns in order to provide access on whatever grasp is currently desired to be performed. There is a various number of such patterns, the so-called *switch methods* (Section 4.4.2.1). In this example, let the defined switch method for changing from one grasp to the other be the *Short Cocontraction*. As Figure 4.7 shows, the user has to contract both observed muscle regions synchronously for a short period of time, before he has to relax the limb again in order to make the EMG signal fall below the required threshold again. This pattern is defined by three settable parameters: On the one hand, there are thresholds called *Cocontraction borders* (x_{C_1} and x_{C_2} in Figure 4.7), which have to be overshot by the signals, measured by electrode 1 and 2 within a certain, internally defined period of time c_{T}. On the other hand, there is the parameter of *signal length*, defining the maximal amount of time, the user is allowed to use in order to bring his or her signal below the internally defined threshold of recognition c_{OFF} again. The challenge hereby is to define the Short Cocontraction parameters individually for each user, such that his or her way of generating a Short Cocontraction signal is recognized by his or her device without any severe problems.

Same concepts of EMG signal pattern generation is used for getting access on all other prosthesis functionality components such as pronation and supination in case of a connected rotation joint or extention and flexion of an elbow joint in case of an upper limb treatment.

2.1 State of the Art – Manual Prosthesis Adjustment

In orthopedic technology, the proper parameter setting of a myoelectric prosthesis is a difficile topic. State of the art is the manual parameter adjustment based on visually observed and interpreted EMG signals of the user. Therefore, the user is asked to generate a range of different movements, contraction and relaxation patterns, which are recorded by the surface electrodes. The orthopedic technician must then tell from the EMG signal patterns the user generated, which patterns are best to connect to which functionality components and to what values to set related parameters to, in order to make an easy and comfortable use of a maximum of functionality components possible. To provide the technician a convenient work environment, *Ottobock* has developed a graphical user interface called *Ottobock Data Station* (Figure 2.2), where EMG signals can in real-time be tracked, paused and zoomed within the *Myo-Graph* (Section 4.3). Furthermore, all connections between patterns and functionality components, as well as their related parameters can be set easily via button clicks and sliders within *Ottobock*'s Data Station.

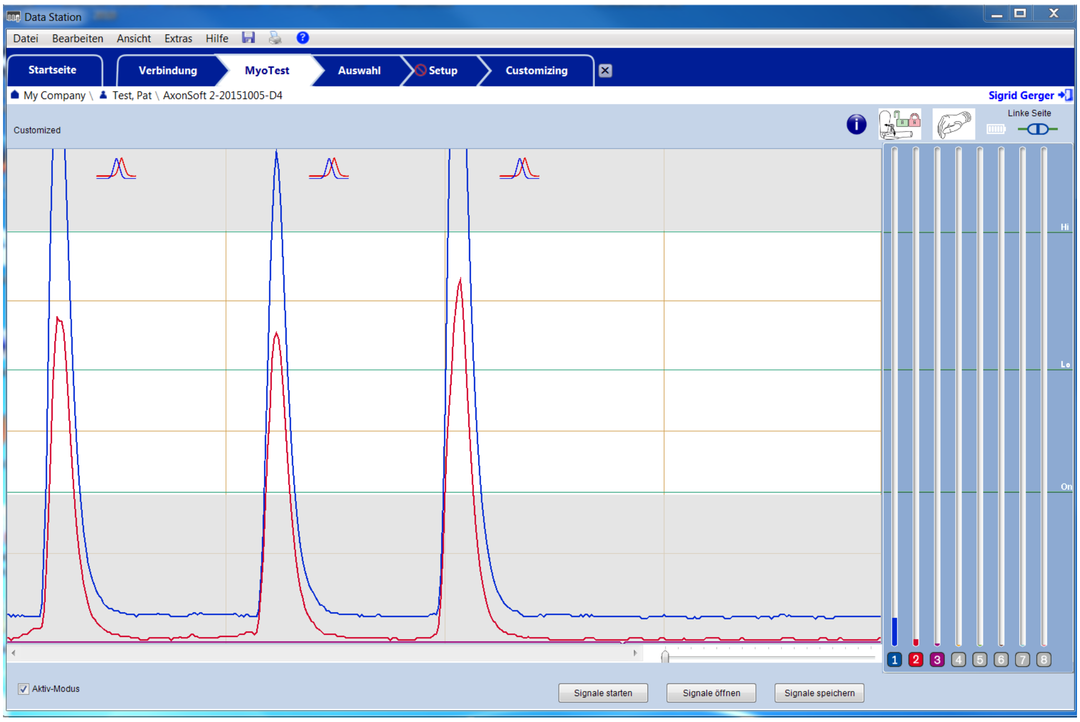

FIGURE 2.2: *Ottobock* Data Station – graphical user interface for manual parameter setting

Nevertheless, analyzing myo-signals is a very sophisticated issue, since a lot of interconnected aspects concerning functionality components and related parameters have to be considered synchronously in order to provide a working adjustment of the myo-prosthesis. This again demands a lot of experience in myo-prosthesis adjustment of the orthopedic technician, reaching far beyond the standard workshop of instruction, an orthopedic technician receives in the course of becoming certified for myoelectric prosthesis adjustments.

2.1.1 Occurring Problems

Amputations are rare surgical interventions. Therefore myoelectric prosthetic treatments do not occur prevalently in orthopedic technicians' daily routine. Due to this lack of experience of many orthopedic technicians, parameters and adjustments are set poorly in many cases, causing an inconvenient use of the prosthesis or even making a use of the device impossible despite sufficient physical and technical conditions. The resulting unpleasant prosthesis experience often leads to frustration of the user, prompting him or her to stop utilizing the myo-prosthesis' functionality, rather using it as a cosmetic prosthesis only. This not only wastes the great benefit of such myoelectric treatments, but also causes impairment of the muscle regions' condition, since the usage of myo-prostheses needs to come along with a lot of training and routine.

2.1.2 Solution Ideas

Due to this doom loop of failure in parameter setting, which often results in frustration and worsening of the user's myo-signals, caused by a lack of motivation, training and routine, the desire to an automated way of prosthesis adjustment arose and lead to *Ottobock*'s master degree project of the **Guided APS-Software**, which received the aim to provide a fully automated, guided path, beginning from user data recording, leading over data analysis towards the final values for settable parameters, which in the end should be enabled to be programmed to the user's prosthesis by a simple button click. The idea was to avoid unnecessary failure of prosthesis adjustment and further usage by providing the orthopedic technician a graphical user interface, which is on the one hand comfortable and easy to use, on the other hand automatically providing appropriate values for settable parameters by making use of mathematical considerations, which should be capable of calculating proper adjustment parameters, based on previously collected user data.

2.2 Guided APS-Software

Therefore, within a working period of six months, a prototype of a software package has been developed, which for one thing supports technician and user in myo-signal recording by interactively leading through a data recording schedule, demanding various movements and muscle activities by giving visual and written explanations and examples. Furthermore, these collected user data is then analyzed by an algorithm, detecting the user's strengths and weaknesses, determining all possibilities of prosthesis adjustment and suggesting it to the technician as a list in the software's graphical user interface. A simple button-click lets technician and user choose from the possible settings, depending on the user's preferences of prosthesis control and the technician's recommendations. As a last step, related parameter values for desired setting are calculated and written to the prosthesis' internal control unit via a connection between software and device.

The way to success in designing mentioned algorithm has turned out to be the idea of mathematical **optimization**, so before specific terms and tools of given problem of prosthesis adjustment are defined and introduced in Chapter 4, a brief overview of the mathematical background is given in Chapter 3.

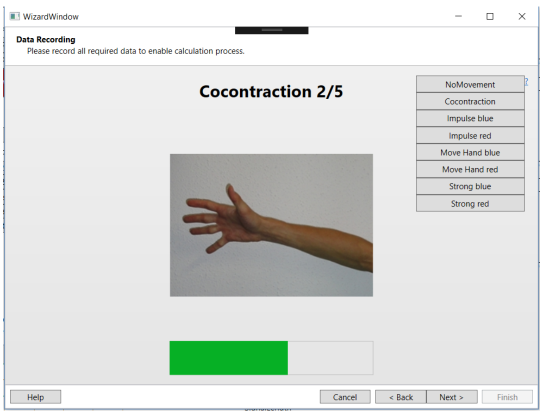

FIGURE 2.3: Instructional data collection – the user is prompted to provide certain movements by visual and written instructions. Provided data is recorded by properly positioned electrodes and stored for further processing.

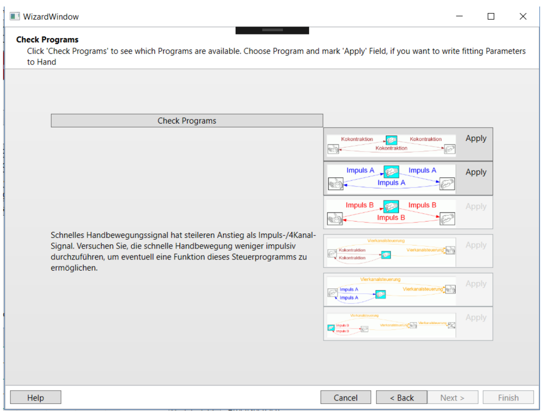

FIGURE 2.4: Program Choice – a list of available programs is shown and possible reasons for the non-working of certain programs are given. Via button-click, suitable parameters for available programs can directly be transported to the prosthesis.

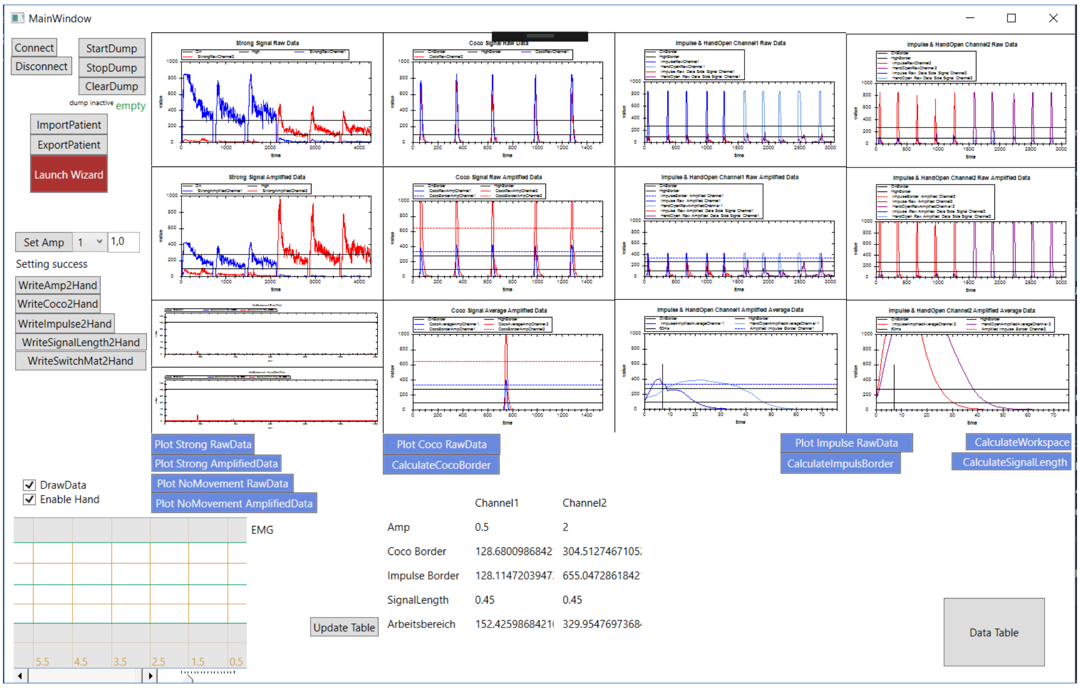

FIGURE 2.5: Guided APS-Software: main window – graphical user interface, granting visual insight of collected data and access to parameter determining functionality of the software.

Chapter 3

Optimization – Mathematical Foundation

Optimization is a major topic in mathematics, since it applies in a large range of working fields such as economy, engineering, finance, telecommunication [14] and computer vision [35]. Very many tasks can be formulated as optimization problem by making use of given data, provided by preveiling real-life circumstances by formulating an appropriate **Objective Function** $g : S \longrightarrow \mathbb{R}$ such that its maximization or minimization (Section 3.4.1) on a certain domain, the so-called **Feasible Set** $S \subseteq \mathbb{R}^n$ containing all **Feasible Solutions** $x \in S$ leads to desired **Optimal Solution** solution $x^* \in S$ of given task [15]:

$$x^* = \arg\max_{x \in S} g(x) \tag{3.1}$$

Of course, modeling a real-world application can be a tricky task and solvability of a formulated optimization problem is not guaranteed. If latter is the case, given optimization problem is called **infeasible**. Infeasibility can occur either when the feasible set is empty, i.e. $S = \emptyset$, or if the optimization problem is **unbounded**, meaning an infinitely high value of the objective function can be achieved by elements $x \in S$ [33]. If the optimization problem formulated by maximization is **bounded** however, lower bounds are usually called **Primal Bounds** and are given by any feasible solution $x \in S$, whereas upper bounds are called **Dual Bounds** and require other methods, e.g. *Relaxation* (Section 3.4.7) for determination [38]. In case of feasibility, the optimization problem either has a unique optimal solution, or a bounded or unbounded set of optimal solutions [12].

In [38] a systematic scheme for translating a given real-life problem description into an optimization problem formulation is postulated:

i) Define what appear to be necessary variables. ii) Use these variables to define a set of constraints so that the feasible points correspond to the feasible solutions of the problem. iii) Use these variables to define the objective function. iv) If difficulties arise, define an additional or alternative set of variables and iterate.

TABLE 3.1: Systematic formulation of an optimization problem [38]

By demanding linearity of the objective function and the constraints or restricting some or all of the **Variables**[1] to be integer or binary, special cases of optimization problems occur, which will be discussed in the following and formulated based on notations in [24] and [38].

3.1 Linear Program

Usually, the feasible set of an optimization problem is formulated by equlity- and inequality constraints, restricting the value of some function, which is dependent on in the objective function's variables [33]. If given objective function along with these constraints are restricted to be of a linear form, a general optimization problem turns into the special case of a **Linear Program** (**LP**). When modeling latter, it is important to respect certain obligatory assumptions in order to guarantee a correct mathematical, linear formulation of the real-world task description. Hence, it is essencial to keep *Direct Proportionality* of decisions to its values, i.e. the decision variables may only be raised to the first power only. Furthermore, the assumption of *Divisibility* demands the decision variables to be allowed to take on any real number. *Additivity* claims the independence of a decision variable towards any other decision variable within the objective function and the constraints. Finally, the insistence of *Certainty* concerning the correctness of data used to model given task is the foundation of receiving a proper formulation for given problem [18], [21].

With the assumption of linearity, the objective function in Equation 3.1 can be written as $g(x) = c^T x$ with **decision vector** $x \in \mathbb{R}^n$, where $c \in \mathbb{R}^n$ is an n-dimensional column vector called **cost vector**, and the feasible set $S \subseteq \mathbb{R}^n$ can be described by the $m \times n$ **Constraint Matrix** $A \in \mathbb{R}^{m \times n}$ and the **right-hand-side vector** $b \in \mathbb{R}^m$: $S = \{x \in R^n : Ax \leq b\}$ [18]. Due to the possibility of reformulating constraints in appropriate ways (Section 3.4), the assumption of non-negativity of the components – the so-called **decision Variables** – of the decision vector x, can be postulated without loss of generality.
All in all, a linear program can be written in the form:

$$\max_{x \in \mathbb{R}^n_+} \{c^T x : Ax \leq b\} \tag{3.2}$$

[1] Also called **Unknowns**. Components making up the elements in S.

where an **Instance** of the problem is defined as the tuple (c, A, b), consisting of the data which emerges from given task [38].

Due to efficiency in certain algorithms, a transformation of the notation in Equation 3.2 towards **Standard Form**[2] [8]:

$$\min_{x \in \mathbb{R}^n_+} \{c^T x : Ax = b\} \tag{3.3}$$

is required, which is easily achievable however by transformation rules and tricks explained in Section 3.4.

3.1.1 Geometric Interpretation

An equation of the form $\sum_{j=1}^{n} a_{ij}x = b_i$ with variable $x \in \mathbb{R}^n$ defines a hyperplane in $\mathbb{R}^n$. Thus, an inequation of the form $\sum_{j=1}^{n} a_{ij}x \leq b_i$, such as it is given in a linear program defining a restriction to the system which is desired to be optimized, divides the space into a half-space of feasible-, and another half-space of infeasible points. The intersection of all half-spaces of feasible points, defined by the rows $(a_{ij})_{i \in \{1,\ldots,m\}}$ of constraint matrix $A = (a_{ij})_{(i,j) \in \{1,\ldots,m\} \times \{1,\ldots,n\}}$ in problem 3.2, make up a convex polyhedron and define the feasible set S.

The cost vector c indicates a direction, to which the hyperplane $\{x \in S : c^T x = 0\}$ representing the objective function is ought to be moved, in order to find optimal solutions, which are placed at vertices of the polyhedron (Figure 3.1) [8], [35]. Latter fact is used, for instance, by the *Simplex Algorithm* (Section 3.1.2.1), whose idea is to move along edges of the feasible set, searching iteratively for an optimal solution.

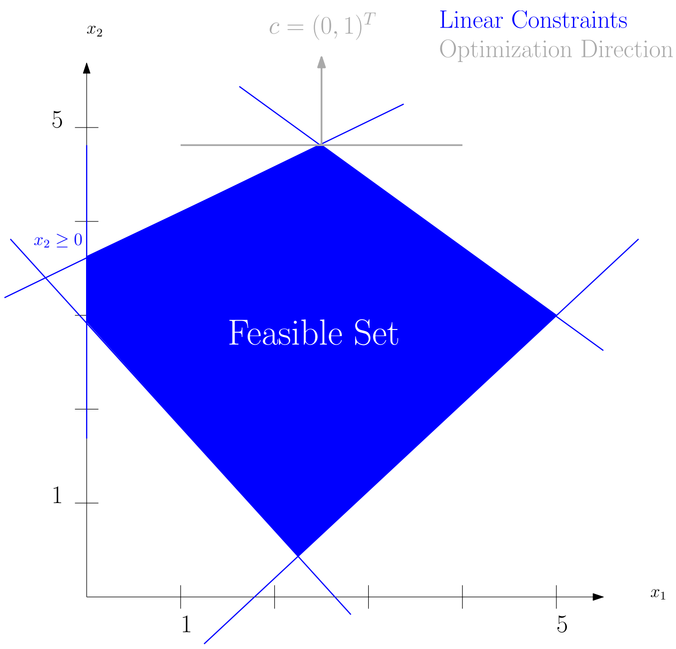

FIGURE 3.1: Graphical interpretation of a linear program – 2-dimensional polyhedron defined by LP's linear constraints.

3.1.2 Algorithms

Due to the existence of algorithms, which solve linear programs even in polynomial runtime, which makes

[2]Note, that problem 3.2 and problem 3.3 indeed differ in both instance-defining data and decision variables usually, due to the transformations which are performed in order to translate one formulation to the other. For the sake of readability however, the vectors c, b and x, respectively the matrix A are used as notation in both problems equally and should therefore be read with consciousness.

them a member of the so-called $\mathcal{P}$-**Class** in the field of **Complexity Theory**, linear programming is seen as a rather easy task in the field of optimization. Examples of such efficient algorithms are **Interior Point Methods**, the **Ellipsoid Method**[3] or the **Projective Method**[4] [8]. The most commonly used algorithm for solving linear programs in practice is the **Simplex Algorithm** and its variants, which makes use of the geometry of an LP's feasible set, but is not guaranteed to always achieve polynomial runtime. [16].

Since in practice, the task of solving linear programs is often part of solving more demanding problems such as **Non Linear Optimization Problems**, **Mixed-Integer Programs** (Section 3.2) or **Integer Programs** (Section 3.2.1.1), the understanding and improvement of solution approaches in linear programming affects a wide range of applications [16]. In the following, basic ideas of the main algorithms of linear programming are given.

3.1.2.1 Simplex Algorithm

One of the most commonly used algorithm in practice in solving linear programs is the **Simplex Algorithm**, pioneered by George Danzig in 1947 [12], which, despite the existence of examples of linear programs provoking runtime up to an exponential level, from the average point of view solves a majority of problems most efficiently compared to other methods.
Another reason of this algorithm's popularity is its capability of **Warm-Starts** [10], i.e. it handles slight changes of given problem such as adding a constraint during an already started iteration process efficiently by using already determined information, usually taking only a few further calculations to solution with no need of starting a whole new solution process [16].

The idea of the Simplex Algorithm makes use of the fact, that the feasible set of a linear program complies with a polyhedron, whose vertices make up optimal solutions of given problem. Graphically, the algorithm starts at one vertex, called the **(Initial) Basic Feasible Solution**, iteratively moving along the polyhedron's edges to neighboring vertices, observing entailing behavior of the objective function, until no improvement concerning latter is generable (Figure 3.2 [11]) [35], thus finding an optimal solution or determining the case of unboundedness [12].

[3] Leonid G. Khachijan, 1979
[4] Narendra Karmarkar, 1984

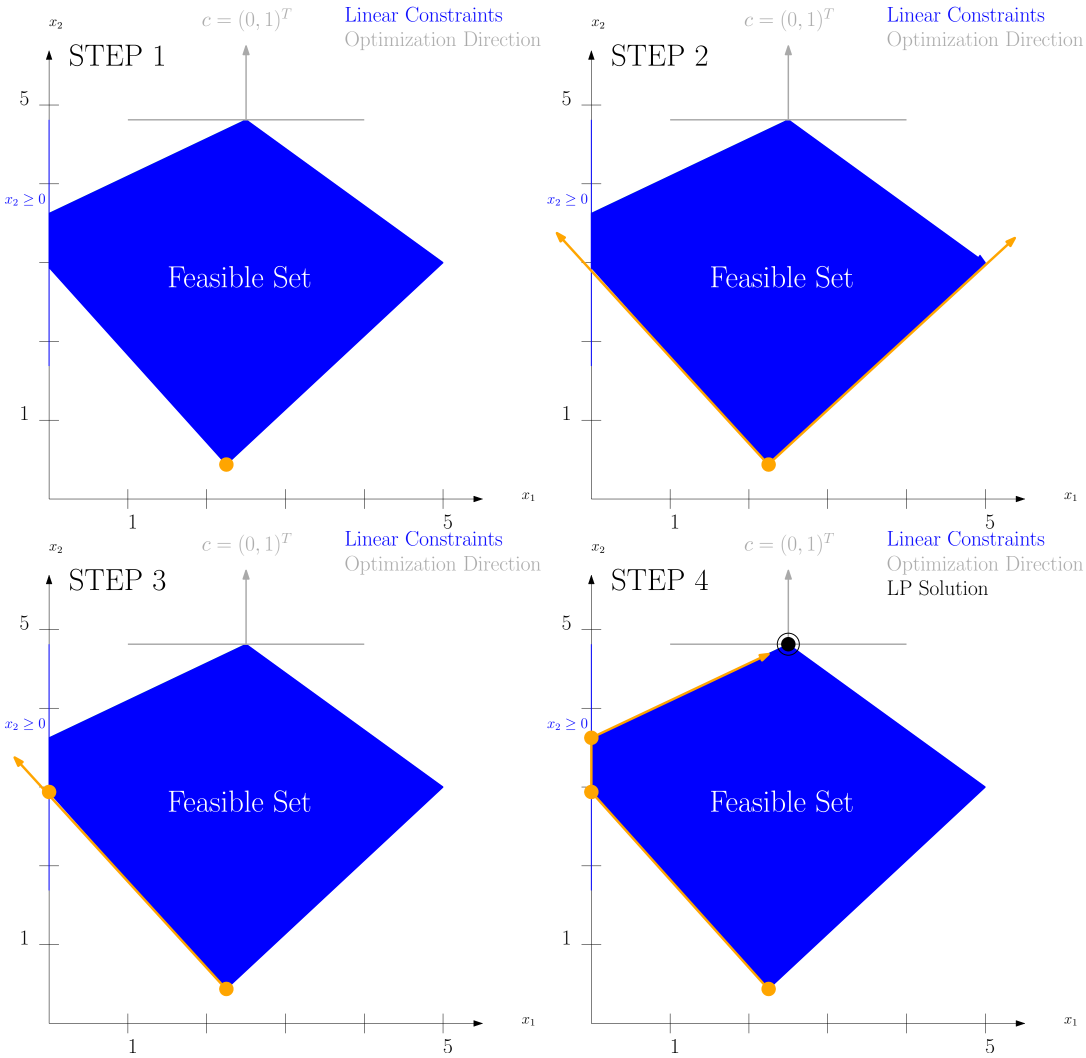

FIGURE 3.2: Graphical illustration of the Simplex Algorithm

There are several variations of the Simplex Algorithm and different approaches to describe the method mathematically. The algorithm requests the linear program to be formulated in standard form as defined in 3.3, the approach can however also be made by observing the linear program formulated as maximization problem. If additionally the constraints are given as inequalities, a transformation towards equalities can easily be performed by introducing *Slack Variables* (Section 3.4.2.2), even simplifying the finding of an inital basic feasible solution, since the so-called **Canonical Form** of appearing only in one single equation with coefficient 1, in which by definition the slack variables natually appear within the reformulted constraints, leads to a feasible solution by setting all but the stack variables to zero [16].

If given problem is of the form 3.2, the problem can be reformulated to following system of equations, the so-called **Starting Dictionary**:

$$z = \sum_{j=1}^{n} c_j x_j$$

$$x_{n+i} = b_i - \sum_{j=1}^{n} a_{ij} x_j, \ i = 1, \dots, m$$
$$x_j \geq 0, \ j = 1, \dots, n+m$$

where $x_{n+i}, \ i = 1, \dots, m$ are the slack variables, introduced at the beginning to obtain equality constraints.
In the case of introduced slack variables, by setting all non-slack variables to zero, an initial basic feasible solution, i.e. $(0, \dots, 0, b_1, \dots, b_n)$, which corresponds to one vertex of the feasible set, is found easily and provides a certain value of the objective function. Variables set to zero are called **Non-Basic Variables**, whereas the remaining ones are named **Basic Variables**. The aim in every interation step is to exchange exactly one basic variable, which in this situation of exchange is called **Leaving Variable**, with exactly one non-basic variable, the so-called **Entering Variable**, with the intention to increase resulting value of the objective function. So with $\mathcal{B}$ as the set of indices of the basic variables and $\mathcal{N}$ as the set of indices corresponding to all non-basic variables, one iteration of the Simplex Algorithm provides the change to a new dictionary by choosing entering and leaving variables and completing appropriate row operations on the equations in order to bring the system to a canonical form again [37]:

$$z = \tilde{z} + \sum_{j \in \mathcal{N}} \tilde{c}_j x_j$$
$$x_i = \tilde{b}_i - \sum_{j \in \mathcal{N}} \tilde{a}_{ij} x_j, \ i \in \mathcal{B}$$

where the value of the entering variable x_k is chosen such that the basic variables $x_i = \tilde{b}_i - \tilde{a}_{ik} x_k, \ i \in \mathcal{B}$ remain non-negative:

$$x_k = \min_{i \in \mathcal{B}, \tilde{a}_{ik} > 0} \frac{\tilde{b}_i}{\tilde{a}_{ik}}$$

Figure 3.3 shows a flowchart of the Simplex Algorithm, where x_{in} and x_{out} represent the entering- and the leaving variable.

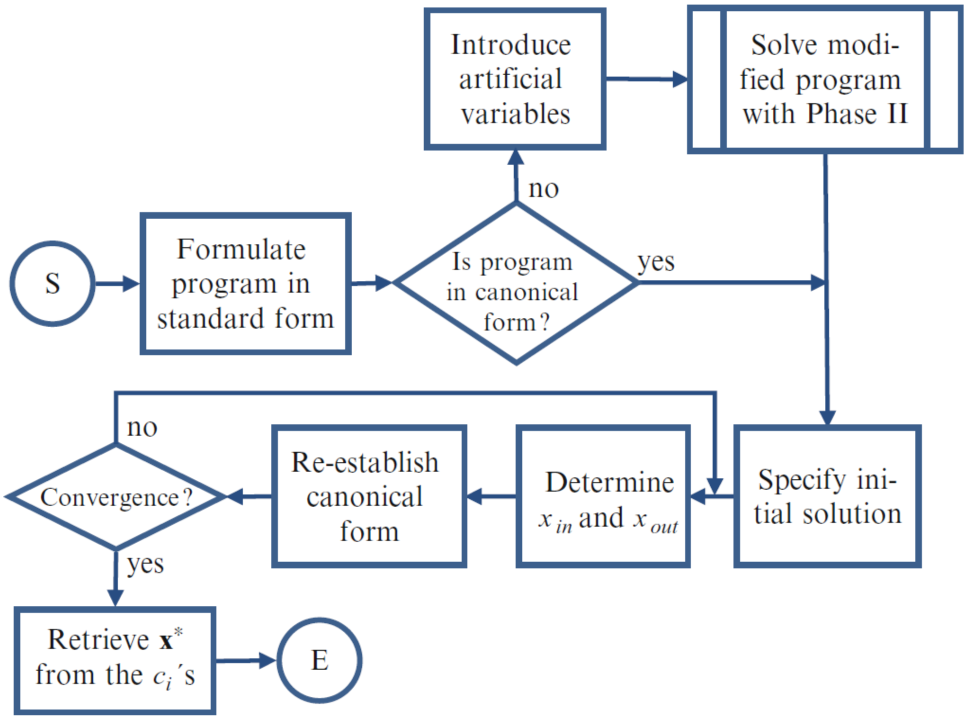

FIGURE 3.3: Scheme of the Simplex Algorithm [35]

3.1.2.2 Interior Point Methods

Another way of solving linear programs are **Interior Point Methods** (**IPM**s), sometimes also called **Barrier Methods** [35]. Originating from non-linear programming, they build concepts especially suitable for problems represented by sparse matrices and, from the theoretical point of view, due to their polynomial runtime are preferable to the Simplex Algorithm. In practice however, since latter in some cases can take many, but still cheap operations [16], and IPMs are not capable of warm-starts, which especially is of interest in solving *Mixed-Integer Programs* (Section 3.2), decisions between the Simplex Algorithm and IPMs strongly depend on given problem [35]. In practice, satisfying runtime is also achieved by hybrid approaches, combining IPM's

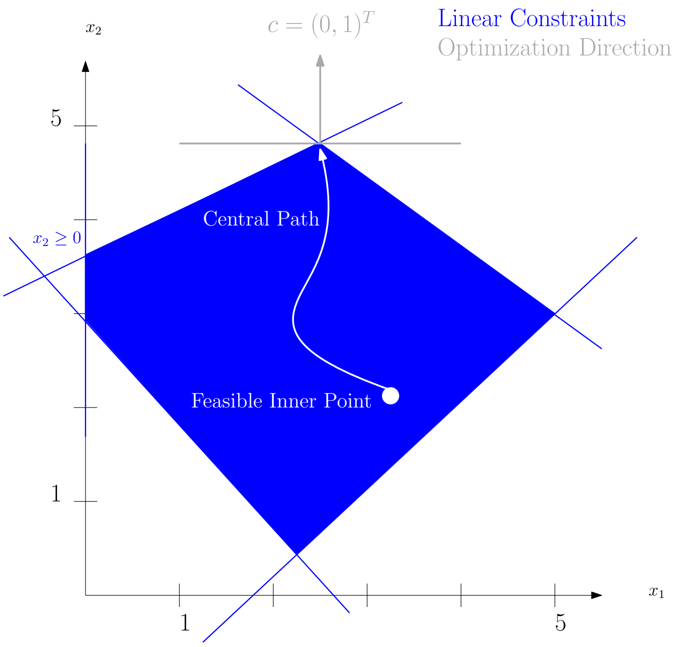

FIGURE 3.4: Graphical interpretation of Interior Point Methods

with the Simplex Algorithm in the solver's implementation [16], [21].

In contrast to the Simplex Algorithm, which starts at a vertex of the feasible set and performs its movements along the polyhedron's edges, an IPM's basic idea is to start *within* the feasible set [35], remaining inside the polyhedron when moving towards an optimal solution, i.e. a polyhedron's vertex, during the process of iteration (Figure 3.4 [16]). Finally, after the iteration process of an IPM determined an inner point of the feasible set, satisfying predefined termination criterion, methods such as **Cross-Over Schemes** or **Pivoting Procedures** are required in order to reach an actual vertex from within the feasible set [16].

IPMs can be classified into three categories of solution strategy:

i) An **Affine Scaling Method**[5] is an iterative, two-phase method for solving linear programs by moving through the feasible set, making use of the **Steepest Ascent Direction**, i.e. the gradient of the objective function – which is the direction of the objective function's best improvement – modifying it by additional restrictions and scaling, such that resulting directions do not lead outside the feasible region [37].

ii) **Potential Reduction Methods**[6] base on reformulating given problem towards the so-called **Potential Function** as objective function, optimizing latter along with linear constraints, which are postulated to ensure feasibility [16].

iii) **Central Trajectory-**, or also called **Central Path Methods** find optimal solutions to a given linear program by starting at an appropriate inner point of the feasible set, iteratively moving along the so-called *Central Path* [16], using linear combinations of the **Direction** towards **Optimality**, towards **Feasibility** and towards **Centrality** [37].

An example of a central trajectory method is the **Logarithmic Barrier Method**. Given a problem in standard form 3.3, the approach of this method is to replace initial linear program by a sequence of non-linear minimization problems

$$\tilde{x}_k^* = \min_{x \in \mathbb{R}_+^n} \left\{ c^T x - \mu \sum_{j=1}^{n} \ln x_j : Ax = b,\ \mu = \mu_k \right\} \tag{3.4}$$

with logarithmic **Penalty Term** $\mu \sum_{j=1}^{n} \ln x_j$ by iteratively generating an appropriate sequence[7] $\mu_k \in \mathbb{R}$, such that the solutions $\tilde{x}_k^*$ of the sequence of non-linear problems converges to an optimal solution x^* of the original problem 3.3 [16]:

$$\lim_{k \longrightarrow \infty} \mu_k \sum_{j=1}^{n} \ln x_j = 0$$
$$\lim_{k \longrightarrow \infty} \tilde{x}_k^* = x^* := \min_{x \in \mathbb{R}_+^n} \{c^T x : Ax = b\}$$

[5] I. I. Dikin, 1967 [37]

[6] Narendra Karmarkar, 1984 [16]

[7] There are several heuristics available in order to calculate appropriate choices of μ, to be found for example in [6], [19] and [22]

The objective function of the family of non-linear auxiliary problems 3.4 is called **Logarithmic Barrier Function** and, as a function of μ, generates a path though the feasible set, the so-called **Central Path** [37].

3.2 Linear Mixed-Integer Program

If the additional claim of some, but not necessarily all of the decision variables to be integer is postulated, the LP turns into a **Linear Mixed-Integer Program** (**MIP**). In this special case of a linear optimization problem, the objective function as well as the part of the formulation made up by the Feasible Set is divided into an integer- and a continuous part and the problem can be written as:

$$\max_{x_{\text{real}} \in \mathbb{R}^p_+, x_{\text{int}} \in \mathbb{Z}^{n-p}_+} \{c^T x_{\text{real}} + h^T x_{\text{int}} : A x_{\text{real}} + G x_{\text{int}} \leq b\} \tag{3.5}$$

where x_{real} is a p-dimensional column vector of decision variables which can take up any value in $\mathbb{R}$, x_{int} is an $n-p$-dimensional column vector of decision variables required to be integer $c \in \mathbb{R}^p$ and $h \in \mathbb{R}^{n-p}$ are related cost vectors and $A \in \mathbb{R}^{m \times p}$ and $G \in \mathbb{R}^{m \times (n-p)}$ the Constraint Matrices of the optimization problem.

3.2.1 Special Cases

If further restrictions to the decision variables are postulated, e.g. banning non-integer values totally from being feasible or even restricting them to take on binary values only, following special cases of MIPs occur:

3.2.1.1 Linear Integer Program

In case all decision variables are demanded to be integer, the optimization problem 3.5 reduces to a **Linear Integer Program** (**IP**):

$$\max_{x \in \mathbb{Z}^n_+} \{c^T x : Ax \leq b\} \tag{3.6}$$

with n-dimensional decision vector x only permitted to take on integer values and $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$ related Constraint Matrix respectively right-hand-side vector.

3.2.1.2 Binary Integer Program

In many applications, decision variables bound to the restriction to only take on binary values lead to a successful problem formulation, so by assuming the decision vector to be in $\{0,1\}^n$, one receives the special case of a **Binary Integer Program** (**BIP**):

$$\max_{x \in \{0,1\}^n} \{c^T x : Ax \leq b\}$$

3.2.2 Geometry

As opposed to an LP and its feasible set's favorable properties and well-known possibilities of solutions' locations, a MIP or an IP is of much more difficult nature [37]. Despite the fact, that an IP's feasible set only contains a finite number of points in contrast to an LP's infinite number of probable solutions, the restriction to some or all decision variables to be integer causes a discontinuity within a (M)IP's feasible set, making the traditional use of calculus impossible, thus leaving questions such as the number or location of potential solutions a major issue to answer. Furthermore, discontinuity of the feasible area refuses a "free" movement amongst potential solutions, raising the challenge to remain feasible during the process of optimization [21].

Figure 3.5 gives an idea about the difference between the feasible set of an LP and that one of an IP. While the feasible set of the LP forms a polyhedron with its favorable property of having placed optimal solutions at its vertices, the finitely many feasible solutions making up the IP's feasible set lie within constrained-defined polyhedron, generally with no additional information about their specific relation to optimality. Figure 3.5 also gives a first insight into the idea of *Relaxation* (Section 3.4.7) being used in solving (M)IPs (Section 3.2.3.2, Section 3.2.3.3), since the LP, generated by neglecting the integer-constraints of a given (M)IP, forms a feasible set, which is superset of the initial (M)IP's feasibe set, thus providing dual bounds of latter.

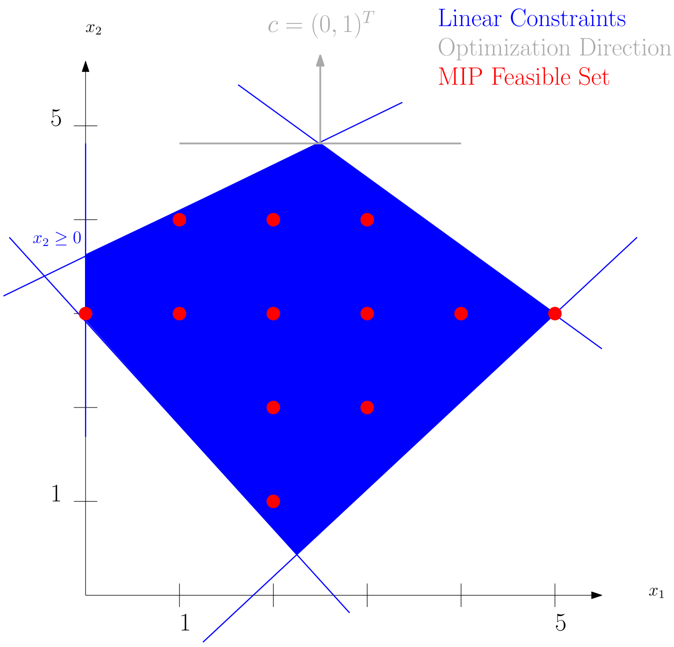

FIGURE 3.5: Graphical interpretation of an integer program – feasible set of an LP vs. feasible set of an IP

3.2.3 Algorithms

The geometry of a (M)IP (Section 3.2.2) already leads to an educated guess, that – in contrast to linear programs belonging to class $\mathcal{P}$ of problems, which can be solved in polynomial runtime (Section 3.1.2) – a (M)IP brings along a much wider range of difficulties concerning solvability [37]. Indeed, most (M)IPs are of the class of $\mathcal{NP}$-**Hard Problems**, meaning, that there no yet exists any algorithm, solving latter in polynomial, but rather in exponential runtime [8].

There are several different approaches in order to find optimal solutions to a given MIP, which can be categorized into the principle of **Exact Methods**, which provide optimal solutions by perfoming a finite number of algorithmic steps, and **Heuristic Methods**, which are based on the idea of using specific rules for specific problems which have appeared to be sensible in these certain situations, but do not guarantee optimality [8] or give an idea about the closeness to latter in general, but can sometimes be judged in quality by certain bounding statements derived from methods such as the **Lagrangian Relaxation Technique** [21]. Figure 3.6 shows a classification of the principles developed in order to solve MIPs, a more detailed overview is given in [8].

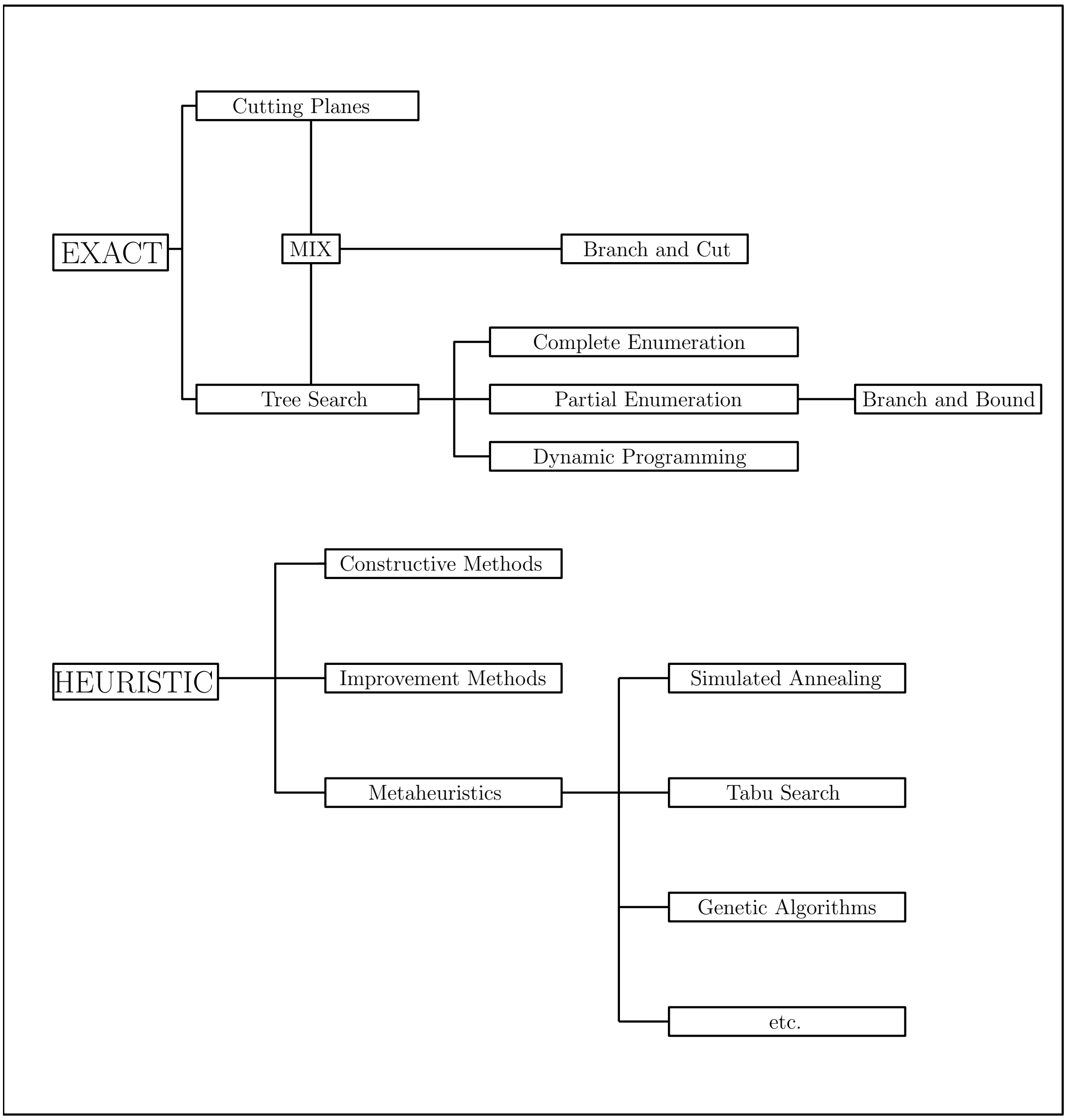

FIGURE 3.6: Classification of MIP solvers

While *Rounding* (Section 3.2.3.1) might be the most naive approach, generally not leading to success [4], the first algorithms occuring in order to attain solutions to (M)IPs were *Cutting-Plane Algorithms*[8], followed by the introduction of the concept of *Branch and Bound* (Section 3.2.3.3) by A. H. Land and A. G. Doig [21]. Due to the tremendous advantages of combining latter two approaches, most of the commercial software packages have so-called *Branch and Cut* algorithms implemented (Section 3.2.3.4), and also certain methods of **Preprocessing** can be used in order to simplify a given model for better solvability [16]. However only certain methods are suitable for certain problems and for a given problem not every approach leads to satisfying results [21].

[8] Danzig (1954), Fulkerson (1954), Johnson (1954), Gomory (1958, 1960, 1963) [21], [16]

3.2.3.1 Rounding

A very intuitive and all the same naive idea of receiving a solution to a MIP is to consider the corresponding LP by ignoring the constraint to some or all decision variables to be integer, solve it by one of the many LP solution approaches and round the result to the nearest integer [21].

Sounding promising and comfortable in implementation, this approach unfortunately often provides infeasible or non-optimal solutions (Figure 3.7) and thus does not lead to success for discrete problems in general [16]. Therefore, when being used in practice, the solution derived from rounding must be analyzed with care, checking for a reasonable gap between continuous and rounded result as well as ensuring, that all constraints remain satisfied [21].

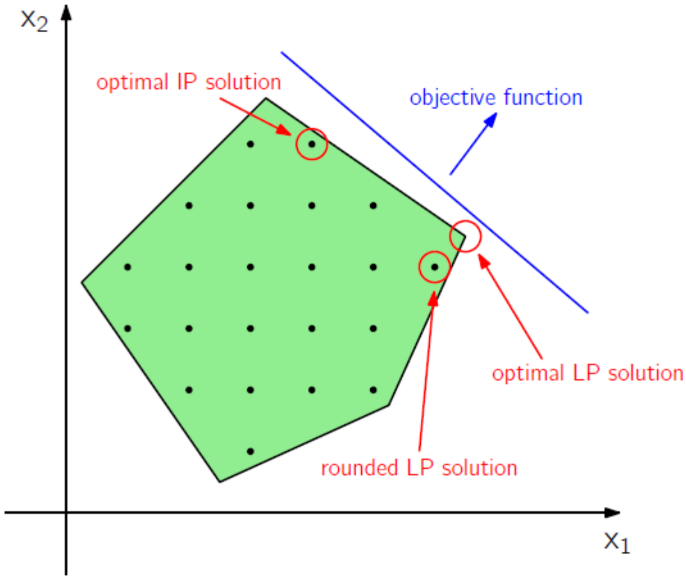

FIGURE 3.7: Rounding – occurring problems [4]

3.2.3.2 Cutting-Plane Methods

One of the first concepts occurring in algorithms developed for solving (M)IPs was the idea of **Cutting Planes** [21]. Based on the idea of LP relaxation (Section 3.4.7) and its generation of dual bounds to corresponding (M)IP, it calculates an optimal solution of the LP relaxed problem (Figure 3.8), which – when not being integer and thus not being feasible for inital discrete problem – provides the opportunity to find so-called **Valid Inequalities** or **Cutting Planes**, which are added to initial program such that the previously found optimal solution of the LP relaxation becomes infeasible, while the feasible set of the initial (M)IP maintains unaffected (Figure 3.9) [16]. This **Cutting** of the feasible set is kept up iteratively, until the relaxed problem generated by all added valid inequalities provides an integer solution, which indeed is an optimal solution to inital discrete problem automatically [21]. Thus, the algorithm successively approximates desired integer solution, but can in fact take a long time for its final convergence, which is the reason for its rather restrainted implementation in nowadays' commercial software [16].

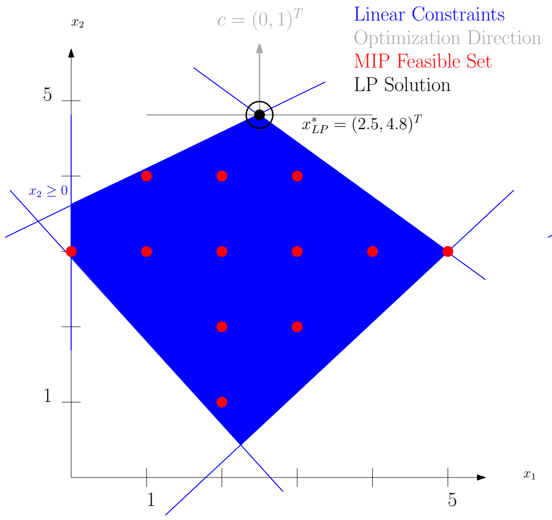

FIGURE 3.8: Cutting Plane Method – solution of LP relaxation. Calculated non-integer value not feasible for initial, discrete problem, additional valid inequalities required.

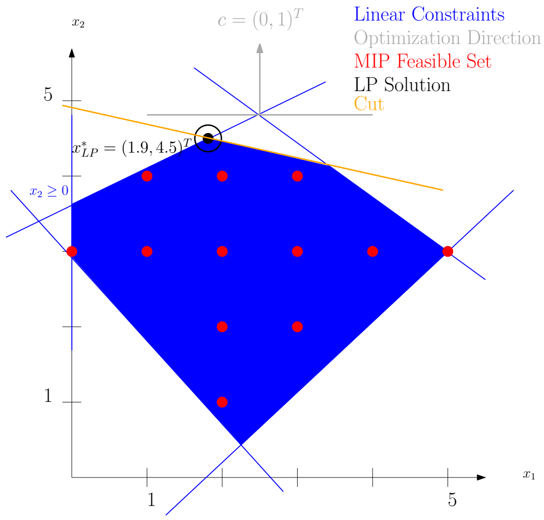

FIGURE 3.9: Cutting Plane – valid inequation divides LP relaxed solution from feasible set and generates new LP relaxed solution. (M)IP's feasible set remains unaffected.

There are many different choices of cuts leading to different performances and a lot of methods have been developed and improved in order to achieve satisfactory results. A more detailed treatment of the idea of cutting planes is given in [24] and [38].

3.2.3.3 Branch and Bound

Potentially very expensive – since in many cases providing near optimal solutions quickly but taking larger effort to verify optimality – but nonetheless commercially most commonly used algorithms are of **Branch and Bound** (**B&B**) type [21]. B&B methods are **Divide and Conquer Algorithms**, meaning that they find a solution by dividing initial problem recursively into smaller, hopefully easier handlable sub-problems, solving latters and generating a final solution by putting together the information gained from these partial solutions [38]. Hereby – in order to avoid **Complete Enumeration** (Figure 3.6), which has an exponential rise in expense and therefore is suitable for minimally-sized problems only – rejecting criteria are postulated by finding upper and lower bounds to an optimal solution, which distinguish considerable sub-problems from those which can be skipped in further recursion, making it a more efficient, a so-called **Implicit-** or **Partial Enumeration Method** [16].

As the name indicates, the basic idea of this method consists of two steps:

i) **Branching:** The initial problem is divided into disjunctive sub-problems, such that the union of the subsets' solution sets is equal to that one of the inital problem [8].

   In order to be able to sketch the idea of branching in mathematical terms, let S be the feasible set of problem 3.5 and $g := (c, h)$ be the concatenation of the cost vectors of its continuous and discrete part. Furthermore, let $\{S_i \subseteq S : i = 1, \ldots, k, \cup_{i=1}^{k} S_i = S, S_i \cap S_j = \emptyset \ \forall i \neq j\}$ be a **Partition** of S, and

$$\max_{x \in S_i} \{c^T x\} \tag{3.7}$$

be the i-th sub-problem of initial problem

$$\max_{x \in S} \{c^T x\} \tag{3.8}$$

The idea of branching makes use of the fact, that an optimal solution of initial problem 3.8 is distinguishable from the optimal solutions of the generated smaller sub-problems 3.7 [24]:

$$x^* := \max_{x \in S} \{c^T x\} = \max_{i=1,\ldots,k} \left(\max_{x \in S_i} \{c^T x\} \right) \tag{3.9}$$

Figure 3.10 [24] shows the tree arising from the idea of branching initial problem into smaller sub-problems by dividing the feasible regions into subsets and continuing calculations on these smaller domains.

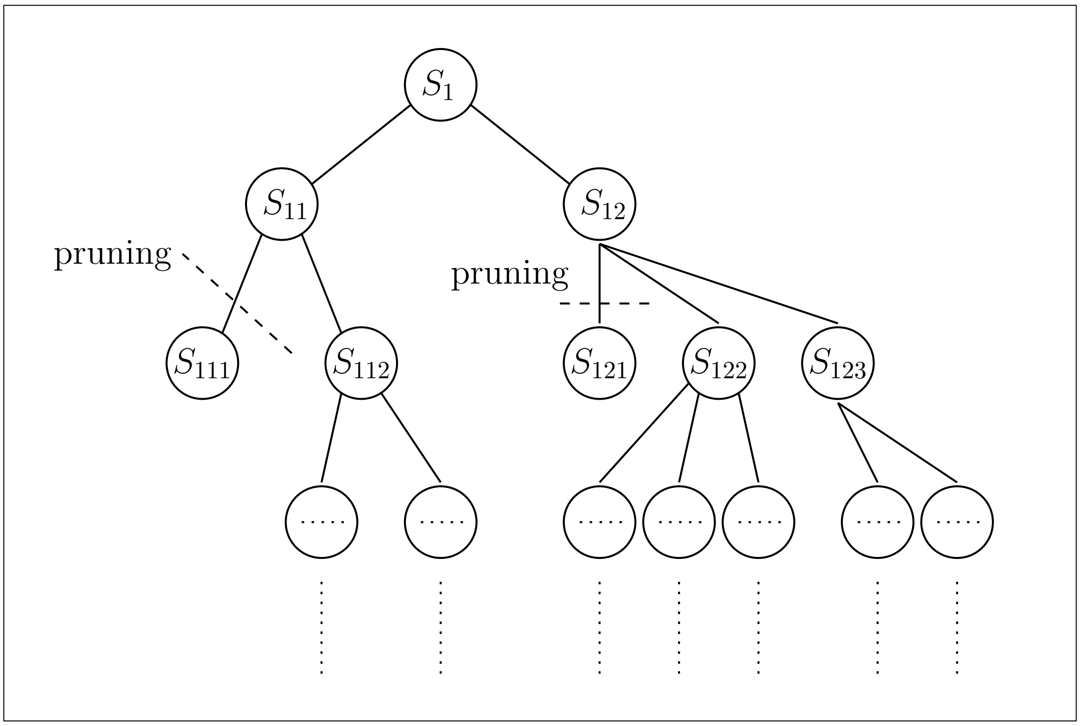

FIGURE 3.10: Branch and Bound tree

Without any further considerations, branching recursively deeper from root to base would lead to total enumeration, engendering enormous costs of calculation. This leads to the method's secondly performed step:

ii) **Bounding:** For every sub-problem, upper and lower bounds to the solution are determined, which lead to the capability of deciding whether or not a sub-problem might contain an optimal solution and requires further examination or can be neglected in following processing steps. The detection and further neglection of a tree's irrelevant node S_i is called **pruning** (Figure 3.10) and can be performed in either one of following three cases [24]:

- *Infeasibility*: $S_i = \emptyset$
- *Optimality*: An optimal solution to 3.7 is known

- *Value Dominance*: $x_i^* := \max_{x \in S_i}\{c^T x\} \leq x^*$

In practice, LP relaxation (Section 3.4.7) is used most commonly in order to calculate dual bounds [33], whereas primal bounds can simply be generated by any already known feasible solution [8]. Furthermore, **Binary Branching** is a common way to implement the branching procedure within a $B\&B$ method. Hereby, the calculated optimal solution x^*_{relaxed} of the LP relaxation of **parental** problem S_j is calculated (Figure 3.11) and – if $x^*_{\text{relaxed}} \notin \mathbb{Z}^n_+$ – used to postulate additional constraints in order to partition S_j into two subsets S_{j1} and S_{j2} (Figure 3.12) such that [4]:

$$S_{j1} := S_j \cap \{x \in \mathbb{Z}^n_+ : x \leq \lfloor x_j \rfloor\}$$
$$S_{j2} := S_j \cap \{x \in \mathbb{Z}^n_+ : x \leq \lceil x_j \rceil\}$$

Indeed, all solutions of parental problem are contained in either one of thus newly generated sub-problems.

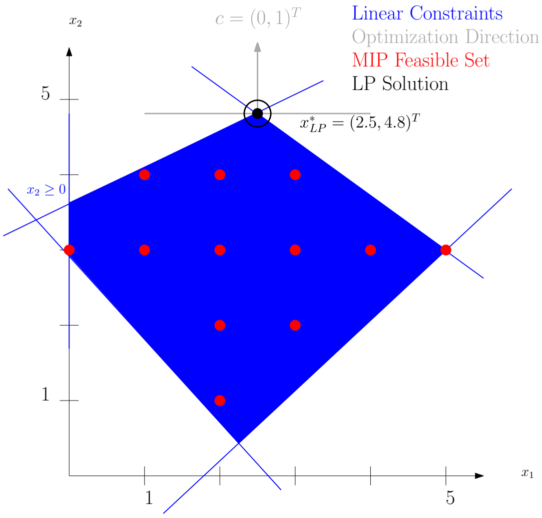

FIGURE 3.11: Feasible region of LP relaxation

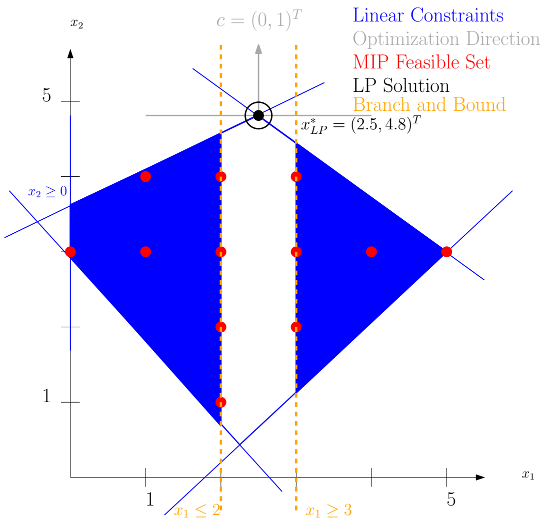

FIGURE 3.12: Feasible regions of sub-problems

These steps are proceeded such that optimal solutions of remained sub-problems can continuously be compared with each other and result in an optimal solution to initial problem based on property 3.9 of the sub-problems.

3.2.3.4 Branch and Cut

Branch and Cut Methods (**B&C**) combine the two aspects of Cutting Planes (Section 3.2.3.2) and $B\&B$ methods (Section 3.2.3.3), which by themselves often cause troubles and produce numerical stability issues [33]. $B\&C$ methods make use of the $B\&B$ idea's advantage to divide the big initial problem into smaller sub-problems, additionally diminishing the considered feasible set by adding constraints, retaining all required demands to the discrete problem's feasible set in order to find an optimal solution of latter. This exploitation of these two techniques' virtue makes

$B\&C$ a very strong tool, making it a method being implemented most commonly in commercial solvers [16].

3.3 Combinatorial Optimization Problem

With index set $N = \{1, \ldots, n\}$ and weights $w_j \in \mathbb{R}$ for every index $j \in N$, another type of optimization problem, the **Combinatorial Optimization Problem** (**COP**) can be formulated as follows:

$$\max_{S \in \mathcal{F}} \left\{ \sum_{j \in S} w_j \right\}$$

where $\mathcal{F} \subset \mathcal{P}(N)$ is the set of feasible subsets of the index set N.

Remark: Of course, all introduced cases of optimization problems can be combined, just as the application requires, in order to be able to model given task in an appropriate way. This leads to the general formulation of a linear optimization problem [21]:

$$\max_{\substack{x_{\text{real}} \in \mathbb{R}_+^p, \\ x_{\text{int}} \in \mathbb{Z}_+^{n-p-q}, \\ x_{\text{bin}} \in \{0,1\}^q, \\ S \in \mathcal{F}}} \{c^T x_{\text{real}} + h^T x_{\text{int}} + l^T x_{\text{bin}} + \sum_{j \in S} w_j : Ax_{\text{real}} + Gx_{\text{int}} + Hx_{\text{bin}} \leq b\}$$

3.4 Reformulation Methods

There is a variety of algorithmic solvers available, which, when given task is formulated in an adequate way, are able to solve given optimization problem efficiently. In practice however, the first trial of modeling a given task often leads to an inappropriate formulation with regards to the expression of constraints or the objective function, making an easy solvability of given problem impossible. Problems can also occur within the modeling process itself, when for instance constraints are desired to exclude each other or in case non-linear expressions appear.

Nevertheless there are tricks at hand in many cases, which allow a remodeling of constraints and the objective function towards an appropriate formulation. In the following, a few tricks, relevant for this thesis' approach specifically, are introduced.

3.4.1 Minimization vs. Maximization

Since an optimization problem modeled as a maximization problem of the objective function can be transformed to a minimization problem by exchanging the objective function:

$$\max_{x \in S} g(x) = \min_{x \in S} -g(x)$$

equvalent approaches for the same task can be made by either formulating an objective function to be maximized or its negative version to be minimized, which is of great importance when for instance being in need of reformulating a given optimization problem towards Standard Form 3.3 [8].

3.4.2 Equalities and Inequalities

The feasible set within a linear program is modeled by equality and inequality constraints [37]. In order to switch amongst those possible constraint formulations, there are several obvious, but still noteworthy methods at hand:

3.4.2.1 Transforming Inequalities

By multiplying both sides of a $\leq$ constraint by -1, it is transformed into a $\geq$ constraint, and vice versa:

$$\sum_{j=1}^{n} a_{ij}x_j \leq b_i \xrightarrow{\cdot(-1)} \sum_{j=1}^{n} \tilde{a}_{ij}x_j \geq \tilde{b}_i$$
$$\sum_{j=1}^{n} a_{ij}x_j \geq b_i \xrightarrow{\cdot(-1)} \sum_{j=1}^{n} \tilde{a}_{ij}x_j \leq \tilde{b}_i$$

for $i = 1, \ldots, m$, where a_{ij} is the i-th component of the j-th column of Constraint Matrix A, b_i is the i-th component of the right-hand-side vector b in linear program 3.2, $a_{ij} = -\tilde{a}_{ij}$ and $b_i = \tilde{b}_i$ [8].

3.4.2.2 Eliminating or Receiving Equalities

If given linear program modeled by equalities is required to be formulated by inequality constraints, a simple exchange of all equality constraints with two inequalities leads to success:

$$\sum_{j=1}^{n} a_{ij}x_j = b_i \qquad \Leftrightarrow \qquad \begin{array}{l} \sum_{j=1}^{n} a_{ij}x_j \leq b_i \\ \sum_{j=1}^{n} a_{ij}x_j \geq b_i \end{array}$$

On the other hand, if a transformation from an inequality to an equality is required, a **Slack Variable** $s \in \mathbb{R}$ with an additional constraint of non-negativity provides desired result:

$$\sum_{j=1}^{n} a_{ij}x_j \leq b_i \qquad \Leftrightarrow \qquad \begin{array}{c} \sum_{j=1}^{n} a_{ij}x_j + s = b_i \\ s \geq 0 \end{array}$$

$$\sum_{j=1}^{n} a_{ij}x_j \geq b_i \qquad \Leftrightarrow \qquad \begin{array}{c} \sum_{j=1}^{n} a_{ij}x_j - s = b_i \\ s \geq 0 \end{array}$$

3.4.3 Elimination of Free Variables

The non-negativity constraints to a decision variable, primarily formulated without any restrictions to its sign, can be generated by substitution of the former variable x_j by decision variables x_j^+ and x_j^- and additional constraints of non-negativity [12]:

$$x_j \in \mathbb{R}^n \quad \Leftrightarrow \quad \begin{array}{c} x_j^+ - x_j^- \\ x_j^+, x_j^- \in \mathbb{R}_+^n \end{array}$$

3.4.4 Formulating OR-Statements

In many applications – just as in the task of this thesis – a possibility of connecting two constraints by an OR relation, such that at least one, but not necessarily both of them, have to be considered within the process of optimization, is required to being modeled. This can be established by introducing two binary variables z and $\tilde{z}$ in connection with a sufficiently large number $M \in \mathbb{R}_+$:

$$\begin{array}{c} \sum\limits_{j=1}^{n} a_{ij}x_j \leq b_i \\ or \\ \sum\limits_{j=1}^{n} \tilde{a}_{ij}x_j \leq \tilde{b}_i \end{array} \quad \Leftrightarrow \quad \begin{array}{c} \sum\limits_{j=1}^{n} a_{ij}x_j - M \cdot z \leq b_i \\ \sum\limits_{j=1}^{n} a_{ij}x_j - M \cdot \tilde{z} \leq b_i \\ z + \tilde{z} \leq 1 \\ z, \tilde{z} \in \{0, 1\} \end{array}$$

For $z = 1$ (respectively $\tilde{z} = 1$), the first (respectively the second) constraint is automatically fulfilled and thus *inactive*. In this way, it is possible to model a neglection of either one of the constraints. The constraint to the sum of z and $\tilde{z}$ however makes sure that not more than one of them is denied, thus generating desired OR-behavior.

If an exclusive OR relation between two constraints is required, such that one and only one of them is *active*, while the other one is desired to be neglected in the process of optimization, the introduction of a single variable z is sufficient for an appropriate formulation:

$$\begin{array}{c} \sum\limits_{j=1}^{n} a_{ij}x_j \leq b_i \\ or \\ \sum\limits_{j=1}^{n} \tilde{a}_{ij}x_j \leq \tilde{b}_i \end{array} \quad \Leftrightarrow \quad \begin{array}{c} \sum\limits_{j=1}^{n} a_{ij}x_j - M \cdot z \leq b_i \\ \sum\limits_{j=1}^{n} a_{ij}x_j - M \cdot (1 - z) \leq b_i \\ z \in \{0, 1\} \end{array}$$

where the first constraint is active if $z = 0$, while it is ignored and constraint two is active in case of $z = 1$ [21].

3.4.5 Eliminating Non-Linearity

Since in linear programming the objective function as well as all constraints are required to be of a linear form, every occurring non-linearity poses a major problem. In many cases however there are tricks at hand, which, by introducing new decision variables and reformulating particular parts of the objective function respectively the constraints, tide over these troubles, leading to appropriate formulations of linear programs.

3.4.5.1 Interpolation

Any continuous function can be approximated by a piecewise linear function. Therefore, a non-linear objective function $g : S \longrightarrow \mathbb{R}$ can easily be replaced by its linearily interpolated version $\tilde{g} : S \longrightarrow \mathbb{R}$, specified by chosen **Partition Points** $(x_j, g(x_j)), j = 1, \ldots, r$:

$$g(x) \approx \tilde{g}(x) = \sum_{j=1}^{r} \lambda_j g(x_j), \qquad \sum_{j=1}^{r} \lambda_j = 1$$
$$\lambda_j \in \mathbb{R}_+ \; \forall j = 1, \ldots, r$$

By claiming $x = \lambda_i x_i + \lambda_{i+1} x_{i+1}$ and $\lambda_i + \lambda_{i+1} = 1$ for $x_i \leq x \leq x_{i+1}, i = 1, \ldots, r-1$, uniqueness of the λ_j's to a certain degree can be generated. By introducing binary decision variables $z_j, j = 1, \ldots, r-1$, a way to control the "active" segment of the interpolating, piecewise linear function as replacement of former, non-linear objective function can be modeled by adding following constraints to the program:

$$\begin{aligned}
&\lambda_1 \leq z_1 \\
&\lambda_j \leq z_{j-1} + z_j \; \forall j = 1, \ldots, r-1 \\
&\lambda_r \leq z_{r-1} \\
&\sum_{j=1}^{r-1} z_j = 1 \\
&x_j \in \{0, 1\} \; \forall j = 1, \ldots, r-1
\end{aligned}$$

where for one and only one $i \in \{1, \ldots, r-1\}$ $x_i = 1$, thus, $\lambda_j = 0 \; \forall j \neq i \; \wedge \; j \neq i+1$, thus only one segment is chosen to be "active" [21], [24], [37].

3.4.5.2 Substitution

If the trial of formulating a given task as optimization problem leads to a non-linearity in a constraint being induced by a bounded decision variable, either continuous or integer, being multiplied by an arbitrary number of binary decision variables, substitution of the constraint by a list of other appropriate constraints indeed provides the opportunity of reformulating the inadequate non-linear constraint in a linear way. So for the product of decision variable x, either being continuous or integer, bounded by some number $u \in \mathbb{R}$: $x \leq u$, with binary decision variables $z_j, j = 1, \ldots, k$, following equivalence holds:

$$
\begin{array}{c}
\left(\prod_{j=1}^{k} z_j\right) \cdot x \\
z_j \in \{0,1\},\ j = 1,\dots k \\
x \in \mathbb{R}_+ \vee x \in \mathbb{Z}_+
\end{array}
\quad \Leftrightarrow \quad
\begin{array}{c}
w \\
w \leq z_j \cdot u,\ j = 1,\dots,k \\
w \geq 0 \\
w \leq x \\
w \geq u \cdot \left(\sum_{j=1}^{k} z_j - k\right) + x \\
w \in \mathbb{R} \\
z_j \in \{0,1\},\ j = 1,\dots k
\end{array}
$$

The first constraint derives from the fact, that, since the product of x, which is bounded by u, multiplied by either 0 or 1 – thus the substitution variable w – can not be larger than the bound u of x.
On the other hand, if all z_j in the former formulation are equal to 1, it must be guaranteed that $\left(\prod_{j=1}^{k} z_j\right) \cdot x = x = w$, which is done by two inequalities: Since w can not take on values larger than x itself per definition, this direction of inequality is easily comprehensible. For the other inequality making up the fourth constraint, the sum of all binary variables z_j guarantees, that, in prevailling case of all $z_j = 1$, the difference with k results in the whole term in parentheses being 0, thus, constraint 4 restricts w to be greater or equal to x, which, together with constraint 3 leads to $w = x$, which was the aim to achieve [33].

3.4.6 Duality

Given linear program of Equation 3.2, which in the context of Duality is called the **Primal Problem**, its related **Dual Problem** is defined as:

$$\min_{u \in \mathbb{R}_+^m} \{u^T b : u^T A \geq c\} \tag{3.10}$$

where u is a non-negative, m-dimensional column vector of decision variables, resulting from the transformation towards the dual formulation. Since the dual problem of the dual problem is the primal problem again, the roles of related problems can be swapped unconcernedly.

The importance of Duality in linear programming becomes clear when stating the resulting fact of **Weak Duality**, which postulates, that a feasible solution $\hat{u}$ of dual problem 3.10 provides an upper bound, whereas a feasible solution $\hat{x}$ of 3.2 provides a lower bound for desired, optimal solution x^* of problem 3.2:

$$c^T \hat{x} \leq \max_{x \in \mathbb{R}_+^n} \{c^T x : Ax \leq b\} = x* \leq \min_{u \in \mathbb{R}_+^m} \{u^T b : u^T A \geq c\} \leq (\hat{u})^T c$$

respectively the even more important result of **Strong Duality**, stating that if either one of the optimal solutions x^* of 3.2 or u^* 3.10 is finite, both 3.2 and 3.10 are finite and equal[9]

[9] This does not hold for MIPs in general, however [16].

$$x^* = \max_{x \in \mathbb{R}^n_+} \{c^T x : Ax \leq b\} = \min_{u \in \mathbb{R}^m_+} \{u^T b : u^T A \geq c\} = u^* < \infty$$

These principles of Duality occur in many different aspects of linear programming, for instance in the field of **Sensitivity Analysis**, where influences of changes of input data on the final outcome are observed [16], or in a version of implementation of the Simplex Algorithm, where its dual version is consulted in case a basic solution of the dual problem, but none of the primal one is known [24], making it a very important topic in theory and application of linear programming.

3.4.7 Relaxation

The idea of **Relaxation** is to enlarge the set of feasible solutions of a given optimization problem respectively replacing given objective function with a function taking on equal or larger values everywhere. Therefore, with sets $S \in \mathbb{R}^n$ and $T \in \mathbb{R}^n$, and functions $g : S \longrightarrow \mathbb{R}$ and $r : T \longrightarrow \mathbb{R}$, the problem $\max_{x \in T}\{r(x)\}$ is a relaxation of $\max_{x \in S}\{g(x)\}$ if

i) $S \subseteq T$

ii) $r(x) \geq g(x) \; \forall x \in S$

An important special case of relaxation is the **Linear Program Relaxation** of an integer program. For program 3.6, it is defined as

$$\max_{x \in \mathbb{R}^n_+} \{c^T x : Ax \leq b\}$$

Relaxation is a strong tool when being interested in obtaining dual bounds, for instance as way of defining a termination criterion in an algorithm or to use it in solution approaches such as the Branch And Bound Method (Section 3.2.3.3) [38].

Part II

Solution Approach

Chapter 4

System Modeling

In order to get to the point of being able to formulate an algorithm solving the problem of prosthesis adjustment described in Section 2, it is necessary to define several terms, give objects and circumstances of reality a mathematical name and in this way build tools and ways to connect it with a language, the algorithm can be formulated in.

Therefore, in the following, a way of mathematically interpreting **myosignals**, which are measured by the prosthesis' surface electrodes (Section 4.1), the **amplify factor**, which serves as one of the most important parameters within prosthesis adjustment (Section 4.2), as well as all other **settable parameters** are defined and mathematically described (Table 4.3). In addition, internally defined **hard-coded constants** are listed (Table 4.2), which also make up an essential part of control pattern defining. Furthermore, all **functionality components** such as different **grips**, possible **movements** and the ways of getting access to them by certain **myo-signal patterns** and a number of available **Switches** is listed and described in terms of mathematical sets and functions (Section 4.4.1). In the end, all previously described tools and components can then be used in order to describe **programs** which consist of options of access to the functionality components and related settable parameter values, finally leading to the ability of formulating an algorithm which calculates actual optimal programs for desired prosthesis adjustment (Section 4.4.3).

4.1 Raw Myo-Signals

Let $\mathcal{T} \subseteq \mathbb{R}_0^+$ be the set of possible time-values as subset of all non-negative real numbers, $s_{\text{min}}, s_{\text{max}} \in \mathbb{R}_0^+$ be the minimum and maximum value of a myo-signal an electrode can measure, $\mathcal{V}_{\mathcal{S}_{\text{raw}}} := [s_{\text{min}}, s_{\text{max}}]$ and $\mathcal{S}_{\text{raw}} := \{s : \mathcal{T} \longrightarrow \mathcal{V}_{\mathcal{S}_{\text{raw}}}, t \mapsto s(t)\}$ the set of all raw myo-signals, interpreted as functions mapping values of time to values within the measurement range of the surface electrodes. Further, let $\mathcal{S}_{\text{C1}_{\text{raw}}} \subset \mathcal{S}_{\text{raw}}$ be the subset of all raw myo-signals measured by electrode 1 ($\hat{=}$ channel 1) and $\mathcal{S}_{\text{C2}_{\text{raw}}} \subset \mathcal{S}_{\text{raw}}$ be the subset of all raw myo-signals measured by electrode 2 ($\hat{=}$ channel 2). Then the output of a two-electrode-construction of a myoelectric prosthesis can be interpreted as a vectorial-valued function

$$\vec{s}_{\text{raw}} = \begin{pmatrix} s_{\text{raw}_{\text{C}_1}} \\ s_{\text{raw}_{\text{C}_2}} \end{pmatrix} : \mathcal{T} \longrightarrow \mathcal{V}^2_{\mathcal{S}_{\text{raw}}}, t \mapsto \begin{pmatrix} s_{\text{raw}_{\text{C}_1}}(t) \\ s_{\text{raw}_{\text{C}_2}}(t) \end{pmatrix}, s_{\text{raw}_{\text{C}_i}} \in \mathcal{S}_{\text{C}i_{\text{raw}}}, i = 1, 2$$

Figure (4.1) shows the graph of a vectorial-valued myo-signal.

Remark: Depending on the context, by using the term *myo-signal*, either both components of the vectorial-valued function or just one single component of the vectorial-valued function is meant.

By setting up a system of constraints to the shape of myo-signals with the help of already mentioned settable parameters (Table 4.3), together with internally defined hard-coded constants (Table 4.2), and thus making them being recognized as control patterns by the prosthesis' internal control unit, they are used to get access on the prosthesis' functionality components, e.g. open or close the prosthetic hand, pronate or supinate a rotation joint, extend or flex adapted elbow joint or switch between modes (Table 4.1).

4.2 Amplifier

The **Amplifier** can be interpreted as a vectorial-valued function $\vec{\mathbf{F}}_{\mathcal{A}}$ which suppresses or heightens a signalvector's image (i.e. the output of the vectorial-valued function $\vec{s}_{\text{raw}}$) by the **amplify factor** $(f_{\mathcal{A}_1}, f_{\mathcal{A}_2}) =: \vec{f_{\mathcal{A}}} \in \mathcal{V}_{\mathcal{A}}^2 \subset \mathbb{R}^+ \times \mathbb{R}^+$

$$\vec{\mathbf{F}}_{\mathcal{A}} = \begin{pmatrix} \mathbf{F}_{\mathcal{A}_1} \\ \mathbf{F}_{\mathcal{A}_2} \end{pmatrix} : \mathcal{S}_{\text{raw}}^2 \longrightarrow \vec{\mathbf{F}}_{\mathcal{A}}\left(\mathcal{S}_{\text{raw}}^2\right) =: \mathcal{S}^2, \vec{s}_{\text{raw}} = \begin{pmatrix} s_{\text{raw}1} \\ s_{\text{raw}2} \end{pmatrix} \mapsto \begin{pmatrix} \mathbf{F}_{\mathcal{A}_1}(s_{\text{raw}1}) \\ \mathbf{F}_{\mathcal{A}_2}(s_{\text{raw}2}) \end{pmatrix}$$

where

$$\mathbf{F}_{\mathcal{A}_i}(s_{\text{raw}_i})(t) = s_{\text{raw}_i}(t) \cdot f_{\mathcal{A}_i}, t \in \mathcal{T}, i = 1, 2 \tag{4.1}$$

After being measured by the surface electrodes, every raw myo-signal $\vec{s}_{\text{raw}} \in \mathcal{S}_{\text{raw}}$ is immediately transformed to the amplified myo-signal $\vec{\mathbf{F}}_{\mathcal{A}}(\vec{s}_{\text{raw}}) \in \mathcal{S}^2$ before any other signal processing within the prosthesis' internal control unit happens. By default, $\vec{f_{\mathcal{A}}} = (1,1)^T$ and therefore $\vec{\mathbf{F}}_{\mathcal{A}} = id_{\mathcal{S}_{\text{raw}}^2}$ in case the amplify factors are not changed from default settings.

Since multiplication is linear, the co-domain of signals in $\mathcal{S}$ indeed is $[s_{\text{min}} \cdot f_{\mathcal{A}_i}, s_{\text{max}} \cdot f_{\mathcal{A}_i}] =: \mathcal{V}_{\text{S}_i}$ for $i = 1, 2$. For better readability, these channel-wise definition of co-domains is summarized by

$$\mathcal{V}_{\mathcal{S}} := [s_{\text{min}} \cdot \min_{i=1,2}(f_{\mathcal{A}_i}), s_{\text{max}} \cdot \max_{i=1,2}(f_{\mathcal{A}_i})]$$

and $\mathcal{S}_{\text{C}_i} \subset \mathcal{S}$, $i = 1, 2$ is set as the already amplified myo-signals of channel i as subset of $\mathcal{S}$.

As already mentioned $f_{\mathcal{A}}$ is part of the set $\mathcal{X}$ of settable parameters (Table 4.3), so its value can be set individually, depending on the prosthesis' user's strength of muscles and resulting heights of myo-signals.

Remark: In real-life application, the set of amplify factors available within *Ottobock*'s Data Station (Section 2.1) to be selected by the orthopedic technician for prosthesis adjustment is the discrete, finite set $\mathcal{V}_{\mathcal{A}} = \{0.25, 0.5, 0.75, 1, \ldots, 3.5, 3.75, 4\}$.

4.3 Myo-Graph

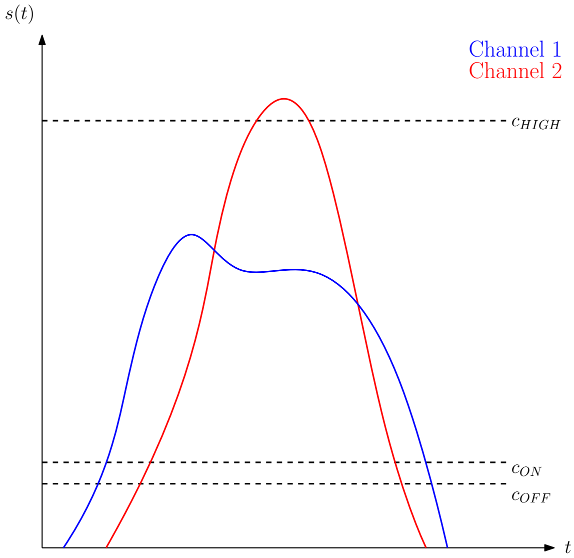

FIGURE 4.1: Myo-Graph

Within the company's graphical user interface for prosthesis setting, the *Ottobock* Data Station (Figure 2.2), myo-signals are plotted in the **Myo-Graph** (Figure 4.1). In the Myo-Graph, signals $s \in \mathcal{S}_{C_1}$ are drawn in blue, $s \in \mathcal{S}_{C_2}$ in red. For the sake of clarity, this habit is carried out through all figures in this thesis.

4.4 Functionality

Within the six-month work of setting up the software for automated prosthesis adjustment, the concept has been conceived for *Ottobock*'s *Michelangelo* Hand (Figure 2.1) in combination with *Ottobock*'s rotational joint, which currently forms a usual treatment for transradial amputees. The developed concepts can easily be adapted to more advanced treatments for example involving elbow joints, or can be used in cases of simpler device compilations where for instance a rotational joint is not adapted. In the following however, all definitions, descriptions and listings refer to this project's compilation of a *Michelangelo* hand connected to a rotational joint.

This certain prosthesis setup comes along with the capability of performing a variety of movements: Hand opening and closing in two different ways of grasping, i.e. **opposition grasp** and **opposition grasp** and pronation and supination due to the added rotational joint (Section 4.4.1.1). These movements can even be varied in their velocity by making use of the intensity of myo-signals.

In Section 4.4.1, all functionality components are defined and described in terms of mathematical sets and functions, before Section 4.4.2 explains the way of getting access to the different components by introducing all myo-signal patterns used in myo-prosthesis control.

4.4.1 Functionality Components

The essential functionality component of a myoelectric prosthesis is hand movement. It gives the user a lot of opportunities to compensate lost extremity by not only supporting residual limb statically, but also being able to take and hold things with the artificial hand, even being capable of varying speed of opening, closing or rotation. On the other side, a very beneficial functionality component in case of a treatment with *Michelangelo* hand, hand opening and closing can even be performed in two different ways of grasping, making it possible to grab and hold things more precisely, depending on their shape and constitution.

In the following, these three aspects of functionality – the prosthesis' **modes**, the **Movements** which can be performed in latters, and the movements' variable **Velocity** – are discussed.

4.4.1.1 Modes – Grasping and Rotation

A hand's mode is a functionality state, in which the hand performs it's movements (Section 4.4.1.2). In this project's device compilation 3 modes are available:

Symbol	Name of mode
M_{Lat}	*opposition* grasp (Figure 4.2)
M_{Opp}	*opposition* grasp (Figure 4.3)
M_{Rot}	*rotation* mode (Figure 4.4)

TABLE 4.1: Hand modes

Lateral Grasp: In lateral grasp mode, the artificial hand is in a position, where the thumb is held sideways to the remaining fingers. With this finger position, the hand can be opened and closed within this mode's mechanically possible moving range.

Opposition Grasp: The opposition grasp mode is another hand position, where the thumb is opposed to its remaining fingers. With this finger position, the hand can be opened and closed within this mode's mechanically possible moving range.

Rotation Mode: When switched to rotation mode, the prosthesis freezes in its previous grasp mode and position and is able to pronate and supinate withing its mechanically possible moving range.

The prosthesis can only be in one mode at a time, if the user wants to switch to another mode, certain myo-signal patterns, the so-called **switch methods** have to be generated (Section 4.4.2.1). Let for later formulation $\mathcal{M} := \{M_{\mathrm{Lat}}, M_{\mathrm{Opp}}, M_{\mathrm{Rot}}\}$ be the set of all prosthesis' available modes.

FIGURE 4.2: Lateral grasp mode – closed

FIGURE 4.3: Opposition grasp mode – closed

FIGURE 4.4: Lateral grasp – rotation mode

4.4.1.2 Movements

Being in a currrent prosthesis' mode, the device is capable of performing several movements depending on which mode is currently active. In grasp modes, the *Michelangelo* hand can be opened and closed, in rotation mode, it can be pronated and supinated. Of course, the range of movement is mechanically limited, so once the artificial hand is opened respectively closed fully or rotation has been accomplished to the joint's maximal displacement, movement stops, even in case of validly provided myo-signal patterns. The way of getting access to movement is explained in Section 4.4.2.2.

4.4.1.3 Velocity

All movements can also be varied in speed, regulated by the intensity of the user's myo-signals. Therefore, consistent training for good muscle control is needed, in order to be able to exploit the prosthesis functionality to the fullest. Section 4.4.2.3 defines the term *Velocity* as mathematical function and explains the way of controlling it with the user's myo-signals.

4.4.2 Access to Functionality Components

As explained in Section 1.1, muscle contractions and relaxations lead to a voltage change of nerve fiber's membrane, which can be measured by surface electrodes and used as way of communication between user and prosthesis. Contraction of a muscle leads to a rise of the myo-signal, whereas relaxed muscles usually generate steady myo-signals around a close-to-zero equilibrium (Figure 1.15, Figure 1.16).
Hence, the idea of myoelectric prostheses is to make use of the user's ability to consciously contract and relax several muscle regions, i.e. generate specific shapes of myo-signals and thus tell the prosthesis' internal control unit which actions to perform. By a number of **Hard-Coded Constants**[1] (Table 4.2) and individually **Settable Parameters** (Table 4.3), constraints to the shape of myo-signals can be formulated such that the internal control unit can react, whenever these constraints are held by the myo-signals provided by the user.

Symbol	Explanation
$c_{\text{ON}} \in \mathcal{V}_{\mathcal{S}}$	ON-threshold of myo-signal recognition
$c_{\text{HIGH}} \in \mathcal{V}_{\mathcal{S}}$	maximal upper proportional bound
$c_{\text{OFF}} \in \mathcal{V}_{\mathcal{S}}$	OFF-threshold of myo-signal recognition
$c_{\text{T}} \in \mathbb{R}^+$	length of control unit's internal counter
$c_{\text{LHOV}} \in [0, 1]$	value of control unit's internal long Hand-open threshold

TABLE 4.2: Hard-coded constant values

Symbol	Explanation
$x_{\text{I}_i} \in [c_{\text{ON}}, c_{\text{HIGH}}]$	*Impulse/4-channel border* for channel $i, i = 1, 2$
$x_{\text{C}_i} \in [c_{\text{ON}}, c_{\text{HIGH}}]$	*Cocontraction border* for channel $i, i = 1, 2$
$x_{\text{SL}} \in \mathcal{V}_{\mathcal{SL}}$	*Maximal signal length* for both channels
$x_{\text{LHOL}} \in \mathcal{V}_{\text{LHOL}}$	*Minimal Long Hand Open Length* for channel 2
$\vec{f_{\mathcal{A}}} = \begin{pmatrix} f_{\mathcal{A}_1} \\ f_{\mathcal{A}_2} \end{pmatrix} \in \mathcal{V}_{\text{A}}^2$	*amplify factor* for each channel
$\vec{f_{\text{W}}} = \begin{pmatrix} f_{\text{W}_1} \\ f_{\text{W}_2} \end{pmatrix} \in [0, 1]^2$	*Workspace factor* for each channel

TABLE 4.3: Settable parameters

Moreover, let $\mathcal{X} := \{x_{\text{I}_1}, x_{\text{I}_2}, x_{\text{C}_1}, x_{\text{C}_2}, x_{\text{SL}}, x_{\text{LHOL}}, \vec{f_{\mathcal{A}}}, \vec{f_{\text{W}}}\}$ be the set of all settable parameters and define following values for $s \in \mathcal{S}$ and $\vec{s} = \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} \in \mathcal{S}^2$ for better readability within further formulation:

$$t_{s_0} := \min\{t \in \mathcal{T} : s(t) \geq c_{\text{ON}}\} \tag{4.2}$$

$$t_{s_\infty} := \min\{t \in \mathcal{T}, t \geq t_{s_0} : s(t) < c_{\text{OFF}}\} \tag{4.3}$$

$$t_{\vec{s}_0}^{\text{sync}} := \min_{i=1,2}\{t \in \mathcal{T} : s_i(t) \geq c_{\text{ON}}\} \tag{4.4}$$

$$t_{\vec{s}_\infty}^{\text{sync}} := \max_{i=1,2}\left\{\min\{t \in \mathcal{T}, t \geq t_{\vec{s}_0}^{\text{sync}} : s_i(t) < c_{\text{OFF}}\}\right\} \tag{4.5}$$

$$t_{H_\infty} := \min\{t \in \mathcal{T} : h(t) = H_\infty \wedge s_2(t) \geq (c_{\text{ON}} + (c_{\text{HIGH}} - c_{\text{ON}}) \cdot f_{\mathcal{W}} \cdot c_{\text{LHOV}})\} \tag{4.6}$$

[1]Fixed internal values which are not changeable.

Remark: For a myo-signal $s \in \mathcal{S}$ provided by the user, (4.2) defines the moment in time, in which it exceeds the threshold value c_{ON} for the very first time within the period of observation.
For a myo-signal, which has already exceeded the threshold value c_{ON} at some point and currently takes on values above c_{ON}, (4.3) defines the point time, the myo-signal s falls below the threshold value c_{OFF} again.
(4.4) and (4.5) are defined analogously for myo-signals $\vec{s} \in \mathcal{S}^2$, determining the time of the first exceeding of c_{ON} and the last falling below c_{OFF} of either one of the components of the vectorial-valued myo-signal $\vec{s}$.
Definition (4.6) indicates the point in time, the prosthesis reaches the position of mechanical maximal openness and cannot perform any further movement of opening even in case of validly provided myo-signal patterns.

For later calculation, in case a myo-signal $s \in \mathcal{S}$ or $s \in \mathcal{S}^2$ does not exceed the threshold value c_{ON} at all, t_{s_0} and $t_{\vec{s}_0}^{\text{sync}}$ are defined as $-\infty$.

4.4.2.1 Access to Modes – Switches

The concept of **Switches** is the way of changing from one hand mode (Section 4.4.1.1) to another by generating specific myo-signal patterns, the so-called **switch methods**.

A switch method can be interpreted as a set of myo-signals, holding a number of constraints formulated with the help of hard-coded constants (Table 4.2) and parameters which can be set individually for every user (Table 4.3).

When $\mathcal{SM}$ is the set of all switch methods, a switch can be interpreted as a transformation function $sw : \mathcal{M} \times \mathcal{SM} \longrightarrow \mathcal{M}$. that defines a way to reach a certain hand mode $M \in \mathcal{M}$ from another one by a certain switch method $SM \in \mathcal{SM}$. This means, a myo-signal $s \in \mathcal{S}$ respectively $s \in \mathcal{S}^2$, which holds all constraints bound to a certain switch method SM, is recognized as its member, i.e. $s \in SM$ and, if latter is contained by the prosthesis' currently selected **Program** (Section 4.4.3) as part of a Switch sw, the defined mode-changing event within the prosthesis' internal control unit is triggered.

Switch Methods: Table 4.4 shows all available switch methods for the *Michelangelo* hand in order to change from one mode to another. In the following, every switch method is defined in terms of mathematical sets and shown graphically in order to provide a better idea of the constraints to the myo-signal shapes to the reader.

Symbol	Explanation
$S_{4\text{C}_i}$	4-*Channel-Switch-Method* of channel $i, i = 1, 2$
S_{CS}	*Short Cocontraction-Switch-Method*
S_{CL}	*Long Cocontraction-Switch-Method*
S_{I_i}	*Impulse-Switch-Method* of channel $i, i = 1, 2$
S_{LHO}	*Long Hand-Open switch method*

TABLE 4.4: Switch methods

4-Channel Switch Method

$$S_{4\mathrm{C}_i} := \{s \in \mathcal{S}_{\mathrm{C}_i} : \max_{t \in [t_{s_0}, t_{s_0}+c_{\mathrm{T}}]} s(t) \geq x_{\mathrm{I}_i}\}, i = 1, 2 \tag{4.7}$$

In words: A myo-signal s measured by electrode i is recognized as 4-Channel switch method of channel i, if s exceeds the (settable) threshold x_{I_i} within the (hard-coded) amount of time c_{T} after having firstly exceeded the (hard-coded) threshold c_{ON} (Figure 4.5, Figure 4.6).

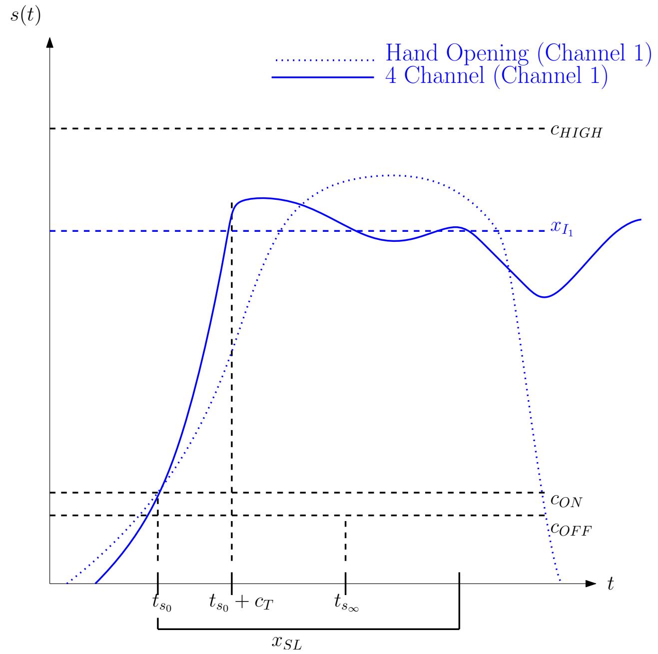

FIGURE 4.5: 4-Channel switch method channel 1

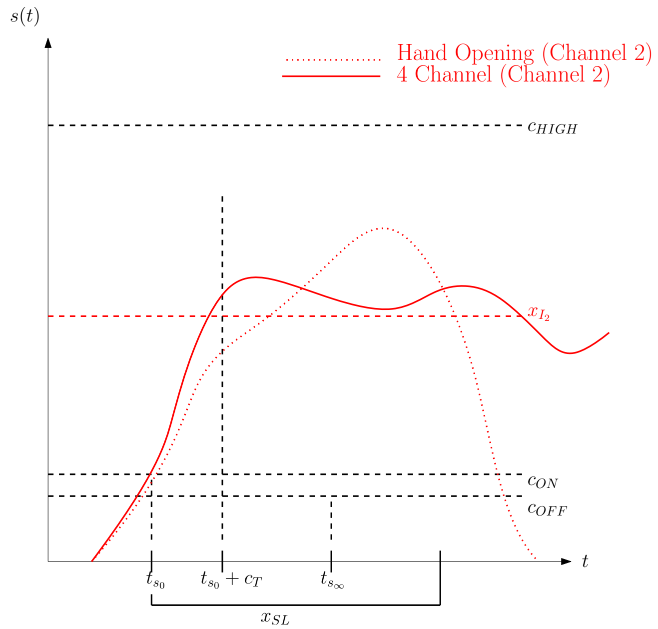

FIGURE 4.6: 4-Channel switch method channel 2

Short/Long Cocontraction

$$S_{\text{CS}} := \{\vec{s} = \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} \in \mathcal{S}^2 : \max_{t \in [t^{\text{sync}}_{s_{i_0}}, t^{\text{sync}}_{s_{i_0}} + c_{\text{T}}]} s_i(t) \geq x_{\text{C}_i} \wedge |t^{\text{sync}}_{s_{i_0}} - t^{\text{sync}}_{s_{i_\infty}}| \leq x_{\text{SL}}, i = 1, 2\}$$

In words: Three conditions have to be fullfilled by the myo-signal $\vec{s}$ in order to be recognized as Short Cocontraction switch method. Both components s_i of the myo-signal $\vec{s}$ have to exceed the (settable) thresholds x_{C_i} within a certain (hard-coded) amount of time c_{T} after firstly exceeding c_{ON}. Additionally, the period of time between the first c_{ON}-passover and the last c_{OFF}-undershoot of both components must not be longer than the (settable) time x_{SL} (Figure 4.7).

$$S_{\text{CL}} := \{\vec{s} = \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} \in \mathcal{S}^2 : \max_{t \in [t^{\text{sync}}_{s_{i_0}}, t^{\text{sync}}_{s_{i_0}} + c_{\text{T}}]} s_i(t) \geq x_{\text{C}_i} \wedge |t^{\text{sync}}_{s_{i_0}} - t^{\text{sync}}_{s_{i_\infty}}| > x_{\text{SL}}, i = 1, 2\}$$

In words: The Long Cocontraction-Switch-Method is defined very similar to the Short Cocontraction-Switch-Method. For being recognized as Long Cocontraction switch method, the thresholds x_{C_i} must still be exceeded by both myo-signal components s_i within the (hard-coded) amount of time c_{T} after firstly exceeding c_{ON} and both have to be kept above x_{C_i} for a longer period of time than x_{SL} before falling below c_{OFF} again (Figure 4.8).

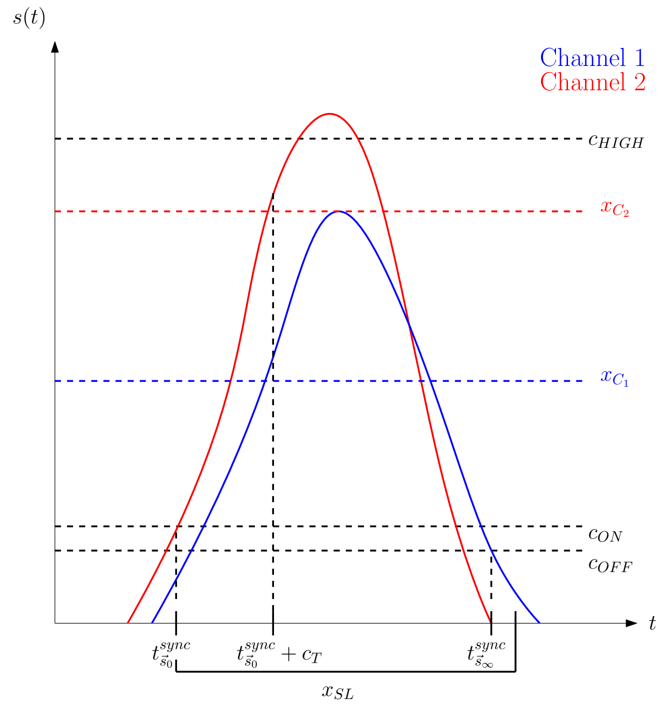

FIGURE 4.7: Short Cocontraction switch method

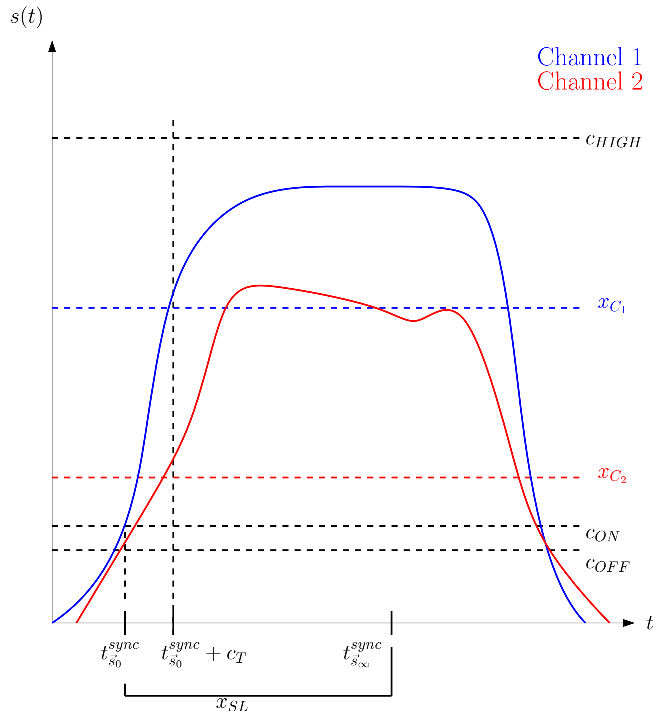

FIGURE 4.8: Long Cocontraction switch method

Impulse

$$S_{\mathrm{I}_i} := \{s \in \mathcal{S}_{\mathrm{C}_i} : \max_{t \in [t_{s0}, t_{s0}+c_{\mathrm{T}}]} s(t) \geq x_{\mathrm{I}_i} \wedge |t_{s0} - t_{s\infty}| \leq x_{\mathrm{SL}}\}, i = 1, 2 \tag{4.8}$$

In words: The constraints of Impulse switch method follow the same principle as the constraints of Cocontraction switch method, but with the observation of a myo-signal's single component only. This means, the myo-signal's component s measured by electrode i is recognized as Impulse switch method of channel i, if it exceeds the (settable) threshold x_{I_i} within the (hard-coded) amount of time c_{T} after firstly exceeding c_{ON}, and if it falls below c_{OFF} again within the (settable) amount of time x_{SL} (Figure 4.9, Figure 4.10).

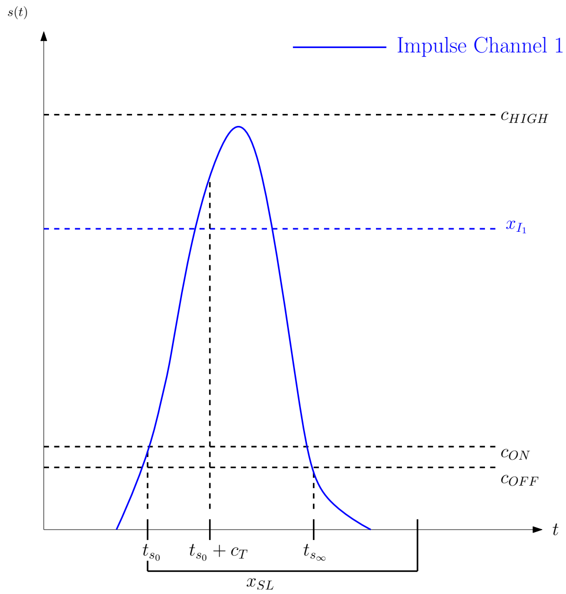

FIGURE 4.9: Impulse switch method channel 1

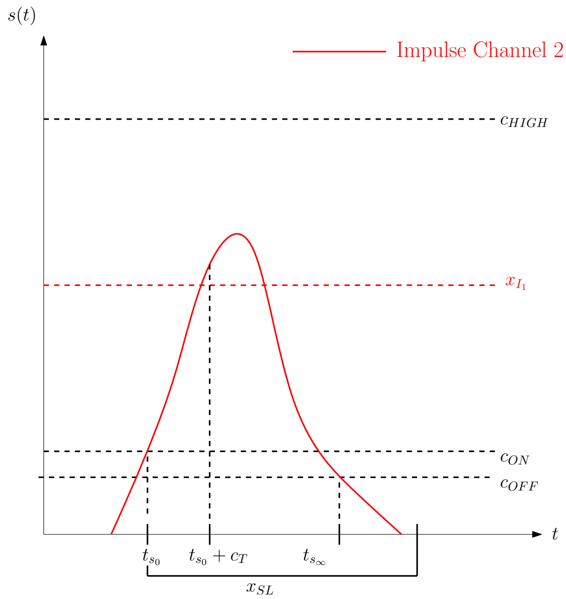

FIGURE 4.10: Impulse switch method channel 2

Long Hand-Open Switch Method

$$S_{\text{LHO}} := \{s \in \mathcal{S}_{\text{C}_2} : \min_{t \in [t_{H_\infty}, t_{H_\infty} + x_{\text{LHOL}}]} s(t) \geq c_{\text{ON}} + (c_{\text{HIGH}} - c_{\text{ON}}) \cdot f_{\text{W}_2} \cdot c_{\text{LHOV}}\} \quad (4.9)$$

In words: The Long Hand-Open switch method can only be recognized when given constraints are held by the myo-signal's component measured by electrode 2. This fact is based on the idea that the myo-signal component of channel 2 is used to perform hand opening within movement generation (Section 4.4.2.2). Thus, at some point of an appropriately provided myo-signal measured by electrode 2, the prosthesis is at its position of mechanical maximal openness H_∞. At this point, the Long Hand-Open switch method can then be triggered by keeping the myo-signal's values of the 2. component above the (hard-coded) ratio c_{LHOV} of the (settable) **Workspace** (Section 4.4.2.3) $W_2 = [c_{\text{ON}}, c_{\text{ON}} + (c_{\text{HIGH}} - c_{\text{ON}}) \cdot f_{\text{W}_2}]$ of channel 2 within the (settable) period of time x_{LHOL} after firstly reaching the position of mechanical maximal openness H_∞ (Figure 4.11).

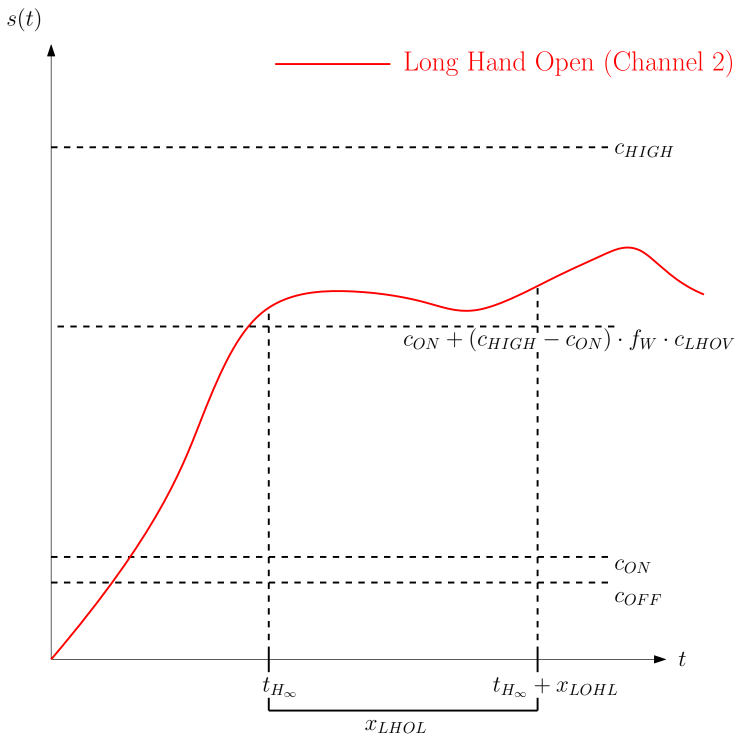

FIGURE 4.11: Long Hand-Open switch method

Switch Matrix: Within *Ottobock*'s Data Station (Figure 2.2), a Switch can defined by the orthopedic technician with the help of the so-called **Switch Matrix** (Figure 4.12). Due to the practical graphical user interface, the way of changing from one mode to another can be set easily via button-clicks to a well-defined function, preventing from complications and errors.

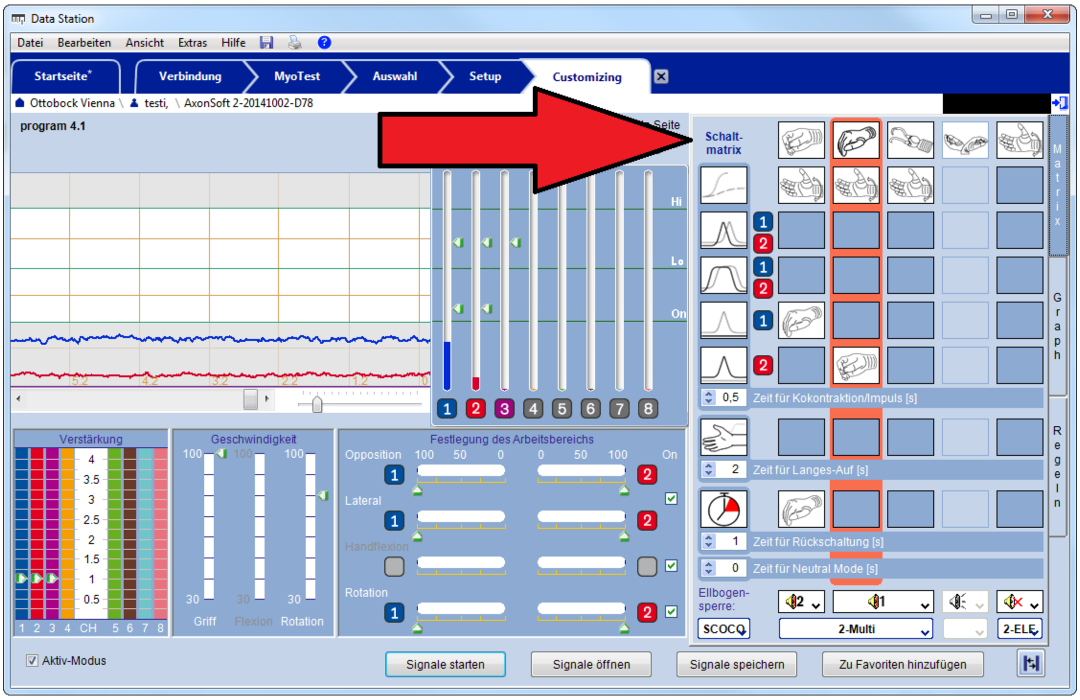

FIGURE 4.12: Switch Matrix in *Ottobock*'s Data Station

4.4.2.2 Access to Movement – MultiGrip

A **Control Setting** defines the requirements to a myo-signal in order to be recognized as intended prosthesis movement within a certain mode. Thus, it can be interpreted as a function cs receiving a vectorial-valued myo-signal $\vec{s}$, mapping it to either a non-zero value of movement-velocity v_{OUT} in case a valid myo-signal is provided, or 0:

$$cs : \mathcal{S}^2 \longrightarrow \mathcal{V}_{\mathrm{V}} : \vec{s} \mapsto \begin{cases} v_{OUT} & \text{if } \vec{s} \text{ holds all constraints of selected Control Setting \&} \\ & \text{if prosthesis is in mechanically movable condition} \\ 0 & \text{else} \end{cases}$$

where $\mathcal{V}_{\mathrm{V}} := [-v_{\max}, v_{\max}]$ is the possible range of movement-velocity and v_{OUT} is the actual velocity of the generated movement (Section 4.4.2.3).

For the sake of convenience and because this way of movement transmission is the most common in use, the Control Setting named **MultiGrip** is being explained and considered only.

MultiGrip: For a myo-signal $\vec{s} = \begin{pmatrix} s_1 \\ s_2 \end{pmatrix}$, the Control Setting MultiGrip defines component s_1 measured by electrode 1 as the component causing hand closure in case the prosthesis is currently in either one of the grasp modes, or pronation in case of being in rotational mode. For following mathematical formulation, these movements are construed as negative velocity, i.e. the Control Setting function cs assigns velocity values, labeled with a negative sign. Myo-signal component s_2 measured by electrode 2 is capable of initializing hand opening or supnation, i.e. positive velocity.

In order to generate movement within current mode, provided myo-signal must not be contained in one of the sets of Switch modes which are part of currently selected Switch, since obviously in this case a switching event would be triggered. Furthermore, one of the myo-signal components must exceed the threshold value c_{ON}:

$$(\exists t_0 \in \mathcal{T} : s_1(t_0) \geq c_{\mathrm{ON}}) \vee (\exists t_0 \in \mathcal{T} : s_2(t_0) \geq c_{\mathrm{ON}})$$

If both myo-signal components s_1 and s_2 reach values $\geq c_{\mathrm{ON}}$, a simple First-Come-First-Serve-Method $pr_{1^{\mathrm{st}}} : \mathcal{S}^2 \longrightarrow \mathcal{S}$:

$$pr_{1^{\mathrm{st}}}(\vec{s}) := \begin{cases} s_1 & \min\{t \in \mathcal{T} : s_1(t) \geq c_{\mathrm{ON}}\} \leq \min\{t \in \mathcal{T} : s_2(t) \geq c_{\mathrm{ON}}\} \\ s_2 & \min\{t \in \mathcal{T} : s_1(t) \geq c_{\mathrm{ON}}\} > \min\{t \in \mathcal{T} : s_2(t) \geq c_{\mathrm{ON}}\} \end{cases}$$

determines the first component of the myo-signal to assign a value $\geq c_{\mathrm{ON}}$ and initializes its corresponding movement.

4.4.2.3 Access to Velocity Control

Movements within available modes can be performed in various speed. In the Control Setting MultiGrip (Section 4.4.2.2), the velocity of movement can be controlled by the strength of myo-signal the user generates, i.e. the prosthesis can be closed/pronated faster respectively slower by increasing respectively reducing the strength of muscle contraction measured by electrode 1, analogously for opening/-supination for myo-signals measured by electrode 2.

When one of the myo-signal components has exceeded the threshold value c_{ON} and its values are located within the so-called **Proportional Workspace**, the speed changes linearly with myo-signal strength, beginning from velocity value ($\pm$) 0 up to a maximal velocity value $\pm v_{\text{max}}$. The Proportional Workspace is defined via the hard-coded threshold values c_{ON} and c_{HIGH} together with a settable factor $f_{\mathcal{W}_i} \in (0, 1]$ and can be seen as ratio of the full interval $[c_{\text{ON}}, c_{\text{HIGH}}]$ in which linear velocity change proportional to myo-signal strength is desired to be performed. In this way, a range of proportionality suitable for the user's myo-signals can be set individually, such that prosthesis velocity can be controlled in a subtle, precise way more easily even if provided myo-signals might not be very strong or capable of very fine variations.

So for myo-signals measured by electrode i, $i = 1, 2$, the Proportional Workspace, within which a linear rise of velocity values proportional to the rise of myo-signal strength is performed, is defined by

$$\mathcal{W}_i = [c_{\text{ON}}, c_{\text{ON}} + (c_{\text{HIGH}} - c_{\text{ON}}) \cdot f_{\mathcal{W}_i}]$$

Remark: In the following, for reasons of readability, $f_{\mathcal{W}}$ and $\mathcal{W}$ will be written without myo-signal component indicating index and be interpreted dependent on the context.

By default, $f_{\mathcal{W}} = 1$, so the default range of proportionality is the whole range from c_{ON} to c_{HIGH}, well seen in Figure 4.1.

The velocity $v : \mathcal{V}_{\mathcal{S}} \longrightarrow \mathcal{V}_{\text{V}}$ theoretically being indicated by provided myo-signal s for every $t \in \mathcal{T}$, but not yet considering the prosthesis' mechanical position, i.e. its current mechanical capability of actually performing movement can be seen in Figure 4.13 respectively Equation 4.10.

$$v(s(\cdot)) = \begin{cases} -v_{\text{max}} & s \in \mathcal{S}_{\text{C}_1} \wedge s(\cdot) \geq \max\{x : x \in \mathcal{W}\} \\ -\frac{v_{\text{max}}(s(\cdot) - c_{\text{ON}})}{f_{\mathcal{W}} \cdot (c_{\text{HIGH}} - c_{\text{ON}})} & s \in \mathcal{S}_{\text{C}_1} \wedge s(\cdot) \in \mathcal{W} \\ 0 & s(\cdot) \leq c_{\text{ON}} \\ \frac{v_{\text{max}}(s(\cdot) - c_{\text{ON}})}{f_{\mathcal{W}} \cdot (c_{\text{HIGH}} - c_{\text{ON}})} & s \in \mathcal{S}_{\text{C}_2} \wedge s(\cdot) \in \mathcal{W} \\ v_{\text{max}} & s \in \mathcal{S}_{\text{C}_2} \wedge s(\cdot) \geq \max\{x : x \in \mathcal{W}\} \end{cases} \tag{4.10}$$

The actual movement velocity of the prosthesis also depends on the prosthesis' mechanical position. Therefore, it is necessary to define the range of mechanical prosthesis movement $\mathcal{H} = [H_0, H_\infty]$ within a mode, where H_0 is the prosthesis' mechanical position of maximal closeness respectively pronation, and H_∞ its mechanical position of maximal openness respectively supination.
In this way the actual movement velocity can be formulated as function v_{OUT} : $\mathcal{V}_{\text{V}} \times \mathcal{H} \longrightarrow \mathcal{V}_{\text{V}}$:

$$(v(s(\cdot)), h) \mapsto \begin{cases} v(s(\cdot)) & \text{if } h \in (H_0, H_\infty) \vee (h = H_0 \wedge v > 0) \vee (h = H_\infty \wedge v < 0) \\ 0 & \text{else} \end{cases}$$

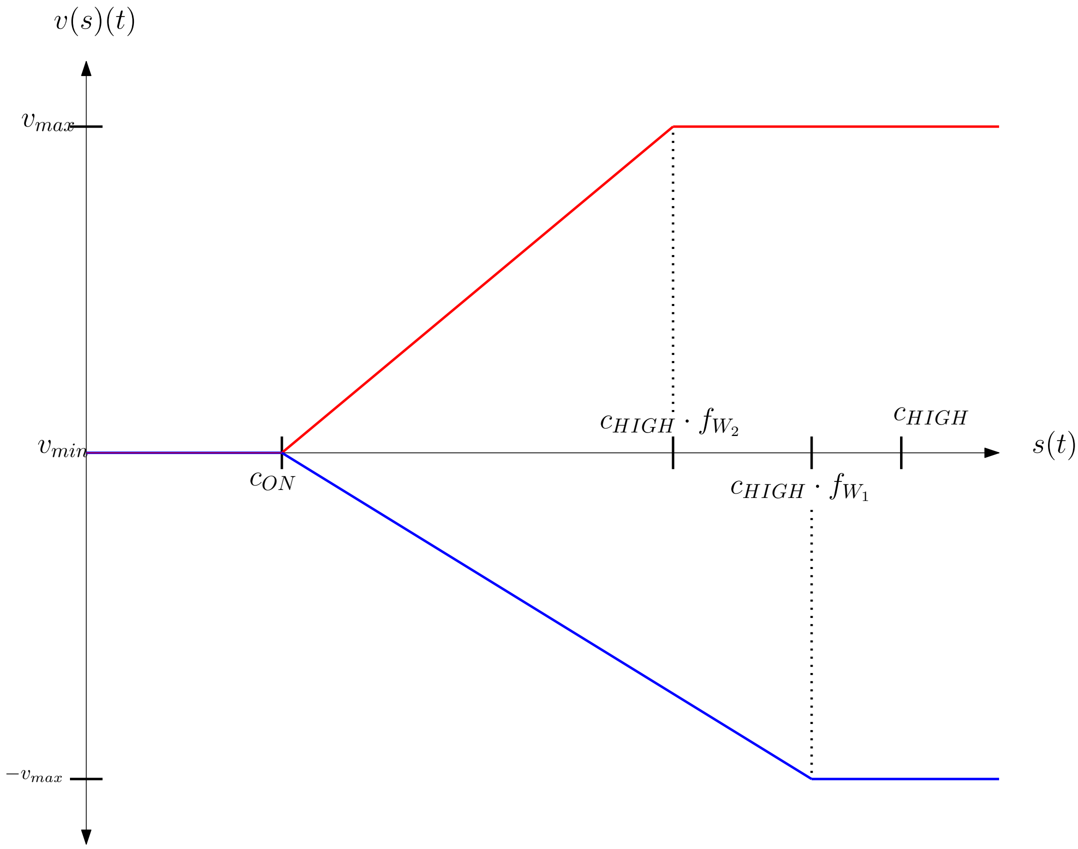

FIGURE 4.13: Velocity function

4.4.3 Programs

Now that the prosthesis compilation of this project, i.e. the *Michelangelo* hand with adapted rotational joint, and all its functionality components and features are described in terms of mathematical sets and functions, there are all tools collected in order to mathematically define what a prosthesis adjustment actually is.

A specific prosthesis adjustment is called **Program** and can be interpreted as a tuple p in the set of all Programs $\mathcal{P}$ containing a specified Control Setting cs[2], a specified Switch $SW \in \mathcal{SW}$ and a vector $\vec{x}$ containing specified values for each settable parameter related to Program p (Table 4.3)

$$p = (cs, SW, \vec{x})$$

Figure 4.14 demonstrates a possible arrangement of a Program. an Impulse with signals of channel 1 provides access from opposition grasp to opposition grasp and vice versa. Being in any of the two Grasp modes, a Switch to rotation mode is triggered by 4-Channel switch method. Movement Transmission is generated by the Control Setting MultiGrip, so in grasp modes the hand can by closed by channel 1 and opened by channel 2, in rotation mode it can be pronated by channel 1 and supinated by channel 2. Velocity behaves linearly within the Proportional Workspace 4.4.2.3.

[2]As mentioned in Section 4.4.2.2, in this project cs is chosen – as it is in the majority of prosthesis adjustment cases – as Control Setting MultiGrip.

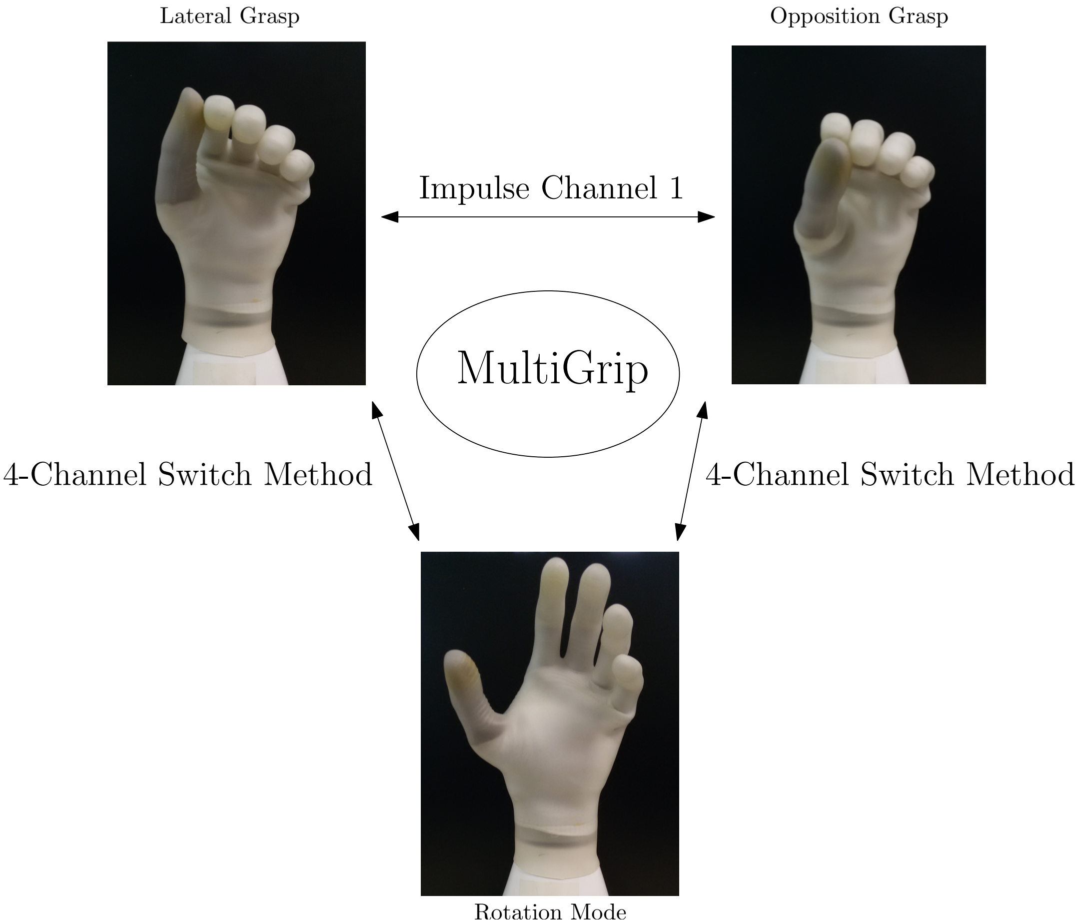

FIGURE 4.14: Possible program definition – Cocontraction to switch between grasps, 4-Channel switch method to switch to rotation mode.

The aim of this thesis is now to design an algorithm, which determines all possible and sensible existing Programs for an arbitrary user by using the information of his or her capability of generating myo-signals. With the help of this algorithm, a list of reasonable ways of prosthesis adjustment can be provided to the orthopedic technician and can either serve as final setup or as foundation for vernier adjustment.

Chapter 5

Optimization Problem – Formulation

Chapter 4 provides all terms and tools in order to be able to continue thinking of a way of designing an algorithm, solving the problem of prosthesis adjustment and thus providing a remarkable progress in myo-prosthesis usage.

At the beginning of this Chapter (Section 5.1), explanations of the basic idea will lead to the understanding, why given issue can be interpreted as a task of optimization and how to receive relevant data and calculate required values in order to be able to formulate an **Optimization Problem**, i.e. a suitable **Objective Function** and a proper **List of Constraints** (Section 5.2) which describes given problem in a mathematically solvable way.
Before formulating the actual components of this thesis' optimization problem in Section 5.3.1, a brief overview of the main concept of optimization problem formulation is given in Section 3 [38] [24].

5.1 Basic Idea

The settable parameters (Table 4.3) are the crucial factor having influence on what is recognized as valid myo-signal pattern for prosthesis control. By asking the user to generate all relevant myo-signal patterns, analyzing their shape and calculating values for all related settable parameters directly from these provided sample-patterns, all settable parameters can be calculated easily.
The obstacle to overcome is on the one hand, that each settable parameter has a bounded range of values it is allowed to be set to, so values calculated directly from the user's myo-signal sample patterns are not guaranteed to be accepted by the internal control unit.
On the other hand, certain settable parameters influence each other and therefore potentially lead to recognition errors in myo-signal pattern recognition within the internal control unit.

First mentioned problem leads to the idea of giving the amplify factor $\vec{f}_A$ a distinctive role within the set of settable parameters. By finding an amplify factor, which manipulates the user's sample myo-signal patterns in a way, such that every amplified myo-signal provides valid values for all related settable parameters, the values can directly be taken from the parameter-calculating functions and written to the internal control unit.
The problem of mutual influence of settable parameters together with latter hurdle

lead to the idea of formulating a set of constraints, which define allowed mutual relations between the settable parameters and further required properties, preventing them from colliding or operating in a destructive way and which need to be held by all calculated values for settable parameters contained in a Program. In this way, an optimization problem, in particular a combination of a **Linear (Mixed-Integer) Program** (Section 3.2) and a **Combinatorial Optimization Problem** (Section 3.3), can be formulated, which, when expressed properly, can be solved by a variety of already existing general purpose solvers.

Section 5.2 shows how to calculate values for each settable parameter, starting by initially collecting relevant user data (Section 5.2.1), subsequently introducing all functions which calculate values for settable parameters from such sample myo-signal patterns (Section 5.2.2). Excluded from this introduction of parameter-calculating functions is the amplify factor $\vec{f}_A$, which is, due to its distinctive role, being dealt with in Section 5.2.3 separately.

After all parameter-calculating functions are defined, they are used in Section 5.3.1 to set up the already mentioned list of constraints, controlling the behavior of values of the settable parameters. A suitable target function, maximized subject to latter list of constraints, finally leads to a suitable amplify factor and thus to values for required settable parameters, also providing information about suitability of a Program for the certain user, which was defined as the aim of this thesis.

5.2 Determination of Settable Parameters

Since a list of Programs is individually determined for each user and the choice of switch methods and parameter values exclusively depend on his or her individual myo-signals, it is essential to collect enough information about the user's capability of generating myo-signal patterns required for prosthesis control. These collected information can then be used as input for certain parameter-calculating functions, providing values for each settable parameter.

5.2.1 Data Collection

Within the process of data recording, the user is asked to generate all myo-signal patterns and other movements involved in myo-prosthesis usage and control. Surface electrodes measure the user's sample myo-signals and gathered information is stored for further processing. Table 5.1 lists all relevant myo-signal patterns and movements the user is asked to generate, names their functionality and aims and gives ideas about their properties.

Pattern Name	Abbr.	Function	Used for	Generated by	Remarks
No-Movement-signal	NM_i	Calculation of amplify factor	Slight movements of the arm as when walking or gesticulating (Figure 1.15)	Unconsciously generated when user makes slight movements. Both components of measured myo-signal Must not exceed specified tolerance threshold in order not to lead to unintended prosthesis activity.	-
Strong signal	S_i	Movement Transmission	-	Contracting observed muscles with high strength continuously for a certain amount of time (Figure 1.16)	Should reach upper bound of Proportional Workspace easily in order to achieve good control of variety of speed.
Strong-Side signal	SS_i	-	Calculation of amplify factor	Unconsciously generated myo-signal of a muscle when providing Strong signal with its antagonist	-
Fast-Hand-Movement signal	FM_i	Movement Transmission	Calculation of 4-Channel border	Fast increasing, strong contraction of observed muscle	Important to consider for 4-Channel border calculation, if latter is contained in a Program, since rise of 4-Channel switch method and fast hand movements have similar myo-signal patterns and therefore potentially lead to misrecognition.

Fast-Hand-Movement-Side signal	FMS_i	-	Calculation of amplify factor	Unconsciously generated myo-signal of a muscle when providing Fast-Hand-Movement signal with its antagonist	Must not exceed threshold of recognition c_{ON} earlier than actually intended component of myo-signal in order to make intended movement possible
Short Co-contraction signal	CS	switch method	Calculation of Cocontraction border (and Impulse border)	Impulsive, strong contraction, fast release of both observed muscles synchronously (4.7)	Often associated with a quick clench and release of the (phantom) fist or a quick spread and relax of the (phantom) fingers.
Long Cocontraction	CL	switch method	Calculation of Cocontraction border	Impulsive, strong contraction of both observed muscles synchronously, holding as long as possible (Figure 4.8)	-
Impulse signal	I_i	switch method	Calculation of Impulse border	Impulsive, strong contraction, fast release of one observed muscle separately (Figure 4.9, Figure 4.10)	Often associated with impulsive, strong extension respectively flexion and fast release of the (phantom) wrist

Impulse-Side signal	IS_i	-	Calculation of Impulse border (and Cocontraction border)	Unconsciously generated myo-signal of a muscle when providing Impulse signal with its antagonist	Can cause problems, when generated myo-signal pattern of provided Impulse signal holds constraints of Cocontraction switch method (Figure 4.7). Therefore also considered within calculation of Cocontraction border.
4-Channel signal	$4C_i$	switch method	Calculation of 4-Channel border	Impulsive, strong contraction of one observed muscle separately, holding as long as possible	Often associated with strong extension respectively flexion and holding of (phantom) wrist.
4-Channel-Side signal	$4CS_i$	-	-	Unconsciously generated myo-signal of a muscle when providing 4-Channel signal with its antagonist	-

TABLE 5.1: Required user data for parameter calculations

Remark: In practice, it is obviously sensible to not only use one single myo-signal sample per pattern, but gather a wide range of information by recording many tries of the user performing each movement. This information can then be compressed and a significant representative for each type of myo-signal pattern can be determined.

5.2.2 Parameter Calculation

The set of settable parameters 4.3 consists of certain threshold values, the so-called Switch-Method **borders**, defining bounds of value ranges or lengths of myo-signal pattern's duration and of multiplying factors influencing value ranges or myo-signals. By analyzing the sample myo-signal patterns the user is asked to generate, values for settable parameters can be discerned such that provided sample patterns would be recognized appropriately, if these calculated values were adjusted within the Program. In the following, functions, calculating values by using the user's information of sample myo-signal patterns, are introduced for each settable parameter.

5.2.2.1 Impulse-/4-Channel Border x_{I_i}

As its name indicates, the Impulse-/4-Channel border is a threshold used in both Impulse and 4-Channel switch method definition (Equation 4.7, Equation 4.8). It stipulates the value, a myo-signal has to exceed within a hard-coded amount of time c_{T} after previously exceeding the hard-coded threshold value of recognition, c_{ON}. In this way, this settable parameter forms the only constraint influencing the 4-Channel switch method, while it makes up one of the two constraints defining an Impulse.

In Programs, where both Impulse and 4-Channel switch method are part of the Switch, it is necessary to consider both sample myo-signal patterns for parameter calculation. If however only one of the switch methods is contained in the current Program, considering information of the non-involved sample pattern would lead to unnecessary additional restrictions to the value and should consequently be ignored. Therefore, three different Impulse-/4-Channel border calculating functions are introduced in the following, in later constraint formulation, the appropriate one for each constraint must be chosen.

By determining the value of the user's sample myo-signal pattern of a 4-Channel switch method respectively an Impulse at the point in time, in which the period of time c_{T} has passed after the myo-signal firstly has exceeded c_{ON}, a value for Impulse-/4-Channel border can be calculated, such that the user's provided sample myo-signal pattern would indeed be recognized properly when using this calculated value as Impulse-/4-Channel border.

Remark: An Impulse-/4-Channel border is calculated separately for each electrode, thus there are two settable parameters x_{I_1} and x_{I_2} providing two different switch methods as access to functionality within prosthesis control.

Case 1 – Considering Impulse:

$$\mathbf{X}_{\mathrm{I}} : S_{\mathrm{I}_i} \to \mathbb{R}_0^+$$
$$s_{\mathrm{I}} \mapsto \begin{cases} \max\limits_{t \in [t_{s_{\mathrm{I}_0}}, t_{s_{\mathrm{I}_0}} + c_{\mathrm{T}}]} s_{\mathrm{I}}(t) \cdot \mathrm{TOL}_{\mathrm{I}}^{\%} & \text{if } t_{s_{\mathrm{I}_0}} \neq -\infty \\ 0 & \text{else} \end{cases}$$

where $\mathrm{TOL}_{\mathrm{I}}^{\%} \in (0, 1]$ is a factor, which eases, when set to a value ≤ 1, the strict choice of the maximum value of the myo-signal within the function formulation. In this way, Impulse stays a reliably recognized switch method, even when a user is

getting tired during prosthesis use and myo-signal strength decreases.
It can be determined by experience and set to a value, which turns out to be comfortable for a majority of users.

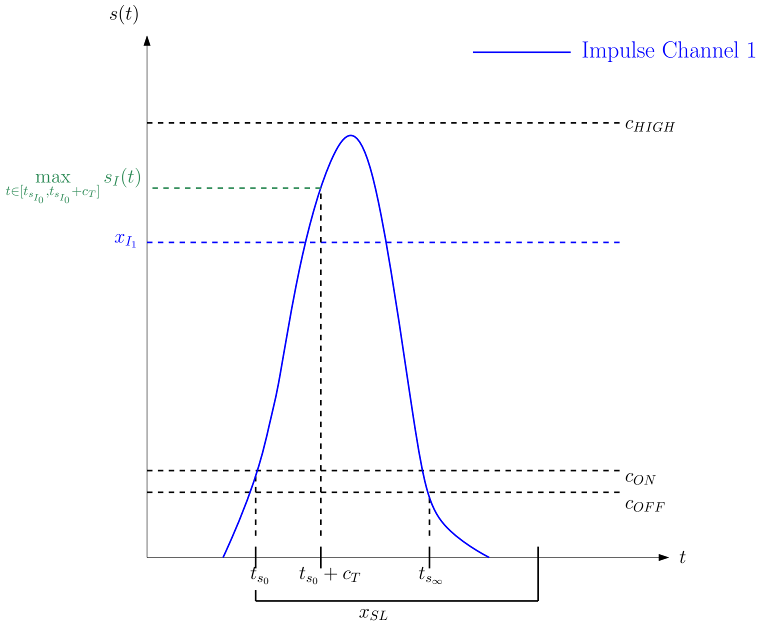

FIGURE 5.1: Calculation of Impulse-/4-Channel border for channel 1 (neglection of 4-Channel switch method)

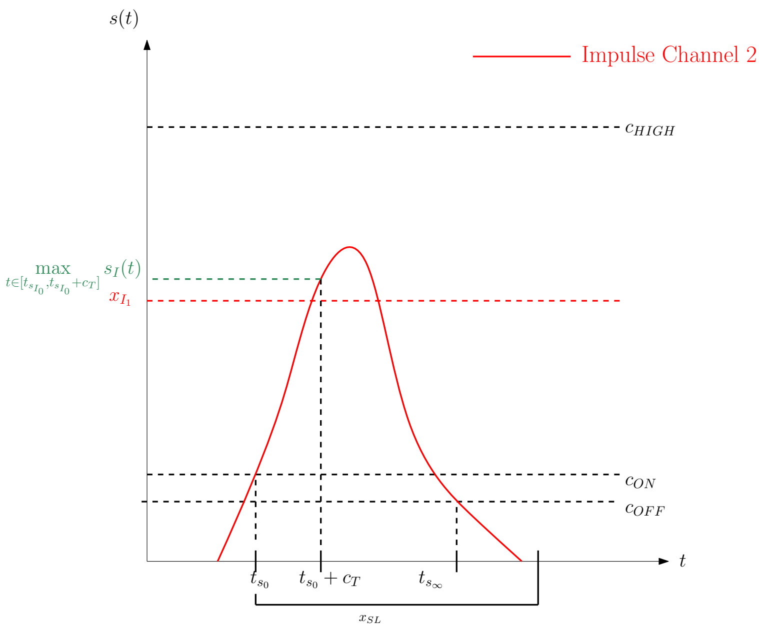

FIGURE 5.2: Calculation of Impulse-/4-Channel border for channel 2 (neglection of 4-Channel switch method)

Case 2 – Considering 4-Channel: In case 4-Channel switch method is part of the Program, it has turned out to be sensible to also consider the myo-signal of intended fast movement. This is due to the fact that myo-signals of intended fast movement and myo-signals of an intended 4-channel switch method have very similar shape and therefore potentially lead to misinterpretation by the internal control unit, if the user is not able to generate sufficiently big differences in myo-signal rise for the two different patterns. Therefore, also the value of the myo-signal of fast hand movement at this latter explained certain point in time is evaluated and leads to following Impulse-/4-Channel border calculating function:

$$\vec{\mathbf{X}}_{\mathrm{I}} : S_{4\mathrm{C}_i} \times S_{\mathrm{FM}_i} \to \mathbb{R}_0^+$$
$$(s_{4\mathrm{C}}, s_{\mathrm{FM}}) \mapsto \begin{cases} \dfrac{\max\limits_{t\in[t_{s_{4\mathrm{C}_0}}, t_{s_{4\mathrm{C}_0}}+c_{\mathrm{T}}]} s_{4\mathrm{C}}(t) + \max\limits_{t\in[t_{s_{\mathrm{FM}_0}}, t_{s_{\mathrm{FM}_0}}+c_{\mathrm{T}}]} s_{\mathrm{FM}}(t)}{2}, & \text{if } \max\limits_{-''-} s_{4\mathrm{C}}(t) > \max\limits_{-''-} s_{\mathrm{FM}}(t) \\ 0, & \text{else} \end{cases}$$

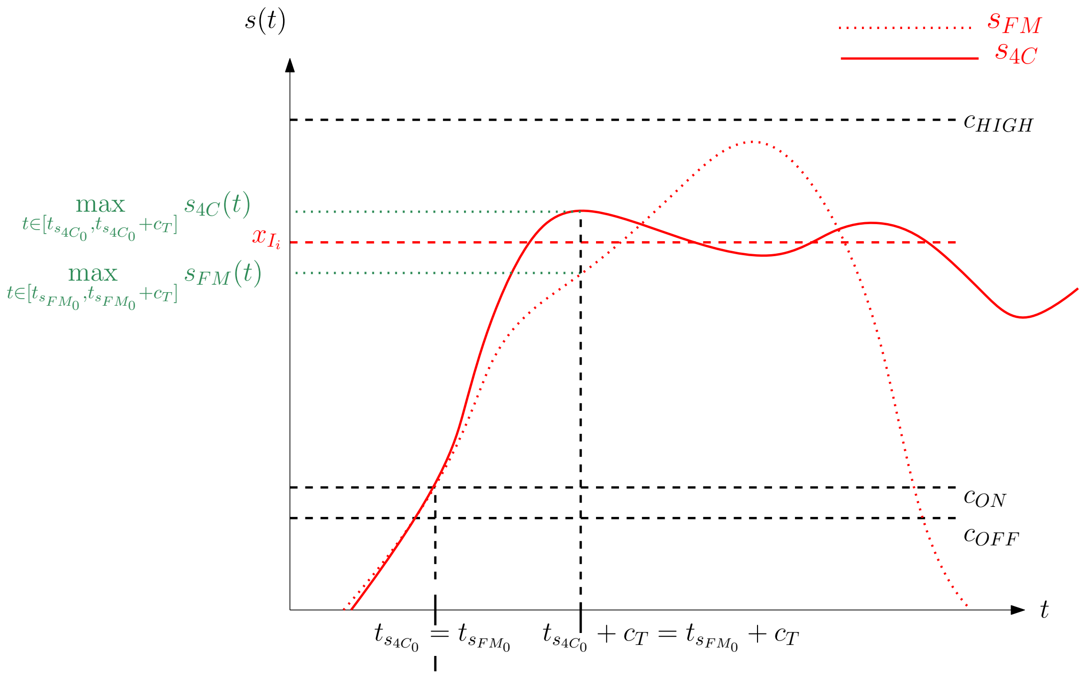

FIGURE 5.3: Calculation of Impulse-/4-Channel border (neglection of Impulse information)

Remark: In this function's definition, the distinction of cases plays a remarkable role in the quality of the main algorithm. This can be understood by considering the situation, where the user is not able to generate a 4-Channel pattern which rises faster than his or her usual fast hand movement myo-signal, i.e.:

$$\max_{t\in[t_{s_{4\mathrm{C}_0}}, t_{s_{4\mathrm{C}_0}}+c_{\mathrm{T}}]} s_{4\mathrm{C}}(t) < \max_{t\in[t_{s_{\mathrm{FM}_0}}, t_{s_{\mathrm{FM}_0}}+c_{\mathrm{T}}]} s_{\mathrm{FM}}(t).$$

In this case, a user would have troubles using a Program containing the 4-Channel switch method, because a Switch would wrongly be triggered whenever the user actually tries to generate a fast hand movement. Exactly this case is caught by previously defined function and labeled by the value 0. In this way, the algorithm is not only able to calculate an Impulse-/4-Channel border from appropriate myo-signals, but can also tell whether or not the myo-signals *are* appropriate.

Case 3 – Considering Impulse and 4-Channel: In case both Impulse and 4-Channel switch method are part of a Program, the Impulse-/4-Channel border is calculated by both parameter-calculating functions $\mathbf{X}_{\mathrm{I}}$ and $\vec{\mathbf{X}}_{\mathrm{I}}$. A good final choice for Impulse-/4-Channel border has turned out to simply be the minimum of both calculated values, since for this particular choice it is made sure that even the weaker myo-signal is capable of being recognized.

5.2.2.2 Cocontraction Border x_{C_i}

The Cocontraction borders are threshold values – one for each myo-signal component – stipulating the value, each myo-signal component has to exceed within the hard-coded amount of time c_{T} after previously exceeding the hard-coded value of recognition, c_{ON}, so they play the exact analogous role as Impulse-/4-Channel border does in the case of Impulse respectively 4-Channel switch method.

The parameter-calculating function $\vec{\mathbf{X}}_{\mathrm{C}}$ excepting a myo-signal $\vec{s}_{\mathrm{CS}} = (s_{\mathrm{CS}_1}, s_{\mathrm{CS}_2})$ and returning its resulting Cocontraction borders as a vector $\mathbf{x}_{\mathrm{C}} = (x_{\mathrm{C}_1}, x_{\mathrm{C}_2})$ is defined by

$$\vec{\mathbf{X}}_{\mathrm{C}} : S_{\mathrm{CS}} \to \mathbb{R}^+ \times \mathbb{R}^+$$
$$\vec{s}_{\mathrm{CS}} \mapsto \begin{cases} \left(\max\limits_{t \in [t^{\mathrm{sync}}_{\vec{s}\mathrm{CS}_0}, t^{\mathrm{sync}}_{\vec{s}\mathrm{CS}_0} + c_{\mathrm{T}}]} s_{\mathrm{CS}_i}(t)\right)_{i=1,2} \cdot \mathrm{TOL}^{\%}_{\mathrm{C}}, & \text{if } t^{\mathrm{sync}}_{\vec{s}\mathrm{CS}_0} \neq -\infty \\ 0 & \text{else} \end{cases}$$

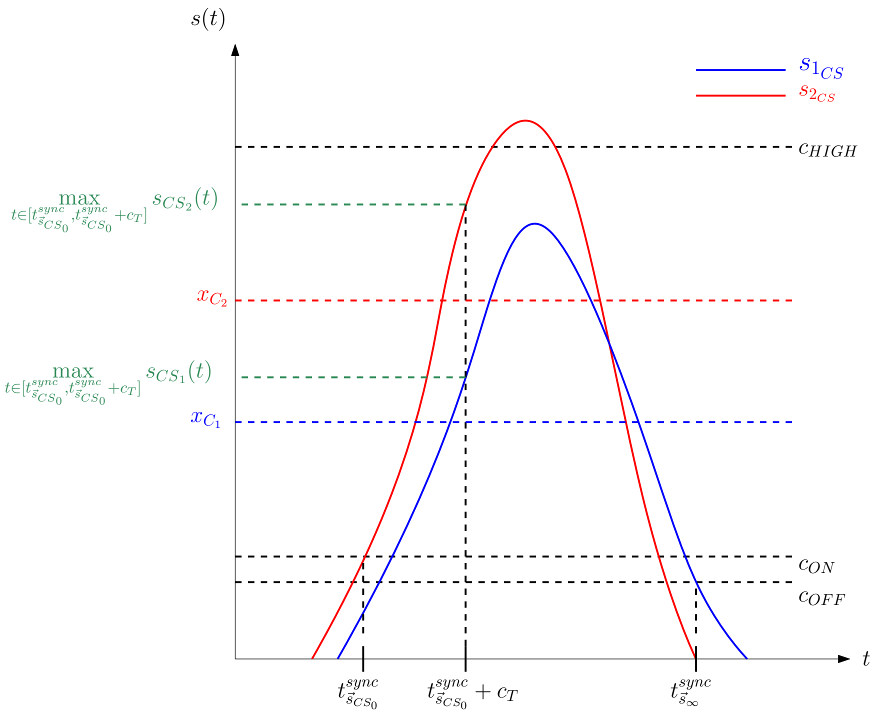

FIGURE 5.4: Cocontraction border calculation

where $\mathrm{TOL}_{\mathrm{C}}^{\%}$ is a tolerance factor leading to same results as $\mathrm{TOL}_{\mathrm{I}}^{\%}$ in Section 5.2.2.1.

5.2.2.3 Signal Length x_{SL}

The signal length, as well as the Impulse-/4Channel border, is another settable parameter which is contained in definitions of two different switch methods: Impulse and Short Cocontraction. It is the threshold value, which stipulates the maximal tolerated amount of time, an Impulse signal respectively a Short Cocontraction signal is excepted to assign values $\geq c_{\mathrm{OFF}}$ after previously exceeding the hard-coded threshold value of recognition, c_{ON}.

Impulse is generated by one single component of the myo-signal, while Cocontraction's myo-signal pattern consists of both components synchronously. Therefore, analogous to the situation of calculating Impulse-/4-Channel border (Section 5.2.2.1), three different functions are defined, depending on if either Impulse, Cocontraction, or both of these two switch methods are contained in the Program.

By measuring the amount of time, a user's sample Impulse respectively Cocontraction needs to deceed the hard-coded threshold value c_{OFF} after previously exceeding c_{ON}, a value for signal length can be calculated, such that the user's provided sample myo-signal pattern would indeed be recognized properly when using this calculated value as signal length.

Case 1 – Considering Impulse:

$$\mathbf{X}_{\mathrm{SL}} : S_{\mathrm{I}_i} \to \mathbb{R}^+$$
$$s_{\mathrm{I}} \mapsto \begin{cases} \left| t_{s_{\mathrm{I}_0}} - t_{s_{\mathrm{I}_\infty}} \right|, & \text{if } t_{s_{\mathrm{I}_0}} \neq -\infty \\ 0 \text{ else} \end{cases}$$

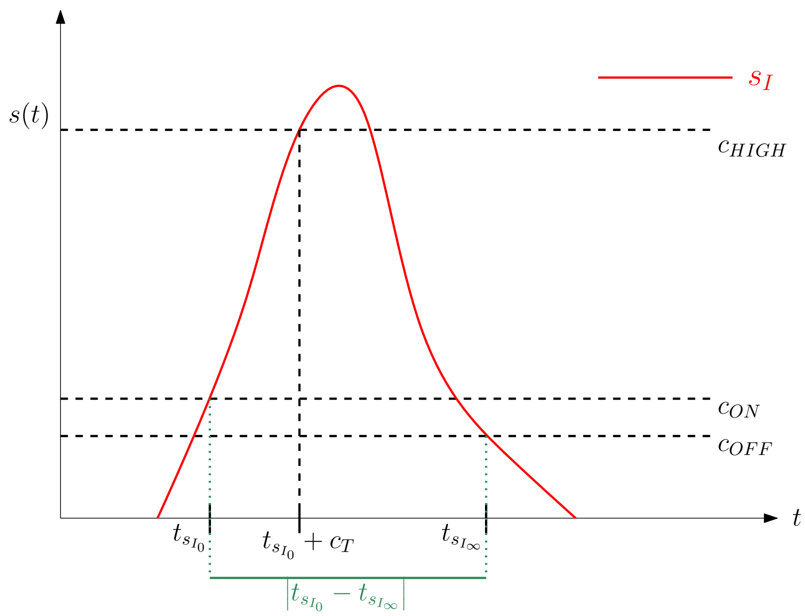

FIGURE 5.5: Calculation of signal length (considering single component only)

Case 2 – Considering Cocontraction:

$$\vec{\mathbf{X}}_{\text{SL}} : S_{\text{CS}} \to \mathbb{R}^{+}$$
$$S_{\text{CS}} \mapsto \left\{ \left| t^{\text{sync}}_{s_{\text{CS}_0}} - t^{\text{sync}}_{s_{\text{CS}_\infty}} \right|, \text{if } t^{\text{sync}}_{s_{\text{CS}_0}} \neq -\infty \right.$$

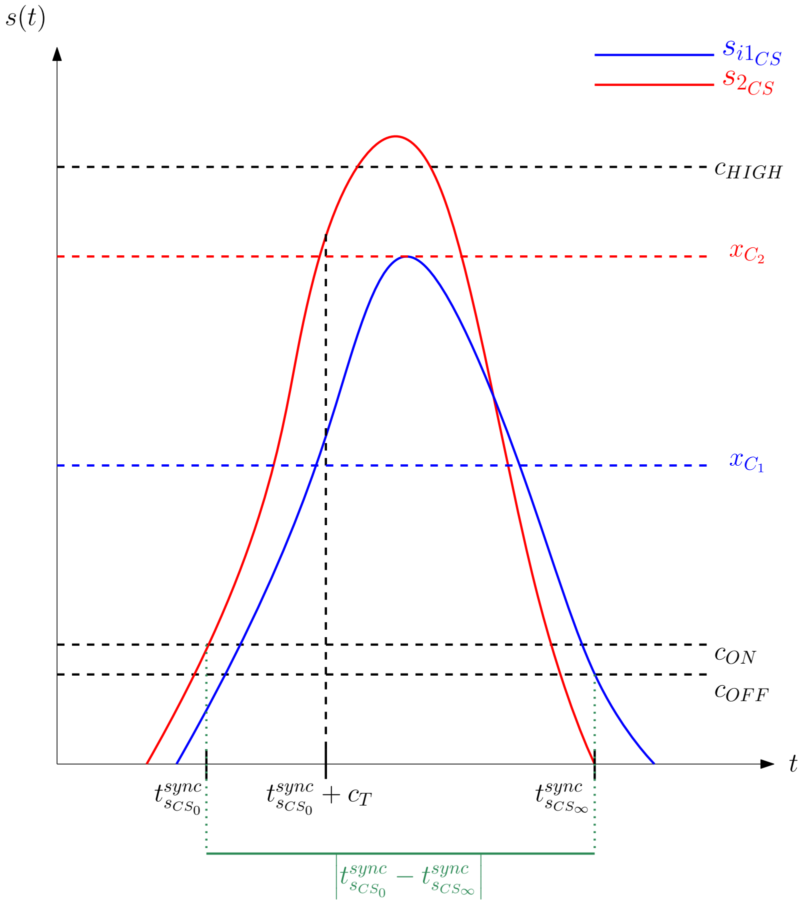

FIGURE 5.6: Calculation of signal length (considering both components)

Case 3 – Considering Impulse and Cocontraction: If both Impulse and Cocontraction are contained in a Program, values with both functions, $\mathbf{X}_{\text{SL}}$ and $\vec{\mathbf{X}}_{\text{SL}}$, are calculated and the signal length is set to the maximum value of these two function outputs in order to make sure, that indeed both switch methods can be recognized by the internal control unit.

5.2.2.4 Long Hand Open Length x_{LHOL}

As explained in Section 4.4.2.1 and shown in Figure 4.11, the Long Hand Open Length is a threshold value, stipulating the amount of time the myo-signal's second component has to keep assigning values within a certain ratio of the Proportional Workspace after having reached the prosthesis' mechanical position of maximum displacement H_∞. So in order to find an appropriate value to set this parameter to, the user's Long Hand-Open signal needs to be analyzed, observing how long he is capable of keeping up the myo-signal to required level. In case the user is limited at fulfilling this task, the Length has to be set to a sufficiently small value, analogously to settable parameters such as Impulse-/4-Channel border or Cocontraction border. If the user does not have any troubles keeping up his or her myo-signal to required level for a long amount of time, the Length has to be set to a reasonably small value, such that the user does not have to wait uncomfortably long until the Switch is triggered.

5.2.2.5 Workspace Factor $\vec{f}_W$

The Workspace factor defines the upper bound of the range of myo-signal values, in which movement velocity is transmitted to the prosthesis proportional to provided myo-signal assigning values within this range (Figure 4.13). It is sensible to set upper bound of latter to a value, the user can reach easily by his or her Strong signal, since in this case he is very likely to be able to have full access on speed control.
By analyzing the user's Strong signal, he was asked to generate during the process of data recording, the strength of myo-signal can be judged by calculating a meaningful statistic value from given myo-signal values and derive a value for the Workspace factor from this value's provided information [1].

Remark: The Proportional Workspace is separately adjustable for the two myo-signal components, so the Workspace factor consists of two parameters, settable individually for each electrode channel.

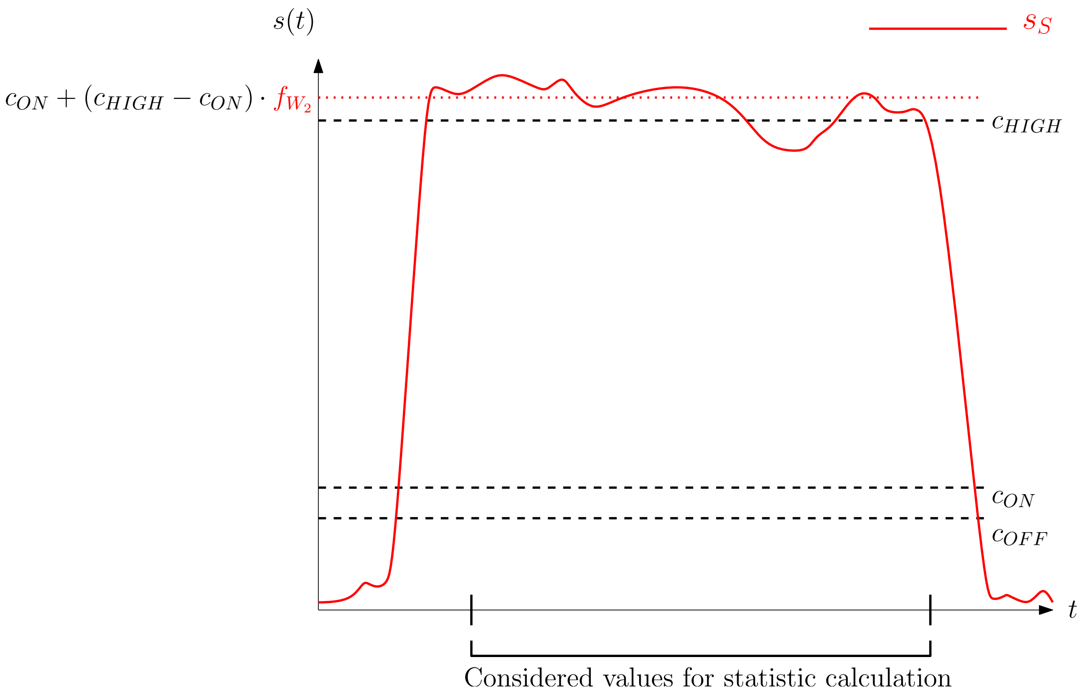

FIGURE 5.7: Result for calculated Workspace factor – sufficiently strong myo-signal, upper bound of Proportional Workspace can be set to c_{HIGH}.

[1] The solution approach of this thesis considers the 95-Percentile as statistic value for parameter calculation, i.e. the 95-Percentile of the user's Strong signal values is determined and set as upper bound of the Proportional Workspace. From these new bounds and the default range for the Proportional Workspace, $[c_{\mathrm{ON}}, c_{\mathrm{HIGH}}]$, the Workspace factor can be calculated easily.

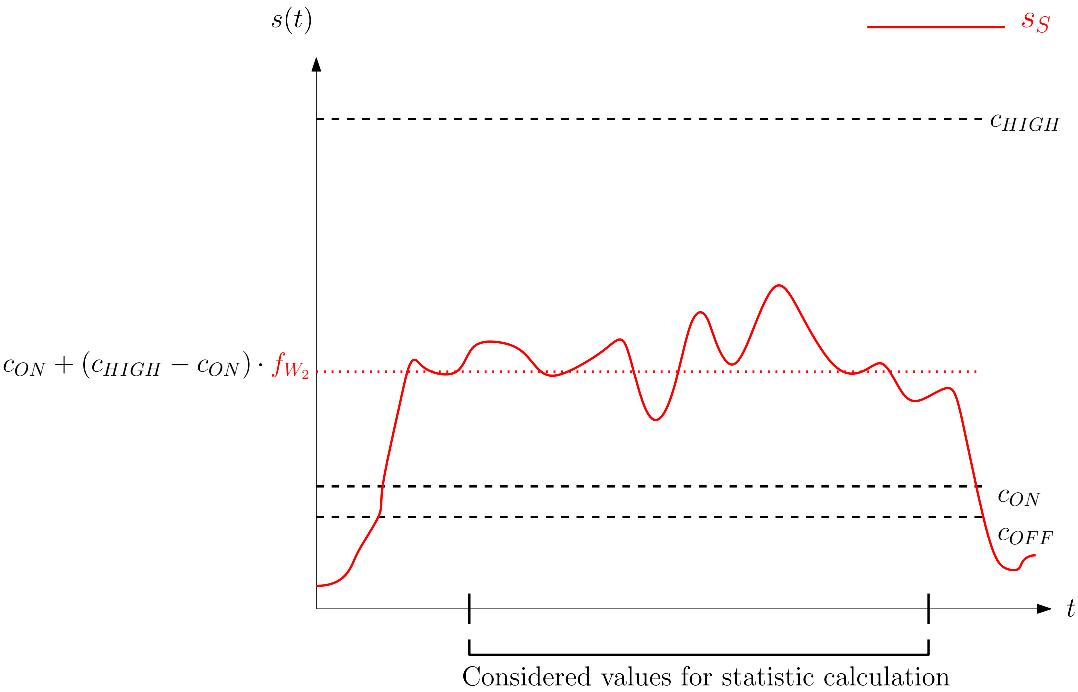

FIGURE 5.8: Result for calculated Workspace factor – sufficiently strong myo-signal, upper bound can be calculated from statistic value within range.

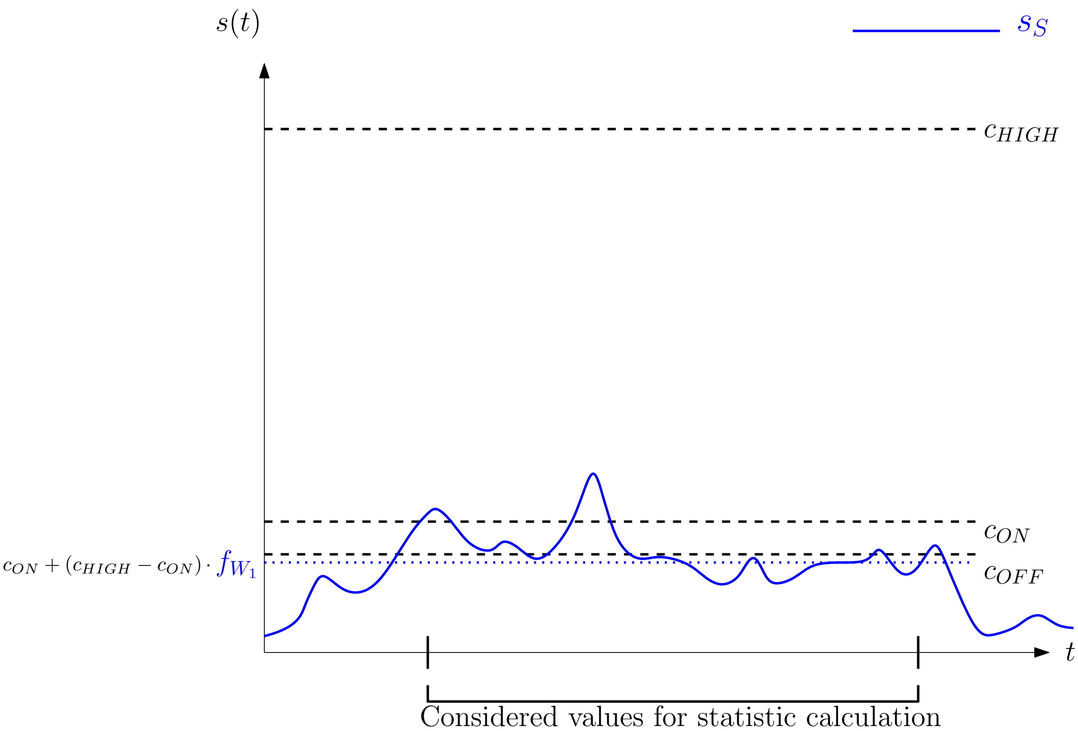

FIGURE 5.9: Result for calculated Workspace factor – insufficient muscle strength, no valid Workspace factor without adequate amplification.

5.2.3 Amplify-Factor $\vec{f}_A$

In Section 5.2.2, functions were introduced, which calculate values for settable parameters (Table 4.3) from sample myo-signal patterns the user was asked to generate

in advance. So for these specifically calculated values – since the value is indeed calculated from the pattern itself – the user's contextless myo-signal pattern would obviously be recognized correctly. In practice however, provided sample myo-signals do not automatically lead to function outputs within the valid ranges for the internal control unit, or potentially interact with each other in an undesired way, leading to erratic behavior in myo-prosthesis control. However, the one settable parameter, which is capable of influencing the myo-signal itself, is the amplify factor. It is able to intensify or weaken a myo-signal component-wise by fixed, but individually settable values and thus can manipulate the user's myo-signal patterns in a way, such that their amplified versions might indeed lead to valid values when used as parameter-calculating function inputs.
Therefore it is sensible to give the amplify factor a distinctive role within the calculation of settable parameters by setting the sights on finding an amplify factor, which manipulates provided myo-signals in a way, such that a maximum of prosthesis' functionality components with a maximum of comfort is accessible.

5.2.3.1 Basic Idea

This leads to the idea of compiling a list of constraints, defining desired behavior of the user's sample myo-signal patterns and related values for settable parameters towards their valid ranges and towards each other and involving the amplify factor (Section 4.2) in a target function, requesting it to manipulate the user's sample myo-signal patterns in a way, such that the list of constraints is held, thus making given problem an optimization problem. In the following, before mathematically precise formulations are given in Section 5.3 the basic idea of this thesis' approach is sketched.

Programs: In myoelectric prosthesis adjustment, there is only a small number of reasonable possibilities to define Switches and Control Settings, i.e. Programs.
Therefore it is sensible to break up given problem of prosthesis adjustment, considering only one Program at a time. For each separate Program, the algorithm should then determine whether or not an appropriate amplify factor exists, such that this certain Program is executable for the user's amplified myo-signals. In this way, not only the complexity of the problem shrinks many times over, but also unnecessary considerations of parameter values, which are redundant for a specific Program and potentially lead to a mistaken disqualification of latter, are avoided.

Constraints: Since a user's myo-signal can be interpreted as an element of a set of functions providing real numbers as output for every $t \in \mathcal{T} \subset \mathbb{R}$ (Section 4.1), since all outputs of the parameter-calculating functions are values for settable parameters, which themselves are elements of $\mathbb{R}$ too (Section 5.2.2), and since given valid ranges for these values are nothing but subsets of real numbers or ordinary intervals (Table 4.3), all these elements can be interrelated and their desired behavior and interaction can be formulated as mathematical constraints.

Combined with the idea of breaking up the problem in separate Programs, a compilation of **Hard Constraints** for each separate Program can be established, which obligatory need to be fulfilled by the user's amplified myo-signal patterns in order to guarantee, that the Program is executable. Additionally, further restrictions to the user's myo-signal patterns can be postulated, which are admittedly unnecessary concerning executability, but would indeed enhance the comfort of prosthesis control. These **Soft Constraints**, when involved properly in the formulation, potentially lead to even better results while at the same time they do not disturb the actual process of problem solution.

Let therefore $p := (cs, SW, \vec{x}) \in \mathcal{P}$ be a Program, S_Q the set of myo-signals of signal pattern Q and $s_Q^U \in \mathcal{S} \cup \mathcal{S}^2$ the representant of user U's provided sample pattern[2] of signal pattern Q. Furthermore, let $\{S_{SW_k} : k = 1, \ldots, |SW|\} \subseteq \mathcal{SW}$ be the set of all switch methods contained in SW, thus $s_{SW_k}^U$ a representant of user U's recorded sample pattern of switch method SW_k for each $k = 1, \ldots, |SW|$.
Then, the list of Hard Constraints for Program p can be sketched in the following way:

$$\vec{\mathbf{F}}_{\mathcal{A}}(s_{SW_k}^U) \in S_{SW_k},\ k = 1, \ldots, |SW| \tag{5.1}$$

$$\vec{\mathbf{F}}_{\mathcal{A}}(s_{SW_k}^U) \notin S_{SW_l},\ k \neq l \tag{5.2}$$

$$\mathbf{F}_{\mathcal{A}_i}(s_{S_i}^U) \in [\text{TOL}],\ i = 1, 2 \tag{5.3}$$

$$\mathbf{F}_{\mathcal{A}_i}(s_{NM_i}^U) \leq \text{TOL}_{NM_i},\ i = 1, 2 \tag{5.4}$$

where [TOL] is an interval, defining the tolerated range of the user's Strong signal, which has to be set reasonably in advance in order to make sure that on the one hand, the user's myo-signal is strong enough to have access to prosthesis control per se, and on the other hand does not have too strong myo-signals, to also have good access on velocity control, and where $\text{TOL}_{NM_i} \in \mathbb{R}^+$ is a threshold value, which has to be set reasonably in advance, such that too noisy or strong No Movement Signals of the user are recognized and an erratic trigger of any functionality of the prosthesis can be avoided.

In words: Amplified sample myo-signal patterns must have several properties, e.g. Equation 5.1 demands from the user's amplified sample myo-signal pattern of switch method SW_k to hold all constraints in order to indeed be element of switch method SW_k, while Equation 5.2 postulates, that an amplified sample myo-signal pattern of one switch method must not hold all constraints defining another switch method, i.e. must be element of one and only one switch method.

Objective Function: In upper constraint formulation, the amplify factor is still an unknown, settable parameter. All sample patterns s_Q^U, as well as [TOL] and c_{ON} are given quantities and the sets of signals S_Q are defined by settable parameters, whose values themselves again depend on the myo-signal patterns, i.e. given s_Q^U. Also, due to Equation 5.3 and 5.4, valid values of amplified myo-signals are bound to a finite interval.
Therefore, and because the height of the amplify factor – within the valid range – correlates with optimality of prosthesis control in a linear way, the target function of desired optimization program with respect to given constraints can be sketched as the amplify factor:

$$\text{Maximize } \vec{\mathbf{F}_A} \text{ with respect to } (5.1) - (5.4)$$

[2]Note, that a sample pattern s_Q^U of switch method Q does not automatically mean $s_Q^U \in S_Q$, since s_Q^U is only a provided try – not necessarily a valid one – of the user to generate switch method Q.

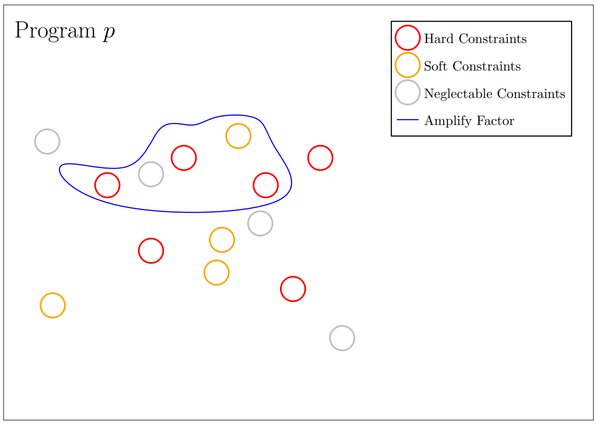

FIGURE 5.10: Inappropriate amplify factor – not all obligatory hard constraints are held, no guarantee for program executability

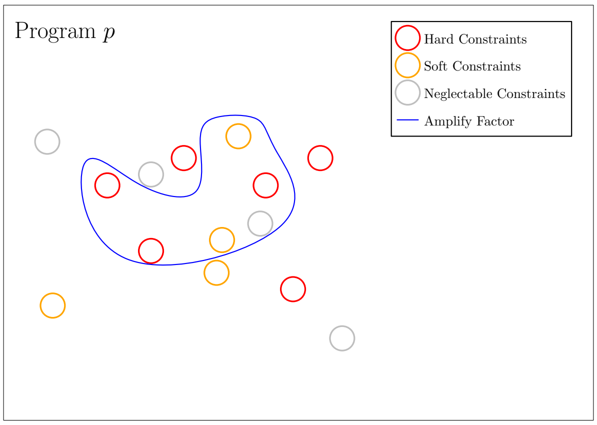

FIGURE 5.11: Inappropriate amplify factor – not all obligatory hard constraints are held, no guarantee for program executability

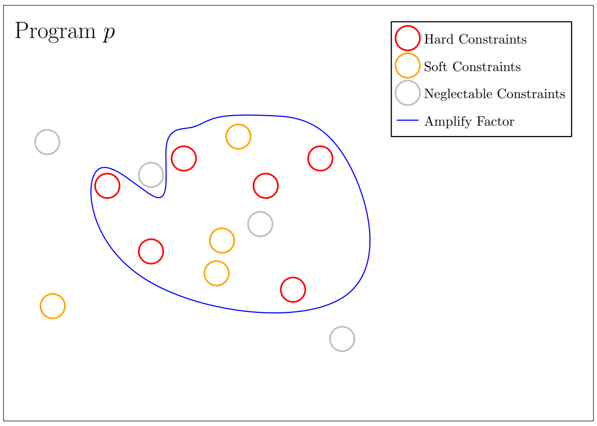

FIGURE 5.12: Appropriate amplify factor – all hard constraints plus a number of enhancing soft constraints are held

Figure 5.10 - 5.12 show the idea of finding an appropriate amplify factor with respect to obligatory Hard Constraints and unessential, but enhancing Soft Constraints. The grey-labeled redundant constraints can be ignored and should just emphasize, that not every Program contains every settable parameter in its formulation, thus, in the following approach, when a complete list of constraints, considering the whole system, is set up, subsets of Hard Constraints and Soft Constraints have to be selected for each Program, which then form this Program's specific optimization problem. Amplify factors of Figure 5.10 and 5.11 are not set to valid values, since not all of red-labeled Hard Constraints are fulfilled, which obligatory have to be held to guarantee executability to the user. In Figure 5.12, a value for the amplify factor has been found, such that all Hard Constraints, and even some of the non-obligatory Soft Constraints are held. So the user's myo-signals can be amplified by determined amplify factor in order to calculate valid values for all settable parameters which are relevant for this Program.

5.3 Prosthesis Adjustment Problem – Formulation

An anatomical idea of the upper extremity and treatments in cases of amputation has been given in Section 1, Sections 2 and 4 have introduced and modeled myoelectric prostheses, their functionality and controllability components. Furthermore, the mathematical foundation of optimization problems and their strategies of being formulated and solved have been given in Section 3. With this given basis of knowledge, the approach of bringing given problem to a form, which is feasible of being solved by available algorithmic software tools, can finally be started.
In the following, for reasons of clarity and comprehensibility, instead of listing a summary of all constraints, which arise within the range of possible programs for hand prosthesis control, one exemplary program is picked, for which the optimization program will be derived in detail. Constraints for Programs, which contain other components of control methods, can analogously be modeled.

5.3.1 Constraints and Objective Function

Considered will be the Program shown in Figure 4.14, which on the one hand uses the Control Setting *Multigrip* in order to perform general movements within the modes, and on the other hand contains two different switch methods – Impulse of channel 1, and 4-Channel switch method for both channel 1 and channel 2 – in order to switch between lateral grasp, opposition grasp and rotation mode.

$\mathrm{TOL}_{\mathrm{NM}} \in [0, c_{\mathrm{ON}}] \cup \mathcal{W}$	Maximum tolerated value of No Movement Signal NM_i
$\mathrm{TOL}_{\mathrm{S}}^{\max} \in [c_{\mathrm{ON}}, \infty)$	Weak maximum tolerated value of Strong Signal S_i
$\mathrm{TOL}_{\mathrm{S}}^{\max^*} \in [c_{\mathrm{ON}}, \infty)$	Strict maximum tolerated value of Strong Signal S_i
$\mathrm{TOL}_{\mathrm{S}}^{\min} \in [c_{\mathrm{ON}}, \infty)$	Minimum tolerated value of Strong Signal S_i
$\mathrm{TOL}_{\mathrm{I}}^{\max} \in [c_{\mathrm{ON}}, \infty)$	Maximum tolerated value of Impulse/4-Channel border x_{I_i}
$\mathrm{TOL}_{\mathrm{I}}^{\min} \in [c_{\mathrm{ON}}, \infty)$	Minimum tolerated value of Impulse/4-Channel border x_{I_i}
$\mathrm{TOL}_{\mathrm{SL}}^{\max}$	Maximum tolerated value of signal length x_{SL}

TABLE 5.2: Tolerance thresholds

This means, all constraints required to be held by the settable parameters of the just mentioned control patterns must be postulated as inequalities and thus generate an optimization program of the form 3.2, such that

$$\mathbf{F}_{\mathcal{A}_i}(\mathrm{S}_i) \in [\mathrm{TOL}_{\mathrm{S}}^{\mathrm{min}}, \mathrm{TOL}_{\mathrm{S}}^{\mathrm{max}}] \tag{5.5}$$
$$\mathbf{t}_0(\mathbf{F}_{\mathcal{A}_i}(S_i)) \leq \mathbf{t}_0(\mathbf{F}_{\mathcal{A}_j}(SS_i)) \tag{5.6}$$
$$\mathbf{F}_{\mathcal{A}_i}(\mathrm{NM}_i) \in [0, \mathrm{TOL}_{\mathrm{NM}}] \tag{5.7}$$
$$\mathbf{F}_{\mathcal{A}_i}(\mathrm{I}_i) \in S_{\mathrm{I}_i} \tag{5.8}$$
$$\mathbf{F}_{\mathcal{A}_i}(\mathrm{I}_i) \notin S_{\mathrm{SW}},\ \mathrm{SW} \neq I_i \tag{5.9}$$
$$\mathbf{F}_{\mathcal{A}_i}(4\mathrm{C}_i) \in S_{4\mathrm{C}_i} \tag{5.10}$$
$$\mathbf{F}_{\mathcal{A}_i}(4\mathrm{C}_i) \notin S_{\mathrm{SW}},\ \mathrm{SW} \neq 4\mathrm{C}_i \tag{5.11}$$

for $i, j = 1, 2,\ i \neq j$, with notations for switch methods of Table 5.1 respectively sets of switch methods of Table 4.4, for the Amplifier of Section 4.2, for tolerance thresholds of Table 5.2, and the $\leq$ operator as well as the amplifying function to be interpreted point-wise. *MultiGrip* requires the user's No-Movement myo-signal NM_i of channel i, amplified by amplify factor $\mathbf{F}_i$, to stay below a sensible threshold $\mathrm{TOL}_{\mathrm{NM}_i}$ (Equation 5.7), such that no unintended movement is generated. Furthermore, it demands the amplified Strong Signal S_i to be capable of resting in a sensible range, such that it is guaranteed to be powerful enough to exploit a maximum of velocity control, but at the same time not exceeding Proportional Workspace (Section 4.4.2.3) too easily in order to keep velocity well controllable (Equation 5.5). At the same time it has to be ensured that, when movement by a certain channel i is intended to be generated, Strong Signal S_i of channel i passes the threshold of recognition c_{ON} earlier than Strong Side Signal SS_i (Equation 5.6), since otherwise erratic hand opening instead of hand closing, pronation instead of supination, or vice versa, would be performed.
The two switch methods – Impulse and 4-Channel switch method – on the other side both require the user's amplified myo-signals to exceed the settable Impulse border x_{I_i} (Table 4.3) within the amount of time defined by counter c_{T} (Table 4.2). Additionally, a last constraint occurs from Impulse switch method, demanding from the user's amplified Impulse myo-signal to undershoot the theshold c_{OFF} within the settable signal length x_{SL}. These restrictions ensure that the current switch method is indeed being recognized (Equation 5.8) and leads to prosthesis action. Since the considered Program does not contain any other switch method, potentially colliding with Impulse and 4-Channel switch method, and these two switch methods themselves do not influence each other erratically, Equation 5.9 and 5.11 do not require further consideration in upcoming constraint formulation.

These considerations lead to a set of constraints $\mathcal{C}$ for the currently considered Program, consisting of constant values (Table 4.2), of the previously defined tolerance threshold values (Table 5.2), parameter calculating functions (Section 5.2.2) and the settable parameters (Table 4.3), with the amplify factor having the distinctive role of being the one main parameter to be capable of influencing the whole system (Section 5.2.3).

5.3.1.1 Continuous Approach

By formulating given task with the help of previous reflections, we can state the following optimization problem:

$$\max_{f_{\mathcal{A}_i}} \sum_{i=1}^{2} f_{\mathcal{A}_i} \tag{O1}$$

subject to:

$$\mathbf{F}_{\mathcal{A}_i}(\mathrm{S}_i) \geq \mathrm{TOL}_{\mathrm{S}}^{\min} \tag{C1}$$
$$\mathbf{F}_{\mathcal{A}_i}(\mathrm{S}_i) \leq \mathrm{TOL}_{\mathrm{S}}^{\max} \tag{C2}$$
$$\mathbf{t}_0(\mathbf{F}_{\mathcal{A}_i}(\mathrm{S}_i)) \leq \mathbf{t}_0(\mathbf{F}_{\mathcal{A}_j}(\mathrm{SS}_i)) \tag{C3}$$
$$\mathbf{F}_{\mathcal{A}_i}(\mathrm{NM}_i) \leq \mathrm{TOL}_{\mathrm{NM}} \tag{C4}$$
$$\mathbf{X}_{\mathrm{I}}\left(\mathbf{F}_{\mathcal{A}_i}(\mathrm{I}_i)\right) \geq \mathrm{TOL}_{\mathrm{I}}^{\min} \tag{C5}$$
$$\mathbf{X}_{\mathrm{I}}\left(\mathbf{F}_{\mathcal{A}_i}(\mathrm{I}_i)\right) \leq \mathrm{TOL}_{\mathrm{I}}^{\max} \tag{C6}$$
$$\mathbf{X}_{\mathrm{SL}}\left(\mathbf{F}_{\mathcal{A}_i}(\mathrm{I}_i)\right) \leq \mathrm{TOL}_{\mathrm{SL}}^{\max} \tag{C7}$$
$$\vec{\mathbf{X}}_{\mathrm{I}}\left(\mathbf{F}_{\mathcal{A}_i}(\mathrm{I}_i), \mathbf{F}_{\mathcal{A}_i}(\mathrm{FM}_i)\right) \geq \mathrm{TOL}_{\mathrm{I}}^{\min} \tag{C8}$$
$$\vec{\mathbf{X}}_{\mathrm{I}}\left(\mathbf{F}_{\mathcal{A}_i}(\mathrm{I}_i), \mathbf{F}_{\mathcal{A}_i}(\mathrm{FM}_i)\right) \leq \mathrm{TOL}_{\mathrm{I}}^{\max} \tag{C9}$$
$$f_{\mathcal{A}_i} \in \mathcal{V}_{\mathcal{A}} \tag{D1}$$

for $i, j = 1, 2,\ i \neq j$. Note that the decision variables to be found are amplify factors $f_{\mathcal{A}_i}$, (Section 3.1) of this optimization problem are contained in amplifying function $\mathbf{F}_{\mathcal{A}_i}$ (Section 4.2). This fact by itself would not cause troubles, since the Amplifier as a multiplicative function does not lead to any non-linearity of the constraints. The problem arises, when non-linear parameter calculating functions are necessary to be added as constraints.

Problem of Non-Commutativity: Taking a closer look at postulated inequalities C1-C9, it is getting clear that problems occur from the fact, that in general

$$\mathbf{X}(\mathbf{F}_{\mathcal{A}_i}(\mathrm{S})) \neq \mathbf{F}_{\mathcal{A}_i}(\mathbf{X}(\mathrm{S})),$$

i.e., amplifying function $\mathbf{F}_{\mathcal{A}_i}$ does not commute with a parameter calculating function $\mathbf{X}$ for a user's myo-signal S in general. This means, postulated inequalities are not capable of being formulated towards the linear form 3.2 with constraint matrix A of constant values and a decision vector containing decision variables $f_{\mathcal{A}_i}$ of this optimization problem.

5.3.1.2 Naive Enumerative Approach

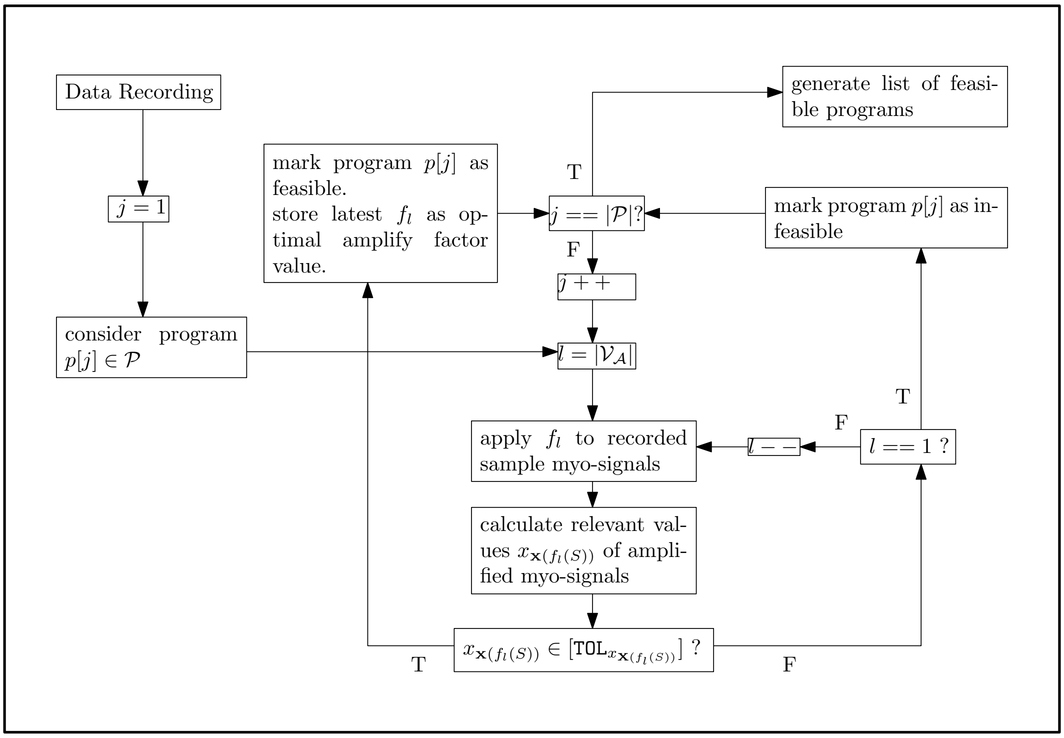

FIGURE 5.13: Naive enumerative approach – flowchart

In our real-life application, amplify factor $f_{\mathcal{A}_i}$ can only be set to a discrete, finite number of different values (Remark Section 4.2). Therefore, one unsophisticated, but however well-working and easily implementable way of overcoming the issue of non-linearity is the brute force approach of solving the maximization problem by enumeratively going through every available value for amplify factor $f_{\mathcal{A}_i}$, starting from the maximum value, gradually going down to the minimum value. By applying it to the user's myo-signals, determining required values with parameter calculating functions and checking for validity of occurring results, the algorithm finds a feasible solution by stopping at the first amplify factor to make all constraints be held, or recognizes infeasibility (Figure 5.13)[3]. In practice however, this naive enumeration is only reasonable in case of a small domain for amplify factors $f_{\mathcal{A}_i}$.

5.3.1.3 Linearized Approach

In the mathematical field of optimization, a lot of methods have been discovered and developed (Section 3) in order to solve problems of higher complexity and size and have been implemented in a variety of programming languanges and computer applications. Therefore it is desirable to overcome former mentioned difficulty of non-commutativity (Section 5.3.1.1) and model given problem towards a formulation, e.g. to a LP or a MIP formulation, which can efficiently be handed by existing program solvers.

Thus, pursuing considerations towards this aim, the fact of the amplify factors taking on values of a discrete, finite set only, leads to the introduction of boolean

[3]This algorithm is also part of current prototype software for automated prosthesis adjustment.

aid variables $f_i^l \in \{0,1\},\ i = 1,2,\ l = 1,\dots,|\mathcal{V}_\mathcal{A}|$, related to the available values for amplify factor $f_{\mathcal{A}_i}$ by running index l.
By pre-calculating every relevant information from the user's amplified myo-signals for each available amplify factor value in $\mathcal{V}_\mathcal{A}$, all required values for setting up linear inequalities and a linear objective function can be stored, thus providing all components in order to set up a linear mixed-integer program of the form 3.5. With storage $x_{\mathrm{X}(f_l(\mathrm{S}))}$ for the value of the settable parameter x, calculated by parameter calculating function **X**, from the user's myo-signal S, amplified by the l-th value f_l in the list of available amplify factor values $\mathcal{V}_\mathcal{A}$, and a sufficiently large **Big M**: $\mathbf{M} \in \mathbb{R}^+$, constraints and objective function can be formulated as follows.

$$\max_{f_i^l} \sum_{i=1}^{2} \sum_{l=1}^{|\mathcal{V}_A|} \left(f_i^l \cdot l \right) \tag{O2}$$

subject to:

$$\sum_{l=1}^{|\mathcal{V}_A|} f_i^l \cdot l \cdot 0.25 \cdot \mathrm{S}_i \geq \mathrm{TOL}_{\mathrm{S}}^{\mathrm{min}} \tag{C1}$$

$$\sum_{l=1}^{|\mathcal{V}_A|} f_i^l \cdot l \cdot 0.25 \cdot \mathrm{S}_i \leq \mathrm{TOL}_{\mathrm{S}}^{\mathrm{max}} \tag{C2}$$

$$f_i^l \cdot x_{\vec{\mathrm{t}}_0(f_l(\mathrm{S}_i))} \leq f_j^l \cdot x_{\vec{\mathrm{t}}_0(f_l(\mathrm{SS}_i))} + \mathbf{M} \cdot (1 - f_j^l) \tag{C3}$$

$$\sum_{l=1}^{|\mathcal{V}_A|} f_i^l \cdot l \cdot 0.25 \cdot \mathrm{NM}_i \leq \mathrm{TOL}_{\mathrm{NM}} \tag{C4}$$

$$\sum_{l=1}^{|\mathcal{V}_A|} f_i^l \cdot x_{\mathrm{X_I}(f_l(\mathrm{I}_i))} \geq \mathrm{TOL}_{\mathrm{I}}^{\mathrm{min}} \tag{C5}$$

$$\sum_{l=1}^{|\mathcal{V}_A|} f_i^l \cdot x_{\mathrm{X_I}(f_l(\mathrm{I}_i))} \leq \mathrm{TOL}_{\mathrm{I}}^{\mathrm{max}} \tag{C6}$$

$$\sum_{l=1}^{|\mathcal{V}_A|} f_i^l \cdot x_{\mathrm{X_{SL}}(f_l(\mathrm{I}_i))} \leq \mathrm{TOL}_{\mathrm{SL}}^{\mathrm{max}} \tag{C7}$$

$$\sum_{l=1}^{|\mathcal{V}_A|} f_i^l \cdot x_{\vec{\mathrm{X}}_\mathrm{I}(f_l(\mathrm{I}_i),(f_l(\mathrm{FM}_i))} \geq \mathrm{TOL}_{\mathrm{I}}^{\mathrm{min}} \tag{C8}$$

$$\sum_{l=1}^{|\mathcal{V}_A|} f_i^l \cdot x_{\vec{\mathrm{X}}_\mathrm{I}(f_l(\mathrm{I}_i),(f_l(\mathrm{FM}_i))} \leq \mathrm{TOL}_{\mathrm{I}}^{\mathrm{max}} \tag{C9}$$

$$\sum_{l=1}^{|\mathcal{V}_A|} f_i^l \geq 1 \tag{C10}$$

$$\sum_{l=1}^{|\mathcal{V}_A|} f_i^l \leq 1 \tag{C11}$$

$$f_i^l \in \{0, 1\} \tag{D1}$$

for $i, j = 1, 2,\ i \neq j, l = 1, \ldots, |\mathcal{V}_{\mathcal{A}}|$. This set of constraints consists of data, generated by all possibly selectable amplify factor values. "Activation" respectively "inactivation" of the part of a constraint, which is set up by calculations with myo-signals amplified by amplify factor value f_l, is controlled by the corresponding boolean aid variable f_i^l. With this formulation, the optimization yields the maximum amplify factor value that satisfies all given constraints, by setting the value of the boolean variable, which corresponds to determined optimal amplify factor value, to 1. Constraints C10-C11 ensure that boolean variables corresponding to all other available amplify factor values remain 0, such that only one amplify factor value per channel contributes to resulting maximization of objective function at the end of optimization process, i.e. only one value per channel is being determined as optimal amplify factor for prosthesis adjustment.

5.3.1.4 Linearized Approach with Soft Constraints

As mentioned in Section 5.2.3.1, a Program's optimization program does not necessarily consist of obligatorily satisfied hard constraints only, but can also be enhanced by potentially fulfilled soft constraints. For currently considered Program for example, constraint C2 makes use of tolerance threshold $\text{TOL}_\text{S}^\text{max}$ to tie the maximum tolerated value of the user's Strong Signal S to a certain domain, in which the prosthesis is controllable. This upper bound of the range defined by $\text{TOL}_\text{S}^\text{max}$ indeed leads to a sufficiently working amplify factor in case of feasibility of the optimization problem, but might be set very generously however. So the choice of a tighter bound would enhance controllability, in case the user's myo-signals are capable of satisfying even these tougher restrictions. Therefore, tolerance threshold $\text{TOL}_\text{S}^{\text{max}^*}$ (Table 5.2) can be used to formulate a soft constraint, tying the user's strong signal to a tighter range, thus potentially leading to another amplify factor value, which might yield even better control to a user, who is capable of more precise variation of myo-signal strength.

So the aim is to find a way to distinguish between hard- and soft constraints within the formulation of inequalities and related objective function, such that given problem again receives the form of a linear mixed-integer program 3.5. This aim makes it necessary to postulate a fixed, linear order within the set of constraints $\mathcal{C}$: C1 $\preccurlyeq$ C2 $\preccurlyeq \ldots \preccurlyeq$ C$|\mathcal{C}|$, in order to be capable of tying them to boolean aid variables $c_i^1, c_i^2, \ldots, c_i^{|\mathcal{C}|} \in \{0, 1\}$ and appropriately chosen priority values $\mathsf{c}_i^1, \mathsf{c}_i^2, \ldots, \mathsf{c}_i^{|\mathcal{C}|} \in \mathbb{R}$ (Section 5.3.2), ranking the importance of all soft constraints within the subset of soft constraints $\mathcal{C}_\text{soft} \subset \mathcal{C}$. Together with the subset of hard constraints $\mathcal{C}_\text{hard} \in \mathcal{C}$, the set of indices $\text{idx}(\mathcal{C}_\text{hard})$ and $\text{idx}(\mathcal{C}_\text{soft})$ of hard constraints respectively soft constraints within predefined linear order, and a sufficiently large $\tilde{\mathbf{M}} \in \mathbb{R}^+$, the system of constraints and corresponding objective function turn to the form:

$$\max_{c_i^n} \sum_{n=1}^{|\mathcal{C}|} (c_i^n \mathsf{c}_i^n) \tag{O3}$$

subject to:

$$\tilde{\mathbf{M}} \cdot (1 - c_i^1) + \sum_{l=1}^{|\mathcal{V}_A|} f_i^l \cdot l \cdot 0.25 \cdot \mathrm{S}_i \geq \mathrm{TOL}_{\mathrm{S}}^{\min} \tag{C1}$$

$$-\tilde{\mathbf{M}} \cdot (1 - c_i^2) + \sum_{l=1}^{|\mathcal{V}_A|} f_i^l \cdot l \cdot 0.25 \cdot \mathrm{S}_i \leq \mathrm{TOL}_{\mathrm{S}}^{\max} \tag{C2}$$

$$-\tilde{\mathbf{M}} \cdot (1 - c_i^3) + f_i^l \cdot x_{\vec{\mathrm{t}}0(f_l(\mathrm{S}_i))} \leq f_j^l \cdot x_{\vec{\mathrm{t}}0(f_l(\mathrm{SS}_i))} + \mathbf{M} \cdot (1 - f_j^l) \tag{C3}$$

$$-\tilde{\mathbf{M}} \cdot (1 - c_i^4) + \sum_{l=1}^{|\mathcal{V}_A|} f_i^l \cdot l \cdot 0.25 \cdot \mathrm{NM}_i \leq \mathrm{TOL}_{\mathrm{NM}} \tag{C4}$$

$$\tilde{\mathbf{M}} \cdot (1 - c_i^5) + \sum_{l=1}^{|\mathcal{V}_A|} f_i^l \cdot x_{\mathrm{x_I}(f_l(\mathrm{I}_i))} \geq \mathrm{TOL}_{\mathrm{I}}^{\min} \tag{C5}$$

$$-\tilde{\mathbf{M}} \cdot (1 - c_i^6) + \sum_{l=1}^{|\mathcal{V}_A|} f_i^l \cdot x_{\mathrm{x_I}(f_l(\mathrm{I}_i))} \leq \mathrm{TOL}_{\mathrm{I}}^{\max} \tag{C6}$$

$$-\tilde{\mathbf{M}} \cdot (1 - c_i^7) + \sum_{l=1}^{|\mathcal{V}_A|} f_i^l \cdot x_{\mathrm{x_{SL}}(f_l(\mathrm{I}_i))} \leq \mathrm{TOL}_{\mathrm{SL}}^{\max} \tag{C7}$$

$$\tilde{\mathbf{M}} \cdot (1 - c_i^8) + \sum_{l=1}^{|\mathcal{V}_A|} f_i^l \cdot x_{\vec{\mathrm{x}}_\mathrm{I}(f_l(\mathrm{I}_i),(f_l(\mathrm{FM}_i))} \geq \mathrm{TOL}_{\mathrm{I}}^{\min} \tag{C8}$$

$$-\tilde{\mathbf{M}} \cdot (1 - c_i^9) + \sum_{l=1}^{|\mathcal{V}_A|} f_i^l \cdot x_{\vec{\mathrm{x}}_\mathrm{I}(f_l(\mathrm{I}_i),(f_l(\mathrm{FM}_i))} \leq \mathrm{TOL}_{\mathrm{I}}^{\max} \tag{C9}$$

$$\tilde{\mathbf{M}} \cdot (1 - c_i^{10}) + \sum_{l=1}^{|\mathcal{V}_A|} f_i^l \geq 1 \tag{C10}$$

$$-\tilde{\mathbf{M}} \cdot (1 - c_i^{11}) + \sum_{l=1}^{|\mathcal{V}_A|} f_i^l \leq 1 \tag{C11}$$

$$-\tilde{\mathbf{M}} \cdot (1 - c_i^{12}) + \sum_{l=1}^{|\mathcal{V}_A|} f_i^l \cdot l \cdot 0.25 \cdot \mathrm{S}_i \leq \mathrm{TOL}_{\mathrm{S}}^{\max^*} \tag{C12}$$

$$\sum_{c_i^n \in \mathcal{C}_{\mathrm{hard}}} c_i^n \geq |\mathcal{C}_{\mathrm{hard}}| \tag{C13}$$

$$\sum_{c_i^n \in \mathcal{C}_{\mathrm{hard}}} c_i^n \leq |\mathcal{C}_{\mathrm{hard}}| \tag{C14}$$

$$f_i^l \in \{0, 1\} \tag{D1}$$

$$c_i^n \in \{0, 1\} \tag{D2}$$

for $i, j = 1, 2,\ i \neq j, l = 1, \ldots, |\mathcal{V}_A|$. Constraints C13-C14 ensure, that all hard constraints remain obligatory conditions, since it forces all boolean aid variables, which are related to hard constraints, to take on value 1. The sum occurring in objective function now, which consists of newly introduced boolean aid variables and connected priority values, each related to a specific constraint in the ordered list of constraints $\mathcal{C}$, leads to acquiescence of a soft constraint, whenever it does not cause

any neglection of another obligatory constraint, or a soft constraint with higher priority. Latter is achieved by making the right choice of priority value for each constraint, such that, within the objective function's additional summation, one soft constraint of higher importance is given preference to an arbitrary number of soft constraints with lower priority (Section 5.3.2).
So the basic idea of finding an optimal prosthesis adjustment by maximizing the amplify factor values subject to given, obligatory constraints has changed to the idea of maximizing the total number of satisfied constraints, with all hard constraints still remaining obligatory, thus potentially providing even more functionality and comfort to the user, even with lower amplify factor values, since obligatory constraints still force amplify factor values to be chosen in appropriate ranges.

5.3.1.5 Generalized, Linearized Approach with Soft Constraints

The so far formulated optimization problem only considered data from one sample myo-signal per movement. In practice however, it is reasonable to think of a way in order to guarantee that no outliers, occurring during data recording, distort the optimization process or lead to a mistaken infeasibility of the optimization program.

One way of treating this issue is to collect a broader number of sample myo-signals of each required signal pattern and use statistical methods in order to generate solid representatives for each movement. These representatives, containing a much wider range of information about the user's average myo-signals, serve for further calculation of required data and lead to more stable results in the formally introduced formulation of given problem[4].

Another approach is again to collect a broader number of sample myo-signals of each movement, summarizing gathered information in several data sets, which themselves consist of one sample myo-signal per required movement only. By formulating the optimization problem towards the idea of finding amplify factor values, such that a maximum number of data sets provide feasibility of corresponding optimization program, a mixed-integer program can be formulated, which again does not only consider information of one single sample myo-signal only, but a much wider range of information, leading to way more significant optimization results in practice.

With the set $\mathcal{U}$ of all data sets $U_i \in \mathcal{U}, i = 1, \ldots, |\mathcal{U}|$, again following a linear order $U_1 \preccurlyeq U_2 \preccurlyeq \ldots \preccurlyeq U_{|\mathcal{U}|}$, corresponding boolean aid variables $u_i \in \{0, 1\}, i = 1, \ldots, |\mathcal{U}|$, a sufficiently large $\hat{\mathbf{M}} \in \mathbb{R}$ and the newly introduced superscript index q for sample myo-signals S^q and newly occurring within the notation of the storage $x^q_{\mathsf{x}(f_l(\mathsf{s}))}$ (Section 5.3.1.3), emphasizing its calculation from myo-signals of the q-th data set, constraints and objective function turn to:

$$\max_{f_i^l, c_i^n, u_q} \sum_{q=1}^{|\mathcal{U}|} \left(u_q + \sum_{n=1}^{|\mathcal{C}|} (c_i^n \mathsf{c}_i^n) \right) \tag{O4}$$

[4]This approach is also part of current prototype software for automated prosthesis adjustment.

subject to:

$$\hat{\mathbf{M}} \cdot (1 - u_q) + \tilde{\mathbf{M}} \cdot (1 - c_i^1) + \sum_{l=1}^{|\mathcal{V}_A|} f_i^l \cdot l \cdot 0.25 \cdot \mathrm{S}_i^q \geq \mathrm{TOL}_\mathrm{S}^{\mathrm{min}} \tag{C1}$$

$$-\hat{\mathbf{M}} \cdot (1 - u_q) - \tilde{\mathbf{M}} \cdot (1 - c_i^2) + \sum_{l=1}^{|\mathcal{V}_A|} f_i^l \cdot l \cdot 0.25 \cdot \mathrm{S}_i^q \leq \mathrm{TOL}_\mathrm{S}^{\mathrm{max}} \tag{C2}$$

$$-\hat{\mathbf{M}} \cdot (1 - u_q) - \tilde{\mathbf{M}} \cdot (1 - c_i^3) + f_i^l \cdot x^q_{\vec{\mathrm{t}}0(f_l(\mathrm{S}_i))} \leq f_j^l \cdot x^q_{\vec{\mathrm{t}}0(f_l(\mathrm{SS}_i))} + \mathbf{M} \cdot (1 - f_j^l) \tag{C3}$$

$$-\hat{\mathbf{M}} \cdot (1 - u_q) - \tilde{\mathbf{M}} \cdot (1 - c_i^4) + \sum_{l=1}^{|\mathcal{V}_A|} f_i^l \cdot l \cdot 0.25 \cdot \mathrm{NM}_i^q \leq \mathrm{TOL}_\mathrm{NM} \tag{C4}$$

$$\hat{\mathbf{M}} \cdot (1 - u_q) + \tilde{\mathbf{M}} \cdot (1 - c_i^5) + \sum_{l=1}^{|\mathcal{V}_A|} f_i^l \cdot x^q_{\mathrm{x_I}(f_l(\mathrm{I}_i))} \geq \mathrm{TOL}_\mathrm{I}^{\mathrm{min}} \tag{C5}$$

$$-\hat{\mathbf{M}} \cdot (1 - u_q) - \tilde{\mathbf{M}} \cdot (1 - c_i^6) + \sum_{l=1}^{|\mathcal{V}_A|} f_i^l \cdot x^q_{\mathrm{x_I}(f_l(\mathrm{I}_i))} \leq \mathrm{TOL}_\mathrm{I}^{\mathrm{max}} \tag{C6}$$

$$-\hat{\mathbf{M}} \cdot (1 - u_q) - \tilde{\mathbf{M}} \cdot (1 - c_i^7) + \sum_{l=1}^{|\mathcal{V}_A|} f_i^l \cdot x^q_{\mathrm{x_{SL}}(f_l(\mathrm{I}_i))} \leq \mathrm{TOL}_\mathrm{SL}^{\mathrm{max}} \tag{C7}$$

$$\hat{\mathbf{M}} \cdot (1 - u_q) + \tilde{\mathbf{M}} \cdot (1 - c_i^8) + \sum_{l=1}^{|\mathcal{V}_A|} f_i^l \cdot x^q_{\vec{\mathrm{x}}_\mathrm{I}(f_l(\mathrm{I}_i)),(f_l(\mathrm{FM}_i))} \geq \mathrm{TOL}_\mathrm{I}^{\mathrm{min}} \tag{C8}$$

$$-\hat{\mathbf{M}} \cdot (1 - u_q) - \tilde{\mathbf{M}} \cdot (1 - c_i^9) + \sum_{l=1}^{|\mathcal{V}_A|} f_i^l \cdot x^q_{\vec{\mathrm{x}}_\mathrm{I}(f_l(\mathrm{I}_i)),(f_l(\mathrm{FM}_i))} \leq \mathrm{TOL}_\mathrm{I}^{\mathrm{max}} \tag{C9}$$

$$\hat{\mathbf{M}} \cdot (1 - u_q) + \tilde{\mathbf{M}} \cdot (1 - c_i^{10}) + \sum_{l=1}^{|\mathcal{V}_A|} f_i^l \geq 1 \tag{C10}$$

$$-\hat{\mathbf{M}} \cdot (1 - u_q) - \tilde{\mathbf{M}} \cdot (1 - c_i^{11}) + \sum_{l=1}^{|\mathcal{V}_A|} f_i^l \leq 1 \tag{C11}$$

$$-\hat{\mathbf{M}} \cdot (1 - u_q) - \tilde{\mathbf{M}} \cdot (1 - c_i^{12}) + \sum_{l=1}^{|\mathcal{V}_A|} f_i^l \cdot l \cdot 0.25 \cdot \mathrm{S}_i^q \leq \mathrm{TOL}_\mathrm{S}^{\mathrm{max}*} \tag{C12}$$

$$\hat{\mathbf{M}} \cdot (1 - u_q) + \sum_{c_i^n \in \mathcal{C}_\mathrm{hard}} c_i^n \geq |\mathcal{C}_\mathrm{hard}| \tag{C13}$$

$$-\hat{\mathbf{M}} \cdot (1 - u_q) + \sum_{c_i^n \in \mathcal{C}_\mathrm{hard}} c_i^n \leq |\mathcal{C}_\mathrm{hard}| \tag{C14}$$

$$f_i^l \in \{0, 1\} \tag{D1}$$

$$c_i^n \in \{0, 1\} \tag{D2}$$

$$u_q \in \{0, 1\} \tag{D3}$$

for $i, j = 1, 2,\ i \neq j,\ l = 1, \ldots, |\mathcal{V}_\mathcal{A}|$ and $q = 1, \ldots |\mathcal{U}|$. Whenever no amplify factor values within the optimization program, generated by data set q, can be found, such that all obligatory hard constraints are satisfied, boolean variable u_q receives value 0, thus indicating, that data set q's optimization program is infeasible. In case of feasibility, the second sum of the objective function over the boolean variables multiplied by related priority values, corresponding to the soft constraints, ensures the inclusion of any further non-obligatory constraint to the optimization process, which does not disturb the obligatory part of the problem.

5.3.1.6 Generalized Interpolation Approach with Soft Constraints

So far, all discussed approaches were based on the simplification of using the finite set of amplify factor values $\mathcal{V}_{\mathcal{A}}$, which is sufficient in practice, since only these specific values are manually settable by an orthopedic technician within *Ottobock*'s Data Station (Figure 2.2). Theoretically however, the prosthesis' internal control unit would be capable of handling a much wider range of amplify factor values. This arises the question of a way to formulate given problem as optimization program, allowing the amplify factor values, as being part of the set of decision variables, namely the decision variables of actual interest, to take on continuous values. This however can just be achieved by compromises due to initially mention problem of non-commutativity of parameter calculating functions (Section 5.3.1.1).

One way to simulate continuity up to a certain degree is, to make use of linear interpolation of all functions, calculating required data, in composition with Amplifier function $\vec{\mathbf{F}}_{\mathcal{A}}$ and use resulting, continuous, piece-wise linear data in order to replace formerly discrete parts of the optimization program.

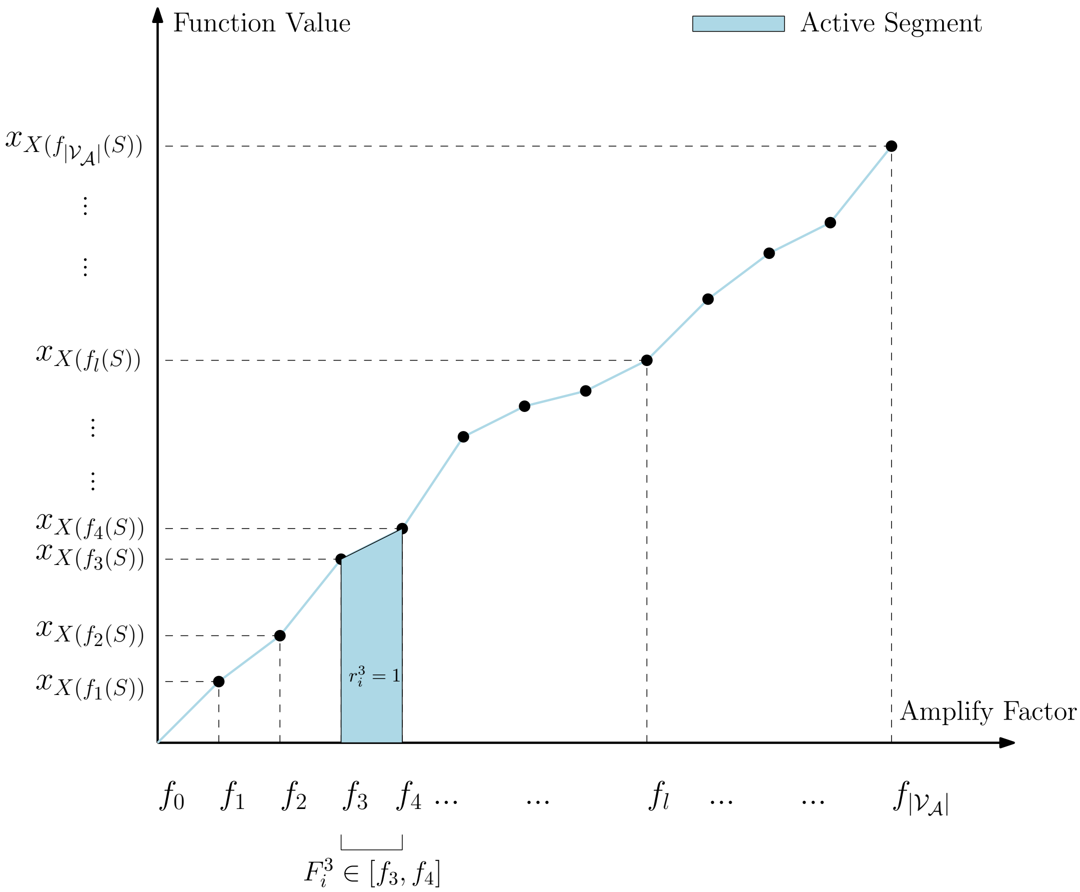

FIGURE 5.14: Linear interpolation of data calcuating function $\mathbf{X}$

As shown in Figure 5.14, a possible choice for the interpolation's decomposition points are these being in the set of available amplify factor values $\mathcal{V}_{\mathcal{A}}$. For all values $f_l \in \mathcal{V}_{\mathcal{A}}$, sample myo-signal S of a certain movement can be amplified, desired values of function **X** can be determined and stored in $x_{\mathrm{X}(f_l(\mathrm{S}))}$, in order to linearly interpolate them by the function:

$$\bar{\mathbf{X}}(F) = \frac{f_{l+1} - F}{f_{l+1} - f_l} x_{\mathrm{X}(f_l(\mathrm{S}))} + \frac{F - f_l}{f_{l+1} - f_l} x_{\mathrm{X}(f_{l+1}(\mathrm{S}))} \tag{5.12}$$

for $F \in [f_l, f_{l+1}],\ l = 1, \dots, |\mathcal{V}_{\mathcal{A}}| - 1$. By replacing all discrete values $x_{\mathrm{X}(f_l(\mathrm{S}))}$ by linear segments 5.12 within the set of constraints and introducing boolean aid variables $r_i^l \in \{0, 1\}$ corresponding to the segment between amplify factor value f_l and f_{l+1} of channel i, formally aiding boolean decision variables f_i^l, indicating the certain amplify factor value being chosen by the optimization process by either taking on value 0 or 1, can now be substituted by continuous decision variables $F_i^l \in [f_l, f_{l+1}], l = 1, \dots, |\mathcal{V}_{\mathcal{A}}| - 1, i = 1, 2$.

Considering left-hand-side of constraint C6, substitution of all just mentioned components would lead to:

$$\dots + \sum_{l=1}^{|\mathcal{V}_{\mathcal{A}}|-1} \bar{\mathbf{X}}_{\mathrm{I}}(F_i^l) \cdot r_i^l =$$

$$\dots + \sum_{l=1}^{|\mathcal{V}_{\mathcal{A}}|-1} \left(\frac{f_{l+1} - F_i^l}{f_{l+1} - f_l} x^q_{\mathrm{X_I}(f_l(\mathrm{I}_i))} + \frac{F_i^l - f_l}{f_{l+1} - f_l} x^q_{\mathrm{X_I}(f_{l+1}(\mathrm{I}_i))} \right) \cdot r_i^l =$$

$$\dots + \sum_{l=1}^{|\mathcal{V}_{\mathcal{A}}|-1} \left(\frac{f_{l+1} \cdot x^q_{\mathrm{X_I}(f_l(\mathrm{I}_i))} - f_l \cdot x^q_{\mathrm{X_I}(f_{l+1}(\mathrm{I}_i))}}{f_{l+1} - f_l} \right) r_i^l + \sum_{l=1}^{|\mathcal{V}_{\mathcal{A}}|-1} \left(\frac{x^q_{\mathrm{X_I}(f_{l+1}(\mathrm{I}_i))} - x^q_{\mathrm{X_I}(f_l(\mathrm{I}_i))}}{f_{l+1} - f_l} \right) \cdot \overbrace{r_i^l \cdot F_i^l}^{non-linearity}$$

with additional constraints

$$\sum_{l=1}^{|\mathcal{V}_{\mathcal{A}}|} r_i^l = 1$$

in order to guarantee, that only one segment of the amplify factor values is active, i.e. one value for amplify factor contributes to given sum. Occurring non-linearity in the second sum can be reformulated towards a linear term by substitution $\tilde{F}_i^l = r_i^l \cdot F_i^l$ and additional constraints as explained in Section 3.4.5.2.

This linearization can analogously be proceeded for every constraint of former optimization program, the set of constraints and related objective function turn to the form:

$$\max_{u_q, c_i^n} \sum_{q=1}^{|\mathcal{U}|} \left(u_q + \sum_{n=1}^{|\mathcal{C}|} (c_i^n \mathrm{c}_i^n) \right) \tag{O5}$$

subject to:

$$\hat{\mathbf{M}} \cdot (1 - u_q) + \tilde{\mathbf{M}} \cdot (1 - c_i^1) + \sum_{l=1}^{|\mathcal{V}_A|} \tilde{F}_i^l \cdot \mathrm{S}_i^q \geq \mathrm{TOL}_{\mathrm{S}}^{\min} \tag{C1}$$

$$-\hat{\mathbf{M}} \cdot (1 - u_q) - \tilde{\mathbf{M}} \cdot (1 - c_i^2) + \sum_{l=1}^{|\mathcal{V}_A|} \tilde{F}_i^l \cdot \mathrm{S}_i^q \leq \mathrm{TOL}_{\mathrm{S}}^{\max} \tag{C2}$$

$$\begin{aligned} &- \mathbf{M} \cdot (1 - r_j^l) - \hat{\mathbf{M}} \cdot (1 - u_q) - \tilde{\mathbf{M}} \cdot (1 - c_i^3) + \\ &\sum_{l=1}^{|\mathcal{V}_A|-1} \left(\frac{f_{l+1} \cdot x^q_{\mathrm{t}0(f_l(\mathrm{S}_i))} - f_l \cdot x^q_{\mathrm{t}0(f_{l+1}(\mathrm{S}_i))}}{f_{l+1} - f_l} \right) r_i^l + \sum_{l=1}^{|\mathcal{V}_A|-1} \left(\frac{x^q_{\mathrm{t}0(f_{l+1}(\mathrm{S}_i))} - x^q_{\mathrm{t}0(f_l(\mathrm{S}_i))}}{f_{l+1} - f_l} \right) \cdot \tilde{F}_i^l \leq \\ &\sum_{l=1}^{|\mathcal{V}_A|-1} \left(\frac{f_{l+1} \cdot x^q_{\mathrm{t}0(f_l(\mathrm{SS}_i))} - f_l \cdot x^q_{\mathrm{t}0(f_{l+1}(\mathrm{SS}_i))}}{f_{l+1} - f_l} \right) r_i^l + \sum_{l=1}^{|\mathcal{V}_A|-1} \left(\frac{x^q_{\mathrm{t}0(f_{l+1}(\mathrm{SS}_i))} - x^q_{\mathrm{t}0(f_l(\mathrm{SS}_i))}}{f_{l+1} - f_l} \right) \cdot \tilde{F}_i^l \end{aligned} \tag{C3}$$

$$-\hat{\mathbf{M}} \cdot (1 - u_q) - \tilde{\mathbf{M}} \cdot (1 - c_i^4) + \sum_{l=1}^{|\mathcal{V}_A|} \tilde{F}_i^l \cdot \mathrm{NM}_i^q \leq \mathrm{TOL}_{\mathrm{NM}} \tag{C4}$$

$$\begin{aligned} &\hat{\mathbf{M}} \cdot (1 - u_q) + \tilde{\mathbf{M}} \cdot (1 - c_i^5) + \\ &\sum_{l=1}^{|\mathcal{V}_A|-1} \left(\frac{f_{l+1} \cdot x^q_{\mathrm{XI}(f_l(\mathrm{I}_i))} - f_l \cdot x^q_{\mathrm{XI}(f_{l+1}(\mathrm{I}_i))}}{f_{l+1} - f_l} \right) r_i^l + \sum_{l=1}^{|\mathcal{V}_A|-1} \left(\frac{x^q_{\mathrm{XI}(f_{l+1}(\mathrm{I}_i))} - x^q_{\mathrm{XI}(f_l(\mathrm{I}_i))}}{f_{l+1} - f_l} \right) \cdot \tilde{F}_i^l \geq \mathrm{TOL}_{\mathrm{I}}^{\min} \end{aligned} \tag{C5}$$

$$\begin{aligned} &-\hat{\mathbf{M}} \cdot (1 - u_q) - \tilde{\mathbf{M}} \cdot (1 - c_i^6) + \\ &\sum_{l=1}^{|\mathcal{V}_A|-1} \left(\frac{f_{l+1} \cdot x^q_{\mathrm{XI}(f_l(\mathrm{I}_i))} - f_l \cdot x^q_{\mathrm{XI}(f_{l+1}(\mathrm{I}_i))}}{f_{l+1} - f_l} \right) r_i^l + \sum_{l=1}^{|\mathcal{V}_A|-1} \left(\frac{x^q_{\mathrm{XI}(f_{l+1}(\mathrm{I}_i))} - x^q_{\mathrm{XI}(f_l(\mathrm{I}_i))}}{f_{l+1} - f_l} \right) \cdot \tilde{F}_i^l \leq \mathrm{TOL}_{\mathrm{I}}^{\max} \end{aligned} \tag{C6}$$

$$\begin{aligned} &-\hat{\mathbf{M}} \cdot (1 - u_q) - \tilde{\mathbf{M}} \cdot (1 - c_i^7) + \\ &\sum_{l=1}^{|\mathcal{V}_A|-1} \left(\frac{f_{l+1} \cdot x^q_{\mathrm{XSL}(f_l(\mathrm{I}_i))} - f_l \cdot x^q_{\mathrm{XSL}(f_{l+1}(\mathrm{I}_i))}}{f_{l+1} - f_l} \right) r_i^l + \sum_{l=1}^{|\mathcal{V}_A|-1} \left(\frac{x^q_{\mathrm{XSL}(f_{l+1}(\mathrm{I}_i))} - x^q_{\mathrm{XSL}(f_l(\mathrm{I}_i))}}{f_{l+1} - f_l} \right) \cdot \tilde{F}_i^l \leq \mathrm{TOL}_{\mathrm{SL}}^{\max} \end{aligned} \tag{C7}$$

$$\begin{aligned} &\hat{\mathbf{M}} \cdot (1 - u_q) + \tilde{\mathbf{M}} \cdot (1 - c_i^8) + \\ &\sum_{l=1}^{|\mathcal{V}_A|-1} \left(\frac{f_{l+1} \cdot x^q_{\vec{\mathrm{X}}\mathrm{I}(f_l(\mathrm{I}_i)),(f_l(\mathrm{FM}_i))} - f_l \cdot x^q_{\vec{\mathrm{X}}\mathrm{I}(f_{l+1}(\mathrm{I}_i)),(f_{l+1}(\mathrm{FM}_i))}}{f_{l+1} - f_l} \right) r_i^l + \\ &\sum_{l=1}^{|\mathcal{V}_A|-1} \left(\frac{x^q_{\vec{\mathrm{X}}\mathrm{I}(f_{l+1}(\mathrm{I}_i)),(f_{l+1}(\mathrm{FM}_i))} - x^q_{\vec{\mathrm{X}}\mathrm{I}(f_l(\mathrm{I}_i)),(f_l(\mathrm{FM}_i))}}{f_{l+1} - f_l} \right) \cdot \tilde{F}_i^l \geq \mathrm{TOL}_{\mathrm{I}}^{\min} \end{aligned} \tag{C8}$$

$$\begin{aligned} &-\hat{\mathbf{M}} \cdot (1 - u_q) - \tilde{\mathbf{M}} \cdot (1 - c_i^9) + \\ &\sum_{l=1}^{|\mathcal{V}_A|-1} \left(\frac{f_{l+1} \cdot x^q_{\vec{\mathrm{X}}\mathrm{I}(f_l(\mathrm{I}_i)),(f_l(\mathrm{FM}_i))} - f_l \cdot x^q_{\vec{\mathrm{X}}\mathrm{I}(f_{l+1}(\mathrm{I}_i)),(f_{l+1}(\mathrm{FM}_i))}}{f_{l+1} - f_l} \right) r_i^l + \\ &\sum_{l=1}^{|\mathcal{V}_A|-1} \left(\frac{x^q_{\vec{\mathrm{X}}\mathrm{I}(f_{l+1}(\mathrm{I}_i)),(f_{l+1}(\mathrm{FM}_i))} - x^q_{\vec{\mathrm{X}}\mathrm{I}(f_l(\mathrm{I}_i)),(f_l(\mathrm{FM}_i))}}{f_{l+1} - f_l} \right) \cdot \tilde{F}_i^l \leq \mathrm{TOL}_{\mathrm{I}}^{\max} \end{aligned} \tag{C9}$$

$$-\hat{\mathbf{M}} \cdot (1-u_q) - \tilde{\mathbf{M}} \cdot (1-c_i^{12}) + \sum_{l=1}^{|\mathcal{V}_A|} \tilde{F}_i^l \cdot \mathrm{S}_i^q \leq \mathrm{TOL}_{\mathrm{S}}^{\max^*} \tag{C12}$$

$$\hat{\mathbf{M}} \cdot (1-u_q) + \sum_{c_i^n \in \mathcal{C}_{\text{hard}}} c_i^n \geq |\mathcal{C}_{\text{hard}}| \tag{C13}$$

$$-\hat{\mathbf{M}} \cdot (1-u_q) + \sum_{c_i^n \in \mathcal{C}_{\text{hard}}} c_i^n \leq |\mathcal{C}_{\text{hard}}| \tag{C14}$$

$$\hat{\mathbf{M}} \cdot (1-u_q) + \tilde{\mathbf{M}} \cdot (1-c_{i,l}^{15}) + \tilde{F}_i^l \geq 0 \tag{C15}$$

$$-\hat{\mathbf{M}} \cdot (1-u_q) - \tilde{\mathbf{M}} \cdot (1-c_{i,l}^{16}) + \tilde{F}_i^l \leq f_{l+1} \cdot r_i^l \tag{C16}$$

$$-\hat{\mathbf{M}} \cdot (1-u_q) - \tilde{\mathbf{M}} \cdot (1-c_{i,l}^{17}) + \tilde{F}_i^l \leq F_i^l \tag{C17}$$

$$\hat{\mathbf{M}} \cdot (1-u_q) + \tilde{\mathbf{M}} \cdot (1-c_{i,l}^{18}) + \tilde{F}_i^l \geq f_{l+1} \cdot (r_i^l - 1) + F_i^l \tag{C18}$$

$$\hat{\mathbf{M}} \cdot (1-u_q) + \tilde{\mathbf{M}} \cdot (1-c_{i,l}^{19}) + \sum_{l=1}^{|\mathcal{V}_\mathcal{A}|-1} r_i^l \geq 1 \tag{C19}$$

$$-\hat{\mathbf{M}} \cdot (1-u_q) - \tilde{\mathbf{M}} \cdot (1-c_{i,l}^{20}) + \sum_{l=1}^{|\mathcal{V}_\mathcal{A}|-1} r_i^l \leq 1 \tag{C20}$$

$$c_i^n \in \{0,1\} \tag{D2}$$

$$u_q \in \{0,1\} \tag{D3}$$

$$r_i^l \in \{0,1\} \tag{D4}$$

$$\tilde{F}_i^l \in \mathbb{R}^+ \tag{D5}$$

$$F_i^l \in [f_{l-1}, f_l] \tag{D6}$$

for $i, j = 1, 2,\ i \neq j,\ q = 1, \ldots |\mathcal{U}|$ and $l = 1, \ldots, |\mathcal{V}_\mathcal{A}|$. Constraint $\mathrm{C}1_1$- $\mathrm{C}14_1$ now are a piece-wise linearized version of former, discrete inequalities. $\mathrm{C}15_l$-$\mathrm{C}18_l$ were added in order to accomplish substitution of the trouble causing product of two of the decision variables. Constraints $\mathrm{C}19_1$-$\mathrm{C}20_1$ finally make sure, that only one boolean variable r_i^l is equal to 1 whereas all others remain 0, thus only one segment of the amplify factor values is considered at a time during the process of optimization.
Constraints $\mathrm{D}2_n$-$\mathrm{D}6_l$ postulate the valid domain of all appearing decision variables.

Remark: Many of the constraints $\mathrm{C}1_1$-$\mathrm{C}20_1$ actually represent a bigger number of constraints, as they are notated channel-wise or amplify-factor-wise by its related index. Therefore, for considered Program, the total number of constraints within linearized, mixed integer program is way higher than probably conjectured at first sight, consisting of several hundreds of equations.

Since every constraint by itself is bound to a decision variable c_i^n, occurring in the optimization program together with decision variables F_i^l for every amplify factor value in combination with its related boolean variable r_i^l for each segment of interpolation, the number of constraints already gives an idea about the large size of the linear mixed integer program, given problem of prosthesis parameter adjustment provokes. This again allows the conclusion, that a huge number of issues have to be considered within the process of parameter determination, giving an idea about the problematic nature of manual prosthesis adjustment. This in turn emphasizes again, that the aim of finding a way towards automatic systems for prosthesis adjustment is absolutely reasonable.

5.3.2 Choice of Priority Values for Soft Constraints

There are two important aspects, which have to be regarded, when seeking for an appropriate choice of priority values for soft constraints contained in the optimization program of a prosthesis' considered Program.

- On the one hand, one assumption within the idea of adding enhancing soft constraints to the system is, that one satisfied soft constraint of higher priority produces more benefits to the prosthesis' handling than an arbitrarily high number of less important soft constraints. One way to achieve this certain ranking of soft constraints is to calculate possible values for priority values recursively:

$$\mathfrak{c}_1 = 1$$
$$\mathfrak{c}_n = 1 + \sum_{k=1}^{n-1} \mathfrak{c}_k$$

  In this way, the constraint of n-th importance receives a higher weight than any arbitrary sum – even the sum of all – weights of less important constraints, thus making it the constraint to choose within the optimization process in case of any occurring conflicts. As it is easily shown per induction, this recursion leads to function $\mathfrak{P}$ (Figure 5.15), which, when receiving integer value n, provides an – in this manner – appropriate value for the soft constraint of n-th importance:

$$\mathfrak{P} : \mathbb{N} \longrightarrow \mathbb{N}$$
$$\mathfrak{P}(n) = 2^{n-1}$$

  Proof:

  IA: $\mathfrak{c}_1 = 1 = 2^{1-1} = \mathfrak{P}(1)$

  IV: $\mathfrak{c}_\mathfrak{n} = 1 + \sum_{k=1}^{n-1} \mathfrak{c}_k = 2^{n-1} = \mathfrak{P}(n)$

  IS: $\mathfrak{c}_{n+1} = 1 + \sum_{k=1}^{n} \mathfrak{c}_k = 1 + \sum_{k=1}^{n-1} \mathfrak{c}_k + \mathfrak{c}_n \overset{IV}{=} 2^{n-1} + 2^{n-1} = 2^n = \mathfrak{P}(n+1)$ □

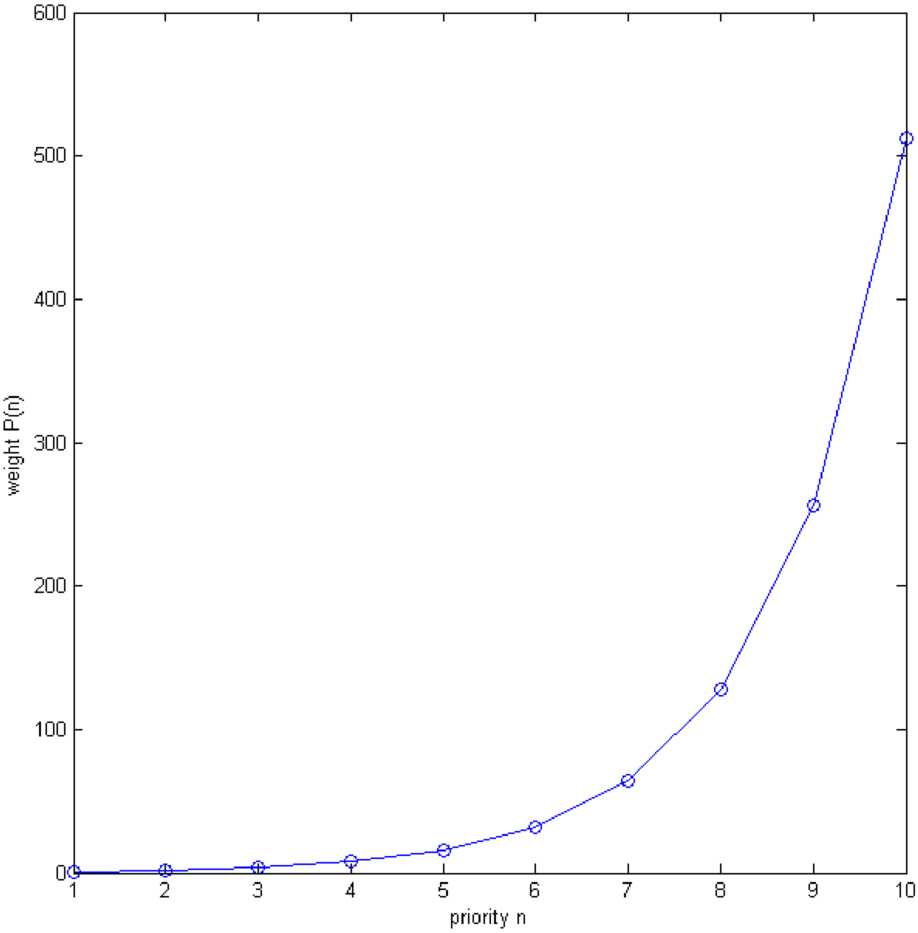

FIGURE 5.15: Calculation of priority values – restriction 1

- On the other hand, the second aspect to consider arises from the sum over the boolean aid variables u_k added up with the sum of the priority values c_i^n of all respected soft constraints in objective function O5. Since $c_i^n \in \{0, 1\}$ and therefore $\sum_{n=1}^{|\mathcal{C}|} c_i^n \mathsf{c}_i^n \leq \sum_{n=1}^{|\mathcal{C}|} \mathsf{c}_i^n$, the sum over the priority values must not dominate over the data-set-related boolean variable u_q, i.e. the value 1 for every $q = 1, \ldots, |\mathcal{U}|$. Therefore it is necessary to restrict priority values c_i^n to the condition $\sum_{n=1}^{|\mathcal{C}|} \mathsf{c}_i^n < 1$, since otherwise dominating sum of the priority values might lead to a mistaken set to 0 of boolean aid variables u_k in order to achieve maximality of objective function.

The combination of these two significant aspects of priority value consideration leads to a way of calculating appropriate values for priority values, which can then be matched to the specific soft constraints as desired.

Chapter 6

Experimental Evaluation

As mentioned in the previous problem-describing and -deriving sections, the aim was to reformulate the former, naive approach of finding suitable settable parameters by browsing through a given, discrete list of amplify factors, choosing for every channel the first appearing value satisfying the hard and soft constraints respectively, without being capable of finding a probably better solution by considering both channels simultaneously (Figure 5.13). This former approach indeed lead to good results in practice, as it, despite its simplicity, turned out to provide usable solutions, which were judged as satisfying outcome by most of the test users (6.4). However, due to the nature of *Ottobock*'s Guided APS-Software project being a six-month prototype approach only, the executing algorithm has been implemented in the form of a naive, enumerative approach, making it on the one hand very inefficient, and on the other hand inflexible for generalizing considerations. Therefore, in order to equip given idea with the flexibility needed for good implementation and eventual changes towards a more general, or enhanced system, it was necessary to formulate given problem in mathematically correct terms, respecting the structure required in order to be able to use commercial solvers.

To receive a small insight into the possible advantages of developing given ideas to more generalized approaches, it was also interesting to check, whether a generalization such as suggested in Chapter 5.3 leads to even better results than the simple discrete and enumerative implementation. In order to be able to compare Guided APS-Software with an implementation using newly formulated constraints and considerations in a sensible way, chosen generalization was confined to the idea of extending the valid amplify factor values to a continuous range. Generalization towards multiple data sets as described in Section 5.3.1.5 is not part of following experiment for reasons of clarity, but can indeed be implemented easily by only expanding constraints and objective function by upper described terms.

So given problem was formulated as a generally usable and easily extendable optimization program, additionally granting more flexibility by expanding the previously small set of permitted values to a continuous range, and these more advanced considerations derived in Section 5.3.1.6 were implemented in order to detect equality or noteworthy differences in the outcome of the two different approaches. This experiment gave indication of whether or not making the amplify factors be continuous decision variables and additionally changing over to a properly formulated optimization problem being capable of being solved in commercial ways is worth the effort. So the constraints to the user data, which have been formulated for the naive approach of the Guided APS-Software, have been reformulated to the form of a MIP as explained in Section 3.2, which additionally made a usage of commercially available MIP solvers possible.

6.1 Collected Data

The basis of all calculations is the collection of user data, which was conducted by a graphical user interface (*GUI*) implemented in the Guided APS-Software (Section 2.2). In this thesis' experimental scenario, a user was asked to perform all required movements (Section 5.2.1) one after the other. Every movement was repeated five times, in order to be able to calculate a sufficient average representative of each movement. The EMG signal resulting from the user's movement was measured by electrodes, appropriately placed at the user's arm, and stored for further processing. In this way, 30 different data sets from amputated and non-amputated test-users were collected and provided information in order to calculate their individual parameters (Section 5.2.2, Table 6.1), which on the one hand were used by Guided APS-Software in order to seek for a solution for every test user, calculated by the enumerative algorithm, on the other hand were used to postulate required constraints in order to formulate the optimization program of the second, generalized approach.

6.2 Tools

As Section 2.2 showed, the Guided APS-Software provides a GUI, leading both the orthopedic technician and the user through a process of data recording in order to be able to use collected information for further parameter calculation for both approaches, so that, in case of feasibility, a satisfying adjustment can be adapted to the user's treatment. This software was implemented in **C#**, collected data was serialized in order to provide an efficient way to store, transport and get access to the information. Table 6.1 lists the coefficients appearing in the inequalities of the set-up optimization program and gives explanation about their origin within the user's collected data.

Abbr.	**Representation**	**Used Data**	**Range of Indices**
S_i	Representative value for user's observed Strong signal	User's Strong signal	$i = 1, 2$
NM_i	Representative value for user's observed No-Movement signal	User's No-Movement signal	$i = 1, 2$
$x_{\vec{\mathbf{x}}_\mathrm{I}(f_l(\mathrm{I}_i)),(f_l(\mathrm{FM}_i))}$	Calculated Impulse border	User's Impulse signal	$i = 1, 2, l = 1, \ldots, 16$
$x_{\mathbf{x}_\mathrm{SL}(f_l(\mathrm{I}_i))}$	Calculated signal length	User's Impulse signal	$i = 1, 2, l = 1, \ldots, 16$

TABLE 6.1: Representative values of collected user data

The chosen tool for the search for a solution of given task by the second, generalized approach formulated as optimization program is **Gurobi** *version* 7.0, a commercial optimization solver, which supports a big variety of programming and modeling languages [28]. In this thesis, the transportation of data, collected by the *C#*-implemented Guided APS-Software, to Gurobi's provided solvers, was chosen to be **MATLAB** *version R*2013*b*, which offers a practical way to set up even expansive systems of constraints by making use of its concept of matrix structure [20]. Gurobi offers implemented MATLAB functions, which accept appropriately set-up systems of constraints in matrix format and provide different optimization methods in order to examine given problem.
By making use of these three tools, the experiment of executing values for settable parameters within the process of prosthesis adjustment as a generalized, linearized approach was finally realized.

6.3 Set-up

The tested prosthesis program was chosen to be the one using the Impulse switch method of channel 1 in order to switch between the opposition and the lateral grasp, and the 4-Channel switch method of channel 1 and channel 2 in order to rotate clock- and counterclockwise, just as described in constraint deriving example in Section 5.3. Due to the lack of some constraints contained in the upper derived formulation within the implementation of the Guided APS-Software prototype, also the system postulated for the experimental, continuous approach was reduced in order to make a reasonable comparison of the two outcomes possible.

Name	Function	Range	#
F_i^l	Actual Amplify Factor	$[f_{l-1}, f_l]$	32
$\tilde{F}_i^l$	Aid Substitution Variable	$\mathbb{R}$	32
c_i^k	Hard-/Softconstraint Indicator	$\{0,1\}$	34
r_i^l	Aid Substitution Variable	$\{0,1\}$	32

TABLE 6.2: Decision variables of experimental approach

With decision variables listed in Table 6.2 and the coefficients

$$K_{SL_i}^{r} := \left(\frac{f_{l+1} \cdot x_{\mathrm{x_{SL}}(f_l(\mathrm{I}_i))} - f_l \cdot x_{\mathrm{x_{SL}}(f_{l+1}(\mathrm{I}_i))}}{f_{l+1} - f_l} \right)$$

$$K_{SL_i}^{\tilde{F}} := \left(\frac{x_{\mathrm{x_{SL}}(f_{l+1}(\mathrm{I}_i))} - x_{\mathrm{x_{SL}}(f_l(\mathrm{I}_i))}}{f_{l+1} - f_l} \right)$$

$$K_{I_i}^{r} := \left(\frac{f_{l+1} \cdot x_{\vec{\mathrm{x}}_{\mathrm{I}}(f_l(\mathrm{I}_i)),(f_l(\mathrm{FM}_i))} - f_l \cdot x_{\vec{\mathrm{x}}_{\mathrm{I}}(f_{l+1}(\mathrm{I}_i)),(f_{l+1}(\mathrm{FM}_i))}}{f_{l+1} - f_l} \right)$$

$$K_{I_i}^{\tilde{F}} := \left(\frac{x_{\vec{\mathrm{x}}_{\mathrm{I}}(f_{l+1}(\mathrm{I}_i)),(f_{l+1}(\mathrm{FM}_i))} - x_{\vec{\mathrm{x}}_{\mathrm{I}}(f_l(\mathrm{I}_i)),(f_l(\mathrm{FM}_i))}}{f_{l+1} - f_l} \right)$$

the involved constraints and corresponding objective function of the optimization program turned to following form:

$$\max_{c_i^n} \sum_{n=1}^{|\mathcal{C}|} (c_i^n \mathsf{c}_i^n) \tag{O5}$$

subject to:

$$\tilde{\mathbf{M}} \cdot (1 - c_i^1) + \sum_{l=1}^{|\mathcal{V}_A|} \tilde{F}_i^l \cdot \mathrm{S}_i \geq \mathrm{TOL}_{\mathrm{S}}^{\mathrm{min}} \qquad \#1 \qquad (\mathrm{C1}_1)$$

$$-\tilde{\mathbf{M}} \cdot (1 - c_i^2) + \sum_{l=1}^{|\mathcal{V}_A|} \tilde{F}_i^l \cdot \mathrm{S}_i \leq \mathrm{TOL}_{\mathrm{S}}^{\mathrm{max}} \qquad \#2 \qquad (\mathrm{C2}_1)$$

$$-\tilde{\mathbf{M}} \cdot (1 - c_i^3) + \sum_{l=1}^{|\mathcal{V}_A|} \tilde{F}_i^l \cdot \mathrm{NM}_i \leq \mathrm{TOL}_{\mathrm{NM}} \qquad \#3 \qquad (\mathrm{C4}_1)$$

$$-\tilde{\mathbf{M}} \cdot (1 - c_i^4) + \sum_{l=1}^{|\mathcal{V}_A|-1} K_{SL_i}^r r_i^l + \sum_{l=1}^{|\mathcal{V}_A|-1} K_{SL_i}^{\tilde{F}} \cdot \tilde{F}_i^l \leq \mathrm{TOL}_{\mathrm{SL}}^{\mathrm{max}} \qquad \#4 \qquad (\mathrm{C7}_1)$$

$$\tilde{\mathbf{M}} \cdot (1 - c_i^5) + \sum_{l=1}^{|\mathcal{V}_A|-1} K_{I_i}^r r_i^l + \sum_{l=1}^{|\mathcal{V}_A|-1} K_{I_i}^{\tilde{F}} \cdot \tilde{F}_i^l \geq \mathrm{TOL}_{\mathrm{I}}^{\mathrm{min}} \qquad \#5 \qquad (\mathrm{C8}_1)$$

$$-\tilde{\mathbf{M}} \cdot (1 - c_i^6) + \sum_{l=1}^{|\mathcal{V}_A|} \tilde{F}_i^l \cdot \mathrm{S}_i \leq \mathrm{TOL}_{\mathrm{S}}^{\mathrm{max*}} \qquad \#8 \qquad (\mathrm{C12}_1)$$

$$\sum_{c_i^n \in \mathcal{C}_{\mathrm{hard}}} c_i^n \geq |\mathcal{C}_{\mathrm{hard}}| \qquad \#9 \qquad (\mathrm{C13}_1)$$

$$\sum_{c_i^n \in \mathcal{C}_{\mathrm{hard}}} c_i^n \leq |\mathcal{C}_{\mathrm{hard}}| \qquad \#10 \qquad (\mathrm{C14}_1)$$

$$\tilde{\mathbf{M}} \cdot (1 - c_i^{15}) + \tilde{F}_i^l \geq 0 \qquad \#11 \qquad (\mathrm{C15}_l)$$

$$-\tilde{\mathbf{M}} \cdot (1 - c_i^{16}) + \tilde{F}_i^l \leq f_{l+1} \cdot r_i^l \qquad \#12 \qquad (\mathrm{C16}_l)$$

$$-\tilde{\mathbf{M}} \cdot (1 - c_i^{17}) + \tilde{F}_i^l \leq F_i^l \qquad \#13 \qquad (\mathrm{C17}_l)$$

$$\tilde{\mathbf{M}} \cdot (1 - c_i^{18}) + \tilde{F}_i^l \geq f_{l+1} \cdot (r_i^l - 1) + F_i^l \qquad \#14 \qquad (\mathrm{C18}_l)$$

$$\tilde{\mathbf{M}} \cdot (1 - c_i^{19}) + \sum_{l=1}^{|\mathcal{V}_A|-1} r_i^l \geq 1 \qquad \#15 \qquad (\mathrm{C19}_1)$$

$$-\tilde{\mathbf{M}} \cdot (1 - c_i^{20}) + \sum_{l=1}^{|\mathcal{V}_A|-1} r_i^l \leq 1 \qquad \#16 \qquad (\mathrm{C20}_1)$$

$$c_i^n \in \{0, 1\} \qquad (\mathrm{D2}_n)$$

$$r_i^l \in \{0, 1\} \qquad (\mathrm{D4}_l)$$

$$\tilde{F}_i^l \in \mathbb{R}^+ \qquad (\mathrm{D5}_l)$$

$$F_i^l \in [f_{l-1}, f_l] \qquad \#17 \qquad (\mathrm{D6}_l)$$

with $i = 1, 2$, $l = 1, \ldots, 16$ and $n = 1, \ldots, 17$.

6.3.1 Data Transportation

A simple straight-forward approach of data transportation has been implemented into the $C\#$ prototyped Guided APS-Software, where all required values were written to ordinary text files and stored appropriately, in order to give the part of the experiment implemented in MATLAB easy access on required numbers.

Furthermore, Gurobi provides an interface between MATLAB and its own solvers, making it comfortable to pass on properly formulated constraints, parameters and the objective function to pre-implemented MATLAB-functions. These functions are capable of communicating with Gurobi, make use of its internal solvers and return a solution or information about infeasibility (Figure 6.1).

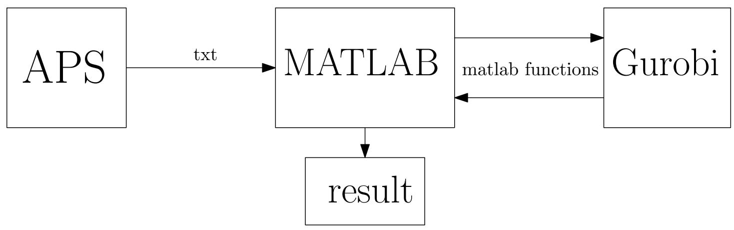

FIGURE 6.1: Data transportation and solution finding

6.3.2 Input Formulation

As already mentioned, version 7.0 of the Gurobi Optimizer provides a pre-implemented interface in order to pass on data in MATLAB format and solve the given optimization program within Gurobi [29] as well as a quick introduction into the usage of latter [26] and additional auxiliary material concerning the usage of provided solution methods [25], [27].

The MATLAB interface requests the implementaion of a matrix of the coefficients of the lefthand-sides and a vector of the righthand-sides of the constraits of given optimization problem. Furthermore, the type of inequality, i.e. *smaller or equal* or *bigger or equal* can be separately defined for every constraint within a string vector, as well as the value type of the decision variables. The coefficients of the objective function, also stored in vector format, finally complete the required input information and a solution of the optimization program can be executed.

Thus, taking the cue from the appearance of the optimization program (3.2), for this experiment's chosen prosthesis program the MIP has the following form:

$$
\begin{array}{r}
\\
\#1\ (C1) \\ \\
\#2\ (C2) \\ \\
\#3\ (C4) \\ \\
\#4\ (C7) \\ \\
\#5\ (C8) \\ \\
(C10) \\ \\ \\
\#6 \\ \\ \\
\#7\ (C11) \\
\vdots \\
\#15\ (C19) \\
\#16\ (C20) \\ \\
(D6) \\ \\ \\
\#17 \\ \\ \\
\end{array}
\begin{array}{c}
\text{Coeff. } F_i^l \qquad\qquad \text{Coeff. } \tilde{F}_i^l \qquad\qquad \text{Coeff. } c_i^k \qquad\qquad \text{Coeff. } r_i^l \\
\begin{pmatrix}
a_1^{\mathrm{C1}_1,F_1^1} & \cdots & a_1^{\mathrm{C1}_1,F_2^{16}} & a_1^{\mathrm{C1}_1,\tilde{F}_1^1} & \cdots & a_1^{\mathrm{C1}_1,\tilde{F}_2^{16}} & a_1^{c_1^1} & \cdots & a_1^{\mathrm{C1}_1,c_2^{17}} & \cdots \\
a_2^{\mathrm{C1}_1,F_1^1} & \cdots & a_2^{\mathrm{C1}_1,F_2^{16}} & a_2^{\mathrm{C1}_1,\tilde{F}_1^1} & \cdots & a_2^{\mathrm{C1}_1,\tilde{F}_2^{16}} & a_2^{\mathrm{C1}_1,c_1^1} & \cdots & a_2^{\mathrm{C1}_1,c_2^{17}} & \cdots \\
a_1^{\mathrm{C2}_1,F_1^1} & \cdots & a_1^{\mathrm{C2}_1,F_2^{16}} & a_1^{\mathrm{C2}_1,\tilde{F}_1^1} & \cdots & a_1^{\mathrm{C2}_1,\tilde{F}_2^{16}} & a_1^{\mathrm{C2}_1,c_1^1} & \cdots & a_1^{\mathrm{C2}_1,c_2^{17}} & \cdots \\
a_2^{\mathrm{C2}_1,F_1^1} & \cdots & a_2^{\mathrm{C2}_1,F_2^{16}} & a_2^{\mathrm{C2}_1,\tilde{F}_1^1} & \cdots & a_2^{\mathrm{C2}_1,\tilde{F}_2^{16}} & a_2^{\mathrm{C2}_1,c_1^1} & \cdots & a_2^{\mathrm{C2}_1,c_2^{17}} & \cdots \\
a_1^{\mathrm{C4}_1,F_1^1} & \cdots & a_1^{\mathrm{C4}_1,F_2^{16}} & a_1^{\mathrm{C4}_1,\tilde{F}_1^1} & \cdots & a_1^{\mathrm{C4}_1,\tilde{F}_2^{16}} & a_1^{\mathrm{C4}_1,c_1^1} & \cdots & a_1^{\mathrm{C4}_1,c_2^{17}} & \cdots \\
a_2^{\mathrm{C4}_1,F_1^1} & \cdots & a_2^{\mathrm{C4}_1,F_2^{16}} & a_2^{\mathrm{C4}_1,\tilde{F}_1^1} & \cdots & a_2^{\mathrm{C4}_1,\tilde{F}_2^{16}} & a_2^{\mathrm{C4}_1,c_1^1} & \cdots & a_2^{\mathrm{C4}_1,c_2^{17}} & \cdots \\
a_1^{\mathrm{C7}_1,F_1^1} & \cdots & a_1^{\mathrm{C7}_1,F_2^{16}} & a_1^{\mathrm{C7}_1,\tilde{F}_1^1} & \cdots & a_1^{\mathrm{C7}_1,\tilde{F}_2^{16}} & a_1^{\mathrm{C7}_1,c_1^1} & \cdots & a_1^{\mathrm{C7}_1,c_2^{17}} & \cdots \\
a_2^{\mathrm{C7}_1,F_1^1} & \cdots & a_2^{\mathrm{C7}_1,F_2^{16}} & a_2^{\mathrm{C7}_1,\tilde{F}_1^1} & \cdots & a_2^{\mathrm{C7}_1,\tilde{F}_2^{16}} & a_2^{\mathrm{C7}_1,c_1^1} & \cdots & a_2^{\mathrm{C7}_1,c_2^{17}} & \cdots \\
a_1^{\mathrm{C8}_1,F_1^1} & \cdots & a_1^{\mathrm{C8}_1,F_2^{16}} & a_1^{\mathrm{C8}_1,\tilde{F}_1^1} & \cdots & a_1^{\mathrm{C8}_1,\tilde{F}_2^{16}} & a_1^{\mathrm{C8}_1,c_1^1} & \cdots & a_1^{\mathrm{C8}_1,c_2^{17}} & \cdots \\
a_2^{\mathrm{C8}_1,F_1^1} & \cdots & a_2^{\mathrm{C8}_1,F_2^{16}} & a_2^{\mathrm{C8}_1,\tilde{F}_1^1} & \cdots & a_2^{\mathrm{C8}_1,\tilde{F}_2^{16}} & a_2^{\mathrm{C8}_1,c_1^1} & \cdots & a_2^{\mathrm{C8}_1,c_2^{17}} & \cdots \\
a_1^{\mathrm{C10}_1,F_1^1} & \cdots & a_1^{\mathrm{C10}_1,F_2^{16}} & a_1^{\mathrm{C10}_1,\tilde{F}_1^1} & \cdots & a_1^{\mathrm{C10}_1,\tilde{F}_2^{16}} & a_1^{\mathrm{C10}_1,c_1^1} & \cdots & a_1^{\mathrm{C10}_1,c_2^{17}} & \cdots \\
\vdots & & & & & & & & & \\
a_1^{\mathrm{C10}_{16},F_1^1} & \cdots & a_1^{\mathrm{C10}_{16},F_2^{16}} & a_1^{\mathrm{C10}_{16},\tilde{F}_1^1} & \cdots & a_1^{\mathrm{C10}_{16},\tilde{F}_2^{16}} & a_1^{\mathrm{C10}_{16},c_1^1} & \cdots & a_1^{\mathrm{C10}_{16},c_2^{17}} & \ldots \\
a_2^{\mathrm{C10}_1,F_1^1} & \cdots & a_2^{\mathrm{C10}_1,F_2^{16}} & a_2^{\mathrm{C10}_1,\tilde{F}_1^1} & \cdots & a_2^{\mathrm{C10}_1,\tilde{F}_2^{16}} & a_2^{\mathrm{C10}_1,c_1^1} & \cdots & a_2^{\mathrm{C10}_1,c_2^{17}} & \cdots \\
\vdots & & & & & & & & & \\
a_2^{\mathrm{C10}_{16},F_1^1} & \cdots & a_2^{\mathrm{C10}_{16},F_2^{16}} & a_2^{\mathrm{C10}_{16},\tilde{F}_1^1} & \cdots & a_2^{\mathrm{C10}_{16},\tilde{F}_2^{16}} & a_2^{\mathrm{C10}_{16},c_1^1} & \cdots & a_2^{\mathrm{C10}_{16},c_2^{17}} & \cdots \\
\vdots & & & & & & & & & \\
\vdots & & & & & & & & & \\
\vdots & & & & & & & & & \\
a_1^{\mathrm{C20}_1,F_1^1} & \cdots & a_1^{\mathrm{C20}_1,F_2^{16}} & a_1^{\mathrm{C20}_1,\tilde{F}_1^1} & \cdots & a_1^{\mathrm{C20}_1,\tilde{F}_2^{16}} & a_1^{\mathrm{C20}_1,c_1^1} & \cdots & a_1^{\mathrm{C20}_1,c_2^{17}} & \cdots \\
a_2^{\mathrm{C20}_1,F_1^1} & \cdots & a_2^{\mathrm{C20}_1,F_2^{16}} & a_2^{\mathrm{C20}_1,\tilde{F}_1^1} & \cdots & a_2^{\mathrm{C20}_1,\tilde{F}_2^{16}} & a_2^{\mathrm{C20}_1,c_1^1} & \cdots & a_2^{\mathrm{C20}_1,c_2^{17}} & \cdots \\
a_1^{\mathrm{D6}_1,F_1^1} & \cdots & a_1^{\mathrm{D6}_1,F_2^{16}} & a_1^{\mathrm{D6}_1,\tilde{F}_1^1} & \cdots & a_1^{\mathrm{D6}_1,\tilde{F}_2^{16}} & a_1^{\mathrm{D6}_1,c_1^1} & \cdots & a_1^{\mathrm{D6}_1,c_2^{17}} & \cdots \\
\vdots & & & & & & & & & \\
a_1^{\mathrm{D6}_{16},F_1^1} & \cdots & a_1^{\mathrm{D6}_{16},F_2^{16}} & a_1^{\mathrm{D6}_{16},\tilde{F}_1^1} & \cdots & a_1^{\mathrm{D6}_{16},\tilde{F}_2^{16}} & a_1^{\mathrm{D6}_{16},c_1^1} & \cdots & a_1^{\mathrm{D6}_{16},c_2^{17}} & \cdots \\
a_2^{\mathrm{D6}_1,F_1^1} & \cdots & a_2^{\mathrm{D6}_1,F_2^{16}} & a_2^{\mathrm{D6}_1,\tilde{F}_1^1} & \cdots & a_2^{\mathrm{D6}_1,\tilde{F}_2^{16}} & a_2^{\mathrm{D6}_1,c_1^1} & \cdots & a_2^{\mathrm{D6}_1,c_2^{17}} & \cdots \\
\vdots & & & & & & & & & \\
a_2^{\mathrm{D6}_{16},F_1^1} & \cdots & a_2^{\mathrm{D6}_{16},F_2^{16}} & a_2^{\mathrm{D6}_{16},\tilde{F}_1^1} & \cdots & a_2^{\mathrm{D6}_{16},\tilde{F}_2^{16}} & a_2^{\mathrm{D6}_{16},c_1^1} & \cdots & a_2^{\mathrm{D6}_{16},c_2^{17}} & \cdots
\end{pmatrix}
\end{array}
\cdot
\begin{pmatrix}
F_1^1 \\ F_1^2 \\ \vdots \\ F_1^{15} \\ F_1^{16} \\ F_2^1 \\ F_2^2 \\ \vdots \\ F_2^{15} \\ F_2^{16} \\
\tilde{F}_1^1 \\ \tilde{F}_1^2 \\ \vdots \\ \tilde{F}_1^{15} \\ \tilde{F}_1^{16} \\ \tilde{F}_2^1 \\ \tilde{F}_2^2 \\ \vdots \\ \tilde{F}_2^{15} \\ \tilde{F}_2^{16} \\
c_1^1 \\ c_1^2 \\ \vdots \\ c_1^{16} \\ c_1^{17} \\ c_2^1 \\ c_2^2 \\ \vdots \\ c_2^{16} \\ c_2^{17} \\
r_1^1 \\ r_1^2 \\ \vdots \\ r_1^{15} \\ r_1^{16} \\ r_2^1 \\ r_2^2 \\ \vdots \\ r_2^{15} \\ r_2^{16}
\end{pmatrix}
\begin{array}{c} \leq \\ \text{resp.} \\ \geq \end{array}
\begin{pmatrix}
\mathrm{rhs}_{\mathrm{C1}} \\ \mathrm{rhs}_{\mathrm{C1}} \\ \mathrm{rhs}_{\mathrm{C2}} \\ \mathrm{rhs}_{\mathrm{C2}} \\ \mathrm{rhs}_{\mathrm{C4}} \\ \mathrm{rhs}_{\mathrm{C4}} \\ \mathrm{rhs}_{\mathrm{C7}} \\ \mathrm{rhs}_{\mathrm{C7}} \\ \mathrm{rhs}_{\mathrm{C8}} \\ \mathrm{rhs}_{\mathrm{C8}} \\
\mathrm{rhs}_{\mathrm{C10}} \\ \vdots \\ \mathrm{rhs}_{\mathrm{C10}} \\ \mathrm{rhs}_{\mathrm{C11}} \\ \vdots \\ \mathrm{rhs}_{\mathrm{C11}} \\
\mathrm{rhs}_{\mathrm{C12}} \\ \mathrm{rhs}_{\mathrm{C12}} \\ \mathrm{rhs}_{\mathrm{C13}} \\ \mathrm{rhs}_{\mathrm{C13}} \\ \mathrm{rhs}_{\mathrm{C14}} \\ \mathrm{rhs}_{\mathrm{C14}} \\
\mathrm{rhs}_{\mathrm{C15}} \\ \vdots \\ \mathrm{rhs}_{\mathrm{C15}} \\ \mathrm{rhs}_{\mathrm{C16}} \\ \vdots \\ \mathrm{rhs}_{\mathrm{C16}} \\ \mathrm{rhs}_{\mathrm{C17}} \\ \vdots \\ \mathrm{rhs}_{\mathrm{C17}} \\ \mathrm{rhs}_{\mathrm{C18}} \\ \vdots \\ \mathrm{rhs}_{\mathrm{C18}} \\
\mathrm{rhs}_{\mathrm{C19}} \\ \mathrm{rhs}_{\mathrm{C19}} \\ \mathrm{rhs}_{\mathrm{C20}} \\ \mathrm{rhs}_{\mathrm{C20}} \\ \mathrm{rhs}_{\mathrm{D6}} \\ \vdots \\ \mathrm{rhs}_{\mathrm{D6}}
\end{pmatrix}
$$

where $a_i^{\mathrm{C}k_l,x}$ represents the coefficient for decision variable x for channel i within constraint $\mathrm{C}k_l$, $i = 1, 2$, $l = 1, \ldots, 16$. In this form and analogue representation for the objective function in vector form, collected data can appropriately be passed on to given solver and the given optimization problem can be treated.

6.4 Results

As already emphasized in Section 2.2, the approach, elaborated during the period of research of six months within the framework of the Guided APS-Software, the algorithm used in order to determine feasibility respectively appropriate parameters for the final prosthesis adjustment is the one of naive enumeration, meaning that the agent starts at the highest available value, browsing downwards, picking the first one to provide desired circumstances. This inefficient way indeed leads to results, which are capable of satisfying previously defined goals. In order to determine differences in the prosthesis usage with manual and algorithmic adjustment respectively, users being experienced with the use of the *Michelangelo* prosthesis with manual adjustment were questioned before and after a readjustment of the prosthesis with the Guided APS-Software. The ease or difficulties of the different functionalities were rated by the testers and they were asked, whether or not they experienced the algorithmic adjustment as more suitable. As can be seen in Figure 6.2, 71.4% of the test users judged the algorithmically determined adjustment as being suitable, 14.2% didn't experience much difference and and only another 14.2% preferred the manual adjustment they were used to so far. Keeping in mind that these results were achieved by only very first, basic considerations and brute force implementation within the prototyped Guided APS-Software, it yields confidence that these basic ideas are of great potential within the finding of a reliable and satisfactory solution of given problem.

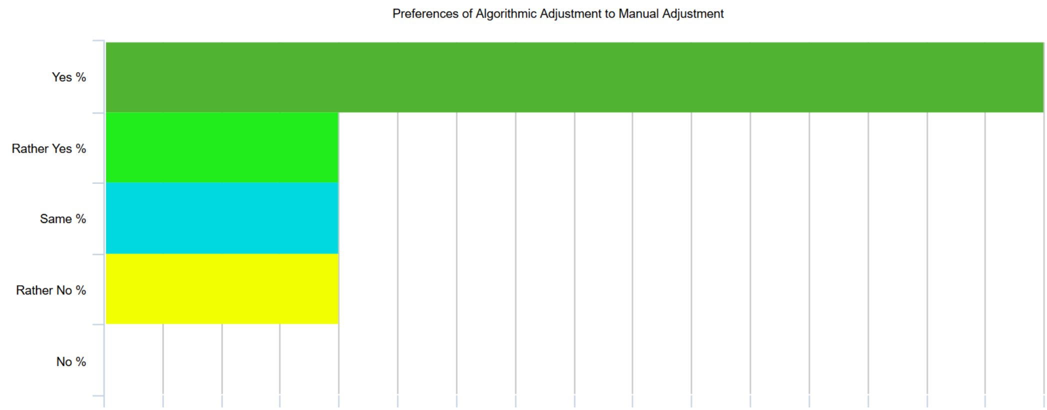

FIGURE 6.2: Comparison APS vs. manual adjustment

Also the amount of time needed for prosthesis adjustment could be diminished significantly. As users experienced manual adjustment as lasting between 30 minutes and three hours, the average time needed in order to go through the guided process of data recording and set calculated parameters afterwards by the Guided APS-Software was approximately 9 minutes.

Nevertheless, a MIP formulation with the additional freedom of letting amplify factors be elements of a continuous interval instead of restricting them to a discrete set of values, as well as the formulation towards an approach, which is capable of putting information into the process of calculation of channel 1 and channel 2 simultaneously, and following trial of determination of a solution with the help of available, commercial solvers turned out to be an approach of even greater advantage.

Table 6.3 shows the outcome of the two different approaches. Amplify factors, calculated with the Guided APS-Software using the naive, enumerating algorithm are listed in column 2, as well as the amplify factors determined by the experimental approach in column 3.
Furthermore, column 4 and 5 display whether or not the soft constraint, which is part of the chosen program for this thesis' experiment, could be included within the process of solving the system by the two different algorithms.

The results indeed show, that in comparison to the former, naive algorithm implemented in the Guided APS-Software, the amplify factors determined by the linear, experimental implementation either lead to the same – thus for given application satisfying – achievements of functionality or were capable of additionally including the one enhancing soft constraint, which is part of the chosen example program to the set of fulfilled constraints, thus deliver a better solution and therefore enhance prosthesis functionality.

User	**APS** Chan. 1/2	**Exp** Chan. 1/2	**APS** Soft Chan. 1/2	**Exp** Soft Chan. 1/2
1	0.5/0.5	0.5/0.25	✓/×	✓/✓
2	0.5/0.5	0.25/0.25	×/×	✓/✓
3	0.75/1	0.25/1	×/✓	✓/✓
4	1.75/3.5	2.0/2.0	×/×	×/✓
5	0.5/2.0	0.5/3.0	✓/×	✓/×
6	−/−	0.25/0.25	×/×	✓/✓
7	1.25/0.75	2.0/0.25	×/×	×/✓
8	−/−	−/−	×/×	×/×
9	0.5/1.0	0.25/1.5	×/×	✓/×
10	2.0/1.5	2.0/2.25	✓/×	✓/×
11	0.5/0.75	0.5/0.5	✓/×	✓/✓
12	0.5/0.5	0.75/0.5	×/✓	✓/✓
13	0.5/0.5	0.5/0.5	✓/✓	✓/✓
14	−/−	−/−	×/×	×/×
15	−/−	−/−	×/×	×/×
16	0.75/0.5	1/0.5	×/✓	✓/✓
17	0.75/0.5	1/0.5	×/✓	✓/✓
18	0.25/0.5	0.25/0.5	✓/✓	✓/✓
19	1.5/0.5	2.25/0.5	×/✓	×/✓
20	−/−	−/−	×/×	×/×
21	−/−	−/−	×/×	×/×
22	0.75/1.25	0.5/1.5	×/×	✓/✓
23	0.75/0.25	0.75/0.5	×/×	×/✓
24	0.75/1.25	0.75/0.5	✓/×	✓/✓
25	0.75/0.75	0.75/0.5	✓/×	✓/✓
26	4.0/4.0	2.25/3.25	×/×	✓/✓
27	0.5/1.25	0.5/0.5	✓/×	✓/✓
28	−/−	0.5/0.5	×/×	✓/✓
29	0.25/0.25	0.5/0.5	×/×	✓/✓
30	0.5/0.25	0.5/0.5	×/×	×/✓

TABLE 6.3: Comparison outcome amplifier Guided APS-Software vs. experimental implementation

Figure 6.3 and 6.4 give an interpretation of calculated numbers and thus impression about the difference in quality of the two algorithms' outcomes. While within the calculation executed by the Guided APS-Software it was only possible to add the soft constraint to a little more than a quarter of all test scenarios, the experimental, reformulated and slightly generalized version of the algorithm already lead to results including the soft constraint in 70% of all scenarios.

Furthermore, as visible in Figure 6.5, 70% of the scenarios themselves could be enhanced by the algorithm which made use of commercial Gurobi solvers. The other 30% of the latter had the same outcome as original, naive approach. For none of them, the Gurobi approach caused worse outcomes than the previous technique.

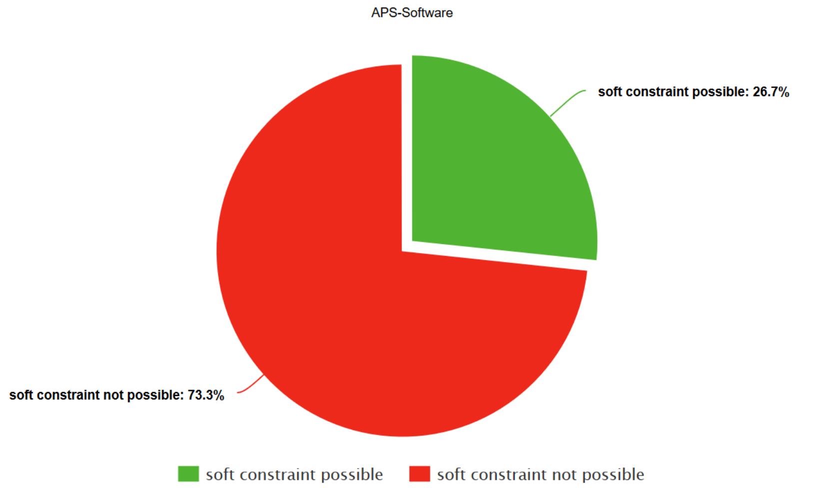

FIGURE 6.3: APS – percentage of inclusions of soft constraint

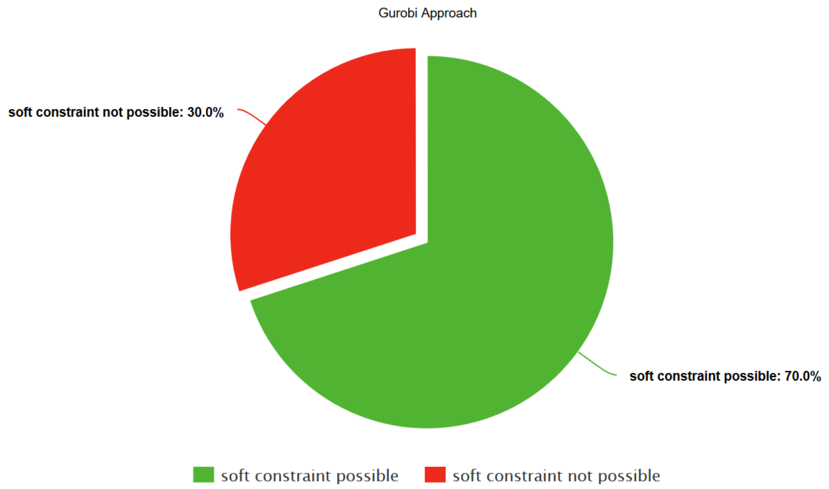

FIGURE 6.4: Gurobi – percentage of inclusions of soft constraint

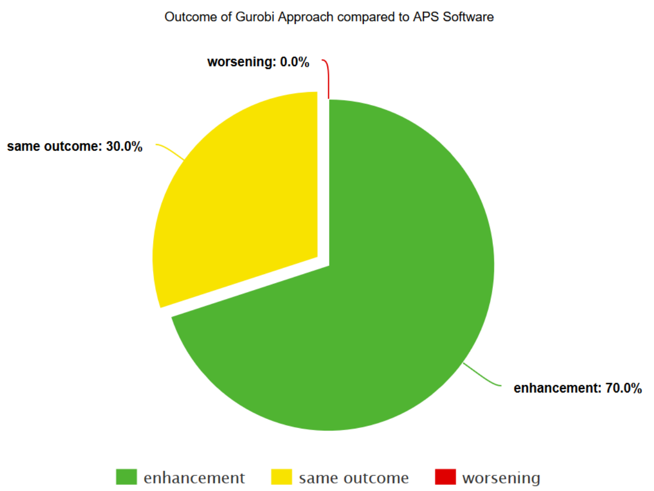

FIGURE 6.5: Comparison APS vs. Gurobi

These results indeed rise the courage for conviction, that turning away from the former, naive but already well-working approach towards a more considered way of implementation is worth the effort, since the differences of the quality of outcome turned out to be significant.

Part III

Conclusion

Chapter 7

Conclusion

The process of prosthesis adjustment is a very sensitive and subtle topic. As the mattering aspect is not only the one that influences the technical process of the device being requested to work in its intended manners, but its way and ease of working indeed has a huge impact on the user and his life and well-being, it is of much importance to take a careful look at this sector of treatment for amputees. As the manual way of adjusting a *Michelangelo* prosthesis turned out to fail in many cases due to a widespread lack of routine and practice of orthopedic technicians caused by the small number of occurring myoelectric treatments, it was sensible to start thinking about alternative ways, which guarantee more reliably a satisfying result for the user.

The outcome of this thesis shows, that the idea of an automated approach, making use of collected user data and determining desired parameters via considerations based on mathematical optimization, and an algorithmic implementation indeed leads to promising new perspectives for proper adjustments of myoelectric prostheses. Even the naive first approach, implemented in the prototype software (Section 2.2), lead to results, which were experienced as enhancements for a majority of tested and subsequently interrogated users. Table 6.3 furthermore grants an insight into the potential of the idea of a well-considered optimization program formulation, since the reformulation towards a MIP and the consequent opportunity of solving latter in reliable and efficient manners achieved an enhancement of solution even despite the simplification of the system for the sake of comparability with the state-of-the-art Guided APS-Software prototype.

These highly positive findings should arise motivation towards a continuation of work and consideration within this field of orthopedic and myoelectric technology and a desire to enhance currently available options. By making use of presented ideas applied to given problem, a huge support within the matter of prosthesis adjustment can be provided to orthopedic technicians.
The approach of mathematical optimization, as it fits well to the structure of the problem and gives it a practical and easily implementable opportunity of being solved with commercial methods, grants a huge flexibility towards generalization. By making use of ideas presented in section 5.3.1.5, it is easy to realize problem solutions also when considering a larger amount of data sets per user, meaning that more information can be used in order to lead to more precise parameter calculations.
Furthermore, due to the flexibility of given problem formulation, the presented solution ideas could even be expanded to be capable of treating prosthesis adjustment for myoelectric prostheses with more than two electrodes. Only an addition of resulting constraints, bound to further electrode channels, have to be added to the

system of constraints, which can be derived analogously to previous presentations. By further development and enhancement of already well-working six months Guided APS-Software project, and introduction to the commercial market, it might be possible to reduce the number of failures in prosthesis adjustment significantly.

As a proper prosthesis treatment after the loss of extremity is not only of technical and practical nature, but has a tremendous impact on the physical and mental state of affected human beings and thus their quality of life, the revealed potential of preceding ideas should indeed be enough encouragement in order to arouse the passion and will to continue putting effort into this topic and lead to a next level of opportunities. In this way, not only a very interesting field of research would receive deserved attention and progress, but also the life of many people having to cope with one of the gravest losses in both the physical and the emotional sense would experience great enhancement.

Bibliography

[1] F. Anderhuber and F. Pera. *Waldeyer Anatomie des Menschen. Lehrbuch und Altas in einem Band*. 19th ed. De Gruyter and Johannes Streicher, 2012.

[2] G. Aumüller, G. Aust, and et. al. *Duale Reihe*. 2nd ed. Thieme, 2007.

[3] R. Baumgartner and P. Botta. *Amputation und Prothesenversorgung*. Thieme, 1997.

[4] B. Biesinger, B. Hu, and G. Raidl. *VU Algorithmics. Part VI: Integer Programming*. Presentation. Vienna University of Technology, Algorithms and Data Structures Group, 2013/2014.

[5] S. Blumentritt and L. Milde. "Exoprothetik". In: *Medizintechnik: Life Science Engineering*. Springer Berlin Heidelberg, 2009.

[6] H. G. Bock and C. Zillober. *Symposium Operations Research (SOR'95). Interior Point Methods and Extrapolation*. Program and Abstract of the Annual Conference of the DGOR, GMÖOR and ÖGOR. Passau, Germany: University of Passau, 1995, p. 39.

[7] J. E. Cheesborough et al. "Targeted Muscle Reinnervation and Advanced Prosthetic Arms". In: *Seminars in Plastic Surgery* (2015), pp. 62–72.

[8] W. Domschke and A. Drexl. *Einführung in Operations Research*. 3rd ed. Springer, 1995.

[9] G. Fitzlaff. "Magische Hände für Amputierte". In: *Myoelektrische Armprothesen*. Enke, 1981.

[10] Gurobi. *Algorithms in Gurobi*. Presentation. Gurobi Optimization, 2016. URL: `http://www.gurobi.com/pdfs/user-events/2016-frankfurt/Die-Algorithmen.pdf` (visited on 08/20/2017).

[11] Gurobi. *Solving Mixed Integer Programs In Practice*. Presentation. Gurobi Optimization, 2017.

[12] B. Hu, G. Raidl, and M. Ruthmair. *VU Algorithmics. Part V: Linear Programming*. Presentation. Vienna University of Technology, Algorithms and Data Structures Group, 2013/2014.

[13] F. Irmay, B. Merzouga, and D. Vettorel. "The Krukenberg procedure: a surgical option for the treatment of double hand amputees in Sierra Leone". In: *The Lancet* (2000), pp. 1072–1075.

[14] M. Jünger et al. *50 Years Of Integer Programming 1958-2008. From the Early Years to the State-of-the-Art*. Springer, 2010.

[15] H. T. Jongen, K. Meer, and E. Triesch. *Optimization Theory*. Springer, 2004.

[16] J. Kallrath. *Gemischt- ganzzahlige Optimierung: Modellierung in der Praxis. Mit Fallstudien aus Chemie, Energiewirtschaft, Paperindustrie, Metallgewerbe, Produktion und Logistik*. 2nd ed. Springer, 2002, 2013.

[17] P. Kampas. "Myoelektrode - optimal eingesetzt". In: *Medizinisch-orthopädische Technik* (2001), pp. 21–26.

[18] R. H. Kwon. *Introduction to Linear Optimization and Extensions with Matlab*. CRC Press, Taylor and Francis Group, 2014.

[19] I. J. Lustig, R. E. Marsten, and D. F. Shanno. “Computational experience with a primal-dual interior point method for linear programming”. In: *Linear Algebra and its Applications* 152 (1991), 191–222.

[20] MathWorks. *Official Homepage MATLAB*. 2018. URL: `https://de.mathworks.com/` (visited on 08/21/2017).

[21] B. A. McCarl and T. H. Spreen. *Applied Mathematical Programming Using Algebraic Systems*. Texas A&M University, 2011.

[22] S. Mehrotra. “On the Implementation of a Primal-Dual Interior Point Method”. In: *Journal on Optimization* 2 (1992), 575–601.

[23] H. G. Näder. *Otto Bock Prothesen-Kompendium. Prothesen für die obere Extremität*. Ed. by L. Milde. 2nd ed. Hans Georg Näder, 2011.

[24] G. L. Nemhauser and L. A. Wolsey. *Integer And Combinatorial Optimization*. Wiley-Interscience, 1988.

[25] G. OPTIMIZATION. *Gurobi Optimizer Example Tour*. 2016. URL: `file:///C:/gurobi701/win64/docs/examples/examples.html` (visited on 08/19/2017).

[26] G. OPTIMIZATION. *Gurobi Optimizer Quick Start Guide - Windows*. 2016. URL: `file:///C:/gurobi701/win64/docs/quickstart/quickstart.html` (visited on 08/19/2017).

[27] G. OPTIMIZATION. *Gurobi Optimizer Reference Manual*. 2016. URL: `file:///C:/gurobi701/win64/docs/refman/refman.html` (visited on 08/19/2017).

[28] G. OPTIMIZATION. *Official Homepage GUROBI OPTIMIZATION*. 2018. URL: `http://www.gurobi.com/index` (visited on 08/20/2017).

[29] G. OPTIMIZATION. *Version 7.0 of the Gurobi Optimizer*. 2016. URL: `file:///C:/gurobi701/win64/ReleaseNotes.html` (visited on 08/21/2017).

[30] F. Paulsen and J. Waschke. *Sobotta. Atlas der Anatomie des Menschen*. Allgemeine Anatomie und Bewegungsapparat. 23rd ed. Elsevier Urban & Fischer, 2012.

[31] H. Pfützner. *Angewandte Biophysik*. 2nd ed. SpringerWienNewYork, 2012.

[32] S. Salminger et al. “Prothetische Rekonstruktion hoher Amputationen der oberen Extremität”. In: *Der Orthopäde* (2015), pp. 413–418.

[33] J. C. Smith and Z. C. Taskin. *A Tutorial Guide To Mixed-Integer Programming. Models and Solution Techniques*. Guide. Department of Industrial and Systems Engineering, University of Florida, 2007.

[34] G. Täger and D. Nast-Kolb. “Amputationen und Prothesenversorgung der oberen Extremität”. In: *Der Unfallchirurg. Organ der Deutschen Gesellschaft für Unfallchirurgie* (2000).

[35] M. A. Treiber. *Optimization for Computer Vision. An Introduction to Core Concepts and Methods*. Springer, 2013.

[36] T. M. of Upper Extremity Amputation Rehabilitation Working Group. *VA/DoD Clinical Practice Guideline for the Management of Upper Extremity Amputation Rehabilitation*. Clinical Practice Guideline. Department of Veterans Affairs and Department of Defense, 2013.

[37] R. J. Vanderbei. *Linear Programming. Foundations and Extensions*. 2nd ed. Kluwer, 2001.

[38] L. A. Wolsey. *Integer Programming*. Wiley-Interscience, 1998.