Control framework for muscle-driven systems: Exploiting bi-articular muscles in antagonistic setups to reduce control complexity and solve the muscle redundancy problem

Von der Fakultät 2: Bau- und Umweltwissenschaften der Universität Stuttgart zur Erlangung der Würde eines Doktor-Ingenieurs (Dr.-Ing.) genehmigte Abhandlung

Vorgelegt von

Simon Wolfen

aus Konstanz

Hauptberichter:	Prof. Dr. Syn Schmitt
Mitberichter	Prof. Dr. Ivan Vujaklija

Tag der mündlichen Prüfung:	04.08.2023

Institut für Modellierung und Simulation biomechanischer Systeme der Universitär Stuttgart

CBB-002-2022

2022

Contents

0 Preface **4**

0.1 Zusammenfassung (deutsch) 5

0.2 Declaration of honor 6

0.3 Structure 7

0.4 Point of view 7

0.5 Math notation 9

0.6 Word definition and delimitation 11

0.6.1 MTC versus MSU 11

0.6.2 System model versus controller model 11

1 Introduction **12**

1.1 Motivation and scientific questions 12

1.2 Bio-robotic Test Bed 22

1.2.1 Ataro 22

1.2.2 Stuttgart Humanoid 1 31
1.2.3 Main Program and Parameters: Implementation 37
1.2.4 Length Sensor Processing 46
1.2.5 Force Sensor Processing 49
1.2.6 Joint Angle Sensor Processing 52
1.2.7 Pressure Sensor Processing 54
1.2.8 Muscle Output Implementation 57

2 Muscle Spring Unit 59

2.1 Mechanical design 59
2.2 Rating of the spring stiffness 61
2.3 Active and passive characteristics 63
2.4 AAS geometry of the SH1 upper ankle 65
2.5 Kinetics of the MSUs in an AAS 68
2.5.1 Force law of single MSUs 68
2.5.2 Resulting torques and variable stiffness in the SH1 ankle 69

3 Controller 73

3.1 Closed Loop Controller 73
3.1.1 Parameter determination 75
3.1.2 Closed-Loop Controller Implementation 87

3.2 Open-Loop Controller 90

3.2.1 Equilibrium Points 90

3.2.2 Open-Loop Performance 92

3.2.3 Open-Loop Controller Implementation 93

3.3 Hybrid Controller 102

3.3.1 Experiment execution 102

3.3.2 Hybrid Control Performance 104

4 Discussion 110

4.1 A2: Muscles in Bio-inspired Setups 110

4.2 A1,A3 and A4: The Control Framework in a Nutshell . . 112

4.3 Parameter Robustness and Stability 115

4.4 Reducing Information and Control 120

4.5 Conclusion 124

4.5.1 Generalization 124

4.5.2 Performance and comparison 126

4.5.3 Critics and outline 131

Chapter 0

Preface

”Jedes Naturgesetz, das sich dem Beobachter offenbart, lässt auf ein höheres, noch unerkanntes schließen.”
- Alexander von Humboldt

”Every law of nature that reveals itself to the observer suggests a higher, still unrecognized.”
- Alexander von Humboldt

0.1 Zusammenfassung (deutsch)

Pneumatische Muskelaktuatoren und deren Verwendung in bio-robotischen Systemen (Muskelgetriebene Roboter) stellen auf Grund ihrer Eigenschaften (Nicht-lineares Verhalten, Hysterese, mono-direktionale Wirkungsrichtung, etc.) eine besondere Herausforderung an einen Regler. Während etablierte Regelstrategien wie Modell-basierte Regelungen oder KI-basierte Regelungen zwar in der Lage sind, muskelgetriebene Robotersysteme mit wenigen Gelenken und wenigen mono-artikularen Muskeln zu handhaben, scheitern diese Ansätze an der Skalierbarkeit (Erweiterung) von weiteren Muskel-Aktuatoren und Gelenken. Besonders bi-artikulare Muskeln in solchen bio-inspirierten Robotersystemen lassen sich mit den etablierten Regelstrategien nur mit einer Steigerung der Komplexität (bei Modell-basierten Regelungen) oder Datenquantität (KI-basierten Regelungen) meistern. Dies liegt daran, dass diese Ansätze zwar Lösungen zu den bekannten „Problemen“ wie Multi-Redundanz von Aktuatoren oder bi-artikulare Muskeln allgemein bieten, jedoch diese generell als Problem definieren, anstatt ihre Eigenschaften zu nutzen.

In dieser Arbeit wird ein alternativer Regelungsansatz vorgestellt, der die nativen Eigenschaften von Muskel-Feder Aktuator Systemen nutzt, welche eine technische Repräsentation des biologischen Muskel Sehnen Komplexes darstellt. Dieser Regelungsansatz besitzt ein mathematisches Regler Modell, ohne jedoch ein mathematisches Aktuator Modell. Durch die Nutzung der systemischen Eigenschaften von bi-artikularen Muskeln in Gelenknetzwerken löst er das Skalierungs- und Parameterproblem. Durch die geometrischen Eigenschaften der Gelenk-Netzwerke können wenige zu bestimmende Parameter auf alle Muskeln des Gesamtsystems angewendet werden. Das vorgestellte Regelsystem stellt daher in einer bio-inspirierten Regler Hierarchie die unterste Regler Schicht dar, jene, welche aus Gelenk Positionssollwerten zugehörige Muskelkommandos generiert. Dieses Regler System wird an Hand von zwei robotischen Systemen untersucht und die Regler Leistung als Zeit in der eine stabil Position mit einer bestimmten resultierenden Regel Abweichung (Genauigkeit) resultiert, definiert.

Dieses Regelsystem fokussiert sich damit darauf, anwendbare robotische Systeme in Echtzeit unter biologischen Gegebenheiten wie

Sensorverzögerung zu Regeln. Das Regelsystem stellt nicht den Anspruch an sich eine Replikation des biologischen Regelsystems zu sein.

0.2 Declaration of honor

I declare that this thesis has been composed solely by myself and that it has not been submitted, in whole or in part, in any previous application for a degree. Except where stated otherwise by reference or acknowledgment, the work presented is entirely my own.

0.3 Structure

This work starts with a quotation from Alexander Humboldt. To understand why I chose this citation, it is important to understand my general point of view, which is explained in the next section. The math notation is introduced afterward. Because this work challenges interdisciplinary questions, algebraic conformity is difficult to apply. The section *"Math notation"* is therefore intended to establish clarity.

The main content of this work starts with chapter 1, which introduces the topic, its scientific questions, and the necessary information regarding the test-beds. The scientific questions are marked with a **Q** and correspond to the answers to these questions, marked with an **A**. The answers are summarized in the chapter 4 *"Discussion"* and are based on the results of the methodology. The method includes chapters 2 *"Muscle Spring Unit"* and 3 *"Controller"*.

0.4 Point of view

I studied electrical engineering as a major and sports science as a minor. When I reflect on the last century, I see that the technical world has changed tremendously, especially in the field of electronics. Based on the thermionic triode invented in 1907, the transistor and many other electronic components were invented in the following years. Today, we are used to communicating with others wherever we are via portable high-performance computers, called smart phones, and we can barely remember what the work environment was like without computers.

Back in the early 20th century, probably nobody could have imagined how life would be today, but just because no one could imagine it does not mean that it was not possible or could not happen. This work was written during the coronavirus pandemic, and also in the beginning 2020, nobody could imagine how a virus could turn everyone's lives upside down. A technical example is the operating limit of optical microscopes. Scientists determined the limit of such microscopes until the early 21st century to be around 200nm. This limit was taught in schools and universities and was considered a given rule until Stefan

Hell and colleages surpassed it with a new technique, for which they recieved the Nobel Prize in 2005.

I try to avoid considering limits as given just because I do not know a method or technology that surpasses that limit. Just because I do not know something does not mean it does not exist, just like the people in the early 20th century who could not imagine how dependent on electronics life would be in the 21st century.

This makes sense for things that humankind may or may not invent in the future. However, it is more difficult to transfer this way of thinking to already existing entities not created by humanity like our body or biology in general. A lack of understanding structures, behavior, or events often leads us to the conclusion that these things play a minor role or can be neglected. The function of the muscle fascia or the power of the mRNA illustrate such misjudgment. Even the parts of our DNA we do not understand are summarized as *crap DNA* as if they were only meaningless amino acids.

In my opinion, this widespread way of viewing things is counterproductive. That is why I introduced this work with the quote from Humboldt, which expresses the infinity of obtainable knowledge if we consider things as possible even if we do not understand them yet. Especially in the field of biology, I am strongly convinced that every structure has one ore more functions, even if we do not understand them yet. The ultimate premise of my point of view is that there is no useless structure in our biological system. Some structures can have evolutionary reasons that can be less useful under given circumstances, but in my viewpoint, there is no such thing as *crap DNA*. That is what I want to illustrate with the sentence *"It is not a bug; It is a feature"*, which can be found in the introduction as well as the conclusion.

This hypothesis can be considered as my base assumption and will most likely never be verified nor falsified since, we do not know what we do not know. Furthermore, I do not claim my point of view to be optimum because there may be existing perspectives that I do not know yet.

0.5 Math notation

Mechanical components are represented by Greek capital letters (e.g., the joints Φ and the muscles M). To prevent confusion, the letters M and N are only used as the Greek capital letters ”my” and ”nu”. The information that sensors provide or actuators receive is represented by its associated Greek lowercase letters accordingly. For example, the joint Φ_1 provides the joint value information φ_1. The information of all n joint angle sensors as a vector is illustrated in bold:

$$\textbf{State variables:} \qquad \boldsymbol{\varphi} = \begin{pmatrix} \varphi_1 \\ \vdots \\ \varphi_n \end{pmatrix}$$

Accordingly, overall actuation is defined as $\boldsymbol{\mu}$, the stimulation vector containing $\mu_1 ... \mu_j$ of all j muscles M.
Matrices containing system parameters are represented by bold Latin capital letters (e.g., the control Matrix $\boldsymbol{P}$).

$$\textbf{Matrices containing parameter:} \qquad \boldsymbol{P} = \begin{bmatrix} p_{11} & \cdots & p_{1k} \\ \vdots & \ddots & \vdots \\ p_{j1} & \cdots & p_{jk} \end{bmatrix}$$

Its elements are the associated Latin lowercase letters p_{jk}. Parameters in general are illustrated in Latin lowercase letters (e.g., the spring rate d). Sets of parameters containing the same type of n-parameters are in bold Latin lowercase letters:

$$\textbf{Parameters:} \qquad \boldsymbol{d} = \begin{pmatrix} d_1 \\ \vdots \\ d_n \end{pmatrix}$$

(e.g., the spring rates of all muscle spring units as a vector). If a set of parameters contains different parameters, these are indicated and referenced to its definition:

Different parameters with reference: $$\boldsymbol{a}^{(5)}_{\boldsymbol{d},\ell} = \begin{pmatrix} d_1 \\ \vdots \\ d_n \\ \ell \end{pmatrix}$$

This is a parameter set with different parameters, which has been introduced earlier as equation (5). Systems and networks are represented by calligraphic Latin capital letters (e.g., the robot Ataro $\mathcal{A}$). Overall information that a system provides or receives is consistently given as:

Different state variables with reference: $$\boldsymbol{\alpha}^{(8)}_{\boldsymbol{\varphi},\boldsymbol{\mu}} = \begin{pmatrix} \varphi_1 \\ \vdots \\ \varphi_n \\ \mu_1 \\ \vdots \\ \mu_n \end{pmatrix}$$

The information is defined in bold Greek lowercase letters with reference to its introduction. Functions and correlations are represented in calligraphic Latin lowercase letters (e.g., the correlation function $\mathscr{w}$):

Functions: $$\mathscr{w}(\boldsymbol{b}) = \frac{\mathscr{l}_2(\boldsymbol{b}) \cdot \mathscr{f}_2(\mu_2, \lambda_2)}{\mathscr{l}_1(\boldsymbol{b} \cdot \mathscr{f}_1(\mu_1, \lambda_1))}$$

It must be mentioned that system variables like the muscle length λ can also be a function due to their dependencies ($\lambda(F, \mu)$). Sets of functions are in bold. Physical units (e.g., the force F or torque T) are, in Latin capital letters; hence, most physical units are established in this way. Exceptions are the acceleration A, which is also in Latin Capital letters in this work to keep the notation consistent.

0.6 Word definition and delimitation

The scope of this work is a technical control solution for bio-robotic applications, but the scientific questions arise from the interdisciplinary field of bio-robotics. In the context of this work, bio-robotics is understood as the intersection between biology and engineering containing many sub fields of both. Therefore, the following definitions are made:

0.6.1 MTC versus MSU

The muscle tendon complex (MTC) describes the biological actuator, while the muscle spring unit (MSU) is its technical pendant.

0.6.2 System model versus controller model

The word ”model” generally describes the model of the system if not specified. Each controller has its controller design or policy, which can be seen as a controller model, but the word ”model” without specification never describes the controller in this work.

Chapter 1

Introduction

1.1 Motivation and scientific questions

High-tech arm robots such as the M-2000iA/2300 (figure 1.1) six-axis industrial robot, uses six electrical drives and less than 100 sensors to perform motion. The control unit of such a robot requires high-tech GHz processor technology to enable control in real time.

Figure 1.1: The M-2000iA/2300 six-axis industrial robot performing motion. *(Source: https://www.fancu.eu)*

The human motor control system uses more than 600 muscles to drive motion of the human body (figure 1.2). To do so, the motor control system receives millions of pieces of sensor information simultaneously and has a hardware structure (neurons) that saturates at 200 Hz.

Figure 1.2: The human body performing motion *(Source: https://www.eurosport.de)*

It may seem obvious that the design of the motor control structure is different from the design of modern industrial controller [18, 16]. The model-based approach of industrial controllers enable extreme precision and speeds that can sometimes even outperform equivalent human motion for basic tasks. On the other hand, this control approach is also the reason for the need for high-performance hardware (e.g., giga hertz processors) for industrial robots due to the high calculation effort of their underlying models. Models also exist for biological morphology, and various biomechanical models have been developed in the last decades to address different motion challenges. Schumacher et al. [97] categorized these models according to the challenges they adress: the stance models [35, 55, 11, 72, 36, 1, 71, 106], the balance models [121, 37, 92, 102, 70], and the swing models [86, 89, 57, 65, 112]. All of these mathematical models share in common, the complexity scale with the number of actuators and joints. As a result, the complexity of calculating the control parameters with a model-based control approach increases as well.

The following example illustrates this issue:

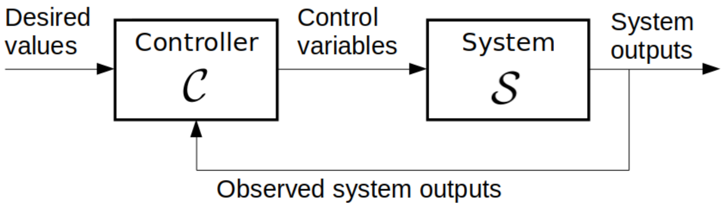


Figure 1.3: General closed-loop control circuit

Figure 1.3 shows the general control scheme for all technical control approaches with $\mathcal{C}$ as the control policy and $\mathcal{S}$ as the controlled system. A model-based controller uses a mathematical model of the controlled system to calculate the control parameters of the controller. That means the controller $\mathcal{C}$ has a control law, and for the system $\mathcal{S}$, a mathematical model is available. Figure 1.3 can be considered as a joint angle control circuit given by:

- φ_{des} as the desired joint angles
- $\mathcal{C}$ as a basic P-controller with a control parameter matrix $\boldsymbol{P}$
- $\boldsymbol{\mu}$ as the control variables generated by the controller
- $\mathcal{S}$ as the controlled system
- $\boldsymbol{\xi}_{all}$ as all the system outputs
- φ_{obs} as the observed joint angle values out of all possible system outputs $\boldsymbol{\xi}_{all}$

Here, we let the control law be a simple P-controller with the control parameter matrix $\boldsymbol{P}$. The size of $\boldsymbol{P}$ is in this case $j \times k$ containing all P-parameters of all muscles j acting on all joints k:

$$\boldsymbol{\mu} = \boldsymbol{P}(\varphi_{des} - \varphi_{obs}) \quad with \quad \varphi_{obs} = \boldsymbol{\mu} \cdot \mathcal{S} \tag{1.1}$$

$\mathcal{S}$ is the model of the system and must be replaced with a system equation of the applied model. In the model based approach, the elements of

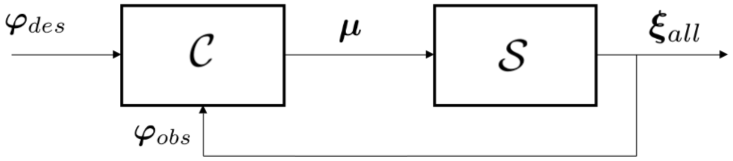


Figure 1.4: Applied control scheme of figure 1.3 for a joint angle controller

the control matrix are defined via the two system equations of (1.1).
The control parameter matrix $\boldsymbol{P}$ increases with the number of joints and actuators, resulting in more parameters to calculate. Each parameter must be calculated by the mathematical description of the applied system model. Therefore, three disadvantages occour in using a model-based approach to solve control tasks of a bio-inspired muscle-driven system:

- The number of parameters to calculate scale with the number of actuators and joints
- The complexity of the applied system model impact the parameter calculation
- The controller performs only as well as the system model used

Bio-inspired muscle-driven systems can be controlled by model-based control approaches but at the cost of higher hardware requirements. Hence, in the model-based approach, these three challenges are solved by increasing calculation power.
Having the premise, that every structure has its function, and that our motor control system uses neurons with firing rates more than one million times slower than our commercial processors, the idea that the morphology is beneficial for control could be the solution. The geometry of the structures [61, 93], the ratios of the muscles [48], and the inter-joint couplings [59] can be considered beneficial for motion control [84]. Furthermore, it can involve a morphological computation, a computation or control reduction performed by the morphology itself in exchange with

its environment [83, 33, 123]. This theory is compliant with the idea of different control levels in the human motor control system [63], where each level solves different tasks of control [69, 114, 44, 91, 108, 48]. The high-level control (HLC) thereby focuses on the superordinate motion control of complex motion [53, 14, 5]. The low-level control (LLC) focuses on the actuation of the single muscles that act within a complex motion.

This work focuses on an applicable LLC framework for muscle-driven robots by exploiting the bio-inspired morphology.

The transfer of the control benefits of biological motor control into the applicable field of robotics is done by observations that lead to assumptions. It is not possible to look into a human motor control system and analyze the controller design, at least not yet. Following this, the superordinate scientific question addressed in this work is:

Q1: Are mathematical models of a complex muscle-driven system mandatory for a controller to perform precise motion, or can the morphology of such systems be exploited to reduce the control complexity?

Different approaches such as fuzzy control [2], AI-based control [24], or a mixture of both [117, 3] already exist to control muscle-driven systems using ”learned” experience to control it. The controller of such systems must be trained to learn the system input and output behavior, which can be considered as experience. Therefore, these control approaches do not need a mathematical model of the controlled system. On the other hand, the downside of such systems are the long training time, and even minor changes to the system can negate the obtained experience, and the training must be repeated. In general, experience-based control approaches (AI and fuzzy) solve control complexity by data quantity. Even if the underlying control policy perfectly fits the control challenge, an increase in complexity by adding more joints or muscles can tremendously increase the training time and data needed. Perhaps, the most significant disadvantage of experience based control approaches is the dependency on the quality of the training data. If a system is not trained with a special situation, which means the quality of the training data does not reflect the situation, it is uncertain how the system will react. For this reason, AI is still excluded in many fields of products such as medical devices directly interacting with humans [116, 32]. Nevertheless, AI-based approaches do not need a model of the system. They simply learn to interact with a system and to choose the right system inputs to achieve the desired system output. AI-based control approaches are, despite their downsides, able to handle bio-inspired muscle-driven systems [24] and their challenging features. Such systems offer several unique features to the controller:

- Muscle redundancy (multiple actuators for the same motion)
- Bi-articular muscles (muscles that act over two joints)
- Scalability (the number of muscles that a control system can handle)

Model-based and AI-based control approaches are able to perform control tasks of muscle-driven systems, but these control approaches consider the features of bio-inspired systems as problems or at least challenges. The model-based control approach "solves" the features of bio-inspired muscle-driven systems by computing power, while the AI- and fuzzy-based control approach "solves" it with data quantity. All of the model-based, fuzzy and AI-based control solutions that target bio-inspired features increase the control system complexity. Therefore, they are limited in scalability. These approaches view these features as drawbacks that require additional control solutions to deal with them. None of these control solutions exploit bio-inspired features to decrease complexity and enable scalability.

In my opinion, this way of thinking is the main reason these approaches are limited and do not benefit from these bio-inspired features. This work thus offers a framework that can be considered as a toolbox for exploiting biological features to reduce complexity and unleash possibilities.

”It is not a bug; It is a feature”

Nature would not have given us muscles if they were the second-best actuators. Electrical drives may be the number-one choice of actuators in many applicable systems and will displace even combustion engines in the future. The main reason they are also used in robotic applications that intend to mimic biology is not because of their innate beneficial properties for such applications. The reason lies in a lack of understanding of bio-inspired actuator setups [67, 97] and applicable technical control solutions for bio-inspired actuator setups [49, 105]. Indeed, electrical drives lack essential features that biological muscle actuators include:

- Compliance (Electrical drives have no passive compliance)
- Stiffness variation (Electrical drives create a torque according to its activation, and the stiffness results from the load acting on that joint)

Compliance is the ability of an actuator to adapt to the environment without external control. The biological muscle tendon complex (MTC) [43, 46] can be stretched passively, which plays a major role in locomotion [96, 34]. Joint stiffness variation is the ability to vary the joint stiffness independent from position. Muscle spring units (MSUs) in antagonistic setups enable this feature [122], which is crucial for locomotion [41, 31]. The MSU is the technical pendant to the MTC and can be found in many bio-robotic setups [27, 49, 50, 62, 76, 82, 88]. The goal of the MSU’s in bio-robotic setups is to attain these biological properties [122, 51, 50, 82].
To further investigate the benefits of muscle spring units in antagonistic joint setups, the scientific question this work addresses is:

Q2: What are the benefits of bio-inspired antagonistic muscle-driven joint setups in applied bio-robotics?

Firstly, in bio-inspired muscle driven systems, an MSU comes with many benefits compared to a single pneumatic muscle. Passive compliance as mentioned before, is just one of these. Furthermore, the ratio between spring and muscle length can be seen as a design element [11, 62, 61, 79, 104] to create the desired force-pressure-length ratio [73, 100, 97]. A long muscle enables higher control ability [10], whereas a long tendon increases the innate passive abilities [39, 40, 52, 90, 109]. However, the ultimate benefits appears in the field, when an MSU is implemented in an antagonistic joint setup with other MSUs [122]. The loss of motion range by using an MSU instead of single pneumatic muscles can be completely neglected in an antagonistic setup and a high range of torque generation and joint stiffness can be achieved [59]. The benefits of MSU in antagonistic setups are further expounded in chapter 2. There, the force-pressure-length relation is investigated through experiments, and the impact of motion range, torque generation, and joint stiffness in antagonistic setups is analyzed (Chapter 2 has already been published by the author of this work in Wolfen 2018 [122]).

Biology designs muscle arrangement in redundant setups. When we consider the elbow joint, for example, there are three muscles that provide elbow flexion: the m. biceps brachii, the m. brachialis, and the m. brachioradialis. Technical industrial solutions like the industrial arm robot in figure 1.1 never use redundant actuators; hence, it is possible to manufacture electrical drives for various torques. If an electrical drive is too weak, it can be replaced by a stronger one. Maximum force creation is also not a feasible reason for biology to use muscle redundancy; thus, a single muscle can be trained, and its maximum force can be increased [81, 110]. Muscle redundancy means that there are more actuators acting on a joint than this joint has degrees of freedom. This theoretically enables infinite actuation possibilities for the same motion. The benefits of this condition is in the scope of scientific investigations [60, 22] but is out of scope of this work. Nevertheless, a controller driving a bio-inspired setup with muscle redundancy is confronted with this condition [47].

Muscle redundancy is connected to the question of scalability. If more muscles provide benefits (and I assume that, as mentioned in the preface), the control system should be scalable with actuators to be an generalizable applicable robotic solution. The same applies to the increase of joints or degrees of freedom provided by additional muscles.

Therefore, another scientific question addressed in this work is:

Q3: How can the muscle redundancy and scalability challenge be solved?

In addition to muscle redundancy, bi-articular muscles are extremely uncommon from an engineering perspective. Bi-articular muscles are muscles that act over two joints. Industrial robots never have actuators that act on two different joints, hence it increases model complexity [58]. However, nature does have this. Experiments and simulations on humans and animals have shown that bi-articular muscles mechanically increase locomotion performance [38, 74, 87, 95, 120] and also have a positive impact on the metabolic costs of locomotion [25, 26, 119]. Therefore, bi-articular muscles in bio-robotic setups have been in the scope of study over the last decade [6, 7, 62, 66, 77, 82, 99, 101, 104]. It is also the case that robotic setups benefit from bi-articular muscles as these show an increase in motion speed [23], locomotion efficiency [54] and robustness [23].
However, the control of bio-robotic setups with bi-articular muscles is still an obstacle because established control approaches treat bi-articular muscles as a problem instead of making use of them, as previously explained. Applying the premise that every biological structure is evolutionary adapted and therefore advantageous for its purpose, bi-articular muscles must be beneficial for control. In fact, there are many indications for control benefits of bi-articular muscles [19, 62, 61, 104, 115]. Assuming that motor control uses different hierarchical levels of control [63], bi-articular muscles show sensory benefits [15, 85, 104, 118] and supportive stabilization benefits [98] for high level control. On the other hand, all these well-investigated control benefits of bi-articular muscles do not provide a control scheme that exploits these bi-articular muscles to reduce control complexity. Therefore, the last and perhaps most interesting scientific question of this work is:

Q4: How can a controller exploit bi-articular muscles to reduce control complexity?

To investigate these scientific questions, two robotic setups are used: the

arm robot Ataro and the one-legged robot SH1. With the robotic test-bed of Ataro, the controller is investigated and its performance output of point-to-point reaching is recorded and serves as a benchmark. The control framework is explained in detail in section 3, and its stability and parameter robustness are investigated in section 4. Point-to-point reaching performance is defined by speed and accuracy. In particular, it is the time a controller needs to reach stable target position within a specified accuracy (joint angle deviation to target position).
The impact of loss in control by reducing the controller input information and its output possibilities is investigated with the stable stance control of SH1. Both robotic test-beds use the same control structure.

The control framework of this work is not intended as a representation of the motor control system. It is inspired by it to exploit the benefits of muscle-driven systems by using its so-called ”drawbacks” (e.g., redundancy, bi-articular actuators) as an advantage for applicable muscle-driven robotic solutions. Furthermore, this work shows a low-level control framework. The inputs of the proposed control framework are target vectors (besides sensor information) generated by a possible upstream high-level control structure. This also enables compatibility for various other upstream control structures. The output of this control framework is muscle commands. It is called a ”framework” and not a ”controller” or ”method” due to its exchangeable components and its general applicability to all systems within scope.

1.2 Bio-robotic Test Bed

1.2.1 Ataro

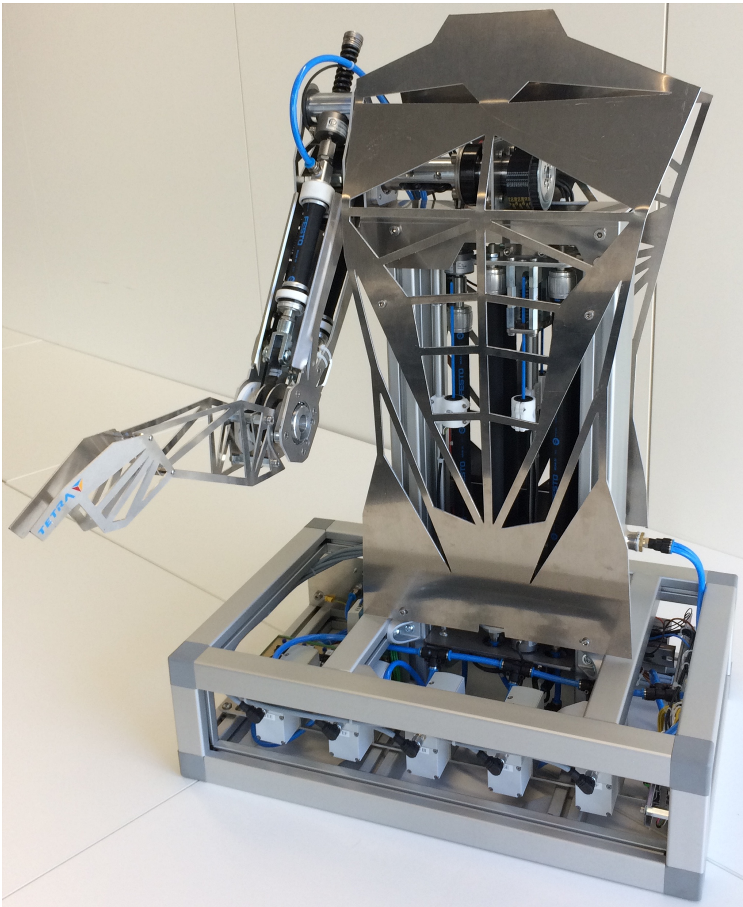


Figure 1.5: The robotic test bed Ataro

My contribution to the robotic test bed Ataro

Ataro was developed in cooperation between the University of Stuttgart and the Hertie Institute for clinical brain science Tübingen. Hereby, the design was developed by Prof. Dr. Syn Schmitt (Stuttgart) and Dr. Daniel Häufle (Tübingen) and Ataro manufactured by an external supplier. Furthermore, several improvements have been made by me until today:

- Interface hardware between Ataro, computer system and remote control was developed, manufactured, and installed by me.
- Security test mode was added by me, to enable a simulation mode in which all sensor information are transferred but the actuation is blocked on the hardware level (manual switch).
- Finger force Sensor was installed at the finger tip and the amplifier hardware was developed, manufactured, and installed by me.
- An optional panel can be mounted on the arm side of Ataro which illustrates the possible motion area. The area was calculated by me and the panel was manufactured under my supervision.
- The camera system Kinect was added on top of Ataro by Tobias Nadler. The interface and communication hardware and software was developed, manufactured and installed by me.
- Several software modes and motion programs have been developed by me (which are not in scope of this work) to demonstrate the research output of our Institute and increase the visibility of our research group inside and outside the University. Noteworthy are the Hannover fair 2017 and the demonstration of the evaluation for the German excellence University program 2019 to represent the Simtech Cluster of Excellence of the University of Suttgart. Additionally, Ataro was demonstrated at several workshops, colloquiums, and University exhibitions by me.

The robotic test bed Ataro $\mathscr{A}$ is an bio-inspired arm robot with two hinge joints, representing the shoulder and the elbow joint.
The joints are defined as:

$$\boldsymbol{\Phi} = \begin{pmatrix} \Phi_s \\ \Phi_e \end{pmatrix} \tag{1.2}$$

Ataro's motion possibilities are limited to the para-sagittal plane and therefore, the shoulder joint Φ_s only enables the anteversion and retroversion as well as the elbow joint Φ_e only enables the flexion and extension.

Definition of joint direction:
The anteversion of the shoulder and the flexion of the elbow are defined as positive joint directions in this work. The retroversion and extension are defined as negative directions respectively.

The joints are equipped with incremental encoders providing sensor information $\boldsymbol{\varphi}$ about the joint positions, respectively defined as:

$$\boldsymbol{\varphi} = \begin{pmatrix} \varphi_s \\ \varphi_e \end{pmatrix} \tag{1.3}$$

The information of the shoulder joint position φ_s and the elbow joint position φ_e are encoded into increments. The joints are driven by five muscle spring units (MSU) using pneumatic actuated McKibben muscles. The muscle spring units have a bio-inspired arrangement and represent biological pendants:

- m. latissimus (M_1) for shoulder anteversion
- m. deltoideus (M_2) for shoulder retroversion
- m. biceps brachii (M_3) for shoulder anteversion and elbow flexion
- m. triceps brachii caput laterale (M_4) for elbow extension
- m. brachialis (M_5) for elbow flexion

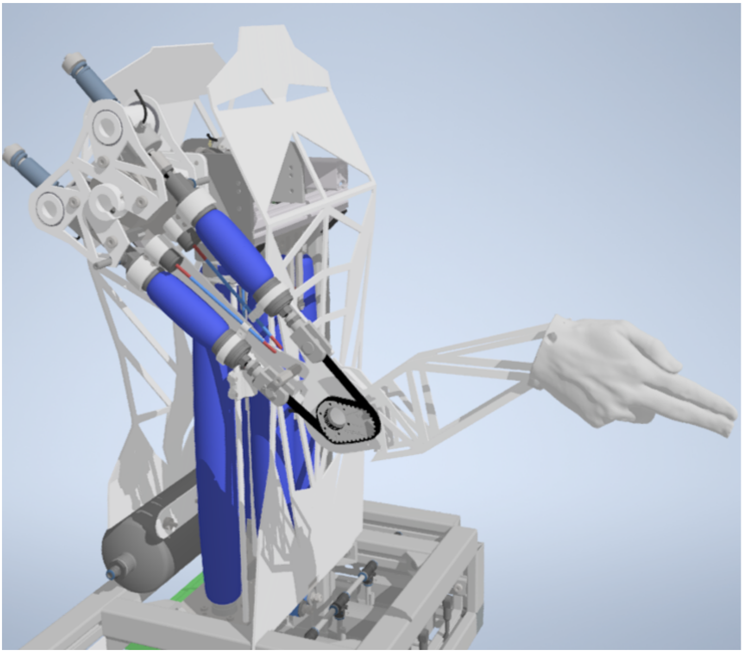


Figure 1.6: CAD Drawing of Ataro and it's muscles. A human arm has more muscles and degrees of freedom than Ataro (figure 1.7). Ataro was build in a cooperation project of the University Stuttgart and the Hertie-Institute for clinical brain research Tübingen to investigate the motor control system of the human body. For that, a minimal setup with two elementary biological drives and a biarticular muscle spring unit was needed. An elementary biological drive consists of two antagonistically arranged muscle actuators that drive a joint.

The muscle spring units **M** are defined as:

$$\mathbf{M} = \begin{pmatrix} M_1 \\ \vdots \\ M_5 \end{pmatrix} \tag{1.4}$$

Figure 1.7 illustrate the biological pendants of the muscle spring units and their locations in the human body.

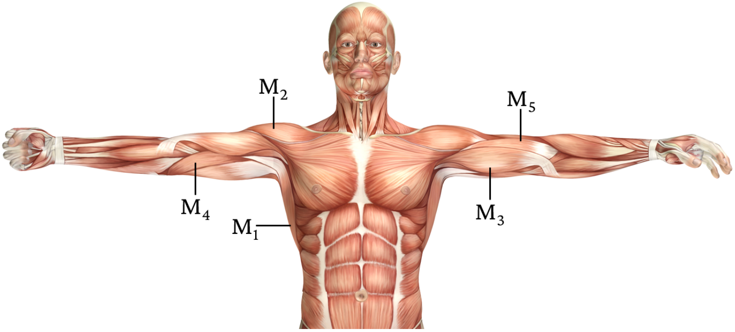


Figure 1.7: Human upper body with the locations of the biological pendants used at Ataro. *Source: https://de.freepik.com*

The bi-articular muscle spring unit M3 and the mono-articular muscle spring unit M2 is located inside the torso of Ataro due to space limitations of the upper arm and shoulder. A rope deflection ensures the correct motion impact. Figure 1.8 shows the muscle spring unit locations of Ataro and the rope deflections.

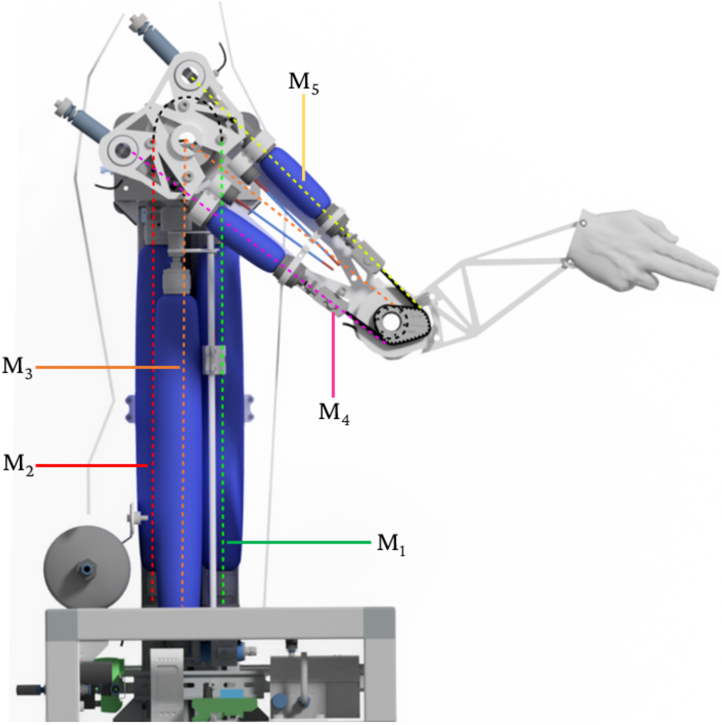


Figure 1.8: CAD Drawing of Ataro with actuator locations.

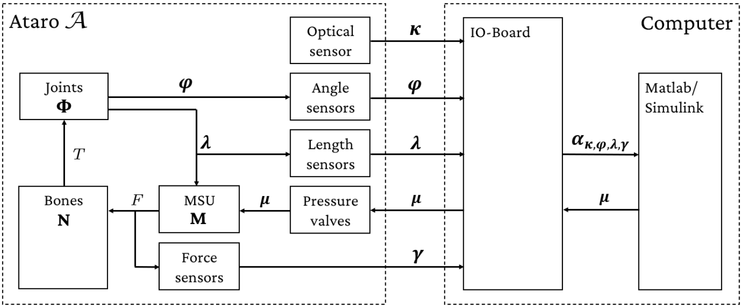


Figure 1.9: Experimental setup of Ataro and the interaction of all physical components

Ataro is a bio-robot, which means it has muscles that can interact with bones $\mathbf{N}$ and joints $\mathbf{\Phi}$ resulting in motion. The muscles generate forces F on the bones resulting in torques T on the joints. That distinguishes muscle driven systems from industrial robots. Industrial robots are com-

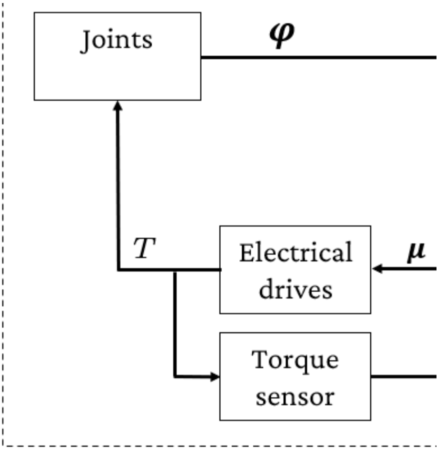


Figure 1.10: Industrial robots commonly uses electrical drives which generate torques directly on the joints. A robotic system with electrical drives as actuators would have figure 1.9 changed to this interaction system.

monly driven by electrical drives which generate a torque directly on the joint (figure 1.10). This is a crucial difference for applied control theories of these systems, hence the joints of a bio-robotic system has a back coupling impact to the muscles via the muscle length $\boldsymbol{\lambda}$. The muscle length $\boldsymbol{\lambda}$ is according to (1.4) defined as:

$$\boldsymbol{\lambda} = \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_5 \end{pmatrix} \tag{1.5}$$

The muscles spring units of Ataro are actuated via the actuation signal $\boldsymbol{\mu}$, which is respectively defined as:

$$\boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_5 \end{pmatrix} \tag{1.6}$$

As figure 1.9 shows, the actuation signal $\boldsymbol{\mu}$ exists in different physical shapes. Inside the computer, $\boldsymbol{\mu}$ is a variable from type double. Between the IO-Board and the pressure valve, $\boldsymbol{\mu}$ is an analog electrical signal. When it reaches the muscle spring unit, $\boldsymbol{\mu}$ is an air pressure between 0...6bar. Hence, the physical shape of $\boldsymbol{\mu}$ plays an underlying role in this work, it is not differentiated in the following and just treated as the muscle activation $\boldsymbol{\mu}$. All muscle spring units $\mathbf{M}$ are equipped with length and force sensors providing the muscle length information $\boldsymbol{\ell}$ and the force acting on the corresponding muscle spring unit against it's contraction:

$$\boldsymbol{\gamma} = \begin{pmatrix} \gamma_1 \\ \vdots \\ \gamma_5 \end{pmatrix} \tag{1.7}$$

In addition, a Kinect optical sensor is mounted on top of Ataro to detect up to two markers. The Kinect sensor provides via 2x 12bit parallel transmission the information $\boldsymbol{\kappa}$ of the two marker positions in space.

$$\boldsymbol{\kappa} = \begin{pmatrix} \kappa_a \\ \kappa_h \end{pmatrix} \tag{1.8}$$

κ_a is the position of Ataro's finger tip and κ_h is the position of the human finger tip. This feature was used as a high level controller for demonstration purposes and is out of scope of this work. All sensor information gathered by the IO-Board is transmitted to the Matlab/Simulink Model and defined as:

$$\boldsymbol{\alpha}_{\boldsymbol{\kappa},\boldsymbol{\varphi},\boldsymbol{\lambda},\gamma} = \begin{pmatrix} \boldsymbol{\kappa} \\ \boldsymbol{\varphi} \\ \boldsymbol{\lambda} \\ \gamma \end{pmatrix} \tag{1.9}$$

1.2.2 Stuttgart Humanoid 1

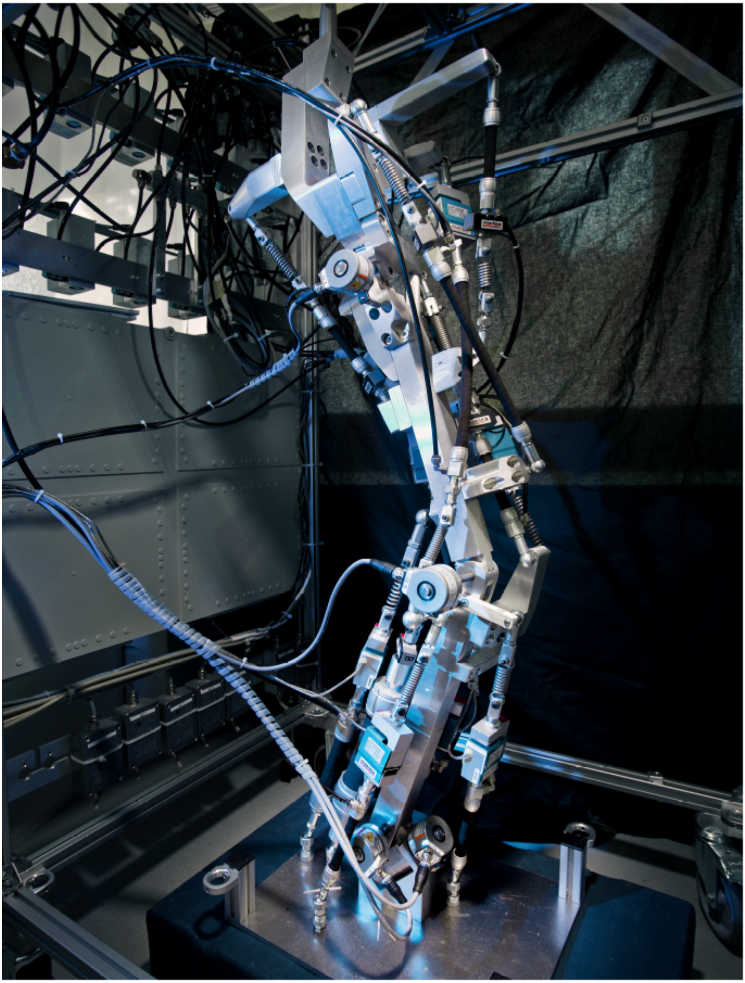


Figure 1.11: Stuttgart Humanoid 1 (SH1) is an one-legged robot mimicking a human left leg

My contribution to the robotic test bed SH1
Stuttgart Humanoid 1 was developed by Prof. Dr. Syn Schmitt and Dr. Daniel Häufle and mechanically manufactured by the University workshops. The initial state when I started my work at SH1 was a mechanical setup including all joints and construction elements representing the bones, 12 muscles connected to pressure valves which could only controlled manually. Several improvements have been made by me until today:

- Interface hardware between SH1 and computer system was developed, manufactured, and installed by me.
- The complete wiring was implemented by me.
- The complete sensor signal processing and amplifier hardware was calibrated or completely new developed, manufactured, and installed by me.
- Additional muscle m. soleus was added and the foot of Ataro has been re-designed by me.
- All MSU were re-designed to improve the actuation ranges and passive abilities.
- Several software modes and motion programs have been developed by me (which are not in scope of this work) to demonstrate the research output of our Institute and increase the visibility of our research group inside and outside the University. Noteworthy are the Hannover fair 2016 and 2018 to represent the Simtech Cluster of Excellence of the University of Suttgart. Additionally, Ataro was demonstrated at several workshops, colloquiums, and University exhibitions by me.

Stuttgart Humanoid 1 represents a technical model of a human left leg which is able to perform movements in 5 degrees of freedom (DOF). The hip joint is designed by two hinge joints which provide abduction/adduction and flexion/extension movements. Nevertheless, the hip joint motion is disabled for measurements of this work and can be considered rigid. The knee joint is a single hinge joint, enabling flexion/extension movement. According to its biological role model, the ankle joint is divided into two hinge joints, the upper and lower ankle. The upper ankle allows flexion/extension movement, the lower ankle supination/pronation. Subsequently, SH1 can be considered as a 3-dimensional inverted pendulum. 13 MSU's representing the muscles:

- m. glutaeus maximus
- m. adductor (lumped)
- m. rectus femoris
- m. iliopsoas
- m. glutaeus medius
- m. sartorius
- m. tibialis posterior
- m. biceps femoris caput breve
- m. tibialis anterior
- m. peronaeus
- m. gastrocnemius
- m. vastii (lumped)
- m. soleus

The geometry of SH1 is based on a 50-percentile 1.78m man [111]. The insertion and origin points of the actuators are close to their biological

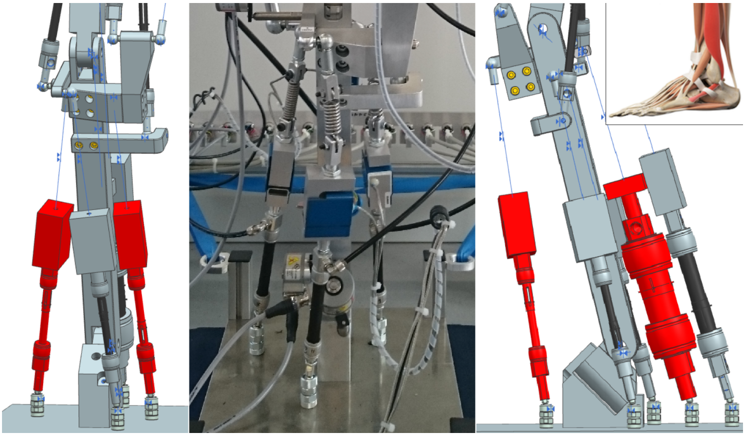


Figure 1.12: SH1 ankle joint and antagonistic muscle arrangement according to a human ankle joint. *Foot source: https://anatomy.lexmedicus.com*

pendant despite some muscles being lumped together for spatial reasons and are represented by a single MSU. The muscle arrangement contains both, antagonistic setups and bi-articular muscles illustrated in figure 1.12. Antagonistic muscle pairs must not equal in terms of maximum force nor lever arm according to its biological pendant. This imbalance is illustrated in figure 1.12 in the right picture. The m. soleus is seven times stronger than the m. tibialis anterior in the biological system (m. soleus with approximately $140Nm$ [13] and the m. tibialis anterior with approximately $20Nm$ [113] voluntarily contraction impact on the ankle joint). Due to mechanical limitations of SH1 (commercially available parts, spatial limitations, etc.), the lack of muscle force is compensated by an increased lever arm for some muscles. This is the case for the m. peronaeus and the m. tibialis posterior (left picture in figure 1.12). In addition to the bio-inspired muscle relations, SH1 also mimics the bi-articular muscles of the lower limb. In example, the m. gastrocnemius impacts both, the ankle joint and the knee joint.

SH1 has 5 incremental encoders in its joints and each muscle spring unit has a force sensor included. It can perform a stable stance with different joint positions without collapsing. SH1 runs with MATLAB/Simulink real-time environment. A given stance position can be achieved with dif-

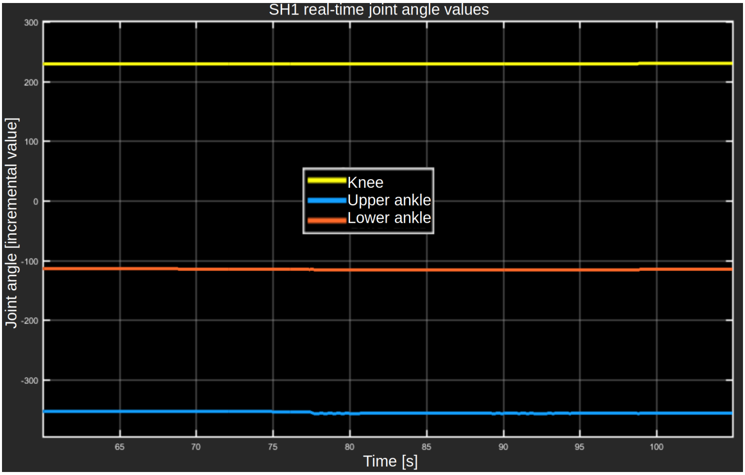


Figure 1.13: Matlab/Simulink real-time environment scope of SH1 in stable stance showing the joint positions while altering muscle activation shown in figure 1.14.

ferent levels of co-contraction of the muscles, as illustrated by the two measurement scopes of figure 1.13 and 1.14. Both data scopes are recorded within the same measurement. The joint angle scope of figure 1.13 shows the joint value from the incremental encoders over time, (10 ticks equals one degree). The bio-inspired control algorithm in hybrid mode (chapter 3) keeps the position while reducing the open-loop activation in steps (10% open-loop reduction per step) illustrated in figure 1.14. This shows that the same joint position can be reached with different actuation levels of the MSUs and therefore different joint stiffness 1.10. The joint stiffness S_i of a joint Φ_i is given by:

$$S_i = \sum_{n=1}^{j} T_n \tag{1.10}$$

S_i is the sum of all torques T generated by all muscles j that impact the joint Φ_i. SH1 collapses completely without actuation.

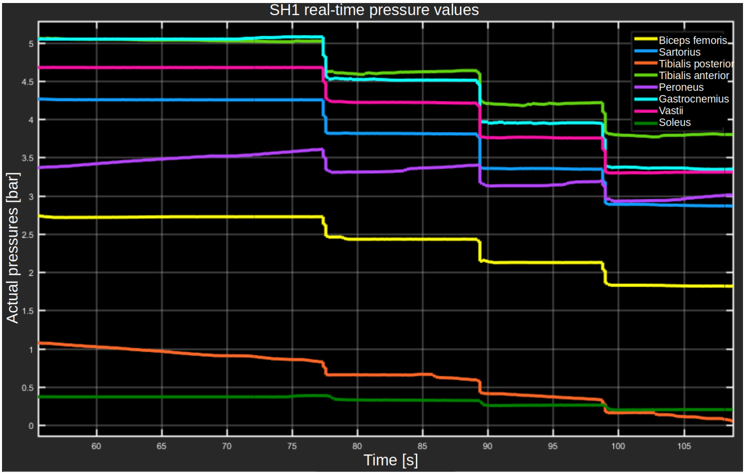


Figure 1.14: Matlab/Simulink real-time environment plot of SH1 in stable stance with different actuation levels and joint stiffness configurations

1.2.3 Main Program and Parameters: Implementation

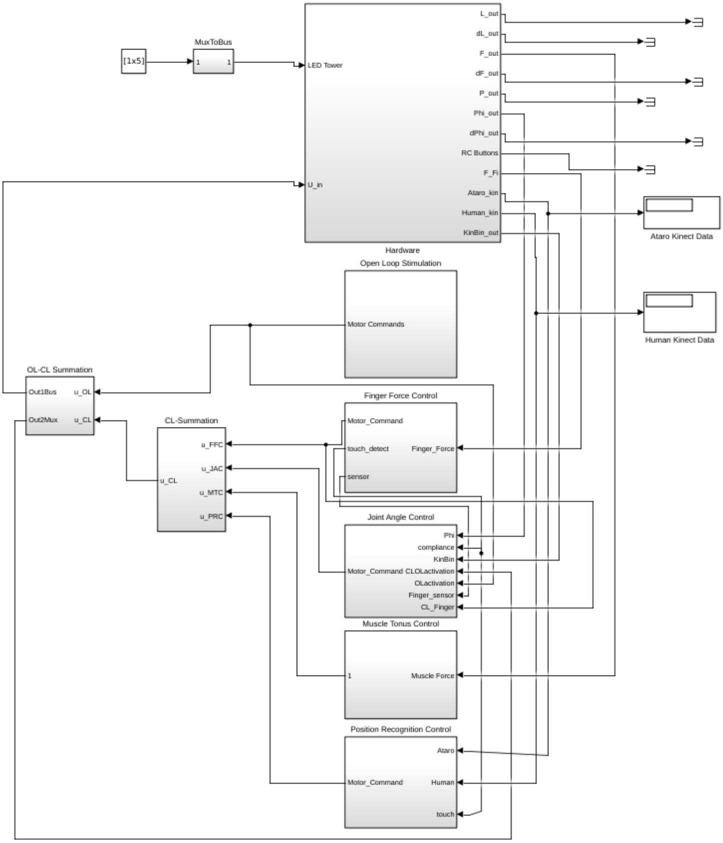


Figure 1.15: Ataro main Program in Matlab/Simulink. This program has more features than explained in the chapters before. There is a finger following program using a kinect as joint value input and a force control program that uses a force sensor to enable active compliance. These features are gimmicks for demonstrations and use the identical controller described in chapter 3. Furthermore, these programs run in parallel or instead of the closed loop controller and did not effect the results shown in this work.

Prior to run the program, several variables have to be initiated. All these variables are summarized in a single script. The script starts with the initialization of the input and output scaling parameters:

```
clear; %refresh workspace
% This is the parameter sheet to initialize all initial
     parameters and allocate necessary workspace

% Simulation and sample time
T_S = 1/500; %Sample time in seconds
T_sim = 300; %Simulation time in seconds

%% Hardware Block Parameters

% Mean Value Filter Parameter Matrix
MVF_para =...
[100  2; ...% 1st: Samples, 2nd: Digits for Length Sensor
     Filter
100  2; ...% 1st: Samples, 2nd: Digits for Force Sensor
     Filter
100  2; ...% 1st: Samples, 2nd: Digits for Pressure Sensor
      Filter
];

%Sensor delay parameters
Fin_para =...
[100 2; ...
400 -100];

% Sensor Scaling Parameter Matrix
SS_para =...
[109/5  109/5   109/5   109/5   109/5; ...% 1V= [mm]: [
     L_SE L_SF L_Bi L_EE L_EF]
200 200 200 200 200; ...%1V= [N]: [F_SE F_SF F_Bi F_EE
     F_EF]
1   1   1   1   1; ...%1V= [bar]: [P_SE P_SF P_Bi P_EE
     P_EF]
360/163000   360/163000   16.03   19.7   0;...% [deg/
     puls.S deg/puls.E off_S off_E 0]
];

% Actuator Scaling Parameter Matrix
AS_para =...
[0   6;...% [min max] P_SE Pressure in bar
0   6;...% [min max] P_SF Pressure in bar
0   6;...% [min max] P_Bi Pressure in bar
0    6;...% [min max] P_EE Pressure in bar
0    6;...% [min max] P_EF Pressure in bar
];
```

The summation parameters are boolean parameters. They enable and disable the different modes:

```
%% Summation Parameters

% CL-Summation
CLSum_Para =...
[0;...  % Finger Force Cotrol
1;...   % Joint Angle Control
0;...   % Muscle Tonus Control
0];     % Position Recognition Control

% OL-CL Summation
OCSum_Para =...
[1;...  % Open Loop
1];     % Closed Loop
```

The open loop block parameters initialize starting values, define equilibrium points, and determine open loop controller behavior:

```
%% Open Loop Stimulation Block Parameters

%B1:     Increase activation with step size
%B2:     Increase activation with step size
%B3:     n.a.
%B4:     set activation smoothly to zero
%B5:     Switch muscle (Active muscle is displayed by the
     tower

%Initialization and Equilibrium Points (EPs)
MRC_para =...
[0.1 0.1 0.1 0.1 0.1;...% Step Size per click [P_SE P_SF
     P_Bi P_EE P_EF]
0.9947 0.9965 0.9974 0.9979 0.9982;...% Switch Position [2
     s 3s 4s 5s 6s] Parameter = Threshold^(T_S/Time to
     reach)|Threshold = 0.005(0.5% Difference)
2.5 0.1 4.5 1 6;...%EP1
1.1 2.2 2.6 6 0.4;...%EP2
2.3 0.2 2.2 2.4 0.5;...%EP3 (Zero)
2.3 1.1 2 2.7 0.3;...%EP4
```

```
2.7 1.5 2 6 0.3;...%EP5
];

%EPs for motion tasks
EP_pool =...
[1 22 0 1.61 4.30 1.24 0;... %EP1-pool
-32 -20 0 0.67 2.07 0.24 0;...% EP2-pool
0 -18 0 1.85 1.80 4.56 0;...%EP3-pool
-33 21 0 0.37 3.55 0 1.37;...%EP4-pool
];

%% Joint Angle Controller

% These are alternative parameter sets for the joint angle
     controller. They are dependent from only one single
    muscle parameter.
% %Closed-Loop P-Matrix (colums are joints, rows are
    muscle)
% %shoulder => (1)
% %elbow => (2)
%
%PM_Pos = 0.015*...
%[-0.05          0            ;  ... %0.5
%0.05           0            ;  ...
%0.05           0.05            ;  ...
%0            -0.05            ;  ...
%0            0.05            ;  ...
%];
%
%p11 = -3.25e-4;
%
%PM_Pos =...
%[p11           0            ;  ... %0.5
%-p11          0            ;  ...
%-p11*3.44    -p11*0.93     ;  ...
%0            p11*0.93      ;  ...
%0            -p11*0.56     ;  ...
%];
%
% %Closed-Loop D-Matrix (colums are joints, rows are
    muscle)
%
% D_JAC= 0.005*...
[0.0035          0            ;  ...
%-0.001           0            ;  ...
%0.0035        0.001            ;  ...
%0             0.001            ;  ...
```

```
%0              -0.001              ; ...
%];
%
% %Closed-Loop I-Matrix (colums are joints, rows are muscle)
% I_JAC= 0.1*     [0.001              0.001]; ... %0.8
%
% %Initial Condition Matrix
% %I_JAC = 0.5* [0.001          0.004];


%% Position Control Parameters

%Closed-Loop P-Matrix (colums are joints, rows are muscle)
%shoulder => (1)
%elbow => (2)
%q1             q2
P_CL= 0.5*...
 [0.023             0               ; ... %0.5
 -0.005             0               ; ...
 0.023              0.02                ; ...
 0              0.02                ; ...
 0              -0.05               ; ...
 ];

%Closed-Loop D-Matrix (colums are joints, rows are muscle)
%q1             q2
D_CL= 0.005*...
 [0.0035            0               ; ...
 -0.001             0               ; ...
 0.0035         0.001               ; ...
 0              0.001               ; ...
 0              -0.001              ; ...
 ];

%Closed-Loop I-Matrix (colums are joints, rows are muscle)
%q1             q2
I_CL= 0.05*...
 [0.001             0               ; ... %0.8
 -0.001             0               ; ...
 0.0001             0.0001              ; ...
 0              0.002               ; ...
 0              -0.002              ; ...
 ];

%Initial Condition Matrix
I_MX = 0.05* [0.001         0.004];
```

```
PC_IM = [0.178;0.12]; %for p=3,25: [0.178;0.12]
M_CL = [0 0 0 0 0; ... %Pos_Set Pos_Clear    t_start
     desPhiS desPhiE
0 0 0 0 0; ... %run_CL  I_Phi   I_Phi            0
     0
0 0 0 0 0; ... %P_SF    P_SE    P_bi    P_EF    P_EE
%          0 0 0 0 0; ... %I_SF    I_SE    I_bi    I_EF
     I_EE
];
```

To ensure smooth motion while changing equilibrium point in open-loop mode, a smoothing filter is used. This filter stores the pre-calculated target positions in a matrix and apply the mean value as the output. This requires pre-allocating memory:

```
%% Open Loop Smoothing

OLS_Filter = [  0 0 0 0 0; ...%1
0 0 0 0 0; ...%2
0 0 0 0 0; ...%3
0 0 0 0 0; ...%4
0 0 0 0 0; ...%5
0 0 0 0 0; ...%6
0 0 0 0 0; ...%7
0 0 0 0 0; ...%8
0 0 0 0 0; ...%9
0 0 0 0 0; ...%10 -10*T_S = 20ms Delay
0 0 0 0 0; ...%1
0 0 0 0 0; ...%2
0 0 0 0 0; ...%3
0 0 0 0 0; ...%4
0 0 0 0 0; ...%5
0 0 0 0 0; ...%6
0 0 0 0 0; ...%7
0 0 0 0 0; ...%8
0 0 0 0 0; ...%9
0 0 0 0 0; ...%10 -20*T_S = 40ms Delay
0 0 0 0 0; ...%1
0 0 0 0 0; ...%2
0 0 0 0 0; ...%3
0 0 0 0 0; ...%4
0 0 0 0 0; ...%5
0 0 0 0 0; ...%6
0 0 0 0 0; ...%7
0 0 0 0 0; ...%8
0  0  0  0  0;  ...%9
0  0  0  0  0;  ...%10 -30*T_S = 60ms Delay
0  0  0  0  0;  ...%1
0  0  0  0  0;  ...%2
0  0  0  0  0;  ...%3
0  0  0  0  0;  ...%4
0  0  0  0  0;  ...%5
0  0  0  0  0;  ...%6
0  0  0  0  0;  ...%7
0  0  0  0  0;  ...%8
0  0  0  0  0;  ...%9
0  0  0  0  0;  ...%10 -40*T_S = 80ms Delay
0  0  0  0  0;  ...%1
0  0  0  0  0;  ...%2
0  0  0  0  0;  ...%3
0  0  0  0  0;  ...%4
0  0  0  0  0;  ...%5
0  0  0  0  0;  ...%6
0  0  0  0  0;  ...%7
0  0  0  0  0;  ...%8
0  0  0  0  0;  ...%9
0  0  0  0  0;  ...%10 -50*T_S = 100ms Delay
0  0  0  0  0;  ...%differential
1  0  0  0  0;  ...%parameters
];
```

This section initialize the force control of the finger tip for active compliance:

```
%% Force Control
%q1  q2 Finger
FC_PM =...%The P-matrix
[    -0.01     0;  ...%m1 0.01
0.01     0;  ...%m2 0.01
0.01     0.01;  ...%m3 0.005 both
0     0.03;  ...%m4 0.03
0     -0.03;  ...%m5 0.03
];

%q1  q2 Finger
FC_IM = 0.05;% The Integrator matrix - alternative value:
    0.03

%Sensor noise filter
MFC_PM =...
```

```
[   0     0;...%m1 Finger
0     0;...%m2
0     0;...%m3
0     -0.01;...%m4
0     0.01;...%m5
];

%Sensitivity
V_MFP = 0.5*...
[0.01;...
0.004;...
0.005;...
0.01;...
0.01;...
];

%Enable/ Disable features
V_MFI =
[0;...
0;...
0;...
0;...
0;...
];
```

While in real-time simulation mode, the different tasks can be set via Matlab Simulink interface. These settings have no impact on the controller and only apply to the interface:

```
%% Interface
Interface_M =...%These parameters apply to the control
     interface of Matlab Simulink
[ 0   0    23   -20  18   -5   15   7    12   18;...
0     0    15   -13  8    3    8    15   0    0;...
12    -22  4    -5   -3   10   -5   20   0    0;...
5     -20  -5   0    -12  13   0    0    0    0;...
-2    -20  -14  0    -22  15   0    0    0    0;...
-8    -20  -20  -2   0    0    0    0    0    0;...
-15   -20  0    0    0    0    0    0    0    0;...
-20   -20  0    0    0    0    0    0    0    0;...
];
```

To model the biological muscle tonus, a minimum activation of each muscle is ensured depending on the actual force acting on the muscle:

```
%% Muscle Tonus Control
tonus_para =...%These are thresholds and control parameter definitions
[  25  10  20  60  10;...                  %Threshold
0.00115 0.00025 0.00107 0.001 0.0025;...      %P-Value
0.15 0.15 0.15 0.15 0.15;...                 %I-Value
];
```

This mode is used with a Kinect optical system using markers on the finger tip to detect and control the finger position:

```
%% Position Recognition Control
%Ataro only acts in within the saggital plain, therefore only the y- and z-axis of the Kinect system is relevant.
%y             z
P_PRC = 0.02*...%P-matrix of this mode
[0.01          0.03             ;  ... %0.5
-0.01          -0.03             ;  ...
0        0             ;  ...
-0.04      0             ;  ...
0.04       0             ;  ...
];
I_PRC = 0.3;%I-matrix of this mode
```

1.2.4 Length Sensor Processing

The length sensors are implemented as input blocks (Figure 1.16) providing the information of the sensors via hardware I/O-board. The processing (Figure 1.17) provides the sensor information and their first derivative.

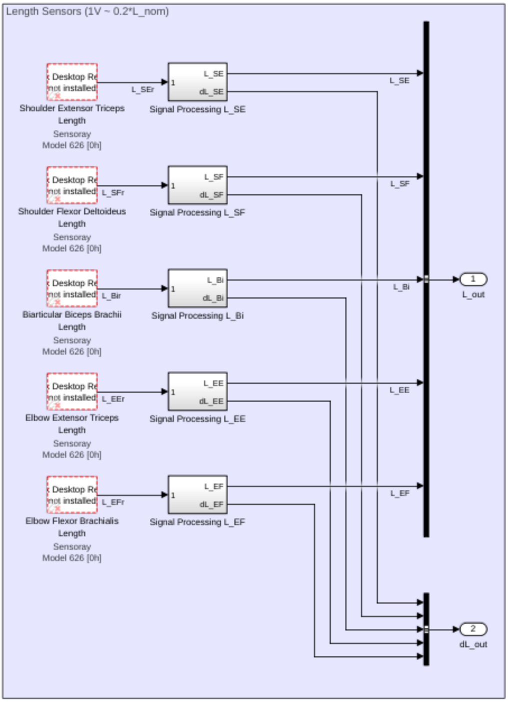


Figure 1.16: Length Sensor Inputs

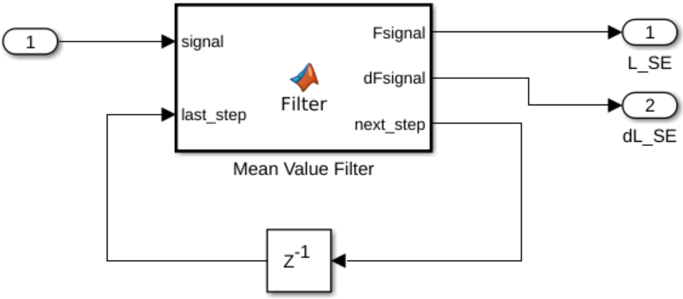


Figure 1.17: Length sensor signal processing

The parameters are initialized by the parameter script explained in section 1.2.3.

```
function [Fsignal, dFsignal, next_step] = Filter(signal,
    last_step, MVF_para, SS_para, T_S)

%Initialization for each time step
i = last_step(1);
value = last_step(2);
hold = last_step(3);
dhold = last_step(4);

%Allocating parameters from the parameter script
MVF_S = MVF_para(1,1);
digit = MVF_para(1,2);
scale = 10^digit;

%Applying the filter and calculating the derivative
if (i < MVF_S)
value = value + signal;
i = i+1;
else
hold = round(value*scale/MVF_S)/scale;
dhold = (hold - last_step(3))/MVF_S;
value = 0;
i = 0;
end

%Determining the output parameters
```

```
next_step = [i value hold dhold];
Fsignal = hold*SS_para(1,1);
dFsignal = dhold*SS_para(1,1)/T_S;
```

The same filter is applied for each length sensor.

1.2.5 Force Sensor Processing

The force sensors are implemented as input blocks (Figure 1.18) providing the information of the sensors via hardware I/O-board. The processing (Figure 1.19) provides the sensor information and their first derivative.

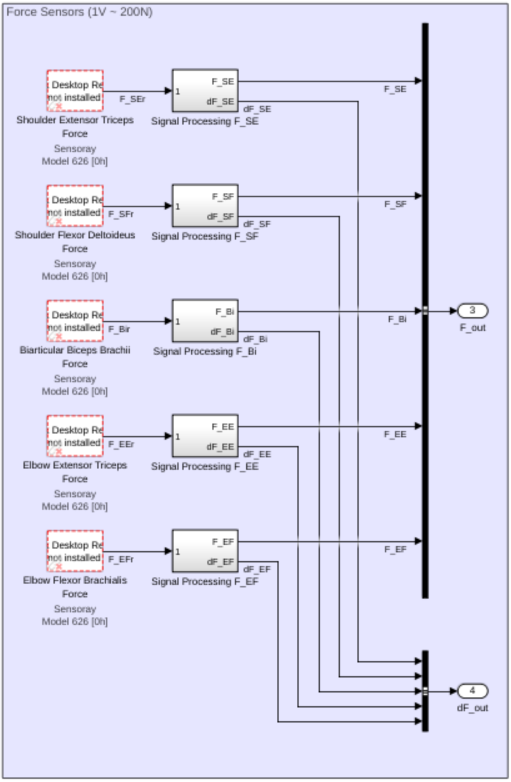


Figure 1.18: Force Sensor Inputs

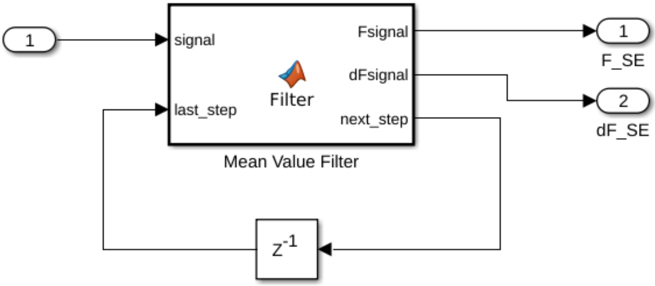


Figure 1.19: Force sensor signal processing

The parameters are initialized by the parameter script explained in section 1.2.3.

```
function [Fsignal, dFsignal, next_step] = Filter(signal,
    last_step, MVF_para, SS_para, T_S)

%Initialization for each time step
i = last_step(1);
value = last_step(2);
hold = last_step(3);
dhold = last_step(4);

%Allocating parameters from the parameter script
MVF_S = MVF_para(2,1);
digit = MVF_para(2,2);
scale = 10^digit;

%Applying the filter and calculating the derivative
if (i < MVF_S)
value = value + signal;
i = i+1;
else
hold = round(value*scale/MVF_S)/scale;
dhold = (hold - last_step(3))/MVF_S;
value = 0;
i = 0;
end
```

```
%Determining the output parameters
next_step = [i value hold dhold];
Fsignal = hold*SS_para(2,1);
dFsignal = dhold*SS_para(2,1)/T_S;
```

The same filter is applied for each force sensor.

1.2.6 Joint Angle Sensor Processing

The joint angle sensors are implemented as input blocks (Figure 1.20) providing the joint position information in increments of the sensors via hardware I/O-board. The processing (Figure 1.21) provides the sensor information and their first derivative.

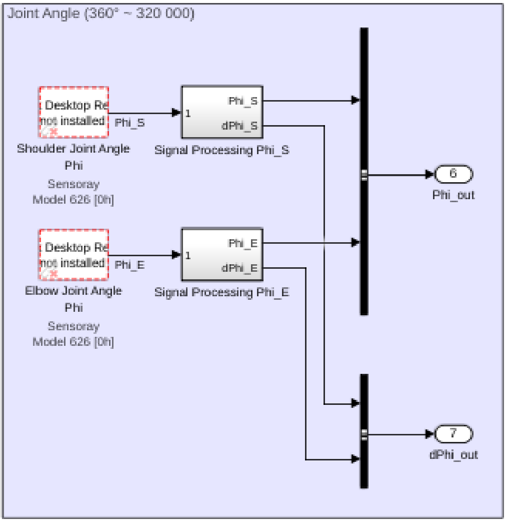


Figure 1.20: Joint angle sensor inputs

The parameters are initialized by the parameter script explained in section 1.2.3.

```
function [Fsignal, dFsignal, next_step] = Filter(signal,
    last_step, SS_para, T_S)

%Initialization for each time step
i = last_step(1);
value = last_step(2);
hold = last_step(3);
dhold = last_step(4);
```

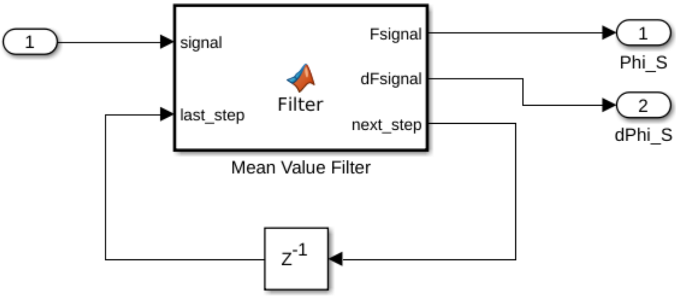


Figure 1.21: Joint angle sensor signal processing

```
%Allocating parameters from the parameter script
MVF_S = MVF_para(2,1);
 digit = MVF_para(2,2);
 scale = 10^digit;

%Applying the filter and calculating the derivative
 if (i < MVF_S)
      value = value + signal;
      i = i+1;
 else
      hold = round(value*scale/MVF_S)/scale;
      dhold = (hold - last_step(3))/MVF_S;
      value = 0;
      i = 0;
 end

%Determining the output parameters
 next_step = [i value hold dhold];
 Fsignal = hold*SS_para(2,1);
 dFsignal = dhold*SS_para(2,1)/T_S;

%Fsignal = -signal*SS_para(4,1) + SS_para(4,3);%
     alternative settings
%dFsignal = (Fsignal - last_step(1))/T_S;%alternative
     settings
%next_step = Fsignal;%alternative settings
```

The same filter is applied for each joint angle sensor.

1.2.7 Pressure Sensor Processing

The pressure sensors are implemented as input blocks (Figure 1.23) providing the information of the sensors via hardware I/O-board. The processing (Figure 1.22) provides the sensor information and their first derivative.

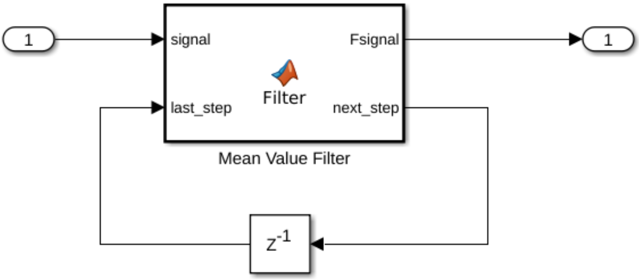


Figure 1.22: Pressure sensor signal processing

The parameters are initialized by the parameter script explained in section 1.2.3.

```
function [Fsignal, next_step] = Filter(signal, last_step,
    MVF_para)

%Initialization for each time step
i = last_step(1);
value = last_step(2);
hold = last_step(3);

%Allocating parameters from the parameter script
MVF_S = MVF_para(3,1);
digit = MVF_para(3,2);
scale = 10^digit;

%Applying the filter and calculating the derivative
if (i < MVF_S)
```

```
value = value + signal;
i = i+1;
else
hold = round(value*scale/MVF_S)/scale;
value = 0;
i = 0;
end

%Determining the output parameters
next_step = [i value hold];
Fsignal = hold;
```

The same filter is applied for each pressure sensor.

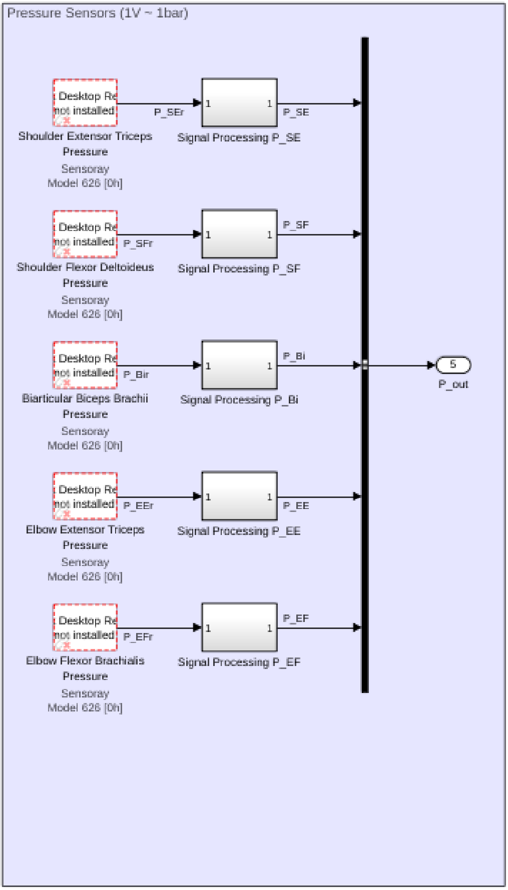


Figure 1.23: Pressure sensor inputs

1.2.8 Muscle Output Implementation

The muscle actuation is implemented as output blocks (Figure 1.24) transferring the controller motor commands via hardware I/O-board into analog signals for the pressure valves. The processing (Figure 1.17) consists of a security check to ensure an applicable operating range for the pressure values.

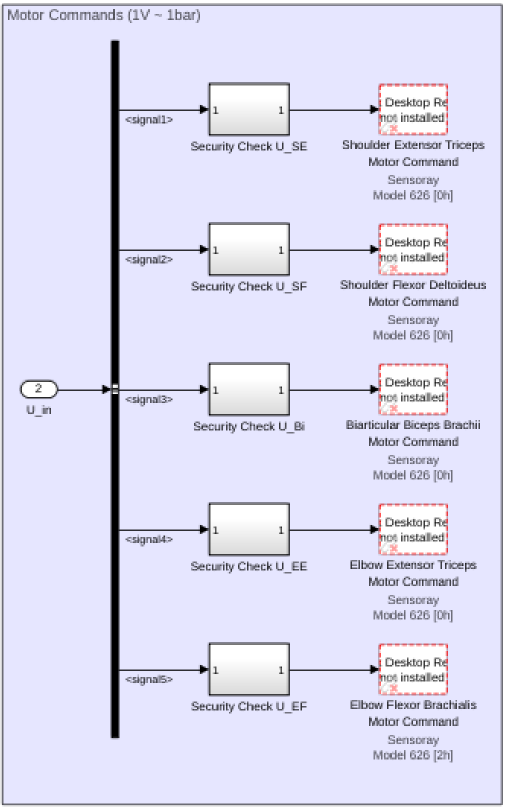

Motor Commands (1V ~ 1bar)
2
U_in
<signal1>
<signal2>
<signal3>
<signal4>
<signal5>
Security Check U_SE
Security Check U_SF
Security Check U_Bi
Security Check U_EE
Security Check U_EF
Desktop Re not installed
Shoulder Extensor Triceps Motor Command
Shoulder Flexor Deltoideus Motor Command
Biarticular Biceps Brachii Motor Command
Elbow Extensor Triceps Motor Command
Elbow Flexor Brachialis Motor Command
Sensoray Model 626 [0h]
Sensoray Model 626 [2h]

Figure 1.24: Motor command outputs

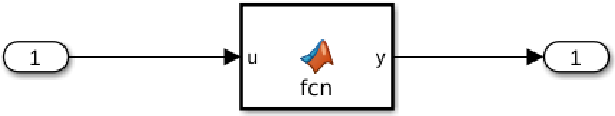


Figure 1.25: Motor command output saturation

The parameters are initialized by the parameter script explained in section 1.2.3. The output saturation prevents damage to the pressure valves by limiting the motor commands. Besides that, it has no function.

```
function y = fcn(u, AS_para)

%Allocating parameters from the parameter script
u_min = AS_para(1,1);
u_max = AS_para(1,2);

%Applying the filter and calculating the derivative
if (u >= u_max)
u = u_max;
elseif (u <= u_min)
u = u_min;
end

%Determining the output parameters
y = u;
```

The same saturation is applied for each motor command output.

Chapter 2

Muscle Spring Unit

This chapter was published in 2018 at the M2VIP Conference [122] by the author of this work.

2.1 Mechanical design

To mimic biological actuation MSUs are used. Each MSU consists of a nonlinear PAM (Festo DMSP fluidic muscle [20]), which can be actuated in a pressure range of $P = 0 \ldots 8$ bar and a linear metal spring in series.

Furthermore, a MSU has additional stiff construction elements to mount the MSU on the robot. The total length of a MSU L_{MSU} is given as:

$$L_{\mathrm{MSU}}(F_{\mathrm{M}}, F_{\mathrm{S}}, P) = L_{\mathrm{M}}(F_{\mathrm{M}}, P) + \Delta L_{\mathrm{S}}(F_{\mathrm{S}}) + L_{\mathrm{R}} \tag{2.1}$$

$L_{\mathrm{M}}(F_{\mathrm{M}}, P)$ is the load and pressure dependent length of the active membrane of the PAM. Under zero pressure condition ($P = 0\,\mathrm{bar}$, $F_{\mathrm{M}} = 0\,\mathrm{N}$), this length is named nominal length $L_{\mathrm{M,nom}}$ and is given in

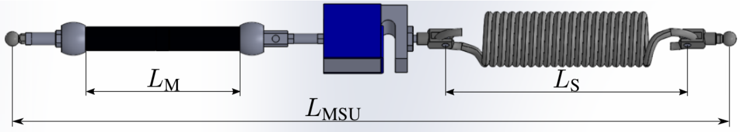


Figure 2.1: CAD drawing of a muscle spring unit with a type DMSP-20-100N PAM, force sensor, spring and stiff construction elements as it is used for the m. gastrocnemius.

the data sheet of the PAM. $\Delta L_S(F_S)$ is the length of the elongation of the spring and L_R is the length of all rigid construction elements including the force sensor and the nominal length of the tensile spring L_S. Therefore, $L_M(F_M, P)$ is the only active part and the pressure P is the only control variable of the MSU. According to the data sheet of the PAM, the length is given in relation to its nominal length depicted as shortening h:

$$h_M = 100\% - \frac{L_M(F_M, P)}{L_{M,\text{nom}}} \cdot 100\% \tag{2.2}$$

This notation is beneficial to assess the active and passive behavior of the muscle hence it only shortens its length while being activated. To describe the force-length-pressure relation and the resulting forces, we symbolize the relation as $F_M(h, P)$ for the muscle membrane and $F_{MSU}(h, P)$ for the entire MSU respectively.

The black membrane (L_M) of the Festo PAM is the only part of the MSU which can be actively contracted. The membrane of this PAM can only be stretched at higher pressure values. Therefore, its passive abilities are strongly dependent on the actuation level. The spring length L_S is the initial length of the spring. It can be passively stretched independent from any pressure value of the PAM. The difference between the initial length of the spring and its actual length is named as ΔL_S and is not shown in this picture. L_{MSU} is defined in (equation (2.1)) as the total length of the MSU between origin and insertion.

2.2 Rating of the spring stiffness

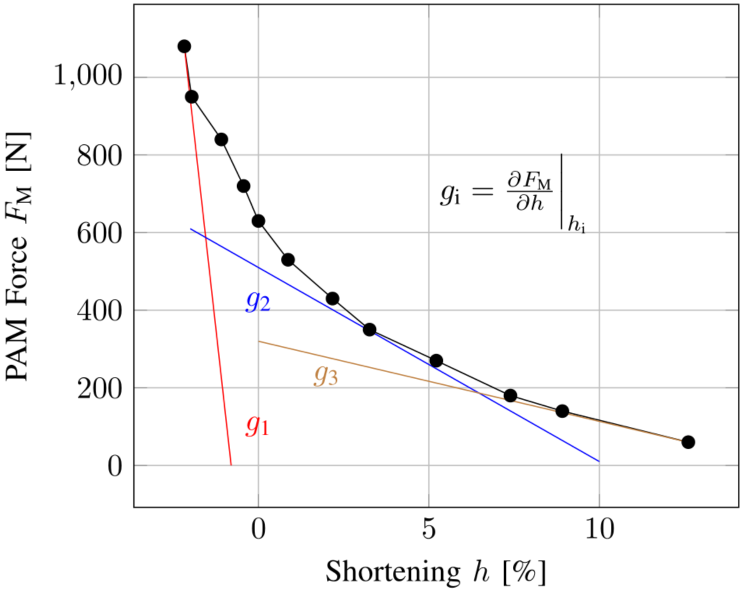


Figure 2.2: Discrete force-length relation $F_M(h, P)$ of an isolated PAM type DMSP-10-230N pneumatic muscle (black) at an operating pressure of $P = 4$ bar. The gradients g_1, g_2 and g_3 are parallel shifted force-length relations of tension springs, where g_1 is a very hard spring, g_2 is a medium hard spring and g_3 is a soft spring. The abscissa only counts to determine the gradients of the lines $g_1..g_3$, not for its h-value.

The rating of the spring stiffness is a trade-off between active and passive movement range. A very stiff spring will keep the active movement range of the PAM alone for the whole MSU but will add a very limited passive ability. A very soft spring will change the force-length characteristics of the MSU nearly to the characteristics of the spring and decrease the active movement range of the MSU to zero. In this case, the spring will compensate every active movement of the pneumatic muscle in a MSU, which transforms the MSU to an almost passive structure. Furthermore, only tension springs are reasonable using in series to the MSU setup from Figure 2.1, because the PAM can only contract. Fig. 2.2 visualizes three different spring stiffness for a type DMSP-10-230N PAM. The absolute value of the gradients of the force-length relation $g_1..g_3$ yield the spring stiffness $k_1..k_3$ ($k_i = |\dot{g}_i|$). The gradients of $g_1..g_3$ are only negative due to the illustration of the shortening h, according to equation ((2.2)). The lines $g_1..g_3$ of the tension springs are right shifted until they match $F_{\mathrm{M}}(h, P)$. To balance the active and passive abilities of the MSU for a defined pressure, it is necessary to find a spring stiffness k_{opt} which depicts the best compromise between active and passive range, dependent on the used pneumatic muscle. If there are no boundary conditions concerning the active or passive range, it is a good compromise to adapt the spring stiffness to the PAMs gradient of shortening $\frac{\partial F_{\mathrm{M}}}{\partial h}$ at its most nonlinear point. At this point, the deviation to the respective isolated PAM curve of $F_{\mathrm{M}}(h, P)$ is low. Therefore, active and passive range are balanced. The most nonlinear point of $F_{\mathrm{M}}(h, P)$ is at the inflection point of its first derivative and therefore at the maximum of its second derivative. We define this point as $F_{\mathrm{M}}(h_{\mathrm{nlin}}, P)$ at the shortening h_{nlin} for a certain pressure P. Hence, h_{nlin} can be calculated by $\frac{\partial^3 F_{\mathrm{M}}}{\partial h^3} = 0$. With h_{nlin} the optimized spring stiffness k_{opt} can be calculated as:

$$k_{\mathrm{opt}} = |\frac{\partial F_{\mathrm{M}}}{\partial h}(h_{\mathrm{nlin}}, P)|$$

Thus, k_{opt} is pressure dependent and not a fixed point over all operating pressures. Finding a solution in the length-dimension still results in a set of $k_{\mathrm{opt}}(P)$-lines for the pressure-domain. A reasonable spring rate for an applied pressure range is given by an average determination of all $k_{\mathrm{opt}}(P)$-curves. Another method to find $k_{\mathrm{opt}}(P_x)$ for a specified pressure P_x could be the the calculation of a line $G(P_x)$ with the least squares of $F_{\mathrm{M}}(h, P_x)$. Thus, the spring rate $k_{\mathrm{opt}}(P_x)$ equals the absolute value of $G(P_x)$ ($k_{\mathrm{opt}}(P_x) = |\dot{G}(P_x)|$). This method is only beneficial for higher pressure operating ranges, because otherwise $F_{\mathrm{M}}(h, P)$ is small for almost the entire shortening range [20], which results in a very flat line $G(P)$ and therefore a too soft stiffness. Taken together, the spring stiffness affect the overall characteristics of the MSU essentially and is therefore a determinable element in an AAS.

2.3 Active and passive characteristics

The violet range bars (Fig. 2.3) show the active movement range for all operating pressures in 500 N steps. The blue, red, brown and black range bars show the passive movement range at the respective operating pressure and a tensile load up to $F_{\mathrm{M}} = 1k\,$N. The isolated PAM (top) has an active movement range of more than 20% of its nominal length under lower tensile load ($F_{\mathrm{M}} \leq 500\,\mathrm{N}$) and less than 15% at forces $F_{\mathrm{M}} \geq 1.5\,$kN (extrapolation of $F_{\mathrm{M}}(h, P)$) within a pressure range up to $P = 7\,$bar. Its passive movement range is below 12% at forces $F_{\mathrm{M}} \geq 1\,$kN which decrease rapidly with low pressure values. The active movement range of the MSU (bottom) amounts to approximately 5% of its nominal length for all operating pressures. Its passive movement range is constantly 8% of its nominal length at forces $F_{\mathrm{M}} \geq 1\,$kN for all operating pressures. The stiff mounting parts are included in the shown results which significantly affect the active and passive abilities by reducing them proportional to L_{R}. The force-length relation of the MSU has a shift in the range of shortening compared to the isolated PAM. This shift illustrates the gain of passive abilities at the cost of active movement range (Fig. 2.3). With zero pressure, the PAM can only be stretched up to 3% (diameter dependent) of its nominal length by forces of $F_{\mathrm{M}} \geq 1\,$kN [12, 21], whereas the MSU crosses this line below an actual force of $F_{\mathrm{M}} = 500\,$N. Hence, an appreciable passive range of approximately 10% of the isolated PAM only exists above operating pressures of $P \geq 3\,$bar. The MSU has a passive movement range of approximately $h = 8\,...10\%$ for all operating pressures and is, compared to the passive range of the PAM, constantly higher over the whole pressure range. However, the maximum active shortening of the MSU is lower than its respective isolated PAM by approximately $10\,...15\%$. The low active range of the MSU is essentially caused by the high length L_{R} of the stiff connection elements which effects the total length L_{MSU} of the MSU by equation ((2.1)). In summary, the reduced active and increased passive range of the MSU is kept constant over the pressure range while the active and passive range of the isolated PAM is highly pressure dependent.

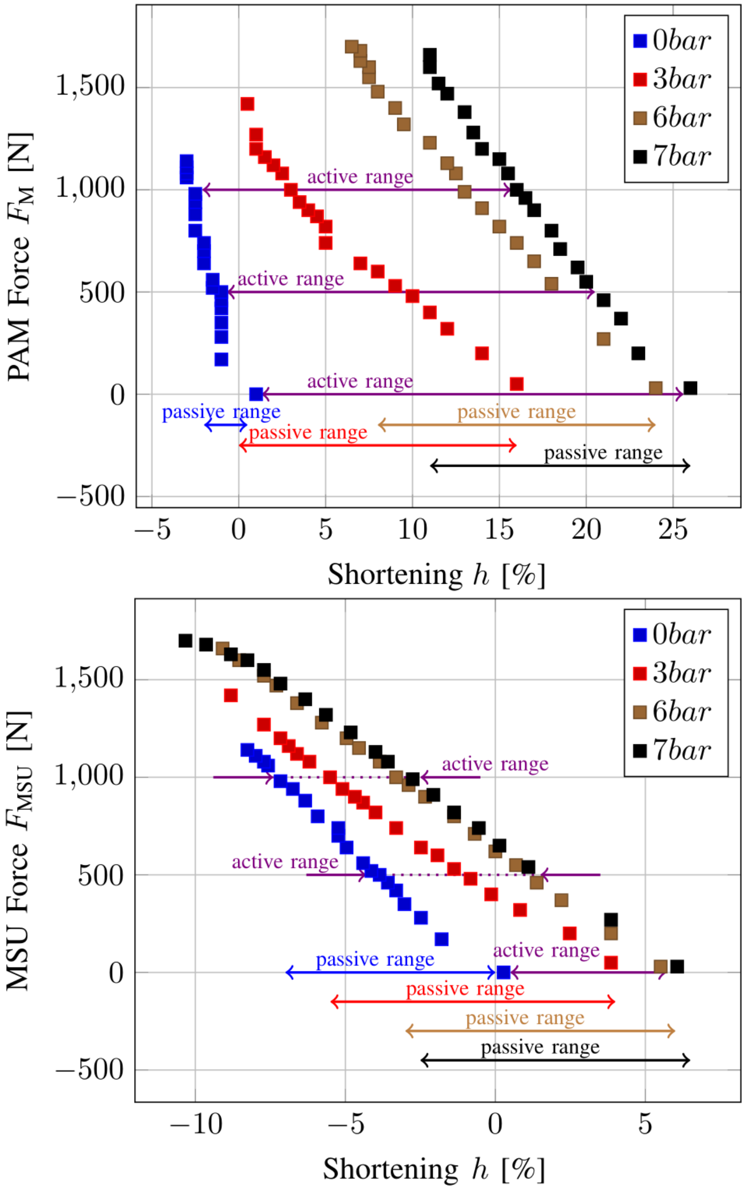


Figure 2.3: Measured force-length relation of a isolated PAM (top) and a MSU (bottom) in four pressure states for a type DMSP-20-100N PAM and a spring with a stiffness of $k_S = 38\,\frac{\mathrm{N}}{\mathrm{mm}}$.

2.4 AAS geometry of the SH1 upper ankle

Fig. 2.4 illustrates the SH1 upper ankle joint with two MSUs, representing the m. soleus (index S) and the m. tibialis anterior (index T) in an AAS. From geometrical constraints, the total muscle lengths can be expressed by

$$L_{\text{MSU}}(\psi) = \sqrt{(L_{\text{O}}^2 + L_{\text{I}}^2 - 2L_{\text{O}}L_{\text{I}}\cos(\psi)}, \tag{2.3}$$

where

$$\psi_{\text{T/S}}(\theta) = 180^\circ - \alpha_{\text{T/S}} - \beta_{\text{T/S}} \mp \theta \tag{2.4}$$

is the effective muscle angle. The distances L_{O} and L_{I} of the muscle's origin and insertion points to the joint axis, as well as the mechanical angles α and β are constant over any motion of the joint angle θ. The moment arms $r_{\text{MSU}} := \frac{\partial L_{\text{MSU}}}{\partial \theta}$ are given by the derivative of ((2.3)) with respect to the joint angle θ:

$$r_{\text{MSU}}(\psi) = \frac{L_{\text{O}}L_{\text{I}}\sin(\psi)}{\sqrt{L_{\text{O}}^2 + L_{\text{I}}^2 - 2L_{\text{O}}L_{\text{I}}\cos(\psi)}}. \tag{2.5}$$

The range of motion (ROM) of an AAS driven joint is depending on all acting MSUs, its angle dependent moment arms ((2.5)) and its MSU parameters L_{R}, $L_{\text{M,nom}}$, and k_{S}.

The benefits of the trade-off for less active and increased passive range of a MSU becomes clear when the MSUs are mounted in an AAS. Fig. 2.5 shows the ROM of the AAS driven upper ankle joint of SH1. By activating an agonist of the upper ankle its antagonist gets stretched, resulting in a joint motion of θ.

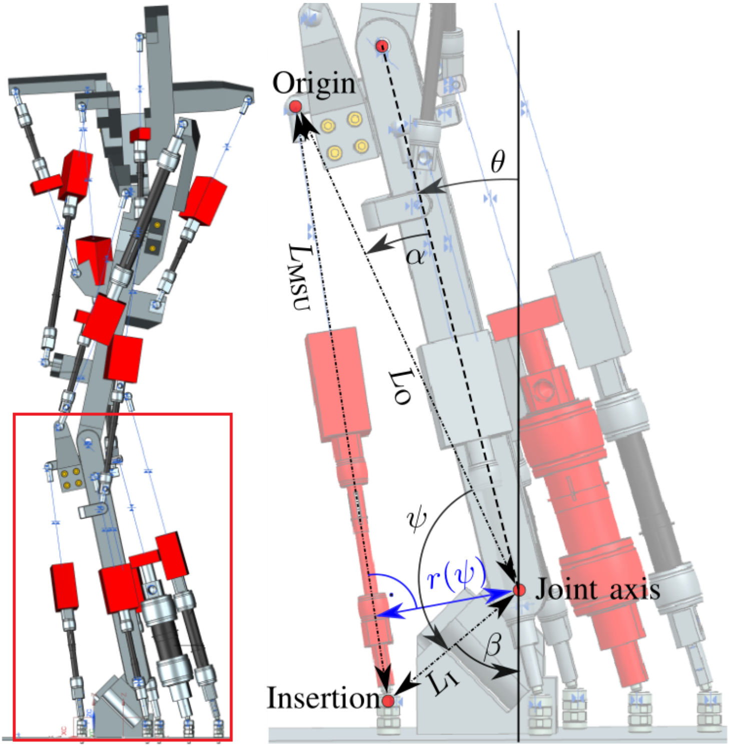


Figure 2.4: Sagittal view of the CAD model of SH1 showing the length L_{MSU} of the m. tibialis anterior and its moment arm. Both, the m. tibialis anterior and the m. soleus (red) are the mono-articular players of the upper ankle.

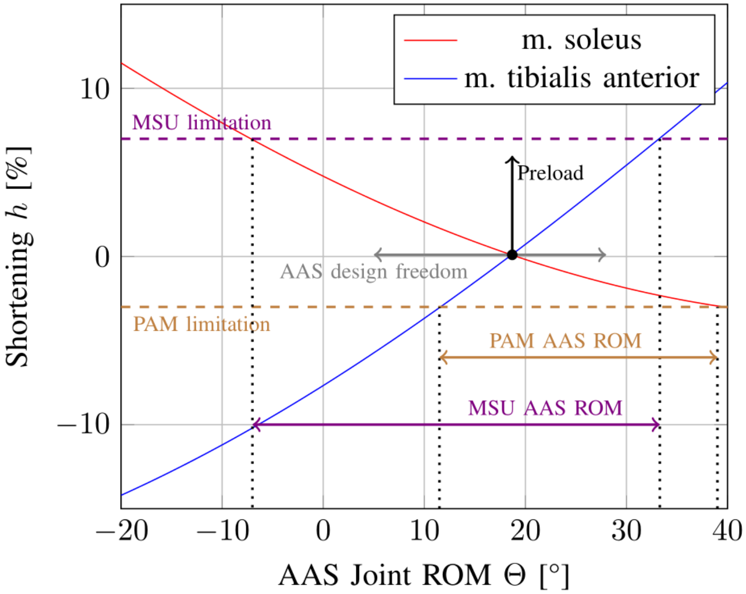


Figure 2.5: The shortening of the upper ankle AAS of fig. 2.4 over the joint ROM.

The AAS ROM is constrained by the limitations of the active and passive abilities of the used actuators. The ROM of an AAS using only PAMs is limited by its passive ranges, while its active range can not be fully exploited. A joint angle change by contracting the agonist of an AAS is only possible if the passive ability of the antagonist allows it. Therefore, both a PAM and a MSU driven AAS have joint ROM limitations. While a MSU driven AAS joint ROM is limited by its active range, a PAM driven AAS with same length and stiff mounting parts is limited by its passive range. The limitations (dashed lines) are shown according to the measurements of fig. 2.3 which are a 3 % passive range limitation (brown) of the PAM and an averaged 7 % active range limitation (violet) of the MSU. Out of this limitations, the AAS joint ROM can be determined. A preloading of the AAS will result in an offset of the shown curves, which can increase the joint ROM of a PAM driven AAS. The geometrical joint design will shift the zero joint position (black dot) over the joint ROM (gray), while the moment arm will affect the gradient of the curves. Furthermore, the 3 % passive ability of an antagonistic PAM can only be reached with forces > 1 kN under zero pressure and forces $\gg 1$ kN for other pressures due to the nonlinear growth of the passive range of a PAM (fig. 2.3). This can not be achieved by the agonist of a bidirectional AAS driven joint and hence, the marked ROM of fig. 2.5 is significantly smaller in reality when using isolated PAMs. By using MSUs in an AAS, the active ranges are reduced and the passive ranges are increased, which results in an overall increased ROM. The only solution of a PAM driven AAS to achieve an approximate high ROM as a MSU driven AAS, is to preload the PAM driven setup. By this, the kinetic adjustment possibilities, e.g. joint torque and stiffness are considerably reduced.

2.5 Kinetics of the MSUs in an AAS

2.5.1 Force law of single MSUs

The force law of an isolated PAM is dependent on pressure P, as well as on its length L_{M}, and can be described by the model created by Chou and Blake [17]:

$$F_{\mathrm{M}}(P, L_{\mathrm{M}}) = \frac{Pb^2}{4\pi n^2}\left(\frac{3L_M^2}{b^2} - 1\right), \tag{2.6}$$

with b and n being specific parameters of the PAMs geometry [17]. Due to the PAM's serial connection to a spring with the linear force law $F_{\mathrm{S}} = k_{\mathrm{S}}\Delta L_{\mathrm{S}}$, the total force of an MSU satisfies $F_{\mathrm{MSU}} = F_{\mathrm{M}} = F_{\mathrm{S}}$. To obtain a force law of the total MSU, the length of the PAM is expressed by L_{MSU}, L_R and ΔL_{S}, according to ((2.1)), and the force equilibrium is used to substitute ΔL_{S}:

$$F_{\mathrm{MSU}}(P, L_{\mathrm{MSU}}) = \frac{Pb^2}{4\pi n^2}\left(\frac{3(L_{\mathrm{MSU}} - \frac{F_{\mathrm{MSU}}}{k_{\mathrm{S}}} - L_{\mathrm{R}})^2}{b^2} - 1\right). \tag{2.7}$$

Considering only positive solutions of (2.7) (the MSU can create only tensile force), the resulting force law of a single MSU is given by

$$F_{\mathrm{MSU}}(P, L_{\mathrm{MSU}}) = \frac{3P(L_{\mathrm{MSU}} - L_{\mathrm{R}}) + 2\pi n^2 k_{\mathrm{S}}^2}{3P} + \frac{k_{\mathrm{S}}}{3P}\sqrt{12Pn^2\pi k_{\mathrm{S}}(L_{\mathrm{MSU}} - L_{\mathrm{R}}) + 3P^2b^2 + 4\pi^2 n^4 k_{\mathrm{S}}^2} \qquad (2.8)$$

2.5.2 Resulting torques and variable stiffness in the SH1 ankle

The kinetic effects of the two MSUs, representing the muscles m. tibialis anterior and m. soleus in the upper ankle joint of the SH1 can hence be analyzed. With the force law (2.8) for a single MSU and its moment arm ((2.5)), the joint torque is given by $T = r_{\mathrm{MSU}}F_{\mathrm{MSU}}$. The resulting AAS torque in the ankle joint of SH1 is

$$T_{\mathrm{A}}(P_{\mathrm{T}}, P_{\mathrm{S}}, \theta) = r_{\mathrm{T}}F_{\mathrm{T}} - r_{\mathrm{S}}F_{\mathrm{S}}. \qquad (2.9)$$

An equilibrium pose of the ankle joint can be achieved if all resulting torques vanish, such that $T_{\mathrm{A}} = T_{\mathrm{grav}}$. For simplicity, we examine situations of significant co-contraction with $T_{\mathrm{grav}} \ll T_{\mathrm{T}}$, T_{S}. With this assumption, the resulting torques of m. tibialis anterior and m. soleus are equal ($T_{\mathrm{T}} = T_{\mathrm{S}}$) in an equilibrium pose. Figure 2.6 shows the torque generation with respect to the ankle joint angle θ for both muscles in the constant pressure setups of 0.1 bar and 7 bar for each muscle. The surface in-between the 0.1 bar and 7 bar lines (gray) represents a torque range of the MSU that can be employed for different pressure setups considering its active range limitations from fig. 2.5. An unpreloaded AAS with PAM's has a reduced torque adjustment range (dashed PAM area) considering its passive range limitations.

Note, that for a desired joint angle θ^* within the range, for which torque equilibrium can be achieved, the solution for the individual pressure val-

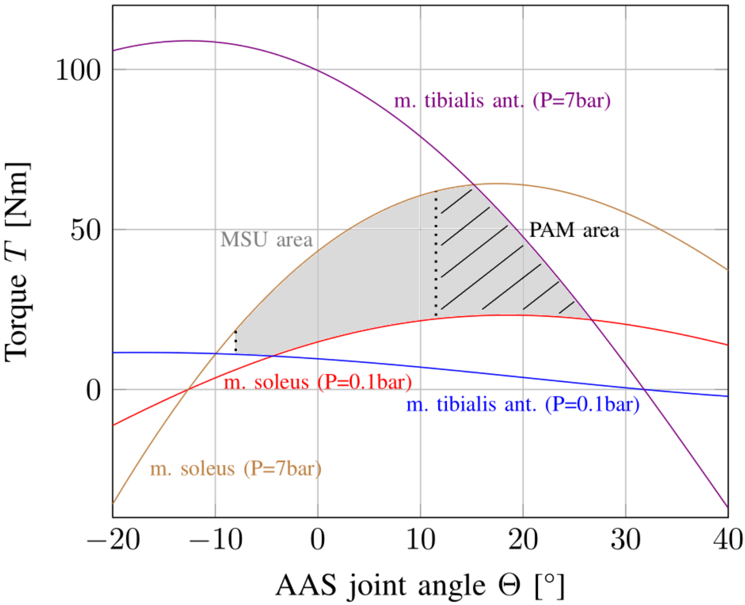


Figure 2.6: Resulting torques on the SH1 ankle joint for different pressure settings of the m. tibialis and m. soleus.

ues is not unique. In other words, there is a manifold of pressure values P_{T} and P_{S} to achieve torque equilibrium for θ^*. This allows to additionally choose the stiffness within a range at θ^*. To further analyze this, we firstly consider the definition of the stiffness $K_{\mathrm{MSU}} := \frac{\partial F_{\mathrm{MSU}}}{\partial L_{\mathrm{MSU}}}$ of a single isolated MSU. With [107]

$$K_{\mathrm{PAM}}(P, L_{\mathrm{M}}) = \frac{3PL_{\mathrm{M}}}{2\pi n^2}$$

the total stiffness of a single MSU can be calculated by $\frac{1}{K_{\mathrm{MSU}}} = \frac{1}{K_{\mathrm{PAM}}} + \frac{1}{k_{\mathrm{S}}}$:

$$K_{\mathrm{MSU}} = \frac{3PL_{\mathrm{M}}k_{\mathrm{S}}}{3PL_{\mathrm{M}} + 2\pi n^2 k_{\mathrm{S}}}. \tag{2.10}$$

The resulting ankle joint stiffness of SH1 is $S := \frac{\partial T_{\mathrm{A}}}{\partial \theta}$. With ((2.9)), and the product rule, the stiffness is expressed by

$$S_{\mathrm{A}}(P_{\mathrm{T}}, P_{\mathrm{S}}, \theta) = \frac{\partial r_{\mathrm{T}}}{\partial \theta} F_{\mathrm{T}} + r_{\mathrm{T}} \frac{\partial F_{\mathrm{T}}}{\partial \theta} - \frac{\partial r_{\mathrm{S}}}{\partial \theta} F_{\mathrm{S}} - r_{\mathrm{S}} \frac{\partial F_{\mathrm{S}}}{\partial \theta}. \tag{2.11}$$

The derivative of a muscle's moment arm $r_{\mathrm{MSU}}(\theta)$ (equation 2.5) with respect to the joint angle $\psi(\theta)$ (equation. (2.4)) is

$$\frac{\partial r_{\mathrm{MSU}}}{\partial \psi} = \frac{L_{\mathrm{I}} L_{\mathrm{O}} (L_{\mathrm{I}} \cos(\psi) - L_{\mathrm{O}})(L_{\mathrm{O}} \cos(\psi) - L_{\mathrm{I}})}{(L_{\mathrm{I}}^2 + L_{\mathrm{O}}^2 - 2 L_{\mathrm{I}} L_{\mathrm{O}} \cos(\psi))}. \tag{2.12}$$

The derivative of F_{MSU} with respect to θ can be calculated by using the definitions of the moment arm $r_{\mathrm{MSU}} := \frac{\partial L_{\mathrm{MSU}}}{\partial \theta}$ and of the muscle stiffness $K_{\mathrm{MSU}} := \frac{\partial F_{\mathrm{MSU}}}{\partial L_{\mathrm{MSU}}}$:

$$\frac{\partial F_{\mathrm{MSU}}}{\partial \theta} = \frac{\partial F_{\mathrm{MSU}}}{\partial L_{\mathrm{MSU}}} \frac{\partial L_{\mathrm{MSU}}}{\partial \theta} = K_{\mathrm{MSU}} r_{\mathrm{MSU}}. \tag{2.13}$$

That is, we can formulate the joint stiffness in closed form , by combining ((2.10)-(2.13)):

$$S_{\mathrm{A}}(P_{\mathrm{T}}, P_{\mathrm{S}}, \theta) = \frac{\partial r_{\mathrm{T}}}{\partial \theta} F_{\mathrm{T}} + r_{\mathrm{T}}^2 K_{\mathrm{T}} - \frac{\partial r_{\mathrm{S}}}{\partial \theta} F_{\mathrm{S}} - r_{\mathrm{S}}^2 K_{\mathrm{S}}. \tag{2.14}$$

Figure 2.7 shows the joint stiffness versus the joint angle θ for four different pressure setups. It illustrates the high range of adjustable joint stiffness of an AAS with MSUs. In summary, the MSUs enable a high ROM of an AAS but not at the cost of a reduced adjustable stiffness or torque range as AAS with PAMs do.

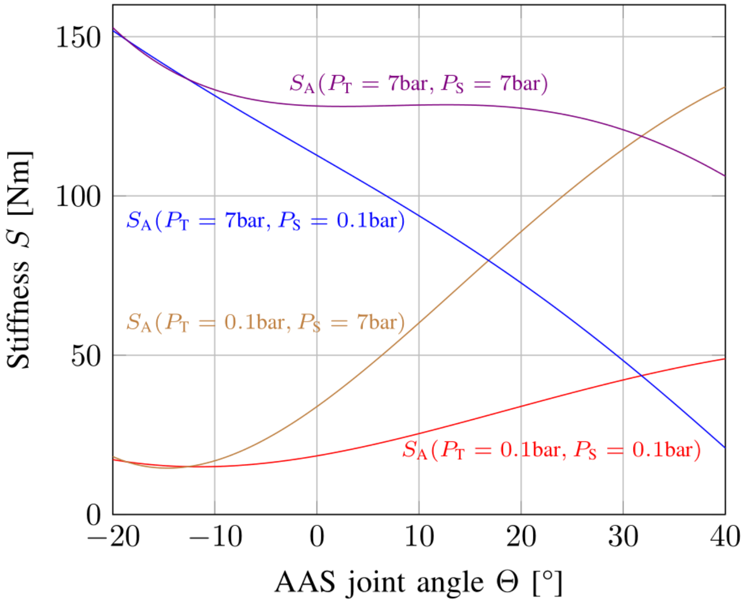


Figure 2.7: Variable joint stiffness of the ankle joint of SH1. Increasing the pressure values in the MSU results in higher stiffness of the joint. All joint stiffness in between the lines can be achieved by an AAS with MSUs.

Chapter 3

Controller

3.1 Closed Loop Controller

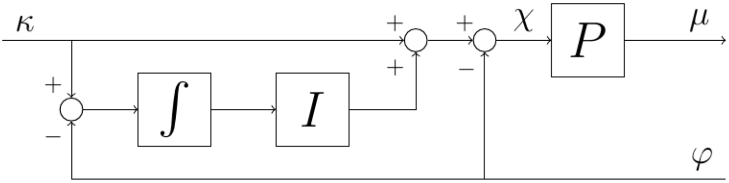


Figure 3.1: Low level control scheme. The control circuit has PI- characteristics (proportional (P) and integral (I)) with an interacting control structure. The control matrix $\boldsymbol{P}$ transforms the resulting desired joint vector $\boldsymbol{\chi}$ into muscle activation $\boldsymbol{\mu}$.

The control law for the muscle stimulation $\boldsymbol{\mu}$ is given by:

$$\boldsymbol{\mu} = \boldsymbol{P}\boldsymbol{\chi} \quad with \quad \boldsymbol{\chi} = \boldsymbol{\kappa} - \boldsymbol{\varphi} + \boldsymbol{I} \int (\boldsymbol{\kappa} - \boldsymbol{\varphi}) \partial t \tag{3.1}$$

The control law is implementation in Matlab and is given by the code listing 3.1. The control matrix $\boldsymbol{P}$ is an $j \times k$ matrix with j as the number of muscles M and k as the number of joints Φ.

$$\boldsymbol{P} = \begin{bmatrix} p_{11} & \cdots & p_{1k} \\ \vdots & \ddots & \vdots \\ p_{j1} & \cdots & p_{jk} \end{bmatrix} \qquad \boldsymbol{P} \rightarrow \mathcal{A} : \boldsymbol{P} = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \\ p_{31} & p_{32} \\ p_{41} & p_{42} \\ p_{51} & p_{52} \end{bmatrix} \tag{3.2}$$

The interacting control structure reduces the size of the integrator matrix $\boldsymbol{I}$ compared to a classic Ideal-PI control structure(fig. 3.2), where the $\boldsymbol{P}$ and $\boldsymbol{I}$ matrices are parallel resulting in the same $j \times k$ matrix size.

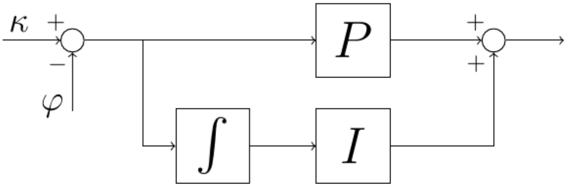


Figure 3.2: Ideal-PI control structure. The matrix $\boldsymbol{P}$ and $\boldsymbol{I}$ have the same size resulting in the same amount of matrix elements.

The $\boldsymbol{I}$-matrix in this controller is only a vector of size k:

$$\boldsymbol{I} = \begin{bmatrix} i_1 \\ \vdots \\ i_k \end{bmatrix} \qquad \boldsymbol{I} \rightarrow \mathcal{A} : \boldsymbol{I} = \begin{bmatrix} i_1 \\ i_2 \end{bmatrix} \tag{3.3}$$

The change from Ideal-PI control to the interacting control structure used has consequences: In an Ideal-PI control, an integration parameter can be set for each muscle and each joint. This enables the possibility for different integration behavior of each single muscle. In the interacting form (fig.3.1), the integration value acts like an offset to the desired joint value. Hence, the integration behavior can now be set up for joints instead of muscles resulting in a parameter reduction of $\boldsymbol{I}$ (equation (3.3)). Considering that no actuator model can be used to calculate parameters, parameter reduction by design is a crucial part of the control concept of this control framework.

3.1.1 Parameter determination

The parameters of the control matrix $\boldsymbol{P}$ can be determined by information the investigated system provides.

Muscle arrangement

By knowing which joints are effected by which muscle, we can decrease the number of p-parameters. Hence, not every muscle effect every joint. Considering that, the $\boldsymbol{P}$-matrix for the control system of $\mathcal{A}$ results in:

$$\boldsymbol{P} \to \mathcal{A} : \boldsymbol{P} = \begin{bmatrix} p_{11} & 0 \\ p_{21} & 0 \\ p_{31} & p_{32} \\ 0 & p_{42} \\ 0 & p_{52} \end{bmatrix} \tag{3.4}$$

According to the control law (equation (3.1)), the parameters p_{12}, p_{22}, p_{41}, p_{51} have to equal zero because they must not have effect to the joint in question.

Muscle operating principle

The MSU used can only contract, and therefore, can only act in the same way like biological muscles. In an antagonistic setup each muscle enables motion in one direction of an effected joint. By defining positive joint directions (see chapter 1.2.1), the sign of the resulting p-parameters can be determined:

$$\boldsymbol{P} \rightarrow \mathcal{A} : \boldsymbol{P} = \begin{bmatrix} -|p_{11}| & 0 \\ +|p_{21}| & 0 \\ +|p_{31}| & +|p_{32}| \\ 0 & -|p_{42}| \\ 0 & +|p_{52}| \end{bmatrix} \tag{3.5}$$

Muscle dependencies

Considering an elementary antagonistic setup (AAS) $\mathcal{H}$ with two muscles acting on a single joint, which is used in many publications [2, 107, 4]. The $\boldsymbol{P}$-matrix would be:

$$\boldsymbol{P} \rightarrow \mathcal{H} : \boldsymbol{P} = \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} \tag{3.6}$$

The control parameters for position control of the two muscles are not independent from each other and hence are physically connected over the joint. By using the exact same muscles with the same lever arms, the p-parameters would be:

$$p_1 = -p_2 \tag{3.7}$$

Hence, the two muscles are antagonistically arranged one muscle will affect the joint in positive and the other in negative direction which determines the sign of the parameter. By changing the properties of only one actuator, e.g. contraction range or lever arm, the equation (3.7) has to be enlarged to:

$$p_1 = -p_2 \cdot w(\boldsymbol{b}) \tag{3.8}$$

$\boldsymbol{b}$ is a set of parameters dependent of actuator and arrangement properties. The function $w(\boldsymbol{b})$ represents the unsymmetrical correlation of the two muscles within $\mathcal{H}$. $w(\boldsymbol{b})$ is defined as the correlation function.
$w(\boldsymbol{b}) < 0$ if the muscles are antagonists and $w(\boldsymbol{b}) > 0$ if they are synergists. $|w(\boldsymbol{b})| = 1$ if the AAS is symmetric (equal muscles equally arranged) and $w(\boldsymbol{b}) = 0$ if the two muscles do not have a joint in common (equation (3.7)). In short, The correlation function defines the ratio of two $\boldsymbol{P}$-parameters and its sign is determined by the role of the muscles, antagonist or synergist. In $\mathcal{H}$ it is:

$$\frac{p_1}{p_2} = -w(\boldsymbol{b}) \tag{3.9}$$

If identical muscles are in use, but unsymmetrically arranged, $w(\boldsymbol{b})$ can be reduced to the ratio between the lever arms $\ell_1(\boldsymbol{b})$ and $\ell_2(\boldsymbol{b})$ of the muscles M_1 and M_2:

$$w(\boldsymbol{b}) = \frac{\ell_2(\boldsymbol{b})}{\ell_1(\boldsymbol{b})} \tag{3.10}$$

Hence, a better lever arm leads to a higher motion impact, the correlation function determining the ratio of control parameters have to be reciprocal to keep the balance between antagonists. A higher joint impact of a muscle must result in a lower control parameter $\boldsymbol{p}$.
If the lever arms of (3.10) are linear, they can be calculated out of the geometry. In this case, $\boldsymbol{b}$ is a set of geometric parameters. Non-linear lever arms $\ell(\boldsymbol{b}, \boldsymbol{\varphi})$ have to be linearized over the range of $\boldsymbol{\varphi}$ used or result in nonlinear $p(\boldsymbol{\varphi}) \in \boldsymbol{P}$ if $|w(\boldsymbol{b})| \neq 1$. Unbalance caused by muscle properties, e.g. muscle length or maximum force, can be solved analog. Hence, $w(\boldsymbol{b})$ represents the unsymmetrical correlation. If $\mathcal{H}$ is a setup with unequal muscles and unsymmetrical lever arms, $w(\boldsymbol{b})$ must be enlarged to:

$$w(\boldsymbol{b}) \to \mathcal{Q} : w(\boldsymbol{b}) = \frac{\ell_2(\boldsymbol{b}) \cdot f_2(\mu_2, \lambda_2)}{\ell_1(\boldsymbol{b} \cdot f_1(\mu_1, \lambda_1))} = \frac{T_2(\boldsymbol{b}_2, \mu_2, \lambda_2)}{T_1(\boldsymbol{b}_1, \mu_1, \lambda_1)} \tag{3.11}$$

where $f(\mu, \ell)$ represents the force-length-pressure relation of the actuator. Pressure is represented by the actuation μ and the actual length of the muscle is represented by λ. According to the Law of the Lever, $l_j(\boldsymbol{b}) \cdot f_j(\mu_j, \ell_j)$ equals the torque T_j that the actuator j can create. It is possible to consider $|w(\boldsymbol{b})| = 1$ in $\mathcal{H}$ when $T_1(\boldsymbol{b}_1, \mu_1, \ell_1) \approx T_2(\boldsymbol{b}_2, \mu_2, \ell_2)$ over the range of possible μ_j. That is the case when a stronger lever arm compensates its weaker actuator. Hence, it is not necessary to know $f_j(\mu_j, \ell_j)$ but only an approximation of the ratio between $l_1(\boldsymbol{b}) \cdot f_1(\mu_1, \ell_1)/l_2(\boldsymbol{b} \cdot f_2(\mu_2, \ell_2))$ in $\mathcal{H}$.

Matrix dependencies

Correlation functions only exist among muscles with a joint in common. According to equation (3.5) and (3.9), a set of correlation functions $w(\boldsymbol{b}) \to \mathcal{A}$ can be found as followed:

$$\frac{p_{11}}{p_{21}} = -w_{11,21}(\boldsymbol{b}) \tag{3.12}$$

$$\frac{p_{11}}{p_{31}} = -w_{11,31}(\boldsymbol{b}) \tag{3.13}$$

$$\frac{p_{42}}{p_{32}} = -w_{42,32}(\boldsymbol{b}) \tag{3.14}$$

$$\frac{p_{52}}{p_{42}} = -w_{52,42}(\boldsymbol{b}) \tag{3.15}$$

The correlation functions $w(\boldsymbol{b})$ of this set can be determined via muscle dependencies as shown before. They are only an approximation of the ratio between two antagonistic muscles based on information of their ability to generate torque. With this set of correlation functions, the elements of each row of the matrix (3.5) can be calculated out of one parameter:

$$p_{21} = -\frac{p_{11}}{w_{11,21}(\boldsymbol{b})} \tag{3.16}$$

$$p_{31} = -\frac{p_{11}}{w_{11,31}(\boldsymbol{b})} \tag{3.17}$$

for the first row of $\boldsymbol{P}$ (out of equation (3.12) and (3.13)) and

$$p_{42} = -p_{32} \cdot w_{42,32}(\boldsymbol{b}) \tag{3.18}$$

$$p_{52} = -p_{42} \cdot w_{52,42}(\boldsymbol{b}) \tag{3.19}$$

for its second row (out of equation (3.14) and (3.15)). As a result of the correlation functions, the parameter determination for the matrix $\boldsymbol{P}$ is reduced to one for each row. On a physical level, this means that we need to find only one parameter for each joint, independent of the number of actuators effecting this joint.
Ataro has a bi-articular muscle (M3) which effects both joints of $\mathcal{A}$ according to (3.5). Both control parameters p_{31} and p_{32} effect the same actuator. Hence, the ratio of possible torque generation to the joints determines the ratio of the parameters:

$$\frac{p_{31}}{p_{32}} = \frac{\ell_{32}(\boldsymbol{b}) \cdot f(\mu, \lambda)}{\ell_{31}(\boldsymbol{b} \cdot f(\mu, \lambda))} = \frac{\ell_{32}(\boldsymbol{b})}{\ell_{31}(\boldsymbol{b})} \tag{3.20}$$

The ratio (3.20) has a positive sign because the muscle M3 affect both joints in positive direction according to (3.5). $f(\mu, \lambda)$ can be cut out because it is the same muscle. Therefore, p_{32} is given by p_{32} multiplied with the reciprocal ratio of its movement arms:

$$p_{32} = p_{31} \cdot \frac{\ell_{31}(\boldsymbol{b})}{\ell_{32}(\boldsymbol{b})} \tag{3.21}$$

$\ell_{jk}(\boldsymbol{b})$ is the movement arm of muscle (M_j) to joint Φ_k.With the bi-articular muscle arrangement of $\mathcal{A}$, all parameters of the matrix $\boldsymbol{P}$ can be represented by only one parameter p_{11} and the correlation functions. p_{21} and p_{31} are given by (3.16) and (3.17). p_{32} is given by the ratio (3.21) and (3.17):

$$p_{32} = p_{31} \cdot \frac{\ell_{31}(\boldsymbol{b})}{\ell_{32}(\boldsymbol{b})} = -p_{11} \cdot \frac{\ell_{31}(\boldsymbol{b})}{w_{11,31}(\boldsymbol{b}) \cdot \ell_{32}(\boldsymbol{b})} \tag{3.22}$$

With equations (3.18) and (3.22) the parameter p_{42} is given by:

$$p_{42} = p_{11} \cdot \frac{w_{42,32}(\boldsymbol{b}) \cdot \ell_{31}(\boldsymbol{b})}{w_{11,31}(\boldsymbol{b}) \cdot \ell_{32}(\boldsymbol{b})} \tag{3.23}$$

Out of equations (3.19) and (3.23) the parameter p_{42} is given by:

$$p_{52} = -p_{11} \cdot \frac{w_{42,32}(\boldsymbol{b}) \cdot w_{52,42}(\boldsymbol{b}) \cdot \ell_{31}(\boldsymbol{b})}{w_{11,31}(\boldsymbol{b}) \cdot \ell_{32}(\boldsymbol{b})} \tag{3.24}$$

As a result, using bi-articular muscle arrangements, which connect all joints, enables control parameter reduction to a single parameter, independent from the amount of muscles:

$$\boldsymbol{P} \rightarrow \mathcal{A} : \boldsymbol{P} = p_{11} \begin{bmatrix} -1 & 0 \\ \frac{1}{w_{11,21}(\boldsymbol{b})} & 0 \\ \frac{1}{w_{11,31}(\boldsymbol{b})} & \frac{\ell_{31}(\boldsymbol{b})}{w_{11,31}(\boldsymbol{b}) \cdot \ell_{32}(\boldsymbol{b})} \\ 0 & -\frac{w_{42,32}(\boldsymbol{b}) \cdot \ell_{31}(\boldsymbol{b})}{w_{11,31}(\boldsymbol{b}) \cdot \ell_{32}(\boldsymbol{b})} \\ 0 & \frac{w_{42,32}(\boldsymbol{b}) \cdot w_{52,42}(\boldsymbol{b}) \cdot \ell_{31}(\boldsymbol{b})}{w_{11,31}(\boldsymbol{b}) \cdot \ell_{32}(\boldsymbol{b})} \end{bmatrix} \tag{3.25}$$

Parameter optimization

The control parameters p, i can be trained with a iterative learning approach using Gradient Descent:

$$\tilde{p}_{n+1} = \tilde{p}_n - \alpha \frac{\partial \tilde{e}}{\partial \tilde{p}} \qquad \tilde{i}_{n+1} = \tilde{i}_n - \alpha \frac{\partial \tilde{e}}{\partial \tilde{i}} \tag{3.26}$$

where the $\tilde{p}$ and $\tilde{i}$ values of the $n+1$ iteration are updated with a learning rate α . The cost-function $\tilde{e}$ is given by the square of the absolute error:

$$\tilde{e} = (\tilde{\varphi}_{target} - \tilde{\varphi}_{actual})^2 \tag{3.27}$$

The discrete derivative $\partial \tilde{e} / \partial \tilde{p}$ can be determined by:

$$\frac{\partial \tilde{e}}{\partial \tilde{p}} = \frac{\tilde{e}_n - \tilde{e}_{n-1}}{\tilde{p}_n - \tilde{p}_{n-1}} \tag{3.28}$$

where n is the actual iteration and $n-1$ is the previous iteration. $\partial \tilde{e} / \partial \tilde{i}$ can be determined respectively. Hence the system is considered as a black box where $\tilde{p}$ and $\tilde{i}$ parameters result in an error of $\tilde{\varphi}$, the SI units are not directly connected. Therefore a normalization is necessary. This can be done by normalizing the control parameters to its starting values:

$$\tilde{p}_n = \frac{p_n}{p_{start}} \qquad \tilde{i}_n = \frac{i_n}{i_{start}} \tag{3.29}$$

The resulting joint angles $\tilde{\varphi}_{actual}$ and the target joint angle $\tilde{\varphi}_{target}$ are normalized to the step size of the target transition:

$$\tilde{\varphi}_{target} - \tilde{\varphi}_{actual} = \frac{\varphi_{target} - \varphi_{actual}}{\varphi_{step}} \tag{3.30}$$

Furthermore a goal criteria has to be defined. On the basis of the goal criteria, the error can be calculated by using equation (3.27). A goal criteria can be a total compensation of the control deviation ($target - actual$) after a defined time. The goal criteria should be chosen optimistically hence it is not important to reach the goal. The control parameters p, i must only be optimized and therefore an optimistic goal criteria leads to a higher error resolution, resulting in more accurate parameter updates for each iteration.

The starting values for the control parameters p, i have to be estimated. In general, if the starting values are too high the system will become unstable. If the starting values are too low the system will act slowly which only increases the number of iterations for parameter optimization. From experience with muscle driven systems, a possible magnitude of p_{start} is:

$$p_{start} \leq magnitude\{t_s \cdot \frac{input}{output}\} \qquad with \qquad t_s = \frac{1}{f_s} \tag{3.31}$$

t_s is the sample time of the controlling system and f_s its sample frequency. Hence we do not have any behavioral information (time constants, order, etc.) about the control system, the starting values of the control parameters p, i have to be chosen appropriate to the control system properties and the magnitude between input and output values of the controller. The i-parameters affect only the time integration of $\kappa - \varphi$ which is additionally multiplied by the p-parameter (3.1). For low perturbed systems (only tare mass), it acts as a weight factor between the absolute control deviation $\kappa - \varphi$ and the integrated control deviation $I \int(\kappa - \varphi)\partial t$. From experience, a possible starting value for i_{start} is 10%:

$$20\% \leq i_{start} < 0\% \tag{3.32}$$

Ataro has two joints resulting in two i-parameters. A bi-articular muscle $M3$ connects both joints and reduces the p-parameter space to one. The control system of Ataro works at a sample rate of $f_s = 500$ Hz.

parameter	value	normalized parameter	normalized value
p_{start1}	$1 \cdot 10^{-4}$	$\tilde{p}_{start}$	1
p_{start2}	$2 \cdot 10^{-4}$	$\tilde{p}_{start}$	2
$i_{1,start}$	0.1	$\tilde{i}_{1,start}$	1
$i_{2,start}$	0.1	$\tilde{i}_{2,start}$	1

Table 3.1: The normalized starting parameters for the p-optimization.

Therefore, the sample time t_s in which data is red, processed and the outputs are updated is $t_s = 2$ ms. The possible joint angle ranges are within a magnitude of 100 degree and the outputs to drive the muscles are within a magnitude of 10 bars. According to (3.31) and (3.32) the starting values for the parameters are:

The iterative parameter training starts with optimizing the p-parameter. The controller (fig. 3.1) is set with the parameters of table 3.1 and performs a specified motion by changing the desired values of the controller input (fig. 3.3). The motion is a point-to-point task where the points are 56cm away from each other.

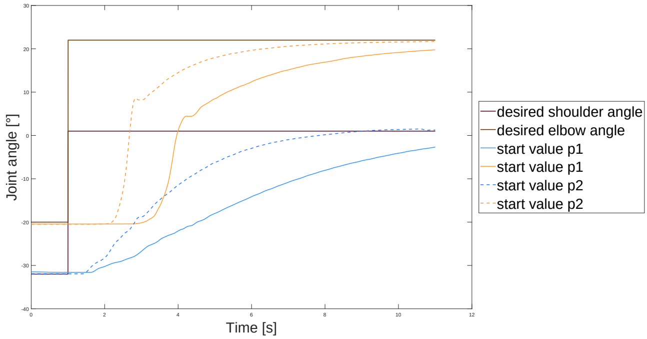


Figure 3.3: Joint angle control with applied start values.

To calculate the discrete derivative (3.28) a second starting value is needed. Hence the parameter p_{start} result in a duration $>$ 10 s to

iteration n	$\tilde{p}_n$	$\varphi_{desired}$	φ_{actual}	$\tilde{e}$	$\partial\tilde{e}/\partial\tilde{p}$
start 1	1	1°	-26.7305°	0.70613465	
start 2	2	1°	-18.7884°	0.35957831	-0.347
1	2.35	1°	-15.9504°	0.26383477	-0.276
2	2.62	1°	-14.1504°	0.21077559	-0.192
3	2.81	1°	-12.9003°	0.17742731	-0.174
4	2.99	1°	-11.8004°	0.15045936	-0.155
5	3.14	1°	-11.0517°	0.13337325	-0.110

Table 3.2: Intermediate data of the p-optimization.

achieve the desired position, the second start parameter p_{start2} has be chosen higher. With $p_{start2} = 2 \cdot 10^{-4}$ (dashed lines in fig. 3.3) the speed of the point-to-point motion increased. By defining a goal criteria, the normalized errors of both parameter sets can be calculated with equation (3.27). The defined goal criteria used is target reaching after 2 s. The next step is to calculate the discrete derivative (3.28) with the calculated errors $\tilde{e}_n$ and update the $\tilde{p}_n$-parameter to $\tilde{p}_{n+1}$ with (3.26). Repeat the process until there is no significant increase. Figure 3.4 illustrates the iterations of the p-optimization to an abort criteria of $\varphi_{n+1} - \varphi_n \leq 0.5°$ at $t_{goal} = t_{step} + 2$ s. Table

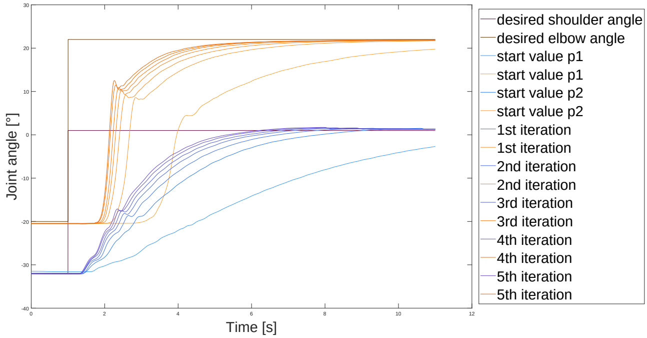


Figure 3.4: Joint angle control iteration for p-optimization

The p-parameter is only optimized with the shoulder joint. The elbow

iteration n	$\tilde{i}_n$	$\varphi_{desired}$	φ_{actual}	$\tilde{e}$	$\partial\tilde{e}/\partial\tilde{i}$
start 1	1	1°	-10.1904°	0.11499087	
start 2	1.5	1°	-3.463°	0.01829051	-0.193
1	1.69	1°	-1.9369°	0.00792046	-0.0536
2	1.75	1°	-1.5614°	0.00602458	-0.0354

Table 3.3: Intermediate data of the shoulder i-optimization.

joint is implicitly optimized by this procedure hence the P-Matrix (3.25) connects both joints with the p-parameter. If there are more than one p-parameters (autonomous joints without bi-articular muscles), the parameters have to be optimized consecutively.
The i-parameters are optimized equally (fig. 3.5 and 3.5). The integrator work on the joint level (fig. 3.1) which enables individual control behavior adjustment for each joint. Both i-parameter affect the bi-articular muscle. Therefore, tuning the shoulder i-parameter also have an impact on the elbow joint (fig. 3.5).

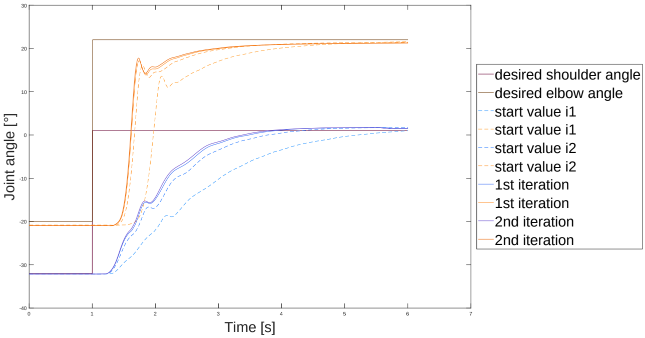


Figure 3.5: i-parameter optimization for the shoulder

Table 3.3 and 3.4 shows the intermediate data after each iteration while optimizing the i-parameters. The shoulder optimization was stopped after the second iteration. The increase in speed at $t_{goal} = t_{step} + 2$ s was below $0.5°$ for that iteration.

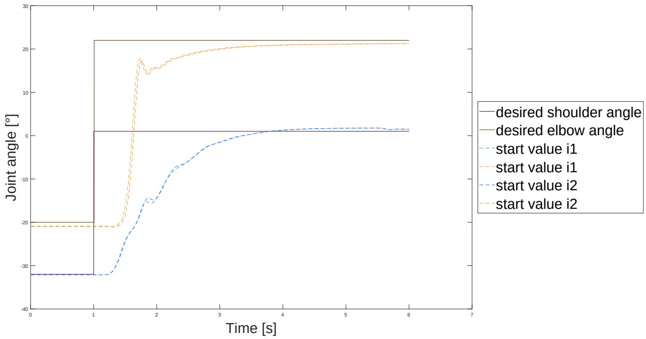


Figure 3.6: i-parameter optimization for the elbow

iteration n	$\tilde{i}_n$	$\varphi_{desired}$	φ_{actual}	$\tilde{e}$	$\partial\tilde{e}/\partial\tilde{i}$
start 1	1	22°	20.0799°	0.00209001	
start 2	1.2	22°	19.9606°	0.0023578	0.00134

Table 3.4: Intermediate data of the elbow i-optimization.

The starting values for the elbow i-iteration result in a performance increase below $0.5°$ at $t_{goal} = t_{step} + 2$ s. If this is the case, the starting value already matches the optimal parameter range for the chosen optimization method.

After parameter optimization, the new parameters are $p_{opt} = 3.14 \cdot 10^{-4}$, $i_{s,opt} = 1.75 \cdot 10^{-1}$ and $i_{e,opt} = 1.2 \cdot 10^{-1}$.

3.1.2 Closed-Loop Controller Implementation

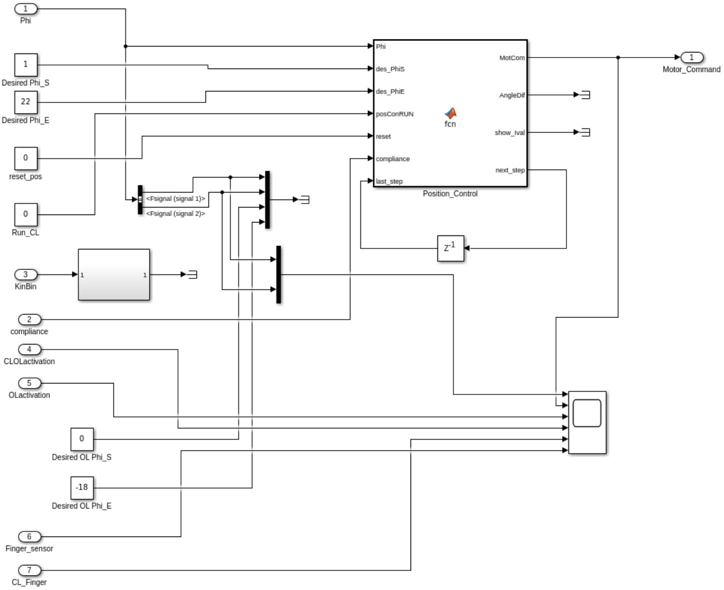


Figure 3.7: Closed-loop controller

The closed-loop controller is implemented as a Matlab function. The parameters are initialized by the parameter script explained in section 1.2.3.

Listing 3.1: closed-loop controller code

```
function [MotCom, AngleDif, show_Ival, next_step] = fcn(
    Phi, des_PhiS, des_PhiE,posConRUN, reset, compliance,
     last_step, PM_Pos, PC_IM, MRC_para)

% Joint angle values within:
% -43 (max flex) ... +25 (max ext) for shoulder
% -26 (max flex) ... +25 (max ext) for elbow

%%Initialization for each time step

last_Int = last_step(1:2,1);
```

```
desPhi = [des_PhiS; des_PhiE];
actPhi = [Phi(1); Phi(2)];

%Reset of the controller
if(reset == 1)
last_Int = last_Int*MRC_para(2,2);
end
if(reset == 2)
last_Int = [0;0];
end

%% Control Law

%Run conditions
if (posConRUN == 1)&&(compliance == 0)
difPhi = desPhi - actPhi;
else
difPhi = [0; 0];
end

dif = desFor - actFor;

if (abs(dif) <= Tresh)
    dif = 0;
end

%I-Controller
Integration = last_Int + difPhi;
I_dif = (Integration.*PC_IM) + difPhi;

F_dif = [dif; dif];
F_Ival = F_dif + Integration;
F_val = [desFor; desFor] + FC_IM*F_Ival - [actFor; actFor
    ];

if (dif == 0)
    F_Ival = F_Ival*0.9980;
end

%P-Controller
P_value = PM_Pos*I_dif;

%% Maximum Value
if (P_value(1) >= 6)
P_value(1) = 6;
end
if (P_value(2) >= 6)
```

```
P_value(2) = 6;
end
if (P_value(3) >= 6)
P_value(3) = 6;
end
if (P_value(4) >= 6)
P_value(4) = 6;
end
if (P_value(5) >= 6)
P_value(5) = 6;
end

%% Minimum Value
if (P_value(1) <= -6)
P_value(1) = -6;
end
if (P_value(2) <= -6)
P_value(2) = -6;
end
if (P_value(3) <= -6)
P_value(3) = -6;
end
if (P_value(4) <= -6)
P_value(4) = -6;
end
if (P_value(5) <= -6)
P_value(5) = -6;
end

%% Outgoings

MotCom = P_value;
next_step = Integration;
AngleDif = difPhi;
show_Ival = Phi;
```

3.2 Open-Loop Controller

The open-loop controller is implemented by applying the actuation $\boldsymbol{\mu}$ of an given equilibrium point to the muscle actuation output. By changing the target equilibrium point, the difference between the old actuation and the new actuation is calculated and decreased over time (code listing 3.2).

3.2.1 Equilibrium Points

In this work, equilibrium points are defined as a set of information related to a specified physical condition.

$$\boldsymbol{ep{:}x}_{\boldsymbol{\varphi},\boldsymbol{\lambda},\boldsymbol{\mu},\boldsymbol{\gamma}} = \begin{pmatrix} \boldsymbol{\varphi}_x \\ \boldsymbol{\lambda}_x \\ \boldsymbol{\mu}_x \\ \boldsymbol{\gamma}_x \end{pmatrix} = \begin{pmatrix} \varphi_{s,x} \\ \varphi_{e,x} \\ \lambda_{1,x} \\ \vdots \\ \lambda_{5,x} \\ \mu_{1,x} \\ \vdots \\ \mu_{5,x} \\ \gamma_{1,x} \\ \vdots \\ \gamma_{5,x} \end{pmatrix} \tag{3.33}$$

Equilibrium points are based on the Equilibrium Point Hypothesis from Feldmann [29, 30]. They describe a certain state of a system in its environment from an information perspective. To transfer this into the field of bio-robotics, equilibrium points are defined as a matched pair of information including actuation according to the available sensor information and actuation possibilities. By this means, an equilibrium point connects e.g. a certain position by a specified actuation for a specified environmental condition with the sensor information to represent that

status. If the same actuation leads to different sensor information, the environmental condition or the system has changed, hence, a different equilibrium point is applied. An equilibrium point must not contain all available information and can be reduced to only relevant information needed by the controller. **Example:**
$\boldsymbol{ep}\mathbf{:1}_{\boldsymbol{\varphi},\boldsymbol{\mu}}$ is an equilibrium point of Ataro matching the muscle actuation $\boldsymbol{\mu_1}$ with a certain joint position $\boldsymbol{\varphi_1}$:

$$\boldsymbol{ep}\mathbf{:1}_{\boldsymbol{\varphi},\boldsymbol{\mu}} = \begin{pmatrix} \boldsymbol{\varphi}_1 \\ \boldsymbol{\mu}_1 \end{pmatrix} = \begin{pmatrix} \varphi_{s,1} \\ \varphi_{e,1} \\ \mu_{1,1} \\ \vdots \\ \mu_{5,1} \end{pmatrix} \tag{3.34}$$

If a load is appended to Ataro's arm, the joint values will change while the muscle actuation is still the same. Hence, Ataro is not in $\boldsymbol{ep}\mathbf{:1}_{\boldsymbol{\varphi},\boldsymbol{\mu}}$ anymore. The new equilibrium point of Ataro is $\boldsymbol{ep}\mathbf{:2}_{\boldsymbol{\varphi},\boldsymbol{\mu}}$ with

$$\boldsymbol{ep}\mathbf{:2}_{\boldsymbol{\varphi},\boldsymbol{\mu}} = \begin{pmatrix} \boldsymbol{\varphi}_2 \\ \boldsymbol{\mu}_2 \end{pmatrix} = \begin{pmatrix} \varphi_{s,2} \\ \varphi_{e,2} \\ \mu_{1,2} \\ \vdots \\ \mu_{5,2} \end{pmatrix} \tag{3.35}$$

even if $\boldsymbol{\mu_1} = \boldsymbol{\mu_2}$.

3.2.2 Open-Loop Performance

To find equilibrium points, the closed-loop controller of section 3.1 is used. Manually tuning the muscles to a certain position is also an option. Once, a desired position is reached, the values of the sensor and actuator information can be stored and represent an equilibrium point. These stored equilibrium points enable open-loop control, hence a certain actuation μ is connected to a certain position φ under the same conditions. On the other hand, the causal behavior (hysteresis) of the pneumatic actuators makes it difficult to reach the same position with the same actuation, hence the previous state of the muscles matters.

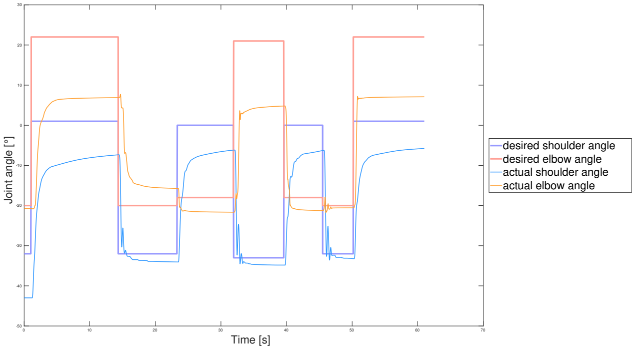


Figure 3.8: Open-loop point reaching with previous defined equilibrium points

Figure 3.8 shows four equilibrium points of Ataro which are targeted successively in an uprising pyramid order (***ep*:1** -> ***ep*:2** -> ***ep*:3** -> ***ep*:4** -> ***ep*:3** -> ***ep*:2** -> ***ep*:1**). All equilibrium points have been defined separately with the closed-loop controller of section 3.1. Then, the calibration sequence has been performed (One maximum shoulder anteversion and elbow flexion, see chapter 3.3.1 - experiment execution) without closed loop point reaching. As a result, the open-loop controller is not able to fulfill point reaching without a deviation.

3.2.3 Open-Loop Controller Implementation

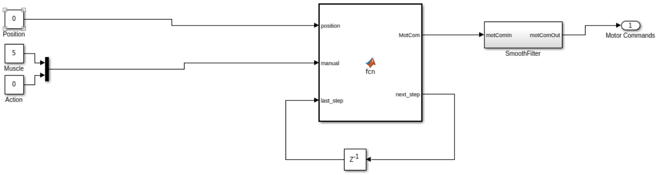


Figure 3.9: Open loop controller

The open-loop controller is implemented as a Matlab function. The parameters are initialized by the parameter script explained in section 1.2.3. The output is modified by a smoothing filter to prevent the motor commands from jumping. (code listing 3.2)

Listing 3.2: open-loop controller code

```
function [MotCom, next_step] = fcn(position, manual,
    last_step, MRC_para, AS_para, EP_pool)

%% Initialization
% Muscle Activation
mAct = [last_step(1,1) last_step(1,2) last_step(1,3)
    last_step(1,4) last_step(1,5)];

% Button Status (remote control and Matlab interface)
if (manual(2) > 0)% Increase selected muscle
b_1 = 1;
else
b_1 = 0;
end

if (manual(2) < 0)% Decrease selected muscle
b_2 = 1;
else
b_2 = 0;
end

if (position == 3)
b_3 = 1;
```

```
else
b_3 = 0;
end

if (position == 4)% Shut down OL
b_4 = 1;
else
b_4 = 0;
end

if (position == 5)%EP1
b_5 = 1;
else
b_5 = 0;
end

if (position == 6)%EP2
b_6 = 1;
else
b_6 = 0;
end

if (position == 7)
b_7 = 1;
else
b_7 = 0;
end

if (position == 8)
b_8 = 1;
else
b_8 = 0;
end

if (position == 9)
b_9 = 1;
else
b_9 = 0;
end

stateB1 = 0;% Increase selected muscle
stateB2 = 0;% Decrease selected muscle
stateB3 = 0;% n.a.
stateB4 = 0;% Shut down OL
stateB5 = 0;% EP1
stateB6 = 0;% EP2
stateB7 = 0;% n.a.
```

```
stateB8 = 0;% n.a.
stateB9 = 0;% n.a.

% Muscle selection
musSelect = last_step(3,1);

% Shut Down Sequence
shutD = last_step(3,2);
checkT = last_step(3,4);

% EP Positions (EP_Loop)
ep1pos = last_step(4,1);
ep2pos = last_step(4,2);
ep3pos = last_step(4,3);
ep4pos = last_step(4,4);

EP_zero = [0 0 0 0 0];
EP_calib = [0 2.3 4.3 0 6];

EP = EP_pool(:,3:7);

%% Edge detection (rising edge) - required for pulse
     control

% Remote Control Buttons
lastB1 = last_step(2,1);
lastB2 = last_step(2,2);
lastB3 = last_step(2,3);
lastB4 = last_step(2,4);
lastB5 = last_step(2,5);
lastB6 = last_step(5,1);
lastB7 = last_step(5,2);
lastB8 = last_step(5,3);
lastB9 = last_step(5,4);


if (lastB1 < b_1) %(rising edge)
stateB1 = 1;
end
if (lastB2 < b_2) %(rising edge)
stateB2 = 1;
end
if (lastB3 < b_3) %(rising edge)
stateB3 = 1;
end
if (lastB4 < b_4) %(rising edge)
stateB4 = 1;
```

```
end
if (lastB5 < b_5) %(rising edge)
stateB5 = 1;
end
if (lastB6 < b_6) %(rising edge)
stateB6 = 1;
end
if (lastB7 < b_7) %(rising edge)
stateB7 = 1;
end
if (lastB8 < b_8) %(rising edge)
stateB8 = 1;
end
if (lastB9 < b_9) %(rising edge)
stateB9 = 1;
end

%% Manual Muscle Stimulation
% Interface
if (manual(1) > 0)
musSelect = manual(1);
end

%Check routine to ensure that musSelect = [1..5]
if (musSelect >= 6)
musSelect = 1;
elseif (musSelect <= 1)
musSelect = 1;
end

% Motor Command Increase/Decrease (B1/B2)
if (stateB1 == 1)
mAct(musSelect) = mAct(musSelect) + MRC_para(1,musSelect);
end

if (stateB2 == 1)
mAct(musSelect) = mAct(musSelect) - MRC_para(1,musSelect);
end

% Security check for activation within allowed pressure
    range
if (mAct(musSelect) >= AS_para(musSelect,2))
mAct(musSelect) = AS_para(musSelect,2);
elseif (mAct(musSelect) <= AS_para(musSelect,1))
mAct(musSelect) = AS_para(musSelect,1);
end
```

```
%% Shut Down (B4)
% single time sequence:

if (stateB4 == 1)
shutD = 1;
checkT = 1;
end

% Execution:
if (shutD == 1)

ep1pos = 0; %Gegenseitige Verriegelung
ep2pos = 0;
ep3pos = 0;
ep4pos = 0;
checkT = checkT*MRC_para(2,5);

if (checkT <= 0.005)
shutD = 0;
checkT = 0;
mAct = EP_zero;
end

mAct = EP_zero - (EP_zero-mAct)*MRC_para(2,5);
end

%% Go to EP1 Position (B5)
% Single execution:

if (stateB5 == 1)
ep1pos = 1;
checkT = 1;
end

%Repeated Execution:
if (ep1pos == 1)

ep2pos = 0;
ep3pos = 0;
ep4pos = 0;
checkT = checkT*MRC_para(2,1);

if (checkT <= 0.005)
ep1pos = 0;
checkT = 0;
mAct = EP(1,:);
end
```

```
mAct = EP(1,:) - (EP(1,:)-mAct)*MRC_para(2,1);
end

%% Go to EP2 Position (B6)
% Single execution:

if (stateB6 == 1)
ep2pos = 1;
checkT = 1;
end

%Repeated Execution:
if (ep2pos == 1)

ep1pos = 0;
ep3pos = 0;
ep4pos = 0;
checkT = checkT*MRC_para(2,1);

if (checkT <= 0.005)
ep2pos = 0;
checkT = 0;
mAct = EP(2,:);
end

mAct = EP(2,:) - (EP(2,:)-mAct)*MRC_para(2,1);
end
%% Go to EP3 Position (B7)

% Single execution:
if (stateB7 == 1)
ep3pos = 1;
checkT = 1;
end

%Repeated Execution:
if (ep3pos == 1)

ep1pos = 0;
ep2pos = 0;
ep4pos = 0;
checkT = checkT*MRC_para(2,1);

if (checkT <= 0.005)
ep3pos = 0;
checkT = 0;
```

```
mAct = EP(3,:);
end

mAct = EP(3,:) - (EP(3,:)-mAct)*MRC_para(2,1);
end

%% Go to EP4 Position (B8)
% Single execution:

if (stateB8 == 1)
ep4pos = 1;
checkT = 1;
end

%Repeated Execution:
if (ep4pos == 1)

ep1pos = 0;
ep2pos = 0;
ep3pos = 0;
checkT = checkT*MRC_para(2,1);

if (checkT <= 0.005)
ep2pos = 0;
checkT = 0;
mAct = EP(4,:);
end

mAct = EP(4,:) - (EP(4,:)-mAct)*MRC_para(2,1);
end

%% Outgoings

next_step = [mAct;...
b_1 b_2 b_3 b_4 b_5;...
musSelect shutD 0 checkT 0;...
ep1pos ep2pos ep3pos ep4pos 0;...
b_6 b_7 b_8 b_9 0;...
];
MotCom = mAct;
```

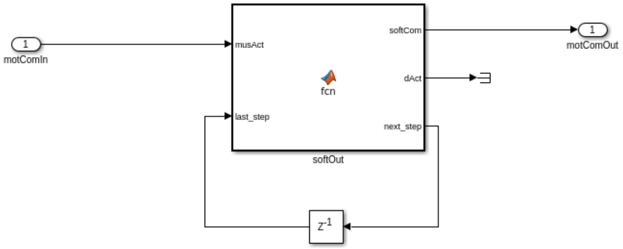


Figure 3.10: Smoothing filter to prevent open-loop jumping

The parameters are initialized by the parameter script explained in section 1.2.3.

```
function [softCom, dAct, next_step] = fcn(musAct,
    last_step, T_S, OLS_Filter)

%% Smoothing parameters

[m,n] = size(OLS_Filter);
steps = m-2;

%% Calculate Differential

dif = (musAct - last_step(11,:))*1/T_S;

%% Edge Smoothing
i = last_step(m,1);

delay = last_step(1:steps,:);

%outCom = delay(i,:); - test setting
outCom = sum(delay).*1/steps;

delay(i,:) = musAct;

i = i+1;
if (i > steps)
i = 1;
```

```
end

%% Outgoings

next_step = [delay;musAct; i 0 0 0 0];
dAct = dif;
softCom = outCom;
```

3.3 Hybrid Controller

3.3.1 Experiment execution

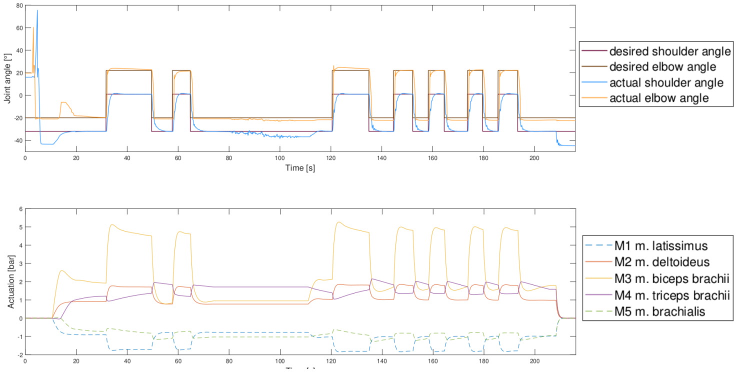


Figure 3.11: Measurement execution

The properties of the control system are investigated via real-time measurements. Each measurement is executed as shown in figure 3.11. At the beginning of each measurement (0s$<$ Time $<$ 10s), the joint angle sensors (incremental encoders) have to be calibrated (peaks at measurement start) by performing a maximum shoulder anteversion and a maximum elbow flexion. The calibration is necessary to cross the zero point positions of the incremental encoders, otherwise the initial position when powering ataro would be considered as zero which can be different. After the calibration, the closed-loop controller (3.1) is started (10s$<$ Time $<$ 70s) and Ataro reaches point one ($\varphi_{shoulder} = -32^\circ$ and $\varphi_{elbow} = -20^\circ$). Ataro is now ready. The first measurements (30s$<$ Time $<$ 70s) are point-to-point reaching without perturbation via weight mounted on the arm. All measurements are executed within the same run. For that, the closed-loop controller was stopped and the actuation was kept with open-loop (figure 3.11 at 70s$<$ Time $<$ 120s). The open-loop controller is not able to perform accurate point-to-point reaching, but once the position is reached via closed-loop control it can easily be kept, as figure 3.11 illustrates. A weight with 700g was mounted on Ataro's wrist ($80s <$ Time $< 120s$). It is important to switch to open-loop control while mounting a weight on Ataro's arm due to high danger while being in range of Ataro during active closed-loop control. Thereafter, the point-to-point reaching was repeated with the additional load. Hence, muscles are under-actuated at the start, the duration of all point reaching executions that targets a point for the first time are increased. This causes an overshooting for the first point to reach, shown at Time $= 122s$. It can be considered as a warm-up and depends on the muscle arrangement and mechanical limitations of the robotic system. This effect will appear each time the closed-loop joint angle control was deactivated and is not considered for the resulting measurements. After that, the performance measurement with perturbation has been executed. The control parameters have not been changed within a measurement. Hence, each change of a control parameter was recorded in a new measurement. The muscle activation for all measurements have been recorded (figure 3.11 bottom graph). The pneumatic muscles can not be activated with negative values. If the actuation falls below zero, the muscle actuation is treated as zero instead.

3.3.2 Hybrid Control Performance

A hybrid controller from Bayer et al. [8] combining open- and closed-loop activation is used to improve control performance (speed and accuracy). The performance is defined as time (speed) in which the target position can be reached. The deviation between target position and actual position at this time step is defined as the accuracy. The resulting muscle activation $\boldsymbol{\mu}_{hyb}$ of the hybrid-controller (open- and closed-loop combination) is given by:

$$\boldsymbol{\mu}_{hyb} = b_{open} \cdot \boldsymbol{\mu}_{open} + \boldsymbol{\mu}_{closed} \tag{3.36}$$

The open-loop part is multiplied by the open-loop weight b_{open}. This enables additional optimization properties by modulating the intensity of the open-loop impact. It can reduce overshooting and increases the performance significantly:

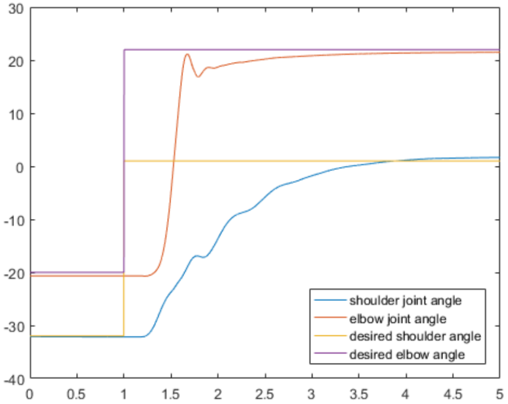


Figure 3.12: Point-to-point reaching in hybrid control with zero open-loop intensity ($b_{open} = 0$) without perturbation. The x-axis shows the time in seconds and the y-axis shows the joint angle in degrees.

Figures 3.12 and 3.13 illustrates the impact of the open-loop control part to the overall control performance. 3.12 shows the single closed-loop performance, hence the open-loop weight b_{open} equals zero.

The performance for single closed-loop mode point-to-point reaching of 0.6m without perturbation is 2.5s for a joint angle deviation of less than 0.5°

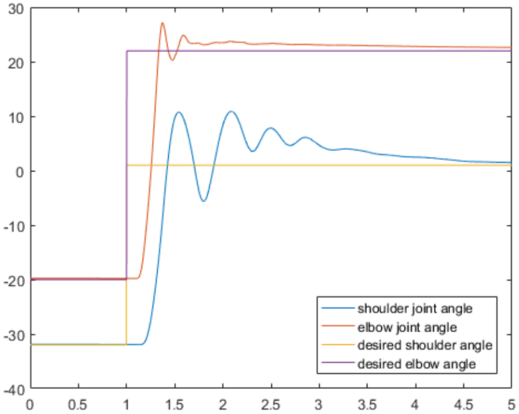


Figure 3.13: Point-to-point reaching in hybrid control with 100% open-loop intensity ($b_{open} = 1$) without perturbation. The x-axis shows the time in seconds and the y-axis shows the joint angle in degrees.

Figure 3.13 illustrates the behavior with full open-loop intensity.

The performance for hybrid mode point-to-point reaching of 0.6m without perturbation and 100% open-loop intensity is 3.5s for a joint angle deviation of less than 0.5°

The overshooting caused by 100% open-loop intensity decreases the overall performance by 1s.

Figures 3.14 and 3.15 illustrates the impact of the open-loop control part to the overall control performance with perturbation of a 700g weight mounted on Ataro's wrist. Figure 3.14 shows the single closed-loop performance.

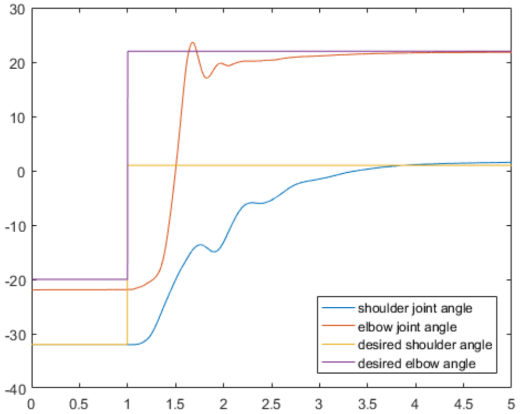


Figure 3.14: Point-to-point reaching in hybrid control with zero open-loop intensity ($b_{open} = 0$) with 700g weight perturbation. The x-axis shows the time in seconds and the y-axis shows the joint angle in degrees.

The performance for single closed-loop mode point-to-point reaching of 0.6m with perturbation is 2.5s for a joint angle deviation of less than 0.5°

The perturbation has therefore no performance impact in single closed-loop mode. Figure 3.15 illustrates the behavior with full open-loop intensity.

The performance for hybrid mode point-to-point reaching of 0.6m with perturbation and 100% open-loop intensity is 3.4s for a joint angle deviation of less than 0.5°

There is no significant difference in the performance with perturbation

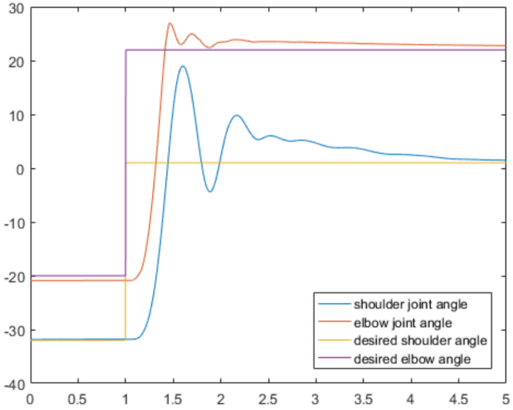


Figure 3.15: Point-to-point reaching in hybrid control with 100% open-loop intensity ($b_{open} = 1$) with 700g weight perturbation. The x-axis shows the time in seconds and the y-axis shows the joint angle in degrees.

compared to the performance without perturbation. The 0.1s performance increase can be caused by the arm weight pushing Ataro's arm down.

Figures 3.16 and 3.17 illustrates the control performance with 50% open-loop intensity for both, with and without perturbation. Figure 3.16 shows the performance without perturbation.

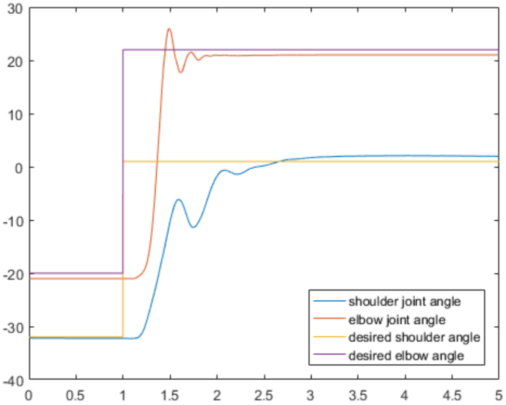


Figure 3.16: Point-to-point reaching in hybrid control with 50% open-loop intensity ($b_{open} = 0.5$) without perturbation. The x-axis shows the time in seconds and the y-axis shows the joint angle in degrees.

The performance for hybrid mode point-to-point reaching of 0.6m without perturbation and 50% open-loop intensity is 1.5s for a joint angle deviation of less than 0.5°

The performance increases significantly by 1s compared to single closed-loop mode and 2s compared to hybrid mode with full open-loop intensity. Figure 3.17 shows the performance with 700g wrist weight perturbation.

The performance for hybrid mode point-to-point reaching of 0.6m with perturbation and 50% open-loop intensity is 1.5s for a joint angle deviation of less than 0.5°

The performance increases significantly by 1s compared to perturbed single closed-loop mode and 1.9s compared to perturbed hybrid mode

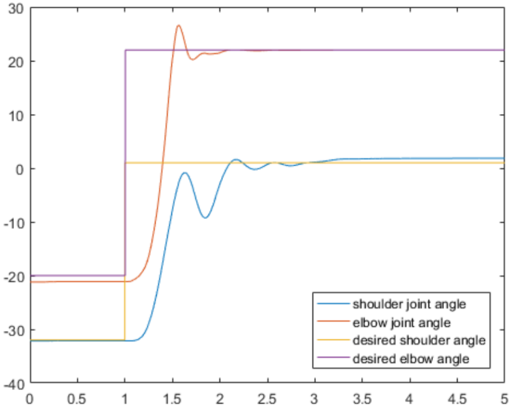


Figure 3.17: Point-to-point reaching in hybrid control with 50% open-loop intensity ($b_{open} = 0.5$) with 700g arm weight perturbation. The x-axis shows the time in seconds and the y-axis shows the joint angle in degrees.

with full open-loop intensity. The perturbation has not shown any significant impact in performance but in the motion trajectory. Figures 3.12 and 3.13 illustrates the oscillation caused by the impact of full open-loop intensity. This oscillation is reduced with perturbation caused by the higher mass inertia as figures 3.14 and 3.15 shows. Using open-loop weights $b_{open} > 0$ will always increase the oscillation of the motion trajectory even if it can increase the overall performance.

Chapter 4

Discussion

4.1 A2: Muscles in Bio-inspired Setups

Muscle spring units are designed to attain properties of biological actuators for reproducing biological motion in AAS. Despite the fact that the biological muscle tendon unit still outperform the muscle spring unit, an antagonistic setup operated by the proposed muscle spring units can be an highly beneficial solution for bio-mimetic robots. With the applied design of chapter 2, I have shown that the loss of active contraction range of a single muscle spring unit does not reduce the range of motion in an antagonistic setup. Moreover, by using muscle spring units, the whole range of motion can be driven with the full pressure spectrum, which is not possible with an antagonistic setup using only pneumatic muscles. An antagonistic setup with only pneumatic actuators without serial elasticity has to be preloaded instead, which leads to a reduction of the available pressure range and therefore reduces the overall range of motion. Furthermore, a reduced available pressure range comes along with a decrease in the range of available joint stiffness values S_{A}. A controllable joint stiffness is the general benefit of the proposed setup compared to other actuators, e.g. electrical drives where stiffness can only be mimicked by control, requiring high band-width control. Pneumatic actuators with serial elasticity in an antagonistic setup do not only provide the possibility of using a maximal pressure spectrum for adjusting the joint stiffness, but do also allow for adjustment of variable joint torque values at a given joint angle. We have shown that deploying muscle spring units provides significant potential for designing joint characteristics for the range of motion, the torque and the stiffness, even across sizes. This enables a wide range of applications for bio-mimetic robots.

A2 in facts:
To sum it up, using muscle spring units in antagonistic joint setups comes along with the following benefits:

- Innate passive ability (figure 2.3)
- Increased range of motion compared to antagonistic muscle only setups (figure 2.5)
- Wide range of joint torque generation (figure 2.6)
- Adjustable joint stiffness (figure 2.7)

Even if the performance of commercially available pneumatic muscles with serial elasticity is presentable, the development of such actuators is still in its infancy. Reducing the length of stiff mounting parts or combining the spring with the muscle to optimize the overall length are promising approaches. The reason, why such actuators lack in development can be found in lack of applicable control strategies.

4.2 A1,A3 and A4: The Control Framework in a Nutshell

The answers to the introduced scientific questions A1 to A4 are strongly connected. Solving the questions Q3 and Q4 implicitly solves the question Q1. As explained in the introduction and the preface of this work, we have to leave our engineering perspective of how we rate properties. Non-linear muscle actuators with hysteresis in series to elastic elements are not unserviceable in general, but only for common model based control policies. The same applies to the design of joints with bi-articular actuators.

”It is not a bug; It is a feature”

Compared to electrical drives, muscles can only create force in one direction, which means we always know how actuation effects a joint motion in terms of direction. Additionally, muscles have always the same force-length-actuation behavior, they only differ in scale. Both features count for biological muscle tendon units as well as bio-inspired muscle spring units. Therefore, the only thing that is needed is to find a control policy that can make use out of the provided information.

A1 solved by A3 and A4:
The control framework that is proposed in chapter 3 make use of the available information in the following steps.

- The arrangement of the muscles in a muscle driven system shows us the overall possibility to effect motion of the system (equation (3.4))
- The knowledge of the motion direction impact of an actuated muscle shows us the sign of the control parameter for its respective muscle (equation (3.5)).
- Muscles share the same force-length-actuation behavior, they only differ in scale. If we know the scale (correlation function

$w(\boldsymbol{b})$), we can create a dependency between two muscles effecting the same joint (equation (3.8)) (A3)

- If we can create correlation functions between two muscles of a joint, we can create correlation functions to all muscles of the joint, including bi-articular muscles (set of equations (3.12) and (3.13)) (A3)
- Bi-articular muscles effecting two joints, that means we can also create correlation functions over multiple joints using a bi-articular muscle (equation (3.14)) (A4)
- If all joints of a system are connected by bi-articular muscles, we are able to create correlation functions to all muscles in our system which reduces the control parameter amount to a single muscle, no matter how many muscles the system uses (equation (3.25)) (A4)
- Finding and optimizing one single parameter implicitly optimizes all other parameters. This can be done by various optimization methods (e.g., Gradient Descent - equation (3.26)) (A1).

Proportional control parameters comes along with a steady control deviation. Therefore additional integral behavior is needed. The popular control scheme of an ideal control (figure 3.2) brings integral behavior to each muscle which makes it necessary to find I-parameters for each muscle. The correlation functions can not be applied to the I-parameters, hence the integral behavior handles the deviation caused by perturbation and inaccuracy and not the relation of actuator activation based on design. But do we need integral behavior on actuator level? - no we don't. The integral behavior must not eliminate the actuator deviation but the deviation of the desired motion or on point of the joints. In this case, we can transfer the integral behavior to the joint level by using an interacting control scheme instead (figure 3.1). The amount of I-parameters are now dependent to the number of joints than to the number of muscles which means a parameter reduction for antagonistic setups with bi-articular muscles especially for systems with more than two-dimensional range of motions like SH1.

Another benefit by using the proposed interacting control scheme is the way it influences the actuation. Instead of biasing the proportional output of an ideal control scheme, it bias the input of the proportional controller instead (equation (3.1)). With ideal control, the integral part can totally replace the proportional part in steady state. This appeared especially with an ideal control scheme driving SH1 to perform a stable stance for a longer period of time. The control law of the interacting control scheme makes this impossible (equation (3.1)).

4.3 Parameter Robustness and Stability

Standard model based control criteria to investigate the stability of a control scheme (e.g. Lyapunov criteria) can not be applied because there is no mathematical model considered for the controlled system. Therefore, the parameter robustness and stability is investigated only on an experimental and phenomenological level. Furthermore, the increase of control parameters in the following experiments are limited to $\pm$30% for the proportional control parameter and $\pm$50% for the integral parameter due to the danger of permanently destroying the robotic setup.

The proposed bio-inspired controller has shown a high parameter robustness on that experimental level as figures 4.1, 4.2 and 4.3, 4.4 illustrate. To investigate the controllers parameter robustness, the closed-loop mode is used (not the hybrid mode). The benchmark of the closed-loop mode was 2.5s to reach desired position. All measurements are point-to-point measurements with the same distance (0.6m) as the results of chapter 4 show. Figures 4.1 and 4.2 shows the anteversion and elbow flexion without load to reach the target position. As expected, an increase in the control parameters will also lead to an increase in the performance but will cause overshooting. Furthermore, by increasing the integral parameters by 50% (figure 4.1 and 4.2, light blue) oscillations appear. The same can be investigated when increasing the proportional parameter by 30% (violet). This oscillation is amplified when applying the 700g wrist weight as perturbation (figure 4.3 and 4.4). Therefore, the perturbation has a significant impact on the parameter robustness and stability. This is remarkable because the perturbation has no significant impact on the performance when using optimized parameters. Another perceptible effect is that a variation in the proportional parameter can be compensated by an increased integral parameter. The dark blue line (p-30%, i+50%) covers nearly the same trajectory as the default values p_{opt};i_{opt}. This is due to the fact, that both are factors in the control law (equation (3.1)). Finally, the bio-inspired control policy do not need exact control parameters to work, instead it tolerates control parameter deviation up to 30% of the optimized parameters. This is not a general rule and applies only to the Ataro setup.

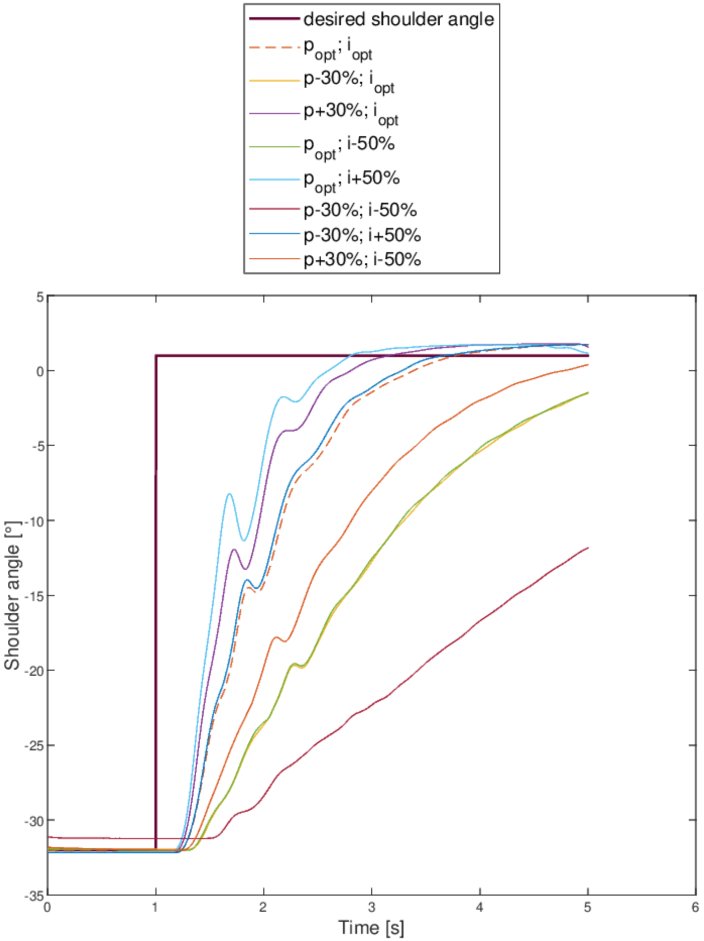


Figure 4.1: Point-to-point reaching with different parameter sets. The performed motion is a shoulder anteversion and elbow flexion. The graph shows the shoulder motion.

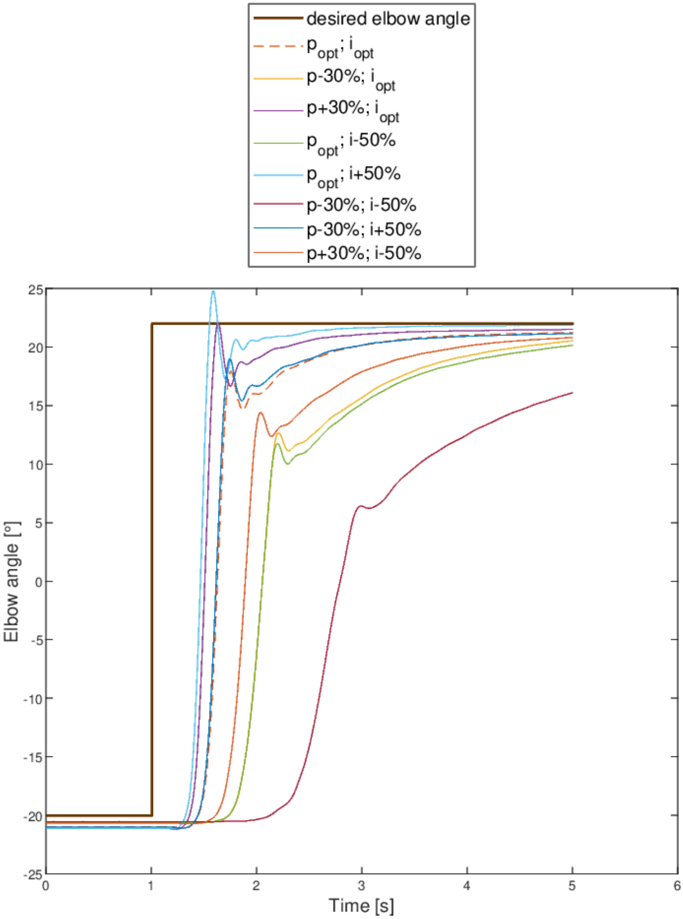


Figure 4.2: Point-to-point reaching with different parameter sets. The performed motion is a shoulder anteversion and elbow flexion. The graph shows the elbow motion.

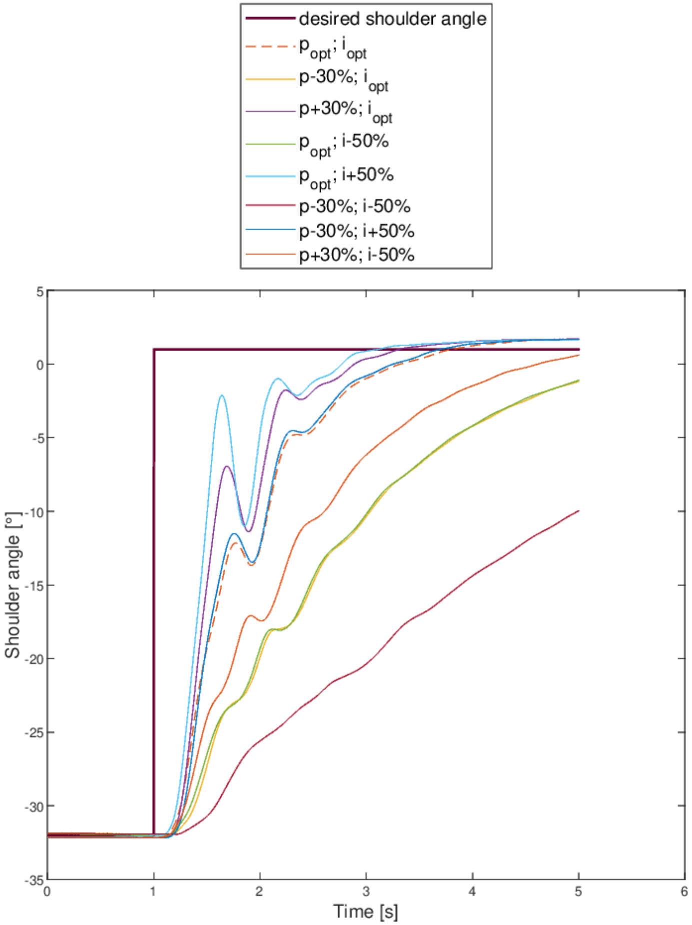


Figure 4.3: Point-to-point reaching with different parameter sets and additional load. The performed motion is a shoulder anteversion and elbow flexion. The graph shows the shoulder motion.

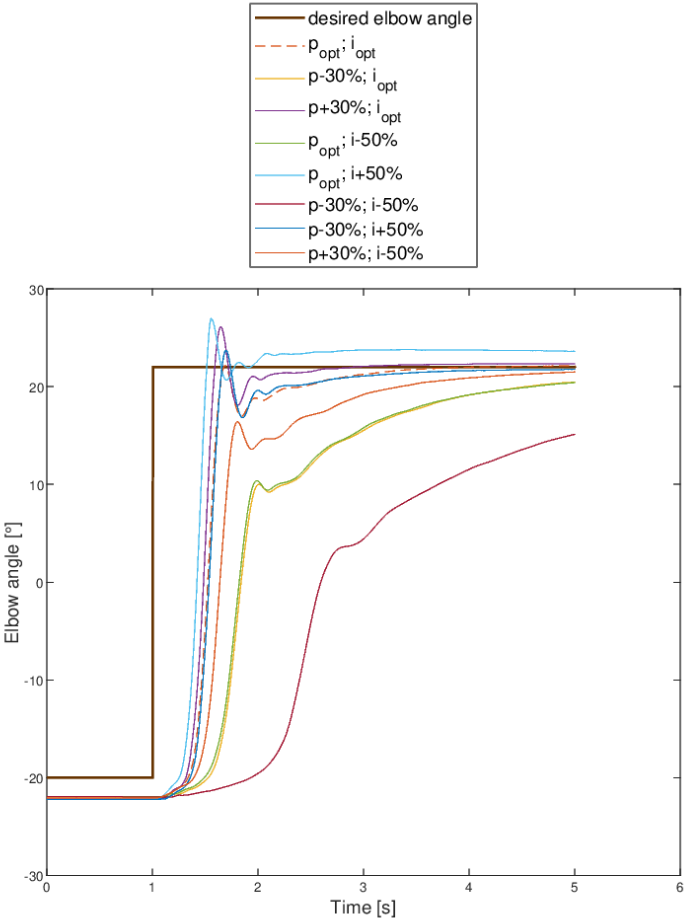


Figure 4.4: Point-to-point reaching with different parameter sets and additional load. The performed motion is a shoulder anteversion and elbow flexion. The graph shows the elbow motion.

4.4 Reducing Information and Control

Biological systems like animals and humans suffer from information reduction when they are tired or under the influence of medication. Furthermore, our possibilities to activate our muscles can also be effected. Up to a certain level, the biological motor control system can handle this challenge and can keep system stability by control. This is an phenomenologically observed stability
To investigate the behavior of the proposed bio-inspired controller, experiments have been made with the leg robot SH1 to observe stability on a phenomenological level. For that, the possible actuation and the sensor information of the joint values were limited to discrete steps. The experiment execution for all measurements has been made in the following order:

- SH1 gets started without control penalties and brings himself into a stable stance position.
- After reaching a stable stance, the penalties have been applied to the controller.
- The measurements starts.
- SH1 was set to perform a one legged squat to a lower equilibrium point position.
- After one second, SH1 was set to stand up to the higher initial equilibrium point and to regain stability (stable stance without falling)

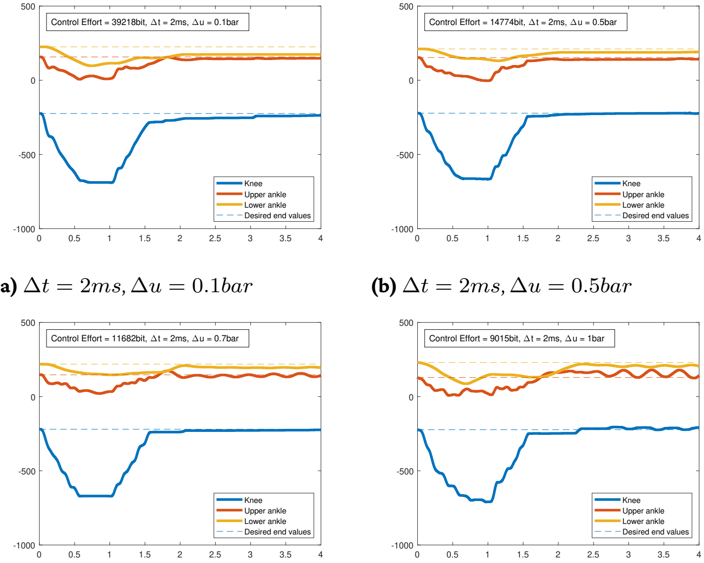
The penalties that have been applied to investigate the control behavior under information and control reduction consist of two parts. First, the input of the controller is penalized by reducing the information the controller receives from sensors. In the default mode, Ataro and SH1 receive sensor information at a sample rate of $500Hz$, therefore every $\Delta t = 2ms$. The biological $200ms$ delay [64] is always applied for both robots in addition to all measurements of this work, as described in Section 1.2 Bio-Robotic Test Bed. The information penalty consist of an increase of Δt. The measurements have been executed with $\Delta t = 2ms$(default),

$10ms$, $14ms$ and $20ms$.

Second, the output of the controller is penalized by increasing the step size of the possible pressure values. The default step size is $\Delta u = 0.01bar$. The applied penalty steps are $\Delta u = 0.1bar$, $0.5bar$, $0.7bar$ and $1bar$. Figure 4.5 shows the behavior of applied actuation penalties without information reduction. The initial joint values of the knee (blue), the upper ankle (orange) and the lower ankle (yellow) are marked as dashed lines. The values of the joint position are in increments, where 10 increments equals 1°. The time axis is in seconds.

The performance is executed by controlling 8 muscles in bio-inspired redundant arrangement in 3-dimensions (Section 1.2). Therefore, the interplay of the muscles prevent jumps of the joint values (e.g. at $\Delta u = 1bar$), hence not all muscles switch their values at the same time.

This measurement can not be executed with Ataro, because the low muscle redundancy (only five muscles for two joints and only one bi-articular muscle) will cause the system to jump under $\Delta u = 1bar$ by creating high torques which destroys the Ataro setup permanently. Figure 4.5 shows, that the system tolerates a reduction of the control variable to regain stance with minor joint angle deviation. By applying an actuation penalty of $\Delta u \geq 0.7bar$, the system starts oscillating permanently but still regains stance.



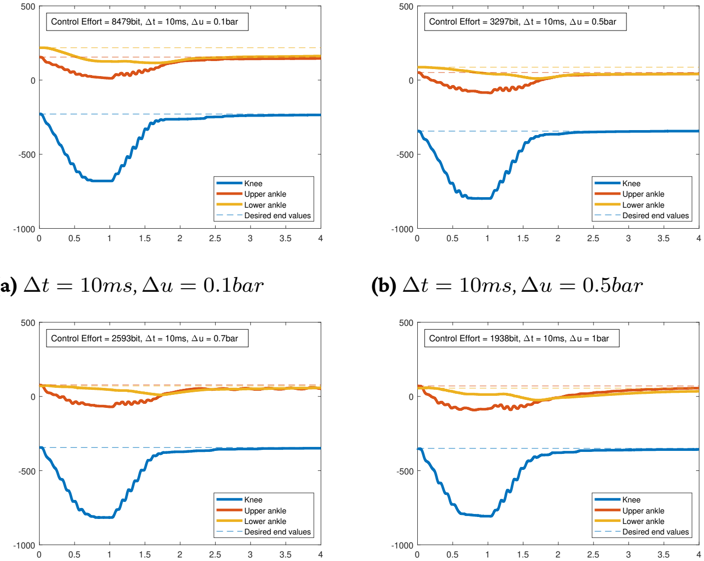
(a) $\Delta t = 2ms, \Delta u = 0.1bar$

(b) $\Delta t = 2ms, \Delta u = 0.5bar$

(c) $\Delta t = 2ms, \Delta u = 0.7bar$

(d) $\Delta t = 2ms, \Delta u = 1bar$

Figure 4.5: SH1 performing a stable stance control after equilibrium point change under reduction of possible actuation



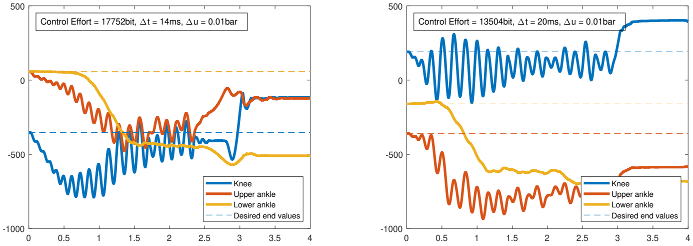
(a) $\Delta t = 10ms, \Delta u = 0.1bar$

(b) $\Delta t = 10ms, \Delta u = 0.5bar$

(c) $\Delta t = 10ms, \Delta u = 0.7bar$

(d) $\Delta t = 10ms, \Delta u = 1bar$

Figure 4.6: SH1 performing a stable stance control after equilibrium point change under reduction of possible information and actuation.



(a) $\Delta t = 14ms, \Delta u = 0.01bar$

(b) $\Delta t = 20ms, \Delta u = 0.01bar$

Figure 4.7: SH1 getting unstable by performing equilibrium point change under reduction of sensor information.

4.5 Conclusion

4.5.1 Generalization

To summarize the benefits of the proposed control scheme of this work, the following statements can be made:

- **Bi-articular muscles are not a ”problem” that causes increased control complexity, in fact, they are the solution to reduce complexity.**
- **The amount of muscle actuators and their non-linearity do not play a significant role in terms of control complexity as long as always the same type (different scales are possible) of muscle actuator is used and all joints are connected with bi-articular muscles.**
- **Bio-inspired actuator setups brings innate bio-inspired features such as compliance.**
- **Non-linear bio-inspired actuators in muscle driven systems can achieve remarkable performance with bio-inspired control policies. Point-to-point performance of 0.6m in 1.5s into a stable steady state independent from perturbation is the best result with this controller so far.**
- **There is no best parameter, only matching sets of parameter which can have the same results**

Bi-articular muscles became an object of high interest in the last years [56, 45, 78, 98, 97]. The complex arrangement of bi-articular muscles seem to be disadvantageous on the first view from a classical engineering perspective. Since they have been investigated more in the last decade, more benefits of such structures have been exposed [23, 54, 62]. Bi-articular muscles can reduce the overall energy consumption while performing motion [68, 28] while energy consumption can be a crucial policy for motion control [42]. Furthermore, bi-articular muscles increase motion ability such as jumping [94]. However, for established controllers, bi-articular muscles seem to be a drawback that has to be overcome via hardware performance (model based approach) or data quantity (AI and Fuzzy), although there is much so suggest that morphology must be beneficial for control [103, 105, 104, 115, 62, 61, 19, 98]. This work fill this gap, by proposing a control framework that exploit biological morphology to reduce complexity instead of compensating the morphology. Furthermore, the proposed control framework can exploit bi-articular muscles to not only reduce system complexity but also to make the whole system scalable to high amounts of muscles or joints.

The control framework is generalizable

Bi-articular muscles enable inter-joint couplings which can be used to minimize the amount of determinable parameters: One parameter per inter-joint coupling network and per controller type (two parameters for PI-controller, three for PID). An inter-joint coupling network is the mechanical construction of all joints that are connected via bi-articular muscles.
The structure of the controller stays always the same. The control parameters may be recalculated depending on the case:

Case: Inter-joint network remains untouched, new muscles are added

If muscles are added to an existing joint network, the control parameters for the previous muscle setup remain, even if these muscles are bi-articular. Just the new parameters of the parameter matrix have to be calculated, because some of the zeros of the p-matrix ((3.2)) turn into parameters (see chapter 3).

Case: Inter-joint network extends or a new joint is added but the main network remain unchanged

If joints are added to a network with new muscles, correlation functions for the new muscles and joint ratios have to be defined and the new added parameters can be calculated. If two inter-joint networks have been connected via bi-articular muscles, for one network, the parameters remain but for the other network, the parameter have to be recalculated. In this case, the added network is not independent anymore and the parameters affecting that network have to be dependent to the main network.

Case: The whole inter-joint network change

In this case, the parameter matrices have to be determined according to chapter 3 and all its correlations have to be redefined. This case creates the most effort, but the controller structure stays exactly the the same. If all joints ob the bio-robotic test-bed are part of an inter-joint network (via bi-articular muscles), only one parameter per controller type (two parameters for PI-controller, three for PID) have to be optimized according to chapter 3.

4.5.2 Performance and comparison

Performance is defined in this work as the time a robotic system needs to reach a certain accuracy of the target position (measured in deviation to target position). The best performance of the Ataro controller is reached in hybrid mode (open- and closed-loop combination [8]):

The performance for hybrid mode point-to-point reaching of 0.6m with perturbation and 50% open-loop intensity is 1.5s for a joint angle deviation of less than 0.5° (Figure 3.17)

Comparing output performance is difficult and can always be seen critical, hence the systems of the different control policies published, differ from each other. A fair comparison can only be made if all comparable policies have been run on the same setup under the same conditions. This is usually not the case and can also be seen critical here.
To compare the control performance of this work, one arm simulation

[75] and one arm robot controller [80] with bi-articular muscles have been chosen. The muscle arrangement is not identical however. The first compared controller drives an arm simulation. It controller is impacted by a delay similar to Ataro. The arm model has two joints, driven by visco-elastic-muscle models. The model-based control approach suffer from the complexity of bi-articular muscles. Its performance results in approximately 4 seconds reaching target position (picture 4.9). Ataro can perform this motion with the proposed control framework in 1.5 seconds.

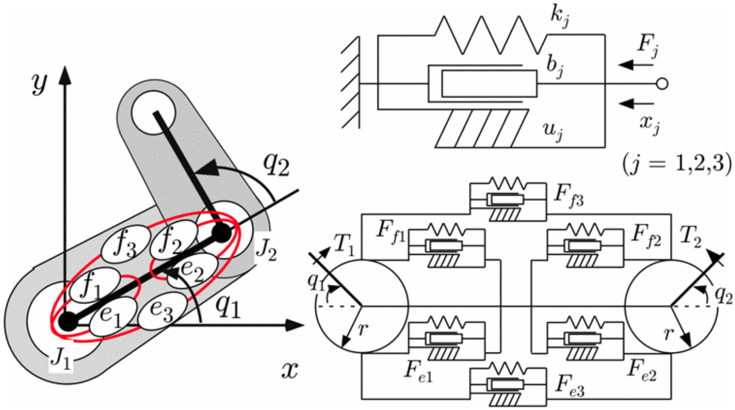


Figure 4.8: Two joint arm model of Murao 2017 [75]

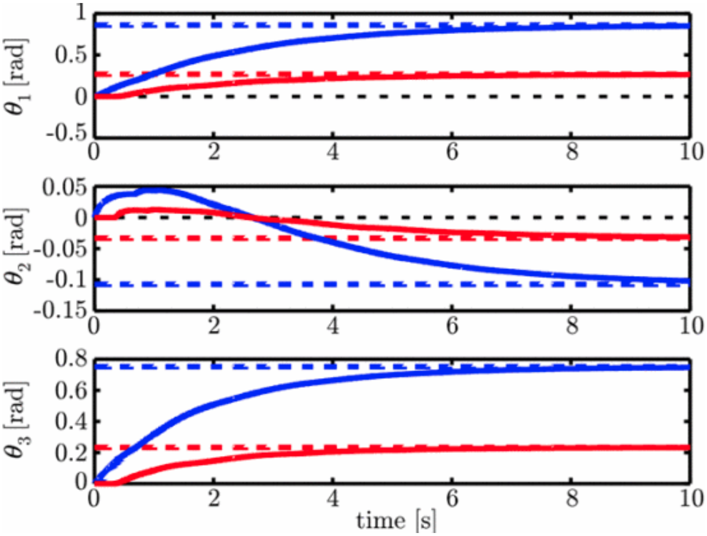


Figure 4.9: Output performance of Murao 2017 [75](solid red lines).

The second compared controller is driving an arm robot shown in picture 4.10. This robot also uses bi-articular muscles. Nishimuras arm robot performs remarkable precision but lack in speed. It performs motion in y-axis of 0.2 meters in 10 seconds (picture 4.11). Ataro performs 0.6 meters in 1.5 seconds. As mentioned, the comparisons have to be seen critical and the proposed controller does not claim to be better in all aspects. Furthermore, the proposed controller should be seen as a toolbox and its single steps such as optimizing can be replaced with other methods, according to the tasks. The potential of the controller framework exceeds

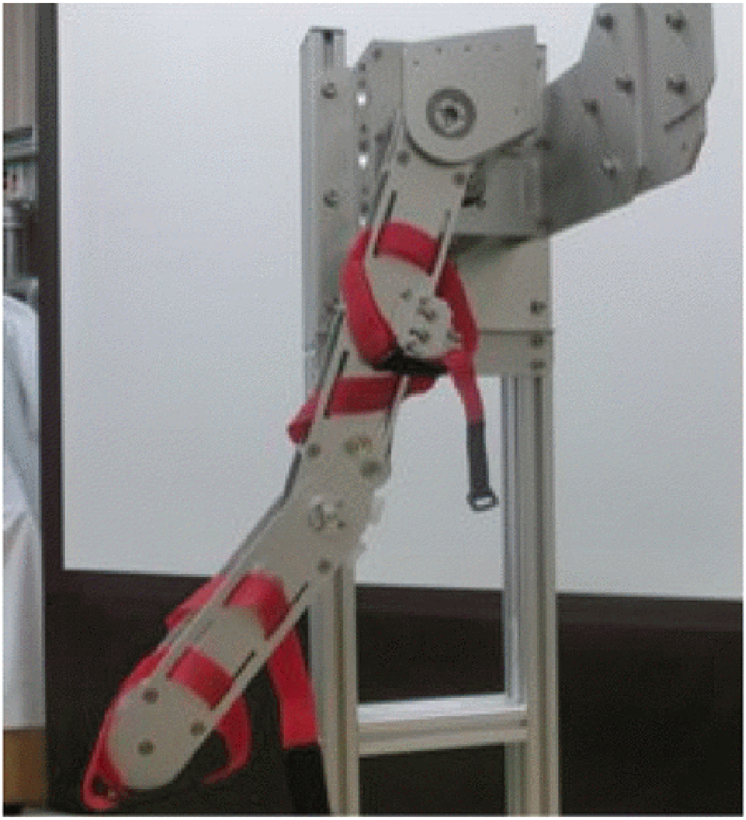


Figure 4.10: Two joint arm model of Nishimura 2021 [80]

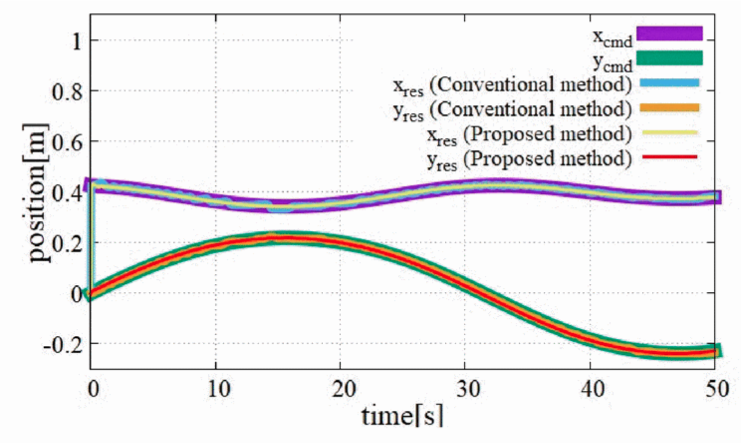


Figure 4.11: Output performance of Nishimura 2021 [80].

the sown applications, hence, it can be further improved (e.g., D-control behavior) nor is this setup limited to joint position control (e.g., force-control).

4.5.3 Critics and outline

"So eine Arbeit wird eigentlich nie fertig, man muß sie für fertig erklären,
wenn man nach Zeit und Umständen das Möglichste getan hat."
- Johann Wolfgang von Goethe

"Properly speaking, such work is never finished; one must declare it so when,
according to time and circumstances, one has done one's best."
- Johann Wolfgang von Goethe

Comparability

The comparisons have to be seen critical and the proposed controller does not claim to be better in all aspects. Furthermore, the compared robots may not even have the same performance criteria nor have they the same technical requirements. The best case to compare controller performance would be, if a standard muscle driven bio-robotic test-bet would be defined so that all control approaches can be compared under equal conditions and under the same performance criteria.

PID-structure

The proposed controller is tested and developed as an PI-controller. Other combinations such as the PD-lambda-controller [8] or a PID-controller is possible and can further increase the performance. Where to include the d-part of a PID-controller has to be investigated and may depend on the goal.

Optimization methods

The tested optimization method may not be the best solution for all applications nor do this work claim that this optimization method is the best solution even for the proposed controller. Optimization is one part out of the toolbox and the chosen method may vary from case to case.

Combination with other control methods

Neural networks are often combined with fuzzy control, hence fuzzy control needs experience of a system and machine learning can serve that demand. It may be possible to reduce complexity by the proposed control framework of this work and afterwards calculate the remaining parameters with model based control approaches (instead of optimizing the remaining parameters). This approach has not been tested yet and can be promising, hence, the proposed framework neglect the weakness of model based approaches.

Processing time is not bio-inspired

The time between reading the input information and applying the output actuation is determined by the available hardware. The sample time was set to $2ms$. The signal processing of the inputs collect 100 samples (see sensor processing section 1.2) and calculates the average of the samples to reduce the sensor signal noise and therefore, applying a bio-inspired $200ms$ delay [64] for the controller inputs. Afterwards, the control, the hybrid mode, additional filters (e.g., smothing filter for open loop) and the output saturation is processed, each within a single sample time of $2ms$. This leads to an overall processing between input and output of $210ms$ which is determined by the hardware and not inspired by biology.

Scope of the framework is limited to 2D motion in this work

The motion performance is investigated by the robotic setup Ataro which has only motion possibilities in the saggital plane. Furthermore, only hinge joints are investigated. The controller setup must be improved to enable control for 3D joints (e.g., ball joints). The challenge of compressing *" the movement system's state space of very many dimensions into a control space of few dimensions"* [9] is not new but can not investigated with the introduced robotic setups. Despite SH1 possibilities to collapse in all 3 dimensions, it only uses hinge joints even if the lower ankle enable 3D motion by two hinge joints acting in different planes. Furthermore, the motion range of SH1 limits the possibility to investigate improved control approaches of the proposed framework.

Final words

The proposed controller should be seen as a toolbox and its single steps such as optimizing or interacting control scheme can be replaced with other methods. It is also not a finished controller setup nor is this setup limited to joint position control. The scope of this work is to find an applicable generalizable control solution for complex muscle driven systems that exploit its features instead of "just dealing with it". It is investigated by its output performance and other observable outputs. Observation in general was my self-imposed limit, leading to the scientific questions of this work and also determine the method of investigation. I demonstrated, that it is possible to design a controller that is able to handle complex systems just by information of observation without intrinsic system knowledge like mathematical system models. Furthermore, this controller do not claim to be "better" than other control frameworks. It is a just another approach and its utility depends on the case.

Bibliography

[1] Emanuel Andrada, Christian Rode, and Reinhard Blickhan. “Grounded running in quails: simulations indicate benefits of observed fixed aperture angle between legs before touch-down”. In: *Journal of theoretical biology* 335 (2013), pp. 97–107.

[2] Ho Pham Huy Anh and Kyoung Kwan Ahn. “Hybrid control of a pneumatic artificial muscle (PAM) robot arm using an inverse NARX fuzzy model”. In: *Engineering Applications of Artificial Intelligence* 24.4 (2011), pp. 697–716.

[3] Ho Pham Huy Anh, Nguyen Ngoc Son, and Cao Van Kien. “Adaptive neural compliant force-position control of serial PAM robot”. In: *Journal of Intelligent & Robotic Systems* 89.3-4 (2018), pp. 351–369.

[4] Yohei Ariga et al. “Novel equilibrium-point control of agonist-antagonist system with pneumatic artificial muscles”. In: *2012 IEEE International Conference on Robotics and Automation*. IEEE. 2012, pp. 1470–1475.

[5] Arick G Auyang, Jasper T Yen, and Young-Hui Chang. “Neuromechanical stabilization of leg length and orientation through interjoint compensation during human hopping”. In: *Experimental brain research* 192.2 (2009), pp. 253–264.

[6] Jan Babic. “Biarticular legged robot: Design and experiments”. In: *2008 IEEE International Conference on Robotics and Biomimetics*. IEEE. 2009, pp. 155–159.

[7] J Babič et al. “A biarticulated robotic leg for jumping movements: theory and experiments”. In: (2009).

[8] A Bayer et al. “The influence of biophysical muscle properties on simulating fast human arm movements”. In: *Computer methods in biomechanics and biomedical engineering* 20.8 (2017), pp. 803–821.

[9] Antonio Bicchi, Marco Gabiccini, and Marco Santello. “Modelling natural and artificial hands with synergies”. In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 366.1581 (2011), pp. 3153–3161.

[10] Andrew A Biewener and Thomas J Roberts. “Muscle and tendon contributions to force, work, and elastic energy savings: a comparative perspective.” In: *Exercise and sport sciences reviews* 28.3 (2000), pp. 99–107.

[11] Reinhard Blickhan. “The spring-mass model for running and hopping”. In: *Journal of biomechanics* 22.11-12 (1989), pp. 1217–1227.

[12] Ivo Boblan. “Modellbildung und Regelung eines fluidischen Muskelpaares”. In: (2010).

[13] Sebastian Bohm et al. “The force–length–velocity potential of the human soleus muscle is related to the energetic cost of running”. In: *Proceedings of the Royal Society B* 286.1917 (2019), p. 20192560.

[14] G Bosco and RE Poppele. “Proprioception from a spinocerebellar perspective”. In: *Physiological reviews* 81.2 (2001), pp. 539–568.

[15] Gianfranco Bosco, A Rankin, and R Poppele. “Representation of passive hindlimb postures in cat spinocerebellar activity”. In: *Journal of neurophysiology* 76.2 (1996), pp. 715–726.

[16] ZY Chen, Yahui Meng, and Timothy Chen. “NN model-based evolved control by DGM model for practical nonlinear systems”. In: *Expert Systems with Applications* 193 (2022), p. 115873.

[17] Ching-Ping Chou and Blake Hannaford. “Measurement and modeling of McKibben pneumatic artificial muscles”. In: *IEEE Transactions on robotics and automation* 12.1 (1996), pp. 90–102.

[18] Vikram Roy Chowdhury. “Internal Model Based Grid Voltage Estimation and Control of a Three-Phase Grid Connected Inverter for PV Application”. In: *IEEE Transactions on Energy Conversion* 36.4 (2021), pp. 3568–3577.

[19] Stacie A Chvatal et al. “Common muscle synergies for control of center of mass and force in nonstepping and stepping postural behaviors”. In: *Journal of neurophysiology* 106.2 (2011), pp. 999–1015.

[20] Festo AG & Co.KG. “Fluidic muscle dmsp datasheet”. In: www.festo.com, 2017.

[21] Frank Daerden et al. “Pleated pneumatic artificial muscles: actuators for automation and robotics”. In: *2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics. Proceedings (Cat. No. 01Th8556)*. Vol. 2. IEEE. 2001, pp. 738–743.

[22] Friedl De Groote et al. “Evaluation of direct collocation optimal control problem formulations for solving the muscle redundancy problem”. In: *Annals of biomedical engineering* 44.10 (2016), pp. 2922–2936.

[23] Jesse C Dean and Arthur D Kuo. “Elastic coupling of limb joints enables faster bipedal walking”. In: *Journal of the Royal Society Interface* 6.35 (2009), pp. 561–573.

[24] Danny Driess et al. “Learning to Control Redundant Musculoskeletal Systems with Neural Networks and SQP: Exploiting Muscle Properties”. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 6461–6468.

[25] Herbert Elftman. “The function of muscles in locomotion”. In: *American Journal of Physiology-Legacy Content* 125.2 (1939), pp. 357–366.

[26] Herbert Elftman. “The work done by muscles in running”. In: *American Journal of Physiology-Legacy Content* 129.3 (1940), pp. 672–684.

[27] Ken Endo, Daniel Paluska, and Hugh Herr. “A quasi-passive model of human leg function in level-ground walking”. In: *2006 IEEE/RSJ international conference on intelligent robots and systems*. IEEE. 2006, pp. 4935–4939.

[28] Mahdy Eslamy, Martin Grimmer, and Andre Seyfarth. “Adding passive biarticular spring to active mono-articular foot prosthesis: effects on power and energy requirement”. In: *2014 IEEE-RAS International Conference on Humanoid Robots*. IEEE. 2014, pp. 677–684.

[29] Anatol G Feldman. “Once more on the equilibrium-point hypothesis (λ model) for motor control”. In: *Journal of motor behavior* 18.1 (1986), pp. 17–54.

[30] Anatol G Feldman and Mindy F Levin. “The equilibrium-point hypothesis–past, present and future”. In: *Progress in motor control* (2009), pp. 699–726.

[31] Daniel P Ferris, Micky Louie, and Claire T Farley. “Running in the real world: adjusting leg stiffness for different surfaces”. In: *Proceedings of the Royal Society of London. Series B: Biological Sciences* 265.1400 (1998), pp. 989–994.

[32] US Food and Drug Administration. “Medical Device Safety”. In: www.fda.gov/medical-devices/medical-device-safety, 2022.

[33] Rudolf M Füchslin et al. “Morphological computation and morphological control: steps toward a formal theory and applications”. In: *Artificial life* 19.1 (2013), pp. 9–34.

[34] Senshi Fukashiro, Dean C Hay, and Akinori Nagano. “Biomechanical behavior of muscle-tendon complex during dynamic human movements.” In: *Journal of Applied Biomechanics* 22.2 (2006).

[35] Mariano Garcia et al. “The simplest walking model: stability, complexity, and scaling”. In: (1998).

[36] Hartmut Geyer, Andre Seyfarth, and Reinhard Blickhan. “Compliant leg behaviour explains basic dynamics of walking and running”. In: *Proceedings of the Royal Society B: Biological Sciences* 273.1603 (2006), pp. 2861–2867.

[37] Mario Gomes and Andy Ruina. “Walking model with no energy cost”. In: *Physical Review E* 83.3 (2011), p. 032901.

[38] L Gregoire et al. “Role of mono-and biarticular muscles in explosive movements”. In: *International journal of sports medicine* 5.06 (1984), pp. 301–305.

[39] Martin Grimmer, Mahdy Eslamy, and André Seyfarth. "Energetic and peak power advantages of series elastic actuators in an actuated prosthetic leg for walking and running". In: *Actuators*. Vol. 3. 1. Multidisciplinary Digital Publishing Institute. 2014, pp. 1–19.

[40] Martin Grimmer and André Seyfarth. "Mimicking human-like leg function in prosthetic limbs". In: *Neuro-Robotics*. Springer, 2014, pp. 105–155.

[41] Michael Günther and Heiko Wagner. "Dynamics of quiet human stance: computer simulations of a triple inverted pendulum model". In: *Computer methods in biomechanics and biomedical engineering* 19.8 (2016), pp. 819–834.

[42] Daniel FB Haeufle et al. "Muscles reduce neuronal information load: quantification of control effort in biological vs. robotic pointing and walking". In: *Frontiers in Robotics and AI* 7 (2020), p. 77.

[43] DFB Haeufle et al. "Hill-type muscle model with serial damping and eccentric force–velocity relation". In: *Journal of biomechanics* 47.6 (2014), pp. 1531–1536.

[44] Thomas Heidlauf et al. "A continuum-mechanical skeletal muscle model including actin-titin interaction predicts stable contractions on the descending limb of the force-length relation". In: *PLoS computational biology* 13.10 (2017), e1005773.

[45] Shuma Hiasa et al. "A Legged Robot With Thigh Bi-Articular Muscle-Tendon Complex". In: *IEEE Access* 9 (2021), pp. 62285–62297.

[46] Archibald Vivian Hill. *First and last experiments in muscle mechanics*. Cambridge University Press, 1970.

[47] Masaya Hirashima and Tomomichi Oya. "How does the brain solve muscle redundancy? Filling the gap between optimization and muscle synergy hypotheses". In: *Neuroscience research* 104 (2016), pp. 80–87.

[48] Koh Hosoda et al. "Actuation in legged locomotion". In: *Bioinspired legged locomotion*. Elsevier, 2017, pp. 563–622.

[49] Koh Hosoda et al. “Pneumatic-driven jumping robot with anthropomorphic muscular skeleton structure”. In: *Autonomous Robots* 28.3 (2010), pp. 307–316.

[50] Fumiya Iida, Juergen Rummel, and Andre Seyfarth. “Bipedal walking and running with spring-like biarticular muscles”. In: *Journal of biomechanics* 41.3 (2008), pp. 656–667.

[51] Fumiya Iida et al. “Toward a human-like biped robot with compliant legs”. In: *Robotics and Autonomous Systems* 57.2 (2009), pp. 139–144.

[52] M Ishikawa, J Pakaslahti, and PV Komi. “Medial gastrocnemius muscle behavior during human running and walking”. In: *Gait & posture* 25.3 (2007), pp. 380–384.

[53] Yuri P Ivanenko et al. “Modular control of limb movements during human locomotion”. In: *Journal of Neuroscience* 27.41 (2007), pp. 11149–11161.

[54] Karen Junius et al. “Biarticular elements as a contributor to energy efficiency: biomechanical review and application in bio-inspired robotics”. In: *Bioinspiration & biomimetics* 12.6 (2017), p. 061001.

[55] Shuuji Kajita et al. “The 3D linear inverted pendulum mode: A simple modeling for a biped walking pattern generation”. In: *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*. Vol. 1. IEEE. 2001, pp. 239–246.

[56] Yongtae G Kim et al. “A voice activated bi-articular exosuit for upper limb assistance during lifting tasks”. In: *Robotics and Computer-Integrated Manufacturing* 66 (2020), p. 101995.

[57] Heidi Knuesel, Hartmut Geyer, and Andre Seyfarth. “Influence of swing leg movement on running stability”. In: *Human movement science* 24.4 (2005), pp. 532–543.

[58] Minayori Kumamoto, Toru Oshima, and Tomohisa Yamamoto. “Control properties induced by the existence of antagonistic pairs of bi-articular muscles—Mechanical engineering model analyses”. In: *Human Movement Science* 13.5 (1994), pp. 611–634.

[59] ARTHUR D Kuo. "The action of two-joint muscles: the legacy of WP Lombard". In: *Classics in movement science* (2001), pp. 289–315.

[60] Jason J Kutch and Francisco J Valero-Cuevas. "Muscle redundancy does not imply robustness to muscle dysfunction". In: *Journal of biomechanics* 44.7 (2011), pp. 1264–1270.

[61] Dominic Lakatos et al. "Design and control of compliantly actuated bipedal running robots: Concepts to exploit natural system dynamics". In: *2014 IEEE-RAS International Conference on Humanoid Robots*. IEEE. 2014, pp. 930–937.

[62] Dominic Lakatos et al. "Dynamic bipedal walking by controlling only the equilibrium of intrinsic elasticities". In: *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. IEEE. 2016, pp. 1282–1289.

[63] Mark L Latash. *Fundamentals of motor control*. Academic Press, 2012.

[64] RD Lewis and JMM Brown. "Influence of muscle activation dynamics on reaction time in the elderly". In: *European journal of applied physiology and occupational physiology* 69.4 (1994), pp. 344–349.

[65] Hyerim Lim and Sukyung Park. "A bipedal compliant walking model generates periodic gait cycles with realistic swing dynamics". In: *Journal of Biomechanics* 91 (2019), pp. 79–84.

[66] Florian Loeffl et al. "The dlr c-runner: Concept, design and experiments". In: *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. IEEE. 2016, pp. 758–765.

[67] Warren P Lombard. "The action of two-joint muscles". In: *American Physical Education Review* 8.3 (1903), pp. 141–145.

[68] Philippe Malcolm et al. "Bi-articular knee-ankle-foot exoskeleton produces higher metabolic cost reduction than weight-matched mono-articular exoskeleton". In: *Frontiers in neuroscience* 12 (2018), p. 69.

[69] Junya Masumoto and Nobuyuki Inui. "Motor control hierarchy in joint action that involves bimanual force production". In: *Journal of Neurophysiology* 113.10 (2015), pp. 3736–3743.

[70] H-M Maus et al. "Upright human gait did not provide a major mechanical challenge for our ancestors". In: *Nature communications* 1.1 (2010), pp. 1–6.

[71] Daniel Maykranz and Andre Seyfarth. "Compliant ankle function results in landing-take off asymmetry in legged locomotion". In: *Journal of theoretical biology* 349 (2014), pp. 44–49.

[72] Thomas A McMahon and George C Cheng. "The mechanics of running: how does stiffness couple with speed?" In: *Journal of biomechanics* 23 (1990), pp. 65–78.

[73] Falk Mörl, Tobias Siebert, and Daniel Häufle. "Contraction dynamics and function of the muscle-tendon complex depend on the muscle fibre-tendon length ratio: a simulation study". In: *Biomechanics and modeling in mechanobiology* 15.1 (2016), pp. 245–258.

[74] JB Morrison. "The mechanics of muscle function in locomotion". In: *Journal of Biomechanics* 3.4 (1970), pp. 431–451.

[75] Toshiyuki Murao et al. "Bilateral control of nonlinear teleoperation for 2DOF robot manipulators with antagonistic bi-articular muscles". In: *2017 11th Asian Control Conference (ASCC)*. IEEE. 2017, pp. 204–209.

[76] Yuto Nakanishi et al. "Design approach of biologically-inspired musculoskeletal humanoids". In: *International Journal of Advanced Robotic Systems* 10.4 (2013), p. 216.

[77] Kenichi Narioka, Toshiyuki Homma, and Koh Hosoda. "Roll-over shapes of musculoskeletal biped walker". In: (2013).

[78] Rezvan Nasiri, Mojtaba Rayati, and Majid Nili Ahmadabadi. "Feedback from mono-articular muscles is sufficient for exoskeleton torque adaptation". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 27.10 (2019), pp. 2097–2106.

[79] Atabak Nejadfard et al. "Moment Arm Analysis of the Biarticular Actuators in Compliant Robotic Leg C arl". In: *Conference on Biomimetic and Biohybrid Systems*. Springer. 2018, pp. 348–360.

[80] Takumi Nishimura and Naoki Motoi. “Motion Control Method Based on Two-link Manipulator Model with Bi-articular Muscle Considering Planetary Gear”. In: *2021 IEEE International Conference on Mechatronics (ICM)*. IEEE. 2021, pp. 1–6.

[81] Nico Nitzsche et al. “Effect of plyometric training on dynamic leg strength and jumping performance in rhythmic gymnastics: A preliminary study”. In: *Isokinetics and Exercise Science* Preprint (2021), pp. 1–9.

[82] Keita Ogawa, Kenichi Narioka, and Koh Hosoda. “Development of whole-body humanoid “Pneumat-BS” with pneumatic musculoskeletal system”. In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2011, pp. 4838–4843.

[83] Rolf Pfeifer and Gabriel Gómez. “Morphological computation–connecting brain, body, and environment”. In: *Creating brain-like intelligence*. Springer, 2009, pp. 66–83.

[84] Rolf Pfeifer, Max Lungarella, and Fumiya Iida. “Self-organization, embodiment, and biologically inspired robotics”. In: *science* 318.5853 (2007), pp. 1088–1093.

[85] RE Poppele, G Bosco, and AM Rankin. “Independent representations of limb axis length and orientation in spinocerebellar response components”. In: *Journal of neurophysiology* 87.1 (2002), pp. 409–422.

[86] Jerry Pratt et al. “Capture point: A step toward humanoid push recovery”. In: *2006 6th IEEE-RAS international conference on humanoid robots*. IEEE. 2006, pp. 200–207.

[87] Boris I Prilutsky and Vladimir M Zatsiorsky. “Tendon action of two-joint muscles: transfer of mechanical energy between joints during jumping, landing, and running”. In: *Journal of biomechanics* 27.1 (1994), pp. 25–34.

[88] Katayon Radkhah et al. “Concept and design of the biobiped1 robot for human-like walking and running”. In: *International Journal of Humanoid Robotics* 8.03 (2011), pp. 439–458.

[89] Aida Mohammadi Nejad Rashty, Maziar Ahmad Sharbafi, and Andre Seyfarth. "SLIP with swing leg augmentation as a model for running". In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2014, pp. 2543–2549.

[90] Thomas J Roberts. "The integrated function of muscles and tendons during locomotion". In: *Comparative Biochemistry and Physiology Part A: Molecular & Integrative Physiology* 133.4 (2002), pp. 1087–1099.

[91] Christian Rode, Tobias Siebert, and Reinhard Blickhan. "Titin-induced force enhancement and force depression: a 'sticky-spring'mechanism in muscle contractions?" In: *Journal of theoretical biology* 259.2 (2009), pp. 350–360.

[92] Juergen Rummel and Andre Seyfarth. "Passive stabilization of the trunk in walking". In: *hip* 180.190 (2010), p. 200.

[93] Alireza Sarmadi et al. "Concerted control of stance and balance locomotor subfunctions—leg force as a conductor". In: *IEEE Transactions on Medical Robotics and Bionics* 1.1 (2019), pp. 49–57.

[94] Ryuki Sato et al. "Vertical Jumping by a Legged Robot With Upper and Lower Leg Bi-Articular Muscle–Tendon Complexes". In: *IEEE Robotics and Automation Letters* 6.4 (2021), pp. 7572–7579.

[95] Gerrit Jan Van Ingen Schenau. "From rotation to translation: Constraints on multi-joint movements and the unique action of bi-articular muscles". In: *Human Movement Science* 8.4 (1989), pp. 301–337.

[96] Syn Schmitt, Michael Günther, and Daniel FB Häufle. "The dynamics of the skeletal muscle: A systems biophysics perspective on muscle modeling with the focus on Hill-type muscle models". In: *GAMM-Mitteilungen* 42.3 (2019), e201900013.

[97] C Schumacher et al. "Biarticular muscles in light of template models, experiments and robotics: a review". In: *Journal of the Royal Society Interface* 17.163 (2020), p. 20180413.

[98] Christian Schumacher et al. "Biarticular muscles are most responsive to upper-body pitch perturbations in human standing". In: *Scientific reports* 9.1 (2019), pp. 1–14.

[99] Steffen Schütz et al. “CARL—A compliant robotic leg featuring mono-and biarticular actuation”. In: *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. IEEE. 2017, pp. 289–296.

[100] A Seyfarth, Reinhard Blickhan, and JL Van Leeuwen. “Optimum take-off techniques and muscle design for long jump”. In: *Journal of Experimental Biology* 203.4 (2000), pp. 741–750.

[101] André Seyfarth et al. “Towards bipedal jogging as a natural result of optimizing walking speed for passively compliant three-segmented legs”. In: *The International Journal of Robotics Research* 28.2 (2009), pp. 257–265.

[102] Maziar A Sharbafi and Andre Seyfarth. “FMCH: A new model for human-like postural control in walking”. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2015, pp. 5742–5747.

[103] Maziar A Sharbafi et al. “Fundamental subfunctions of locomotion”. In: *Bioinspired Legged Locomotion*. Elsevier, 2017, pp. 11–53.

[104] Maziar Ahmad Sharbafi et al. “A new biarticular actuator design facilitates control of leg function in BioBiped3”. In: *Bioinspiration & biomimetics* 11.4 (2016), p. 046003.

[105] Maziar Ahmad Sharbafi et al. “Reconstruction of human swing leg motion with passive biarticular muscle models”. In: *Human movement science* 52 (2017), pp. 96–107.

[106] ZH Shen and JE Seipel. “A fundamental mechanism of legged locomotion with hip torque and leg damping”. In: *Bioinspiration & biomimetics* 7.4 (2012), p. 046010.

[107] Dongjun Shin, Xiyang Yeh, and Oussama Khatib. “Variable radius pulley design methodology for pneumatic artificial muscle-based antagonistic actuation systems”. In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2011, pp. 1830–1835.

[108] T Siebert and C Rode. “Computational modeling of muscle biomechanics”. In: *Computational modelling of biomechanics and biotribology in the musculoskeletal system*. Elsevier, 2014, pp. 173–204.

[109] Erik B Simonsen, Leif Thomsen, and Klaus Klausen. “Activity of mono-and biarticular leg muscles during sprint running”. In: *European Journal of Applied Physiology and Occupational Physiology* 54.5 (1985), pp. 524–532.

[110] Heiki Sooneste et al. “Effects of training volume on strength and hypertrophy in young men”. In: *The Journal of Strength & Conditioning Research* 27.1 (2013), pp. 8–13.

[111] NASA Anthropometric Sourcebook. “NASA Reference Publication No. 1024”. In: *Houston TX* (1978).

[112] Manoj Srinivasan and Andy Ruina. “Computer optimization of a minimal biped model discovers walking and running”. In: *Nature* 439.7072 (2006), pp. 72–75.

[113] Markus Tilp, S Steib, and W Herzog. “Force–time history effects in voluntary contractions of human tibialis anterior”. In: *European journal of applied physiology* 106.2 (2009), pp. 159–166.

[114] André Tomalka et al. “The active force–length relationship is invisible during extensive eccentric contractions in skinned skeletal muscle fibres”. In: *Proceedings of the Royal Society B: Biological Sciences* 284.1854 (2017), p. 20162497.

[115] Gelsy Torres-Oviedo and Lena H Ting. “Subject-specific muscle synergies in human balance control are consistent across different biomechanical contexts”. In: *Journal of neurophysiology* 103.6 (2010), pp. 3084–3098.

[116] European Union. “EU Medical Device Regulation”. In: www.eur-lex.europa.eu, 2017.

[117] Cao Van Kien, Nguyen Ngoc Son, and Ho Pham Huy Anh. “Adaptive Fuzzy Sliding Mode Control for Nonlinear Uncertain SISO System Optimized by Differential Evolution Algorithm”. In: *International Journal of Fuzzy Systems* 21.3 (2019), pp. 755–768.

[118] Sabine MP Verschueren, Paul J Cordo, and Stephan P Swinnen. “Representation of wrist joint kinematics by the ensemble of muscle spindles from synergistic muscles”. In: *Journal of neurophysiology* (1998).

[119] RP Wells. "Mechanical energy costs of human movement: an approach to evaluating the transfer possibilities of two-joint muscles". In: *Journal of biomechanics* 21.11 (1988), pp. 955–964.

[120] Alan M Wilson et al. "Horses damp the spring in their step". In: *Nature* 414.6866 (2001), pp. 895–899.

[121] David A Winter. "Human balance and posture control during standing and walking". In: *Gait & posture* 3.4 (1995), pp. 193–214.

[122] Simon Wolfen et al. "Bioinspired pneumatic muscle spring units mimicking the human motion apparatus: benefits for passive motion range and joint stiffness variation in antagonistic setups". In: *2018 25th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*. IEEE. 2018, pp. 1–6.

[123] Keyan Zahedi and Nihat Ay. "Quantifying morphological computation". In: *Entropy* 15.5 (2013), pp. 1887–1915.