WILLIAM SCAFF

Pneumatic artificial muscles: model, design, fabrication, sensing and control strategies for electromagnetic risk applications

Thesis presented to Polytechnic School of São Paulo University to obtain the title of Doctor of Science.

São Paulo
2023

WILLIAM SCAFF

Pneumatic artificial muscles: model, design, fabrication, sensing and control strategies for electromagnetic risk applications

Versão Corrigida

Thesis presented to Polytechnic School of São Paulo University to obtain the title of Doctor of Science.

Field of concentration:
Control and Automation

Advisor:
Oswaldo Horikawa

São Paulo
2023

Este exemplar foi revisado e corrigido em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, 20 de maio de 2023

Assinatura do autor: William Scaff

Assinatura do orientador:

Catalogação-na-publicação

Scaff, William
Músculos Artificiais Pneumáticos: modelos, projeto, fabricação, sensoriamento e estratégias de controle para aplicações de risco eletromagnético / W. Scaff -- versão corr. -- São Paulo, 2023.
137 p.

Tese (Doutorado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos.

1.controle ótimo 2.otimização não linear 3.ressonância magnética 4.reabilitação I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos II.t.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Artificial muscles are materials or devices that changes shape with a stimulus. These biological inspired actuators are getting popular because of their advantages over conventional actuators, such as electric motors, hydraulic and pneumatic cylinders. Pneumatic artificial muscles, for example, have several advantages over conventional actuators, such as the compliance, actuation flexibility and high power-to-weight ratio, and also have the flexibility to be constructed without conductive and/or ferromagnetic materials. These characteristics makes artificial muscles suitable for many applications where conventional actuators cannot be used or have limited performance, as in high electric and/or magnetic field environments such as inside magnetic resonance imaging or explosion risk environments. However, pneumatic artificial muscles usage is limited because of the complexity of its implementation. Furthermore, designing and controlling a system actuated by artificial muscles have never been done with totally Magnetic Resonance Imaging compatible materials and sensors. To improve the applicability of pneumatic muscles, this thesis develops a methodology for designing, sensing and controlling devices for electromagnetic risk applications. And, to address the control problem, an optimal control approach is used, considering several optimization algorithms to tune the controller, in a simulated environment or in an experimental environment. In this way, parameter tuning can be customized to each specific application, translating its requirements to an objective function. A new optimization algorithm is proposed and used to tune the parameters of the controller, resulting in a 48.15% shorter learning time and a 8% improvement on parameter quality compared to Bayesian Optimization, a state-of-the-art stochastic optimization algorithm.

keywords: Pneumatic artificial muscle, magnetic resonance imaging compatibility, rehabilitation, position control, optimization algorithms.

RESUMO

Músculos artificiais são materiais ou dispositivos que mudam de forma com um estímulo. Esses atuadores bioinspirados estão se tornando populares pelas suas vantagens sobre atuadores convencionais, como motores elétricos e cilindros hidráulicos e pneumáticos. Os músculos artificias pneumáticos, por exemplo, possuem diversas vantagens em relação aos atuadores convencionais como a complacência, flexibilidade de atuação e alta razão de potência-peso, além de permitirem uma fabricação livre de materiais condutores de energia elétrica e ferromagnéticos. Estas características os tornam aptos para aplicações onde atuadores convencionais não podem ser utilizados ou teriam sua performance limitada, como é o caso de ambientes de ressonância magnética, com risco de explosão e com presença de campos elétricos intensos. Contudo, a utilização dos músculos artificiais pneumáticos é limitada pela complexibilidade de sua implementação, dado pelo seu comportamento altamente não linear. Além disso, um sistema atuado por músculos artificiais pneumáticos nunca foi construído com materiais totalmente compatíveis com ambientes de ressonância magnética, por exemplo. Para contribuir no avanço da utilização dos músculos pneumáticos, esta tese desenvolve uma metodologia para o projeto, sensoriamento e controle de dispositivos para aplicações de risco eletromagnético. Para endereçar o problema do controle, uma abordagem de controle ótimo é utilizada, considerando diversos algoritmos de otimização para ajuste do controlador, tanto em ambiente simulado, como em ambiente experimental. Desta forma, o ajuste dos parâmetros pode ser feito de maneira customizada para cada aplicação traduzindo-se os requisitos para uma função objetivo a ser otimizada. Um novo algoritmo de otimização foi proposto e utilizado para sintonizar os parâmetros do controlador, resultando em um aprendizado 48,15% mais rápido e uma melhora de 8% na qualidade dos parâmetros, comparado à Otimização Bayesiana, um algoritmo considerado estado da arte para otimizações estocásticas.

palavras-chave: Músculos artificiais pneumáticos, compatibilidade com ressonância magnética, reabilitação, controle de posição, algoritmos de otimização.

1 INTRODUCTION

The term Artificial Muscle (AM) is used to classify materials or devices that contracts with a stimulus, by analogy with the human skeletal muscle (TONDU, 2015). However, the term is used in a broader manner, where the contraction is often interpreted as a change of shape in a general way (TONDU, 2015). Materials and devices classified as artificial muscles are getting increasingly popular in the literature. This is because conventional actuators, such as electric motors and hydraulic and pneumatic cylinders struggle to achieve the same performance as biological muscles in some tasks.

The study of animals locomotion and some plants gave rise to a new era of biological inspired actuators. These actuators are responsible for a new field of robotics, known as soft robotics. Soft robotics main advantages are the compliant behavior of the robots, versatility, lightweight and compactness compared with conventional robotics. Advances in the field of soft robotics could benefit many areas such as minimally invasive surgical devices, exploration robots, rehabilitation devices, assistive devices and robots for human interaction and cooperation (MADDEN et al., 2004; MAJIDI, 2014).

Artificial muscles are not only related to soft robotics, but also to adding flexibility and efficiency to industry applications and to developing new technologies. There are several types of artificial muscles being developed and studied. One of them is the Pneumatic Artificial Muscle (PAM).

PAMs are interesting actuators that have similar static and dynamic characteristics to the human skeletal muscle and, therefore, have many applications in orthetics and biomimetic systems (DAERDEN; LEFEBER, 2002; TONDU et al., 2009). They are lightweight, easy to fabricate, low cost, compliant and inherits the advantages of pneumatic systems.

The invention of the PAM can be traced back to 1930, by S. Garasiev (DAERDEN; LEFEBER, 2002; TAKOSOGLU et al., 2016). Later, it was used by McKibben in the 1950's to power an orthetic upper limb (TONDU; LOPEZ, 1997; DAERDEN; LEFEBER, 2002). However, it was not given much attention because of the difficulty of working with these muscles with the technology at that time (DAERDEN; LEFEBER, 2002). The progress on valve technology and control techniques have brought back the interest on pneumatic muscles, more specifically, the McKibben PAM.

The McKibben PAM is the most used PAM today (KOTHERA; PHILEN; TONDU, 2012). One insteresting characteristic of the McKibben PAM is that it can be built without ferromagnetic and electric conductive materials. This characteristic makes them suitable to power devices in intense electromagnetic field environments without affecting or being affected by these fields, unlike any other conventional actuator. It is still not widely used in industry or outside academia because it is highly nonlinear and difficult to model and control accurately (THANH; AHN, 2006; REPPERGER; JOHNSON; PHILIPS, 1999; CHAN et al., 2003; ANH, 2010; MEDRANO-CERDA; BOWLER; CALDWELL, 1995; WU et al., 2009a; SITUM; HERCEG, 2008; TANG et al., 2016; GUO et al., 2021).

There are many researches related to PAM's control. Classical linear control techniques such as PI and PID control have been proposed, because they are simple to design and have high adaptability (SAKTHIVELU; CHONG, 2015).

However, many of the PID controllers are tuned based on the Ziegler-Nichols method, which is known to produce poor results in many cases (SAKTHIVELU; CHONG, 2015; ÅSTRÖM; HÄGGLUND, 2001; SCHRÖDER et al., 2003).

Many nonlinear control techniques were also proposed. Sliding-mode control is known to have chattering problems (LIN; YEN; WANG, 2013). Neural network based control may need thousands of training samples before good position control and suffers from time shift in response because of the iterations of the neural network (TANG et al., 2016).

Other control techniques that requires linearized models, may suffer for unmodeled effects and nonlinear behaviour. Sofisticated control techniques, such as Adaptative Fuzzy Non-singular Terminal Sliding Mode Control (AFNTSMC) may achieve robust, stable, smooth and fast control action, but have worse positioning accuracy than other methods and are complex to implement (TANG et al., 2016).

PID controllers are one of the most common controllers used today and may have similar performance to very complex control techniques with adequate gains. A method for choosing good PID gains can make the implementation of PAMs simpler and promote it's use in industry and other areas. Also, it is difficult to tune the controller with custom objectives. Therefore, a method to tune the controller for different performance characteristics, such as slow response but accurate positioning (surgeries) or the fastest response as possible without overshooting, can provide mode flexibility to PAM's application. An easy strategy for designing a controller to meet the application's requirements is still missing.

The physical modeling of the McKibben PAM is very difficult because of the high non-linearity of the muscle. PAMs static and dynamic behavior is mostly dependent on the geometric characteristics as well as the materials used in the fabrication of the muscle. This means that a muscle can be tuned for a specific application, e.g. exert higher force with less pressure and/or have wider contractile range with same muscle length. However, there is little known in the literature about McKibben PAMs characteristics with respect to using different materials and designs. Additionally, material characteristics may vary from suppliers or over time, a slightly difference in the muscle construction may have significant affect on the static and dynamic characteristics of the muscle, contributing to even more complexity to PAM's applicability beyond academia and tasks where positioning accuracy of the actuator is not relevant.

Furthermore, the usage of PAM's in the context of electromagnetic risk applications is not extensively explored, although they can be made without ferromagnetic and conductive materials. There are also sensors and materials that can be used to produce a totally compatible device for very restrictive applications such as functional Magnetic Resonance (fMRI) manipulators.

Because every variation of the muscle and mechanisms can drastically affect the static and dynamic characteristics of the model, and thus, affect the control performance, it is clear that a learning or adaptive method should be used to control the muscle.

1.1 Objectives

Because of the limiting factors discussed briefly in section 1, the objectives of this work are:

- Analise how different materials influence the PAMs dynamic and static behavior;

- Propose a mathematical model capable of predicting muscle performance characteristics, which can be used to design muscles based on specific application's requirements, such as pressure ranges, contraction ranges and maximum forces;

- Analise sensoring mechanisms for the muscle which are compatible with electromagnetic risk applications;

- Propose a controller for the muscle and

- Propose a method for tunning the controller to the desired dynamic characteristics of the specific application.

1.2 Methodology

To achieve the objectives presented in 1.1, this thesis uses the following methodology:

- **Test bench** - Build a test bench to evaluate muscle's characteristics and control performance. Two main test benches were constructed: linear movement bench and angular movement bench. The linear movement bench is used to evaluate muscle characteristics, such as force, hysteresis, pressures, dynamics, and so on, and the angular movement bench was used mainly to evaluate the parameter tuning with different commonly muscle powered systems and other sensing techniques (camera based sensing).

- **PAM prototype** - Analise alternative materials for the braid and, specially, the membrane. Build prototypes with these materials to evaluate the force, contraction and durability. Two main categories of membranes are considered: elastic and plastic (folding membranes).

- **Static model** - Verify if the mathematical models are capable of predicting the static characteristics of the prototypes constructed in the Materials study stage. Verify the limitations and possible improvements to the models. The proposed model should be adequate for designing muscles for specific applications.

- **Sensing** - Propose sensing methods for measuring the muscle's attributes, which can be used in electromagnetic risk environments. Two main techniques were further evaluated: force-based feedback and movement based feedback.

- **Controller** - Test controllers that are flexible and easy to implement. Initially a proportional integrative derivative (PID) is used because of its flexibility, easy implementation and wide usage in academia and industry.

- **Dynamic model** - Implementation of dynamic models for the muscle and for the basic elements of the system. The model will be validated using the test bench built and used to simulate the system.

- **Simulation and optimization** - Propose optimization methods for tuning the controller. The proposed methods will be tested in a simulated environment according to the developed dynamic model. Because optimization methods generally need many iterations before good results can be achieved and the simulation is computationally expensive, the "Águia" supercomputer can be used. The supercomputer is available at the "interNuvem" of the University of São Paulo through the superintendence of information technology. Best algorithms are evaluated in the test bench for real time parameter tuning.

- **Analysis** - Process the data of the experiments and study the efficacy of the proposed methods.

1.3 Thesis organization

This thesis is organized as follows. Chapter 2 describes some potential applications that can be built using the proposed method. Current challenges for these applications are addressed and the general strategy for the proposed method is described. In chapter 3, alternative actuators are reviewed with the emphasis on the PAMs. Several types of PAMs are reviewed and each design characteristic is described. PAMs mathematical models are also reviewed. Chapter 4 reviews control strategies that were used in the literature for positioning control of PAMs. Optimization methods are also covered for tuning the controllers. Chapter 5 reviews compatible sensors used or that can be used in electromagnetic risk environments. An image based method for measuring angles is developed and described. Chapter 6 describes the system modeling and simulation of nonlinear systems. These models and simulations are used to tune the controller in chapter 6.8. Other tuning methods which don't depend on mathematical models are also addressed in chapter 6.8. To address the difficulties in optimizing the controller in the bench, chapter 7 describes alternative optimization algorithms. Chapter 8 describes the conclusions of this thesis and chapter 9 describes future works.

2 POTENTIAL APPLICATIONS, CHALLENGES AND GENERAL STRATEGY

There are some applications where conventional actuators cannot be used or have limited performance. This chapter describes some applications that can benefit from systems that are fully compatible with electromagnetic risk environments.

There are several challenges on applying mechatronic devices in such environments. This chapter addresses these difficulties and proposes a general strategy for overcoming them.

2.1 Robotic applications and current challenges

Robotics is still considered a futuristic term for the population. However, many don't realize that it is part of our daily lives, and will continue to be in the future in a large scale.

The rapid evolution of technology is feared by people and, specially employees, that consider robots as job thieves. Actually, the term "robot" was born in the Czech literature when the brothers Capek wrote stories and plays about robots evolution and revolt against humans (HOCKSTEIN et al., 2007). Robots became more popular from the stories of Isaac Asimov and his three laws of robotics, that dealt with the danger of robots hurting people and taking over the world

(HOCKSTEIN et al., 2007).

Real robots history started in 1958, when General Motors introduced Unimate, a robotic arm, to assist in the production of automobiles (HOCKSTEIN et al., 2007). Since then, many robots were applied in various areas to increase efficiency and lower costs. Robots also have several advantages to humans in specific tasks, such as accuracy, working time, focus, used for dangerous activities, strength and many others.

Robots have various applications in many different areas. The sections below will address some of them, that are relevant for this thesis.

2.1.1 Robotic surgery

The history of robots in surgery started in 1985, when an industrial robotic arm was modified to perform a stereotactic brain biopsy - a biopsy guided by images, typically from Magnetic Resonance Imaging (MRI) or Computed Tomography (CT) - achieving 0.05mm of positioning accuracy (HOCKSTEIN et al., 2007). This robot served as prototype for the Neuromate, surgical robot from Integrated Surgical Systems, and received the Food and Drug Administration (FDA) aproval in 1999 (HOCKSTEIN et al., 2007). Many other robotic systems were created such as Robodoc, Acrobot, RX-130, CASPAR, Evolution 1, Zeus and da Vinci, the most successful surgical robotic system (HOCKSTEIN et al., 2007; BARGAR; BAUER; BÖRNER, 1998; JAKOPEC et al., 2003; FEDERSPIL et al., 2003).

The position accuracy of the robotic arm is one advantage of the robotic surgery. The surgeons hand tremor can be filtered and the movements can be scaled down, for more precise procedures, minimally invasive surgery and lower tissue damage. These advantages makes robots very likely to substitute human hand-made surgery in the near future. Many hospitals in the world already have robotic surgery available. Just in São Paulo there are several options for robotic surgery, such as the 9 de Julho Hospital, Sírio Libanes Hospital, Albert Einstein Hospital, São Luiz Hospital and Cancer Institute of São Paulo.

The ergonomy is also improved because the surgeon is sited in the control console, even if the location of the surgery is of difficult access.

Other advantages of robotic surgery were recognized by Scott Fisher, a scientist at the National Aeronautics and Space Administration (NASA), and Joe Rosen, a plastic surgeon from Stanford University, such as the use of telepresence and remotely controlled surgery (HOCKSTEIN et al., 2007)

Pentagon's Defense Advanced Research Projects Agency (DARPA) started researching remotely controlled surgery to allow a surgeon in a safe place to treat a soldier in the battlefield (HOCKSTEIN et al., 2007). Computer Motion Inc., initially funded by DARPA, created the Zeus surgical system, which in 2001 performed the first telepresence surgical procedure (HOCKSTEIN et al., 2007). It was a laparoscopic cholecystectomy procedure that was performed on a patient in Strasbourg, France, by a surgeon at 6.115 km away, using a connection of 155ms of latency (HOCKSTEIN et al., 2007).

Some drawbacks of robotic surgery include the lack of haptic feedback that a surgeon has in a common surgery or laparoscopy and the lack of imaging techniques compatibility, because these devices are made with ferromagnetic and conductive materials.

2.1.1.1 MRI guided surgery

There are many cases where the surgery or biopsy is guided by images. Some examples of imaging techniques used for guided surgeries and biopsies include MRI, X-ray, CT and ultrasound. Depending on the lesion or region, some imaging techniques are preferred over another. For example, MRI scans are not indicated for detecting calcification, such as kidney stones or breast cancer, and so CT and ultrasound are commonly used for that.

In CT and X-ray patients are exposed to radiation, which is associated with cancer (BRENNER; HALL, 2007). In general, the higher the radiation dose, the higher is the image quality, and so the radiation can be increased to detect small masses. Because of the risks of radiation, alternative techniques should be used if possible.

Ultrasound is a lower cost and widely available technique for guided biopsies and surgeries. However there are some limitations for using ultrasound in regions that absorb or reflect sound waves, causing acoustic shadowing.

MRI is the most expensive technique, but it doesn't expose the patient to ionizing radiation and produces high resolution images, including high contrast images of soft tissue, which is more difficult to see using CT. There are many imaging sequence techniques to observe different tissue characteristics, and new techniques are still being developed. As of an example, it is possible to detect calcification with a new technique using susceptibility-weighted imaging (WU et al., 2009b).

It is also possible to use MRI in real time to guide the procedure. For example, real time MRI was used to guide stem cells delivery to the central nervous system (WALCZAK et al., 2017). The effectiveness of some procedures, such as the mentioned before, is very sensitive to the position accuracy of the instrument that executes the procedure at the target region.

Another example is the stereotactic biopsy. Stereotactic biopsies are potentially fatal procedures that are very sensitive to operator skill and experience (HALL, 1998). Sample error caused by inaccuracy of the sample collection results in diagnostic failure (HALL, 1998). Complex regions such as the brain are very sensitive to the positioning of the needle and the trajectory to reach the sample site (HALL, 1998).

The usage of MRI guidance with robotic surgery is a logical evolution to the current state, because the MRI provides precise information of the target region and the robot is capable of reaching the region with high accuracy. This, however, represents a great engineering breakthrough.

To produce the images, MRI scanners use four main elements: a strong static magnet or coil, a radio-frequency coil, a gradient coil and a receiver coil. The strong static magnetic field is responsible for aligning the protons in the direction of the magnetic field. The radio-frequency coil produces pulses in a certain frequency, called Larmor frequency, to produce the image. The gradient coil is used to produce a magnetic field gradient along inner tube of the scanner, which is used to identify the location of the signals captured by the antenna, or the receiver coils.

The high static magnetic field used by the MRI scanner can cause ferromagnetic materials to turn into projectiles or missiles (BUSHONG; CLARKE, 2014). The gradient magnetic field in the MRI vary with time, which can induce eddy currents in conductors (BUSHONG; CLARKE, 2014). Radio-frequency coils can also induce eddy currents and generate heat (BUSHONG; CLARKE, 2014). Because of these effects, there are many safety procedures and guidelines to prevent accidents.

The American College of Radiology (ACR) defined four zones that identify the electromagnetic risks (BUSHONG; CLARKE, 2014). Zones descriptions are as follows:

- **Zone I** - Areas of free access to the general public;

- **Zone II** - Areas where patients and visitors are controlled and supervised by trained MRI personnel. Is the area of transition between no electromagnetic risk (Zone I) and electromagnetic risk areas (Zone III and Zone IV);

- **Zone III** - Is, in general, the control or computer room that has direct access to the Zone IV. There is a potential for injury from ferromagnetic materials in this area. Access to this zone must be strictly controlled by trained MRI personnel and must have warning signs in the entrance.

- **Zone IV** - Is the MRI scanner room. The entrance to this zone must be visible by MRI personnel at all times to absolutely ensure that no unauthorized person enters the room. Not even during an emergency, unscreened person must not enter this zone. It is the responsibility of the trained MRI personnel to give basic life support assistance and/or move the patient to Zone II or Zone I.

Even though there are many safety procedures, guidelines and training, accidents still occur. There are several reported cases of serious injuries and even death caused by allowing prohibited materials to enter the MRI room. For example, a fourth-degree burn was reported in an extremity of a patient, leading to amputation (HAIK et al., 2009). It was caused by the oximeter in the pulse of the patient, which had conductive wires (HAIK et al., 2009).

Because of the high electromagnetic risk, common robotic devices are not allowed near the scanner, as they represent high risk of injury to the patient, personnel and the equipment. There are classes of MRI compatible devices. Some devices, such as aluminum materials, are not ferromagnetic materials and doesn't represent risk in certain areas of the room. However, aluminum is an electrical conductor which might be heated by induced currents. The induced currents also create a magnetic field, and so forces will act in the aluminum even though it is not ferromagnetic.

Conductive materials must not enter into the scanner even if the material is considered MRI-compatible. One reason is the risk of burn caused by eddy currents, as mentioned before. Another reason is because conductive materials distorts and degrades the image generated by the machine (FISCHER et al., 2008).

The requirements for a MRI compatible robotic device are as follows:

- The device must not represent any risk caused the electromagnetic radiation produced by the MRI;
- The device must not affect the image quality of the MRI, and
- The device must not be affected by the electromagnetic radiation produced by the MRI.

These conditions for MRI safety makes the use of conventional mechatronic devices prohibited.

2.1.2 Robotics in rehabilitation

Stroke is a major cause of mortality and one of the largest cause of disability worldwide (STEWART et al., 2018; DESIKAN et al., 2016). About 1 in each 6 people will suffer from stoke in their lifetime and this statistic is possibly getting worse in the future, with the increase of elderly and the population growth (STEWART et al., 2018; ROTH et al., 2015). For those that survive this event, the rehabilitation process is very important for disability recovery (STEWART et al., 2018).

A stroke can occur in two main ways: a rupture in a weakened blood vessel or an obstruction in the blood vessel that supplies blood to the brain cells. The obstruction can be temporary, called transient ischemic attack or mini-stroke. When the blood supply is interrupted, brain cells start to lose their function and die. If the blood supply is normalized fast enough, brain tissue can be totally recovered (AGUIAR, 2017).

Patients that survive to a stroke may have serious brain damages and severe disability. Symptoms depends on the brain region affected, but many present mobility limitations and movement impairment after the stroke (ANDROWIS; NOLAN, 2017).

The rehabilitation process relies on the neuroplasticity of the brain. With certain activities, the brain relearns the movements lost. However, the quality of the rehabilitation process is important for the recovery (ANDROWIS; NOLAN, 2017; HUANG; KRAKAUER, 2009).

Conventional rehabilitation is challenging. Typically, at least two persons are needed to help a stroke patient walk, and there is still risk of accidents, e.g. falling on the floor. It is also difficult to maintain the patient in a good posture during the rehabilitation. There are also ergonomic concerns for the physiotherapist. Because of the limitating factors associated with conventional rehabilitation, the use of robotic devices grew in the past decade (ANDROWIS; NOLAN, 2017).

After the stroke, there is a period of about three months where the plasticity-dependent recovery is optimum (MORONE et al., 2017). Animal models suggest that the first month after the stroke is where the neuroplasticity is the highest and, therefore, is the best period for recovery from impairment (KRAKAUER et al., 2012; NG et al., 2015). Very early intense rehabilitation, i.e. less than 5 days after stroke, might be harmful though (KRAKAUER et al., 2012).

While conventional rehabilitation may have little impact on impairment recovery over spontaneous biological recovery, robotic rehabilitation have the potential for a greater impact over impairment recovery (HUANG; KRAKAUER, 2009). This is because of the higher dosage and higher intensity protocols achievable by robotic devices and its more reliable measurements over conventional rehabilitation (HUANG; KRAKAUER, 2009; KITAGO et al., 2015).

Typically, the patient waits lying on the hospital bed while in this enhanced plasticity period without any rehabilitation activity. The usage of a robotic device to entertain the patient, e.g. playing a game, while in this period might enhance the impairment recovery, and also gather data for more specific training afterwards.

Measurements monitoring can also enhance the rehabilitation process by tailoring the exercise according to the patients improvements, lowering the assistive force or even exerting a resistive force. Fun, engagement and rewards can also help in the recovery process, providing higher dosage and higher intensity exercises (WIDMER et al., 2017).

Usage of soft robotics can also improve safety and comfort to the current rigid conventional robotic rehabilitation devices. While many robotic devices exists, only a few use soft robotic design. Hard parts, joints and actuators can cause serious injuries to the patient if they are not well placed or the device malfunctions.

Although robotic devices could help enhance the rehabilitation process, currently the effect is not significantly superior (KITAGO et al., 2015). The robotic rehabilitation might be effective for certain patients and not so effective for others (MORONE et al., 2017). Many controversial results are found on the effectiveness of robotic rehabilitation, but different robotic devices stimulates different types of skills in different levels of assistance (MORONE et al., 2017). Using robotics for simple repetition of tasks is not sufficient, and so the question of how the robotic device should act to be most effective for a particular patient is yet to be discovered (MORONE et al., 2017).

2.1.2.1 fMRI-guided rehabilitation

The physiology and mechanisms of rehabilitation is not yet well understood. There are ways to study brain activity during the rehabilitation process, e.g. electroencephalogram (EEG) based monitoring and functional Magnetic Resonance Imaging (fMRI) techniques. EEG signals are captured from the scalp and, therefore, only closer to the scalp regions can be reliably recorded. There are also limitations in capturing lower frequency signals and high noise interference. fMRI measures the blood oxygen-level dependent (BOLD) signal, which occurs when there is an increase of metabolism and blood flow. This signal can be traced to its location of occurrence and, therefore, can be related to regions of the brain being activated by some rehabilitation task.

fMRI has been widely used and still is an important tool for studying the brain and how it behaves in response to certain conditions and diseases, such as Parkinson's disease and stroke (KHANICHEH et al., 2006). Various advances in the technology made possible to capture this information in real time using real time functional magnetic resonance (rtfMRI) (DECHARMS, 2008).

rtfMRI can be used to study and optimize the rehabilitation process. However, there is a limitation with the devices that can be used in the rtfMRI environment to stimulate the patient and, therefore, to study the brain's response to a stimuli (KHANICHEH et al., 2006).

Unfortunately, conventional rigid and soft robotic devices for rehabilitation depend on electronic devices, such as sensors and actuators, ferromagnetic and/or conductive materials that makes them not suitable for usage in a rtfMRI, fMRI or MRI scanner (KHANICHEH et al., 2006). With the growing interest in using MRI for minimally invasive surgeries and fMRI for neuroscience research and rehabilitation, alternative devices such as sensors and actuators were developed (SU et al., 2017; KHANICHEH et al., 2006; STOIANOVICI et al., 2014).

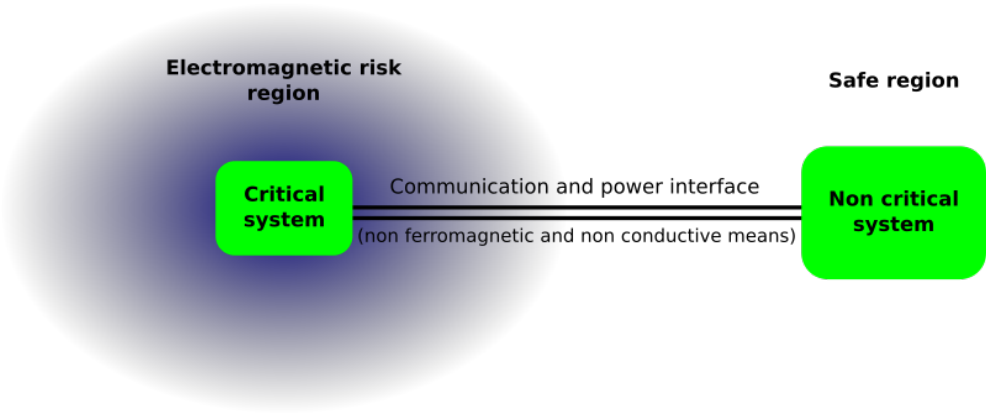
Many devices are classified as compatible, even when using potentially dangerous materials such as conductive materials, as in (KHANICHEH et al., 2006).

2.2 General Strategy

To avoid risks and limitations when designing mechatronic systems for high electric and/or magnetic field environments, the easiest way is to ensure all of the materials used in the system are compatible.

Of course this is not always practical, because generally computers or circuits are used to control mechatronic systems and alternatives for this aren't yet available in a practical way. Therefore, the general strategy when building such systems is to ensure that the critical parts of the systems are built only with compatible materials and that other materials are kept in a safe distance. The communication and power transmission to the critical system is provided by compatible means, as is illustrated in fig. 1.

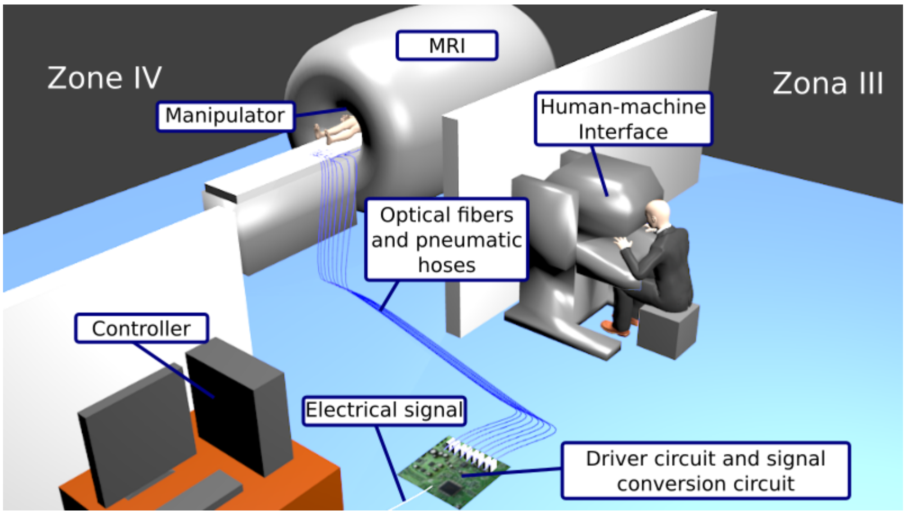
Figure 1: General strategy for designing mechatronic systems for electromagnetic risk applications.


Electromagnetic risk region
Safe region
Critical system
Communication and power interface
(non ferromagnetic and non conductive means)
Non critical system

Source: Own authorship.

For a MRI-guided surgical manipulator, for example, the electromagnetic risk region is inside the MRI scanner and surroundings (zone IV). PAMs built with compatible materials can be used to power the system, such as the one in (SCAFF, 2015), and compatible sensors (which are discussed in chapter 5) can be used in the manipulator, which is the critical system. Pneumatic hoses and optical fiber cables can transmit power and signals, respectively, from the non critical system to the critical system, forming the communication and power interface. At the safe region, air compressors, controllers and valves can be used to process the signals received from the critical system, compute the control law and command the manipulator through the interface. Fig. 2 illustrates this example.

Figure 2: General strategy applied to a MRI-guided surgical system.



Source: Own authorship.

2.3 Chapter conclusion

In this chapter, it was reviewed applications where conventional mechatronic devices cannot be used in a safe way, such as for MRI-guided procedures and fMRI-guided rehabilitation. There are several engineering challenges to overcome when designing systems for these applications. One of the main challenges is to build the system with compatible materials and achieving reasonable performance.

Unfortunately, building a system only with compatible materials is not practical and, therefore, a general strategy for building such systems is proposed, where the critical system is built with compatible materials and all non compatible materials are kept in a safe distance, connected by compatible means.

3 ARTIFICIAL MUSCLES AND MODELS

Artificial muscles are bioinspired actuators which serve as alternative to be used in electromagnetic risk environments. There are several types of artificial muscles, powered by many different forms of energy. Some of them are made from compatible materials and some of them can be powered by means other than electrical or magnetic.

Unfortunately, the majority of artificial muscles that are not electrical or magnetic actuated have limitations of force and/or contraction and, thus, have limited applicability.

Pneumatic artificial muscles, on the other hand, are comparable to human skeletal muscles in terms of force and contraction ranges and can be fabricated with compatible materials. Because of this characteristic, they are the actuator of choice for this thesis. However, they are known to be difficult to control accurately. This challenge will be investigated in the next chapters.

This chapter will describe several types of pneumatic artificial muscles, and how different designs are used to address specific problems. These designs can be used to tune the actuator for the specific application, e.g. favoring contraction or force, over higher hysteresis.

Proposed models for predicting the performance of pneumatic muscles and for the design are also covered in this chapter. These models can also be used to design controllers for the proposed mechatronic system.

3.1 Pneumatic artificial muscles

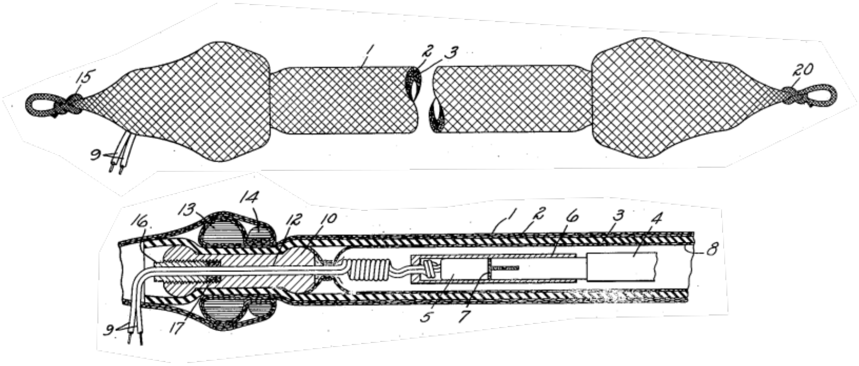
Since the invention of the first pneumatic muscle in 1930, many other types of pneumatic muscles were invented (DAERDEN; LEFEBER, 2002). They can be classified by their working principles and construction as:

- overpressure or underpressure operation;
- braided/netted or embedded membrane and
- stretching membrane or rearranging membrane.

3.1.1 Hugh De Haven muscle

In 1949, Hugh De Haven patented a tensioning device for converting fluid pressure into pulling force. This device is illustrated in fig. 3.

Figure 3: Hugh de Haven muscle schematics.



Source: Adapted from (DE, 1949).

This device was designed for applications where great pulling force and fast action is necessary, e.g. aircraft crash belts and parachutes (DE, 1949).

The muscle is composed of two end fittings, inner and outer braids, elastic bladder, wrappers, protective fireproof layer, lead wires and ignitable gas-producing material. Bowline knots are used to form attachment means at each end of the muscle.

Fluid pressure is provided by black gunpowder, which is ignited by electric cables located inside the muscle's chamber. The gas produced by the gunpowder rapidly fills the chamber, expanding the muscle radially and contracting axially. This conversion of radial expansion to linear contraction is provided by strands, which are around the muscle's chamber in the shape of right and left helices, forming the inner and outer braids. Inside the chamber, a flame resistant material is used to protect the chamber from damage when the gunpowder is ignited (DE, 1949).

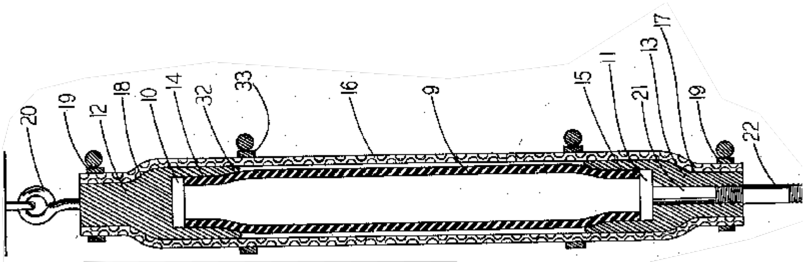
Suggested materials for the muscle are rubber, for the elastic bladder, Nylon, for the braid strands, cotton braid, for the first wrapper, fiberglass tape, for the second wrapper, and cotton braid dipped in ammonium stearate, for the fireproof layer (DE, 1949).

Muscle contraction is about thirty percent of the initial length. It is capable of exerting $6.7kN$ of force with $6.9kPa$ of gas pressure (DE, 1949).

3.1.2 Gaylord muscle

The Gaylord muscle was invented by Richard H. Gaylord and patented in 1958 (GAYLORD, 1958). The muscle is illustrated in fig. 4.

Figure 4: Gaylord muscle schematics.



Source: Adapted from (GAYLORD, 1958).

The muscle is composed of end fittings, a cylindrical expansible tube and a braid (GAYLORD, 1958). When fluid pressure is applied into the expansible tube, it expands pushing the braid radially, which results in an axial contraction due to the way the braid strands are interwoven (GAYLORD, 1958). Another way of actuation is to lower the external pressure, e.g. with the muscle inside a vacuum chamber (GAYLORD, 1958).

The expansible tube is attached to the internal part of the end fittings, while the braid is attached to the external part using clamps (GAYLORD, 1958).

The expansible tube is made of an elastic material such as rubber. The braid can be made with metal wires, Nylon, Orlon, which is an acrylic fiber, or the like (GAYLORD, 1958).

3.1.3 McKibben muscle

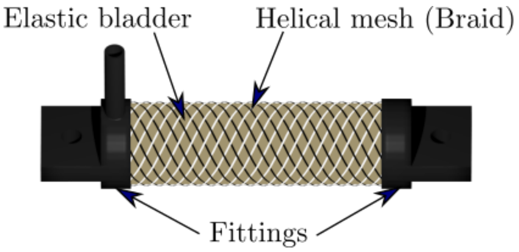
The McKibben muscle is the most used PAM today (DAERDEN, 1999; KOTHERA; PHILEN; TONDU, 2012). It can be classified as an overpressure, braided and stretching membrane muscle. It was introduced in the 1950's by Joseph L. McKibben to produce a prothesis for his daughter, which was a victim of polio (DAERDEN, 1999; BALDWIN, 1969).

Actually, the term "McKibben muscle" is so widespread that many researchers consider other types of muscles as the McKibben muscle or even consider it as a synonym for pneumatic muscle.

The McKibben muscle is characterized by an elastic bladder surrounded by a double helical mesh. The bladder and the mesh is tied in two end fittings at the extremities. Figure 5 ilustrates a McKibben muscle.

The reason why this is the most used PAM today is not clear. Maybe the reason lies in its easy construction, with commercially available materials. On the other hand, as already said above, many researchers use the term "McKibben muscle", when in fact using another type of muscle. One possible reason for this

Figure 5: Example of a McKibben muscle.



Source: Produced by the author.

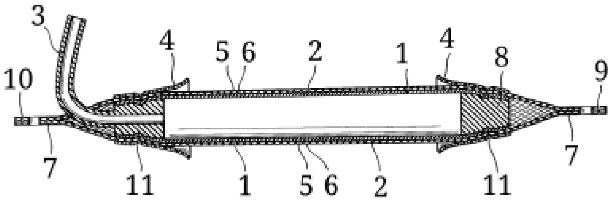
misconception may be because many other types of pneumatic muscles are similar to the McKibben muscle.

The bladder is typically of latex rubber or silicone. Braid strings are usually of Nylon, but other materials such as aramid and fiberglass were already used. The fittings are usually of metallic material, such as aluminum or steel, but engineering plastics can also be used. The fixation of the fittings, braid and bladder are typically of metallic hose clamps or tied with strings.

3.1.4 Beullens muscle

The Beullens muscle is similar to the McKibben muscle, patented in 1989 as a hydraulic or pneumatic drive (BEULLENS, 1989). Figure 6 illustrates a Beullens muscle.

Figure 6: Beullens muscle schematics.



Source: Adapted from (BEULLENS, 1989).

The muscle is, basically, composed by a chamber, a braid, a wall and fittings (BEULLENS, 1989).

In the patent, Beullens describes the chamber material, also referred as bladder in the McKibben muscle, as a resiliently distortable material (BEULLENS, 1989). The braid strings material is made from a flexible and substantially unstretchable material, such as steel wires (BEULLENS, 1989). Therefore, the braid and chamber definition in the patent is compatible with the McKibben muscle.

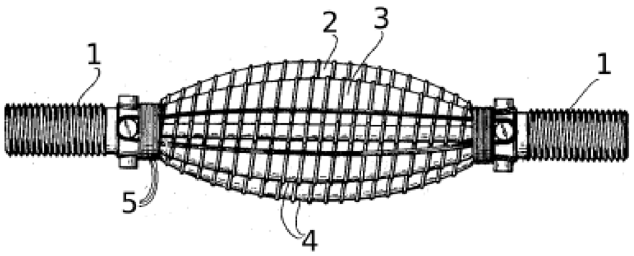
The distinguishing element from the McKibben muscle, the wall, is made from a substantially resiliently distortable material located between the braid and the chamber (BEULLENS, 1989). This is a similar approach of the Rubbertuator to increase the muscle's durability.

The shape of the chamber and the wall is preferably a cylinder, like in the McKibben muscle. The reason for the wall is not clear in the patent, but it serves as a protection to the chamber against the friction. He also considers the use of a lubricant, such as graphite and talc, to lower the friction between the filaments and as a protection for the wall (BEULLENS, 1989).

3.1.5 Yarlott muscle

The Yarlott muscle was invented in 1969 by John M. Yarlott, with patent granted in 1972 (YARLOTT, 1972). Similarly to the McKibben and Beullens muscle, it is composed of a chamber, strings and fittings. The muscle is illustrated in figure 7.

Figure 7: Yarlott muscle schematics.



Source: Adapted from (YARLOTT, 1972).

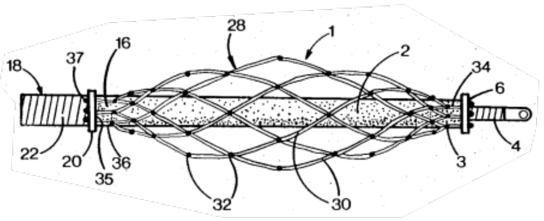
The chamber has a spherical shape. When the chamber is pressurized and the fittings are pulled, the muscle shape is of a prolate (elongated) spheroid (YARLOTT, 1972).

The chamber material is of an elastomer with inextensible strands embedded, to avoid the elastic expansion and possible rupture of the chamber (YARLOTT, 1972). The strands that connects both fittings are responsible for the axial contraction. When the muscle is empty, the strands are straight, which corresponds to the muscle maximum length. When the chamber is inflated, the strands are forced outwards radially, reducing the muscle length.

3.1.6 Kukolj muscle

The Kukolj muscle was patented by Mirko Kukolj in 1988 (KUKOLJ, 1988). This muscle has similar construction to the McKibben muscle. The Kukolj muscle is illustrated in figure 8.

Figure 8: Kukolj muscle schematics.



Source: Adapted from (KUKOLJ, 1988).

The main difference from the McKibben muscle is the braid size. While in the McKibben muscle the braid has similar geometry to the bladder, in the Kukolj muscle the bladder is shorter.

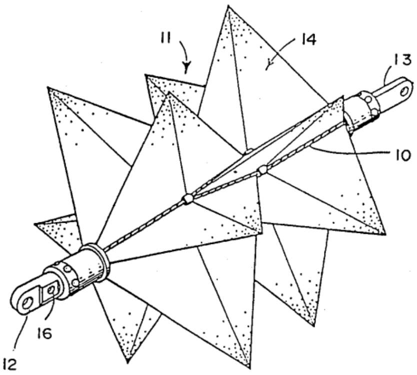
When the Kukolj muscle is not attached to a system, the braid is loose. When attaching the muscle to a system, the bladder is stretched and the braid elongates axially, decreasing its diameter to the bladder diameter, as in the McKibben muscle.

When not pressurized, the Kukolj muscle is exerting an initial pulling force, which increases as the fluid enters the chamber.

3.1.7 Immega/Kukolj muscle

Also known as the ROMAC muscle, it was invented by Guy Immega and Mirko Kukolj and patented in 1990 (IMMEGA; KUKOLJ, 1990). It is basically composed of a chamber and strings, but have have different geometries than McKibben like muscles. The ROMAC muscle is illustrated in fig. 9.

Figure 9: Romac muscle schematics.



Source: Adapted from (IMMEGA; KUKOLJ, 1990).

The chamber membrane is made from woven fibers of nylon or kevlar, a sintetic aramid fiber, bonded with flexible rubber or plastic, thus forming an impermeable membrane (IMMEGA; KUKOLJ, 1990). Chamber shape is formed by various four-sided pyramids, which inflate as fluid pressure increases (IMMEGA; KUKOLJ, 1990). Muliple-strand steel cables connects both fittings, forming a diamond-shaped network on the base of the pyramids (IMMEGA; KUKOLJ, 1990). When the chamber is deflated, the network gets stretched straightly. But when the chamber is inflated, the pyramids pushes the cables, forming the diamond-shaped network, pulling the fittings and contracting the muscle (IM-

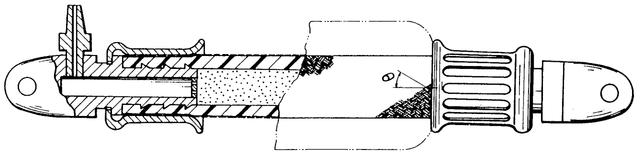
MEGA; KUKOLJ, 1990).

Contractions rates for specific ROMAC muscles can be greater than 45

3.1.8 Brigestone Rubbertuator

The Rubbertuator was invented by Takeo Takagi Yokohama and Yuji Sakaguchi Kawasaki of Brigestone Corporation, and patented in 1986 (TAKAGI; SAKAGUCHI, 1986). The Rubbertuator is illustrated in fig. 10.

Figure 10: Rubbertuator muscle squematics.



Source: Adapted from (TAKAGI; SAKAGUCHI, 1986).

The construction is similar to the McKibben muscle with further improvements. To address the degradation of the bladder by the friction with the helical braid, the Rubbertuator has a protective layer in between these elements. The protective layer is a braid that expands and contracts with the bladder (TAKAGI; SAKAGUCHI, 1986).

Another improvement is the geometry of the filaments in the braid, which have an elliptical shape to avoid damaging the bladder (TAKAGI; SAKAGUCHI, 1986).

To reduce air consumption a filler in the bladder is proposed. The filler is a material that can adapt its shape, such as sand or an incompressible fluid. When the muscle is pressurized, the amount of air needed is reduced by the volume of the filler inside the bladder (TAKAGI; SAKAGUCHI, 1986). This saves air consumption but might cause contamination problems, similar to hydraulic actuators, if the bladder is damaged.

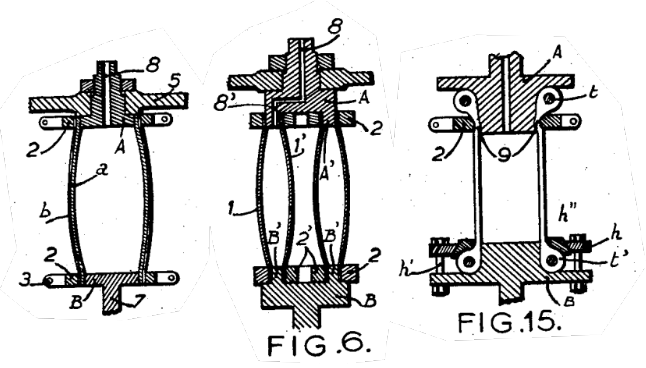
To improve the contraction performance and high fatigue strength, the Rubbertuator has the diameter of the fittings are twice as larger than the mid portion of the muscle (TAKAGI; SAKAGUCHI, 1986).

Contractions of 20-40

3.1.9 Morin muscle

The Morin muscle was patented in 1953 by Alexandre H. Morin as an elastic diaphragm. Fig. 11 illustrates three examples of Morin muscles.

Figure 11: Morin muscle schematics.



Source: Adapted from (HENRI, 1953).

A Morin muscle is basically composed of one or several fluid-tight elastic films, substantially inextensible thread-like elements, forming a chamber which is connected to rigid fittings at the extremities of said elastic chamber (HENRI, 1953). The chamber is fixed at the fittings using clamps as in fig. 11(a) or with the help of a ring as in fig. 11(c).

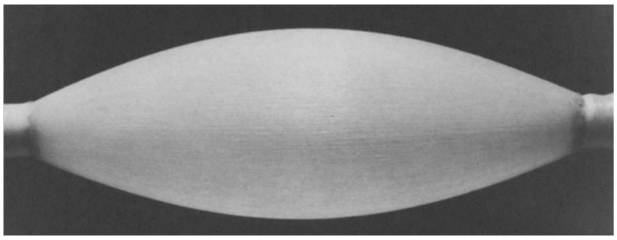
When a fluid enters the chamber, the chamber expands radially and shortens axially. This effect occurs when the thread-like elements are axially disposed between the elastic films or when the thread-like elements are in a shape of a cylindrical helix (HENRI, 1953).

Because this device only shortens axially when pressurized fluid enters the chamber, a compression spring can be used to elongate the device axially when fluid pressure lowers (HENRI, 1953).

3.1.10 Baldwin muscle

The Baldwin muscle is based on the Morin muscle, and was introduced in 1969 by Howard A. Baldwin (BALDWIN, 1969). Fig. 12 is a picture of the prototype of a Baldwin muscle.

Figure 12: Baldwin muscle prototype.



Source: Taken from (BALDWIN, 1969).

This muscle consists of a thin elastic bladder of surgical rubber with embedded glass fibers in the muscle's axial drection. The glass fibers restricts the bladder elongation in the axial direction, while permitting elongation in the radial direction. When fluid pressure is applied inside the bladder, it deforms to a prolate spheroid, contracting the muscle.

Because the bladder is a thin membrane and the glass fibers only restricts elongation in the axial direction, lower pressures have to be used with this muscle to avoid membrane rupture in the radial direction.

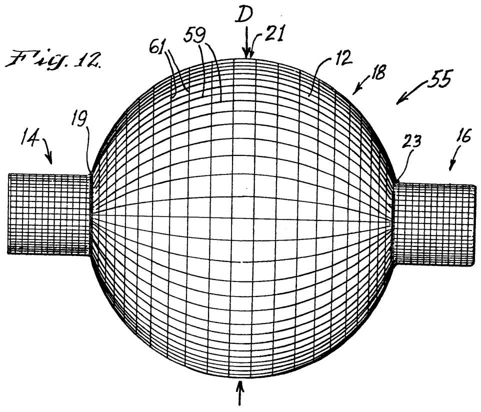
Maximum contraction rates are about 50% of the muscle's rest length, considering a muscle with diameter-to-length ratio of $\frac{1}{10}$. With a $34kg$ load, the muscle contracted $9.5mm$ with a gauge pressure of $138kPa$ (BALDWIN, 1969).

This muscle is capable of 10 to 30 thousand cycles before failure, considering a weight of $4.5kg$ and being pressurized with $103.4kPa$ (BALDWIN, 1969). Details of the geometry of the muscle is not given.

3.1.11 Paynter muscle

The Paynter fluidic muscle was invented by Henry M. Paynter and patented in 1988 (PAYNTER, 1988a). The Paynter muscle is illustrated in fig. 13.

Figure 13: Paynter muscle schematics.



Source: Adapted from (PAYNTER, 1988a).

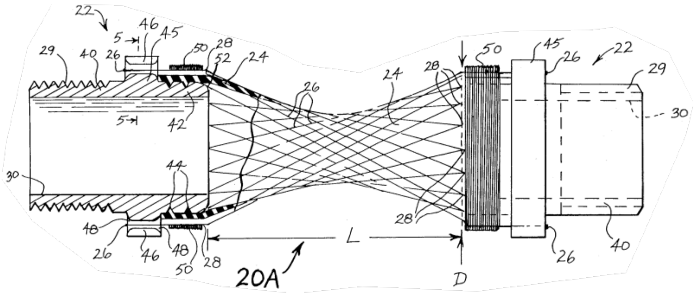
The Paynter muscle is composed of fittings and a bladder with embedded constraining elements. The fluid entrance is located in one or both of the fittings. When inflated, the bladder increases its volume and is constrained by the thread network bonded into it. The network is composed of parallel threads and meridian threads which, when the chamber is inflated, constrains the chamber to a nearly spherical shape (PAYNTER, 1988a).

According to the patent, the advantages of this muscle over the Yarlott muscle are working with higher pressures and increased cycles of operation, due to the network pattern (PAYNTER, 1988a).

3.1.12 Paynter hyperboloid muscle

The hyperboloid muscle was invented by Henry M. Paynter and patented in 1988 (PAYNTER, 1988b). The hyperboloid muscle is illustrated in 14.

Figure 14: Paynter hyperboloid muscle schematics.



Source: Adapted from (PAYNTER, 1988b).

This muscle is composed of two end fittings, inextensible strands and an elastomeric shell (PAYNTER, 1988b). The end fittings are of relatively large size, compared to other muscles, exhibiting less air flow resistance and, thus, faster action response to pressure increase (PAYNTER, 1988b). The strands are disposed in such a way as to form a hyperboloid of revolution in its deflated state (PAYNTER, 1988b). These strands constrains the elastomeric shell which expands into a nearly spherical shape when fluid pressure is provided (PAYNTER, 1988b).

Mentioned materials to construct this muscle are Dracon polyester or Kevlar polymer for the strands, Aluminum, polycarbonate, Debrin acetal resin, Nylon or High density polypropylene for the end fittings and reinforced polyurethane with a plastic square pattern grid (PAYNTER, 1988b).

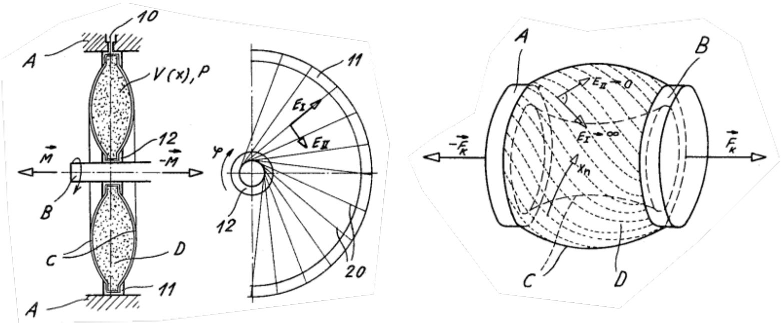
According to the inventor, this muscle is capable of working with higher pressures than the Yarlott muscle and is capable of higher working cycles before failure than the Yarlott and the Morin muscle, while producing a longer and more forceful contraction (PAYNTER, 1988b).

Muscle contraction is of about 37% of the end fittings' diameter (PAYNTER, 1988b).

3.1.13 Kleinwachter muscles

The Kleinwachter muscles were invented by Hans Kleinwachter and Jens Geerk and patented in 1972 (KLEINWACHTER; GEERK, 1972). The muscles are illustrated in fig. 15.

Figure 15: Kleinwachter torsional (left) and linear (right) muscles' squematics.



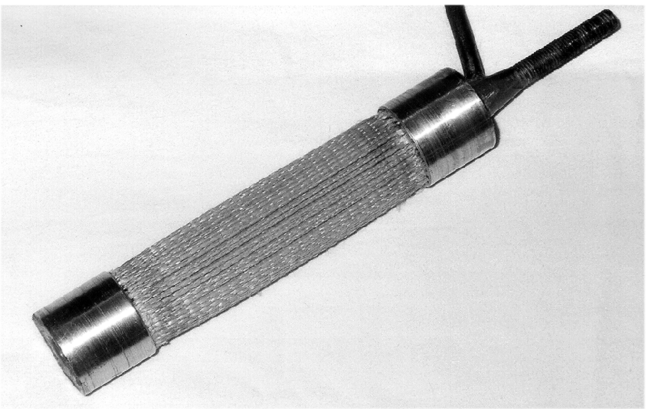
Source: Adapted from (KLEINWACHTER; GEERK, 1972).

Both muscles are composed of 2 rigid parts connected by a stretchable diaphragm. This diaphragm have anisotropic elastic properties. This is achieved by embedding flexible but substantially inextensible fibers or strands in one direction. The chamber is in toroidal shape, attached to the 2 rigid parts by internal and external rings. When fluid pressure is applied inside the chamber, the diaphragm expands radially, bringing both rigid parts toward each other, in the case of the right muscle in fig. 15. In the case of the left muscle in fig. 15, strands are disposed obliquely connecting the rigid parts, where one is an external fixed ring and the other is a movable shaft that turns when fluid pressure is applied in the chamber.

3.1.14 Pleated muscle

The pleated pneumatic muscle was developed in the Vrije University of Brussel, as a result of Frank Daerden doctoral work finished in 1999 (DAERDEN, 1999). The project, however, continued and two more versions of the pleated pneumatic muscle were designed. The pleated muscle prototype is illustrated in fig. 16.

Figure 16: Pleated pneumatic muscle prototype.



Source: Taken from (DAERDEN, 1999).

The pleated muscle is composed of two end fittings and a membrane. The distinctive part of this muscle is the pleated membrane (DAERDEN, 1999). The membrane material is folded in a circular pattern, forming the pleats. When fluid pressure is applied, the membrane material unfolds and the chamber increases in diameter, while contracting axially because of the membrane's high tensile stiffness in the axial direction (DAERDEN et al., 2001; DAERDEN; LEFEBER, 2002).

To equally separate the folds and to keep them in place during operation, teeth are machined in the end fittings. The end fittings also have a conical shape, where the membrane is attached with a resin (DAERDEN, 1999).

Materials used to make the prototype shown in fig. 16 are aluminum alloy for the end fittings and para-aramid, or the commercially known name Kevlar, fabric sealed with poplypropylene film and pressure sensitive rubber-resin adhesive (DAERDEN, 1999).

Maximum contraction of the prototype shown in fig. 16 was 41.5% (DAERDEN; LEFEBER, 2002).

3.2 PAMs models

Because PAMs are very nonlinear devices, physical modeling is complex. There are many researches of PAMs modeling. One of the most used approach to model the muscle's force was developed by Richard H. Gaylord in his 1958 patent. The approach used by Gaylord is the conservation of energy (WANG; WERELEY; PILLSBURY, 2015).

The perfect muscle would be the one that converts all of the pneumatic energy into mechanical work. The muscle's mechanical work W_m can be expressed by

$$W_m = Fl \tag{3.1}$$

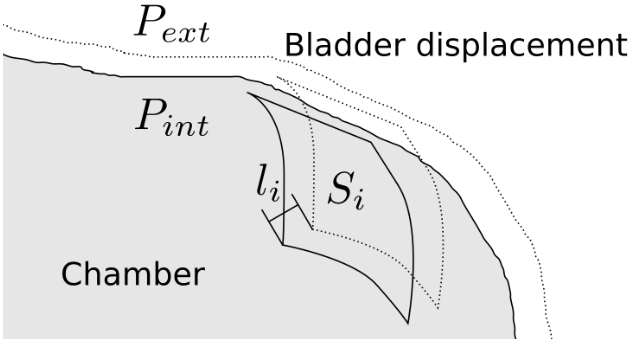
where F is the muscle's force and l the load displacement. Equation 3.1 can also be expressed in infinitesimal form by

$$\mathrm{d}W_m = F\mathrm{d}l \tag{3.2}$$

When pressurized air enters the chamber, it pushes the membrane and braid radially, realizing work as illustrated in fig. 3.4.

The work done by the fluid can be expressed as

Figure 17: Fluid work when there is a pressure difference between the chamber and the exterior.



Source: Own authorship.

$$\mathrm{d}W_f = (P_{int} - P_{ext})\,\mathrm{d}S_i\,\mathrm{d}l_i \tag{3.3}$$

Considering that this is an ideal muscle and that energy is conserved, the principle of virtual work is applied resulting in

$$\mathrm{d}W_m = \mathrm{d}W_f$$

$$F\mathrm{d}\,l = (P_{int} - P_{ext})\,\mathrm{d}S_i\,\mathrm{d}l_i \tag{3.4}$$

Because $\mathrm{d}S_i\,\mathrm{d}l_i$ is the differential volume $\mathrm{d}V$, 3.4 can be rewritten to

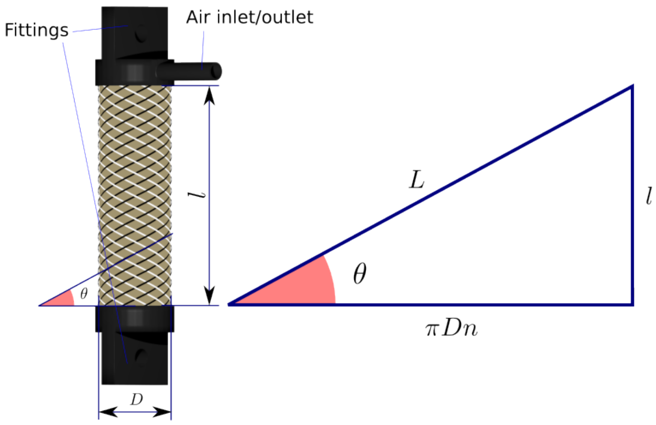
$$F\mathrm{d}l = (P_{int} - P_{ext})\,\mathrm{d}V$$

$$F = (P_{int} - P_{ext})\,\frac{\mathrm{d}V}{\mathrm{d}l} \tag{3.5}$$

From 3.5, basic models are derived solving $\frac{\mathrm{d}V}{\mathrm{d}l}$ for specific muscle volume models, such as for cylindrical bladders (GAYLORD, 1958; CHOU; HANNAFORD, 1996; SCHULTE, 1961; KLUTE; HANNAFORD, 2000).

For a McKibben muscle, the volume can be approximated to a cylinder. This cylinder, however, changes shape according to the double helical braid and, thus, it have to be considered in the model. Fig. 18 illustrates the parameters considered in a typical McKibben muscle volume model.

Figure 18: McKibben muscle volume parameters for a cylindrical membrane.



Source: Own authorship.

The braid is composed of several strings at an angle θ with respect to the fitting, which runs around the membrane n times connecting one fitting to the other. The membrane length l varies according to the angle θ of the braid, or according to the diameter of the braid D. If one braid string is unrolled in a straight line with length L at the angle θ, other muscle parameters can be related with the triangle formed by the braid string, the membrane and the imaginary perimeter line unrolled with length πDn, as illustrated in fig. 18.

Using the parameters illustrated in fig. 18, the membrane volume can be expressed by

$$V = \frac{\pi D^2 l}{4} \tag{3.6}$$

The diameter D can be related to the membrane length, using the right triangle relation

$$L^2 = (\pi D n)^2 + l^2$$

$$D = \sqrt{\frac{L^2 - l^2}{\pi^2 n^2}} \tag{3.7}$$

Substituting 3.7 in 3.6, gives

$$V = \frac{\pi l}{4} \left(\frac{L^2 - l^2}{\pi^2 n^2} \right)$$

$$V = \frac{l\,(L^2 - l^2)}{4\pi n^2} \tag{3.8}$$

Calculating $\frac{\mathrm{d}V}{\mathrm{d}l}$ of 3.8, results

$$\frac{\mathrm{d}V}{\mathrm{d}l} = \frac{L^2 - 3l^2}{4\pi n^2} \tag{3.9}$$

Equation 3.5 can be used to calculate the McKibben muscle force F_m, substituting the $\frac{\mathrm{d}V}{\mathrm{d}l}$ term with 3.9, which results in

$$F_m = (P_{int} - P_{ext}) \frac{(L^2 - 3l^2)}{4\pi n^2} \tag{3.10}$$

This model, however, doesn't predict the force of McKibben muscles accurately for several reasons (KLUTE; HANNAFORD, 2000; WANG; WERELEY; PILLSBURY, 2015). One reason is because the muscle volume is oversimplified. Instead of a cylinder, the membrane shape is rounded in the boundaries because of the fixation at the fittings, which have a constant diameter.

To compensate for the rounded extremities of the muscle, Tondu e Lopez (2000) propose a constant correction factor. This correction factor is determined experimentally (TONDU; LOPEZ, 2000).

Another reason is because McKibben muscles have an elastic bladder that is ignored in this model. In this respect, Klute e Hannaford (2000) use a Mooney-Rivlin model to account for the elastic energy stored in the muscle's membrane. According to Klute e Hannaford (2000), the force prediction is improved but the discrepancy between model and experimental data is still significant.

Friction is also not considered in the model, which causes hysteresis. Tondu e Lopez (2000) believe that the hysteresis is due to the thread-on-thread friction, and that ideally no friction occurs between the membrane and the braid. Wang, Wereley e Pillsbury (2015) believe that the interaction between the braid and the bladder is a source of the observed hysteresis and should be considered.

Many other factors influences the muscle's performance. The membrane material hysteresis, elastic behavior of the threads and temperature and thermal properties of the materials used are other relevant topics to better models.

Because of this rather complex physics involved in the muscle model, many researchers prefer to model by identification techniques. Unfortunately, these models can't be used to design muscles and to study its properties before it is built, and are particular to the very muscle used, at the same conditions that were tested (WANG; WERELEY; PILLSBURY, 2015).

3.3 Chapter conclusions

One of the main components of a mechatronic system are the actuators. When designing systems for electromagnetic risk applications, there are two alternatives: keep the actuator far away and provide power transmission to the critical system or to use compatible actuators.

Pneumatic artificial muscles can be built as compatible actuators for such systems and can provide sufficient force and displacement. These actuators can be designed with many geometric variations and materials, which changes its performance characteristic. It was reviewed several different designs of PAMs and what are the main changes of performance of each design, which can be used to tune the actuator for specific application requirements.

Mathematical models are available for these devices but the physically based models aren't, in general, accurate enough to be used on precise position control systems. Because of that, identification models are used instead. Physically based models, however, can be used to design actuators, considering the maximum force and contraction.

PAMs are also highly nonlinear actuators, which makes them difficult to model and control accurately.

4 CONTROL AND OPTIMIZATION METHODS

Controlling accurately the pneumatic artificial muscle is a challenging task. Many researchers around the world have tested many different techniques for the position control of PAMs. This chapter reviews some of these attempts and discusses some of the drawbacks of each technique.

To address the control problem, machine learning techniques can be used for the position control of PAMs. Because of that, this chapter also considers optimization algorithms and machine learning methods to be used in conjunction with control techniques to tune the controller.

4.1 PAMs control techniques

Because of the nonlinearities of PAMs, accurate position control is challenging (CSIKÓS; SZ; SÁROSI, 2017; CHIANG; CHEN, 2017; XIE et al., 2018). Many classical and nonlinear control techniques were proposed for PAMs position control (CSIKÓS; SZ; SÁROSI, 2017; CHIANG; CHEN, 2017).

Classical proportional-integral-derivative (PID) controllers were proposed for position control of PAMs. The PID controller is simple, versatile and widely used in many academic and industrial systems. It is based on three parameters by which the sum is equal to the control signal. Each parameter adds its contribution to the overall control output. In a typical position control using PID, the proportional parameter gives the coarse position control in the medium term, the integrative parameter is used for fine positioning in the long term and the derivative tries to react very fast to system disturbance. Two main problems of PID controllers are that they are supposed to work with linear systems and finding proper gains (tuning). There are some cases where nonlinear systems can be controlled using simple PID controllers, by choosing adequate gains by trial and error or by some approximation method such as the Ziegler-Nichols method. Because of that, many researchers try to use PID controllers for controlling pneumatic muscles. Tan et al. (2016) use a PID controller for angle position control using an antagonist setup of PAMs. The Ziegler-Nichols method is used to approximate the gains of the controller and fine tuning of the gains are determined by trial and error (TAN et al., 2016).

Another approach is to identify the muscle characteristics, fitting a model, linearizing the model about a working point and calculating the gains for that particular point and the nearby region. This approach is used by Repperger, Johnson e Philips (1999) and is called a gain scheduling approach.

Position control can be obtained using fuzzy logic, known as fuzzy controllers. The system is measured with sensors to provide the inputs. Then membership functions are distributed in the operation range for the input values of the system. According to each range of the membership functions, control rules are established. With the membership functions and the control rules, the control surface can be calculated, which is used to produce the output of the controller. This approach is used by Csikós, Sz e Sárosi (2017) for controlling the position of a ball in beam, tilted by pneumatic muscles.

With sliding mode controllers, the control system can have multiple control laws for specific cases. The controller switches to the control laws, also called modes, to drive the system to the desired position. One mode tends to drive the system to a state where another mode is used. Instead of infinitely switching to both modes, the sliding mode controller determines a combination of the modes, also called induced mode, by which the system state remains between the intersection of the modes states, and "slides" along that intersection where the desired state is. This approach is used by many researchers to control pneumatic muscles. Shen (2010), for example, use the sliding mode controller for a one degree of freedom (DoF) horizontal mass position control. Cao, Xie e Das (2018) used multiple-input-multiple-output (MIMO) sliding mode controller in a gait exoskeleton powered by pneumatic muscles. The controller parameters had to be tuned experimentally and model parameters also had to be determined experimentally (CAO; XIE; DAS, 2018). Lilly e Yang (2005) used sliding mode control for controlling a 1-DoF arm with antagonist setup of pneumatic muscles. Model parameters for the simulation are needed, trial and error parameters were chosen to yield good performance and a strategy for avoiding chattering problems was studied (LILLY; YANG, 2005).

Control algorithms that depend on the model are difficult to implement or lack on performance because of the complex nonlinear nature of PAMs. They often need parameters to be determined experimentally and trial and error tuning. Linearizing the model can also degrade the control performance, introducing linearization errors (XIE et al., 2018).

Some control algorithms are robust to disturbance, model uncertainties and unmodeled dynamics, such as adaptive robust control, fuzzy controllers, sliding mode and artificial neural networks (SENTHILKUMAR, 2010; XIE et al., 2018). But there is always a tradeoff between robust controllers and performance, which have to be taken care according to the application's requirements.

Sliding mode controllers are knwon to have chattering problems and high-frequency control laws that generated high control effort and can make the system unstable (XIE et al., 2018; CARBONELL; JIANG; REPPERGER, 2001). Artificial neural networks may need many samples and exhaustive training before good control performance is achieved (TANG et al., 2016). There are no systematic approaches to design fuzzy controllers, it is time consuming and based on trial and error (CSIKÓS; SZ; SÁROSI, 2017).

When dealing with very complex problems, machine learning techniques are often applied. They can be used to avoid human intervention by trial and error and achieve better performance by learning the control law and/or the control parameters.

4.2 Machine Learning and optimization algorithms

Machine learning is the study of methods that optimizes the performance metric of a task over experience or training by a machine. Machine learning is often used to solve complex problems, e.g. image processing and speech recognition.

One interesting application of machine learning techniques is for controlling nonlinear systems. The difficulty of correct modeling the nonlinear system, solving and designing a controller for it is hidden in the model and the algorithm learning process. The problem, however, is how to train the algorithm. Typically, large datasets are used to train the model. The algorithm uses the dataset to choose the parameters that maximizes some performance metric.

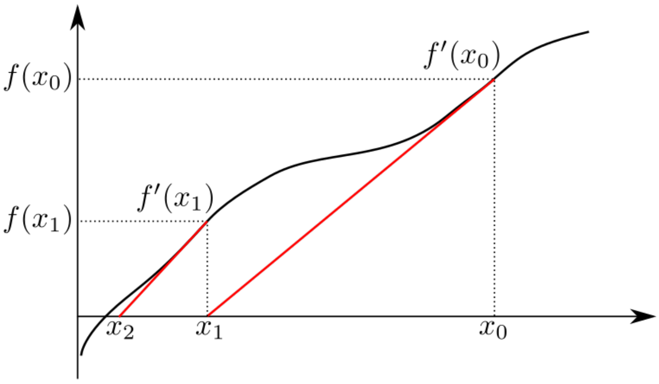
There are several machine learning methods and optimization algorithms. The following sections will address some of them.

4.2.1 Newton-Raphson

The Newton's method, also called Newton-Raphson method, is a root finding algorithm widely used to solve nonlinear systems or in optimization problems (YPMA, 1995). The Newton-Raphson method is also the basis for many other frequently used methods for solving nonlinear equations and optimization algorithms (YPMA, 1995).

The Newton-Raphson method works by using the derivative of the function to approximate an initial guess to the root of the function, as illustrated in fig. 4.3.

Figure 19: Newton-Raphson method for finding an approximate root of the function by iteration.



Source: Own authorship.

From fig. 4.3 it is possible to derive a formula for finding x_1 starting with x_0 using the derivative

$$f'(x_0) = \frac{f(x_0)}{x_0 - x_1} \tag{4.1}$$

Rearranging the terms in 4.1 as to isolate x_1, gives

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} \tag{4.2}$$

Generalizing 4.2 results in the Newton-Raphson iterative method

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{4.3}$$

The approximate solution can be as close as desired, according to the stop criterion. One simple example for the stop criterion would be to stipulate how close to zero is $f(x_{n+1})$. This however, can lead to infinite loops. In fact, the Newton-Raphson method is good when the initial guess x_0 is reasonably close to the root. If this is not the case, convergence can be compromised.

One drawback of this iterative method is that $f'(x_n)$ must be known, which implies that an algebraic equation exists for the system. Of course one can simply substitute the $f'(x_n)$ term to a finite difference approximation

$$x_{n+1} = x_n - \frac{h_n f(x_n)}{f(x_n + h_n) - f(x_n)} \tag{4.4}$$

or using the last iteration result, also known as the secant method (YPMA, 1995)

$$x_{n+1} = x_n - \frac{(x_n - x_{n-1})f(x_n)}{f(x_n) - f(x_{n-1})} \tag{4.5}$$

For optimization problems, one can use the derivative as the function $f(x) = g'(x)$, resulting in

$$x_{n+1} = x_n - \frac{g'(x_n)}{g''(x_n)} \tag{4.6}$$

This is because we can find the critical points of the function by finding where $g'(x) = 0$, which is where the minimum of the function, or the maximum, is.

To account for multi-variable functions, 4.6 can be written as

$$\vec{x}_{n+1} = \vec{x}_n - H^{-1}\nabla_{\vec{x}}G(\vec{x}) \tag{4.7}$$

where $\nabla_{\vec{x}}G(\vec{x})$ is the gradient vector defined as

$$\nabla_{\vec{x}}G(\vec{x}) = \begin{bmatrix} \frac{\partial G(\vec{x})}{\partial x_{n_1}} \\ \frac{\partial G(\vec{x})}{\partial x_{n_2}} \\ \frac{\partial G(\vec{x})}{\partial x_{n_3}} \\ \vdots \\ \frac{\partial G(\vec{x})}{\partial x_{n_i}} \end{bmatrix} \tag{4.8}$$

and H is the Hessian matrix defined as

$$H = \begin{bmatrix} \frac{\partial^2 G(\vec{x})}{\partial x_{n_1}^2} & \frac{\partial^2 G(\vec{x})}{\partial x_{n_1}\partial x_{n_2}} & \cdots & \frac{\partial^2 G(\vec{x})}{\partial x_{n_1}\partial x_{n_j}} \\ \frac{\partial^2 G(\vec{x})}{\partial x_{n_2}\partial x_{n_1}} & \frac{\partial^2 G(\vec{x})}{\partial x_{n_2}^2} & \cdots & \frac{\partial^2 G(\vec{x})}{\partial x_{n_2}\partial x_{n_j}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 G(\vec{x})}{\partial x_{n_i}\partial x_{n_1}} & \frac{\partial^2 G(\vec{x})}{\partial x_{n_i}\partial x_{n_2}} & \cdots & \frac{\partial^2 G(\vec{x})}{\partial x_{n_j}^2} \end{bmatrix} \tag{4.9}$$

If, however, $x \in R^n$ where $n \gg 10000$, then inverting the Hessian matrix can be very expensive or even infeasible. Because of this, other methods were developed, such as truncated-Newton or quasi-Newton methods, which can be used when the Hessian isn't available or is expensive to compute.

Another alternative to Newton based optimization algorithms is the Gradient descent algorithm.

4.2.2 Gradient Descent

The gradient descent algorithm is used to find the local minima of a multivariable function. The algorithm works by going towards the steepest descent of the function F at small steps. The steepest descent is calculated by the negative gradient of F. The update rule, can be expressed as

$$\vec{x}_{n+1} = \vec{x}_n - \gamma \nabla_{\vec{x}} F(\vec{x}_n) \tag{4.10}$$

where γ is the step size factor, also called the learning rate in machine learning.

Of course, the gradient can be substituted by the finite difference when the gradient isn't available.

The gradient descent algorithm is slower to converge than the Newton-Raphson method, but its easier to implement. If the function is convex, then the Newton-Raphson converges faster and can easily be implemented. However, if the function is not convex, there are some implementation problems that needs to be solved, such as saddle point convergence, solution oscilations, root jumping and infinite loops.

Gradient descent is a first order method and, therefore, it is easier to compute at each step than the Newton-Raphson method, which is second order.

Because of this factors, gradient descent is widely used in optimization problems for general functions and in machine learning.

In machine learning, when large datasets are used or many training samples are needed, gradient descent can be expensive. In these cases, a variation of this algorithm is used, called stochastic gradient descent.

4.2.3 Stochastic Gradient Descent

The stochastic gradient descent algorithm is widely used in machine learning (LE et al., 2011). This is because datasets or training samples are getting bigger and bigger over time, which makes gradient descent expensive. Stochastic gradient descent is a simplification of gradient descent when computing the gradient of a function.

The difficulty of computing $\nabla_{\vec{x}}F(\vec{x})$ is that the function F is often unknown. When this is the case, training samples are used to estimate $F(\vec{x})$. After estimating $F(\vec{x})$, the gradient can finally be computed.

When using gradient descent, the function F is estimated by an hypothesis $h(\vec{x})$, such as a linear hypothesis of the form

$$h_{\vec{\theta}(\vec{x})=\sum_{i=0}^{S}\theta_i x_i}(4.11)$$

where θ_i's are the terms to be found by the algorithm to approximate $h_{\vec{\theta}(\vec{x})}$ to $F(\vec{x})$, and S is the number of samples.

The way gradient descent approximates the hypothesis to the function, is by minimizing a cost function that relates how the predictions of the hypothesis differ from $F(\vec{x})$. This cost function can be the squared error function of the form

$$J(\vec{\theta}) = \frac{1}{2S}\sum_{i=0}^{S}\left(h_{\vec{\theta}(\vec{x}_i)-y(\vec{x}_i)}\right)^2 \tag{4.12}$$

where $y(\vec{x}_i)$ is the value of the sample corresponding to the input $\vec{x}_i$.

To calculate the gradient of $J(\vec{\theta})$, its is necessary to compute the summation for each θ_i. And this can be computationally expensive if S is large. The stochastic gradient descent addresses this problem by randomly choosing a sample, and computing the gradient for that sample. This simplified gradient is then used to slightly update the hypothesis feature vector $\vec{\theta}$. This procedure is repeated, at least, for each sample.

4.2.4 Dynamic Mode Decomposition

Dynamic Mode Decomposition, or DMD, was first introduced by Schmid (2010) in the fluid dynamics area. It is a method for learning a linear model of a system from data.

In the same way that a differential equation is useful to predict the future state of a system, the DMD tries to find a matrix A that maps the previous state of a system to a future state. In a mathematical sense, the future state $\vec{x}_{t+\Delta t}$ is assumed to be a linear combination of $\vec{x}_t$ given by

$$\vec{x}_{t+\Delta t} = A\vec{x}_t \tag{4.13}$$

The state $\vec{x}_t$ can be measured data or numerically solved by finite element method, for example. However, only two steps $\vec{x}_t$ and $\vec{x}_{t+1}$ probably won't capture the dominant features of the system (SCHMID, 2010). Moreover, the matrix that maps one state $\vec{x}_i$ to $\vec{x}_{i+1}$ is probably different than the matrix that maps $\vec{x}_j$ to $\vec{x}_{j+1}$.

Therefore, multiple samples of the system state are necessary to find an optimal matrix A. If the additional samples doesn't include additional features of the system, then the rank of the matrix A won't increase (SCHMID, 2010).

To include multiple samples, let a matrix X be the state matrix of the form

$$X = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ \vec{x}_{t1} & \vec{x}_{t2} & \dots & \vec{x}_{tn} \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \tag{4.14}$$

and X' be the future state matrix of the form

$$X' = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ \vec{x}_{t2} & \vec{x}_{t3} & \dots & \vec{x}_{tn+1} \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \tag{4.15}$$

and the goal is to find A such that

$$X' = AX \tag{4.16}$$

It is possible to solve this equation by multiplying both sides by the pseudoinverse of X

$$A = X'X^{\dagger} \tag{4.17}$$

This method gives the "best-fit" linear operator A, according to the pseudoinverse used (e.g. least-squares fit for the Moore-Penrose pseudoinverse).

DMD is commonly used in conjunction with the Singular Value Decomposition (SVD), to extract lower order linear models of the system, because A is often huge and very computationally expensive, or even impractical to work with. Additionally, since A is huge, it's best to avoid computing it at all. This can be done by applying SVD in X and truncated the decomposed matrices with the rows that most captures the system dynamics.

Applying the SVD to X results in

$$X = U\Sigma V^* \tag{4.18}$$

and substituting (4.18) in (4.16) gives

$$X' = AU\Sigma V^* \tag{4.19}$$

Multiplying both sides of (4.19) by U^* gives

$$U^*X' = U^*AU\Sigma V^* \tag{4.20}$$

and exposes a similar matrix to A ($\tilde{A}$), given by

$$\tilde{A} = U^*AU \tag{4.21}$$

which is a much simpler matrix to work with.

And substituing (4.21) in (4.20), gives

$$U^*X' = \tilde{A}\Sigma V^* \tag{4.22}$$

which can be rearranged to isolate $\tilde{A}$

$$\tilde{A} = U^*X'V\Sigma^{-1} \tag{4.23}$$

And because $\tilde{A}$ is similar to A, they have the same eigenvalues. And it's also possible to reproject the eigenvectors of $\tilde{A}$ (W) to match the eigenvectors of A (Φ), also called DMD modes, given by

$$\Phi = X'V\Sigma^{-1}W \tag{4.24}$$

Therefore, it's possible to compute the eigenvectors and eigenvalues of A without having to actually compute A, by working with a truncated similar matrix $\tilde{A}$. With the eigenvalues and eigenvectors, the prediction of the future state of the system can be calculated with

$$\hat{X}(k\Delta t) = \Phi \Lambda^t b_0 \tag{4.25}$$

In summary, DMD can extract a reduced order linear model of a nonlinear system which can be used to design a control law. For a highly nonlinear system, it's not clear if the operator A captures the dominant features and the operator might need to be updated constantly to be representative of the current state.

In contrast of learning a linear model, which is often the goal, other techniques are used to learn a nonlinear model of a system, which can be more representative of the system. To learn a nonlinear model of a system, other approaches are used, e.g. Artificial Neural Networks (ANN).

4.2.5 Artificial Neural networks

Artificial Neural Networks is a very successful algorithm for solving difficult problems. It is inspired by the way the brain learns, mimicking the behavior of neurons.

The neuron have, basically, dendrites and an axon. The dendrites are the inputs of the neuron and the axon is the output. Each neuron can interact with another neuron by connecting the axon to the other neuron's dendrite, forming a synapse. The neuron sends a signal down the axon for specific combinations of input signals in the dendrites, or don't send any signal otherwise.

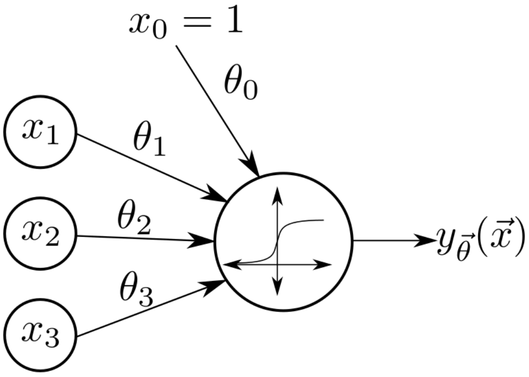
One way to model this behavior mathematically, and which is used in many artificial neural networks, is by a linear combinations of the dendrite signals. If the linear combinations of the inputs are higher than a threshold value, then a signal is emitted down the axon to other neurons, and if it is bellow the threshold nothing is emitted. Turns out that this strategy doesn't work well, and instead of sending or not sending a signal (discrete representation), the artificial neuron sends a signal with a certain intensity (continuous representation), corresponding to the sigmoid function defined as

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{4.26}$$

The functions that represents the signal emitted in the axon are called activation functions, and the sigmoid is just one commonly used one, among many others.

Fig. 20 represents the basic artificial neuron model.

Figure 20: Artificial neuron diagrammatic representation.



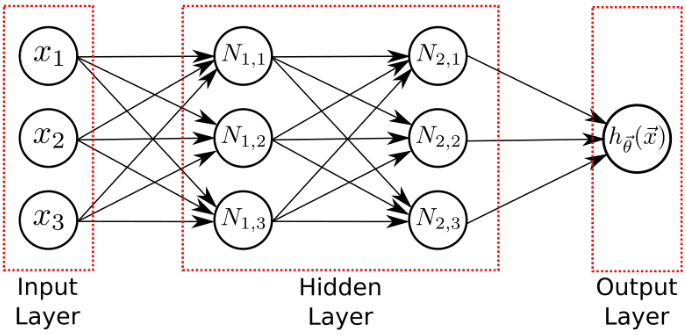
Source: Own authorship.

In fig. 20, x_1, x_2 and x_3 and sometimes x_0, known as the bias unit with the constant value of 1, form the input vector $\vec{x}$. The terms θ_0, θ_1, θ_2 and θ_3 form the weight vector $\vec{\theta}$. Each input value is multiplied by its respective weight value, and the summation of all the input values times their respective weights forms the linear combination of the input values to be evaluated by the activation function. Considering fig. 20 as an example, the output signal down the axon would be

$$y_{\vec{\theta}}(\vec{x}) = \frac{1}{1+e^{-\vec{\theta}\vec{x}}(4.27)}$$

This, however, represents only one neuron, which isn't much useful. A combination of neurons, or a network of neurons, is what makes this approach very useful. There are a lot of ways to interconnect the artificial neurons, each of them is called an architecture. The architectures are composed of three main parts: the input layer, the hidden layers and the output layer. One commonly used architecture is illustrated in fig. 21.

Figure 21: A neural network architecture.



Source: Own authorship.

In fig. 21, only one output is shown, but a neural network can have multiple outputs, e.g. classification neural networks. Typically, each input and neuron is connected with every other input or neuron in the previous layer, like in fig. 21.

The output is calculated by propagating the neuron's signal layer after layer, until it reaches the output layer. This process is known as forward propagation. However, the weights of the neural network have to be tuned for the neural network to work properly. This is the learning, or training, process of the neural network (NN). This is achieved by the backpropagation algorithm. It's is a similar process for finding the θ_i's described in section 4.2.3. There is a cost function that relates how bad the NN, or the NN hypothesis $h_{\vec{\theta}}(\vec{x})$, is performing and an optimization algorithm is used to adjust the θ_i's, e.g. Newton-Raphson method, gradient descent and stochastic gradient descent.

One interesting characteristic of neural networks is that they learn the features by themselves with the backpropagation algorithm from the data. They are also capable of modeling very complex nonlinear functions (DONGARE et al., 2012). Their performance is basically dependent on the architecture used, the input parameters and, specifically, the dataset used. Typically, large datasets are used to train NNs and they can be very computationally expensive to train.

4.2.6 Simulated Annealing

Simulated Annealing (SA) is an optimization algorithm based on the thermodynamic process of metals called Annealing (GENDREAU; POTVIN, 2005). It was first proposed in 1983 by Kirkpatrick, Gelatt e Vecchi (1983).

The annealing process arranges the atoms in an efficient way, creating an arrangement with low energy state. This occurs when the material is slowly and gradually cooled. This process is mimicked to find good solutions (lower state arrangements). It is a probabilistic meta-heuristic with the capacity of "escaping" from a local minima.

In the annealing process, atoms can easily migrate to form different states, generally lower energy states, as the material gradually cools down. In some cases, however, a higher energy state can be formed. Similarly, the SA algorithm allows a bad solution to be accepted with a certain probability. This allows the algorithm to escape from a local minima, as mentioned before.

The algorithm has two main stages: the exploration and the exploitation. The exploration occurs at higher temperatures, where the algorithm explores the domain space, performing a global search. The exploitation stage occurs at lower temperatures, where the algorithm performs a local search for solution refinement. The probability of accepting bad solutions decreases with temperature drop. When no more good solutions are found, the process is considered frozen and the algorithm finishes.

Many solution rejections can lower the exploration area and trap the algorithm to a local minima. To address this problem, a crystallization heuristic can be used. This serves as an instant feedback for the SA algorithm, to refine search when there are rejected solutions and explore more when there are accepted solutions. The SA with crystallization heuristic algorithm is shown in fig. 22.

Figure 22: Simulated Annealing with crystallization heuristic algorithm.

```
$\vec{x} \leftarrow$<random initial solution>
$T_0 \leftarrow$<initial temperature>
while <global condition not satisfied> do
    $T_i \leftarrow T_i * \alpha$
    while <local condition not satisfied> do
        $k \leftarrow$<select parameter to modify>
        $x^* \leftarrow x + \frac{1}{c_k} \sum_1^{c_k} \text{random}(-\frac{1}{2}, \frac{1}{2}) \cdot \Delta r_i \cdot e_k$
        $\Delta E = F(x^*) - F(x)$
        if $\Delta E < 0$ then
            $x \leftarrow x^*$
            $c_k \leftarrow 1$                ▷ Positive Feedback
        else
            if random$(0, 1) < e^{-\Delta E / k_B T_i}$ then
                $x \leftarrow x^*$
                $c_k \leftarrow 1$            ▷ Positive Feedback
            else
                $c_k \leftarrow c_k + 1$      ▷ Negative Feedback
            end if
        end if
    end while
end while
```

Source: Own authorship.

The algorithm works by first selecting an initial solution $\vec{x}$ and an initial temperature T_0. Then it selects a parameter of the current solution to modify, e.g. the third element of the solution vector. The parameter is perturbed with a random distribution. The algorithm then evaluates the objective function, which is the function to be optimized, with the current solution and the perturbed solution. The difference between the solutions ΔE corresponds to the increase or decrease in energy of the system. Therefore, if the perturbed solution is better than the current solution, then $\Delta E < 0$, which corresponds in a decrease in the system's energy, and the perturbed solution is accepted, overwriting the previous solution. If, however, there is an increase in energy, or $\Delta E > 0$, there is a probability that this solution can be accepted. This probability is computed according to the Boltzmann probability factor (KIRKPATRICK; GELATT; VECCHI, 1983) given by

$$P(\Delta E) = e^{-\frac{\Delta E}{k_B T}} \tag{4.28}$$

The local and global conditions can be an established maximum iteration parameter and the existence of an accepted solution, respectively.

4.3 Chapter conclusions

Several attempts for muscle precise position control are described in the literature. Many of them are complex to implement and require highly specialized personnel to design the control system. This makes the application of PAMs practically limited to academic works. Additionally, many of the proposed methods require previous studies and experiments before the actual controller design.

An easy method for designing controllers for PAMs is still missing. One possible solution is to merge the advantages of classical control with machine learning techniques, e.g. using a simple and widely used controller and an optimization method for tuning the controller to the desired performance requirement of the specific application.

In this respect, some machine learning and optimization algorithms were reviewed to address the problem of tuning the controller to the specific application, through minimizing a user defined cost function.

Additionally, previous works on PAMs control didn't focus on electromagnetic risk applications. Thus, the control problem have to be studied in more detail using sensors and equipment for these specific applications.

5 SENSING TECHNIQUES FOR ELECTROMAGNETIC RISK APPLICATIONS

Sensors are essential for precise and robust position control, specially when nonlinear actuators are used. On the previous chapter, some control techniques were reviewed. However, these techniques generally rely on fast, accurate, low noise, electronic sensors. Depending on the application, common electric sensors can be used if properly insulated. Signals can also be sent through wireless networks. But there are cases that every electronic device is prohibited, e.g. MRI and specially fMRI. To address the control problem for electromagnetic risk environments, alternative sensing techniques must be evaluated. Therefore, this chapter will describe sensors that can be applied in general to every electromagnetic risk application.

Also, in this chapter, it is described the development of an image based sensing technique to be used in the validation of the proposed controller.

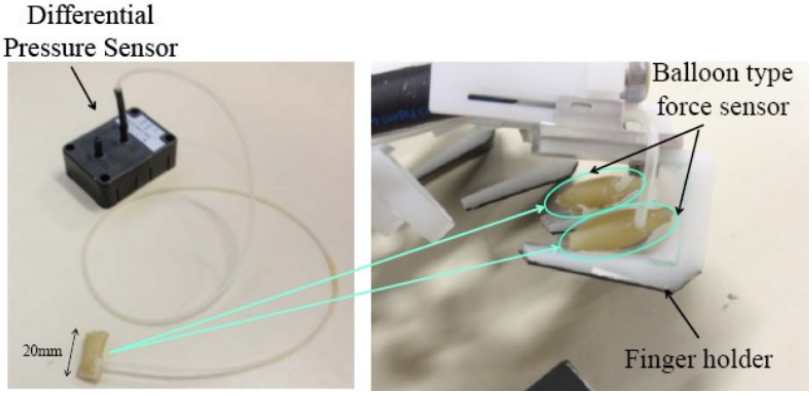
5.1 Fluidic power transmission

5.1.1 Balloon sensor

The balloon sensor was used in a grip amplification glove to measure the force at the finger tips (TADANO et al., 2010). The used sensor is shown in fig. 23.

This device is composed of an elastomeric chamber which is connect through

Figure 23: Balloon sensor to measure the force applied at the finger tips.



Source: From (TADANO et al., 2010).

a pipe to a differential pressure sensor. When force is applied in the chamber, it deforms and decreases the inner volume, raising the pressure in the tube. The pressure is then related to the force applied in the chamber.

This is not an accurate force sensor, but can be used as an end switch or for coarse force measurement. One drawback of this design is that the pressure difference will be smaller for longer pipes.

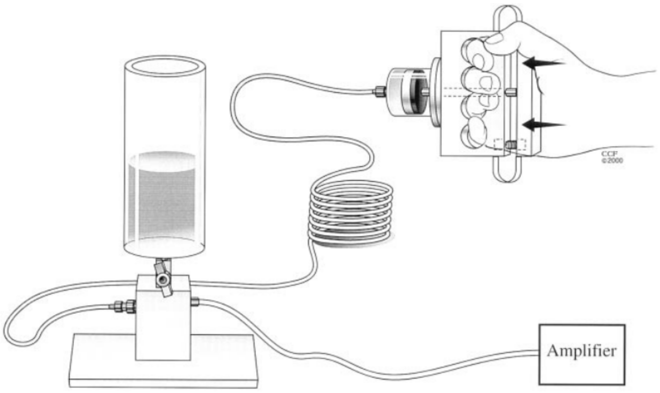
5.1.2 Piston sensor

This type of device uses a piston to transmit fluid pressure to a pressure sensor. One example is the one used by Liu et al. (2000), which is illustrated in fig. 24.

The basic principle is to transmit the force applied at the piston in the form of fluid pressure though the pipe to a pressure sensor. In the device used by Liu et al. (2000), there is a reservoir above the sensor to providing fluid to the transmission system. There is a screw by which is possible to flow the fluid out. The signal of the pressure sensor is then amplified.

Hydraulic systems can cause contamination problems if not properly sealed (COMBER; BARTH, 2011). Air bubbles can also cause calibration and measure-

Figure 24: Hand-grip hydraulic force sensor.



Source: From (LIU et al., 2000).

ment errors.

5.2 Optical fiber signal transmission

5.2.1 Optical fiber encoders

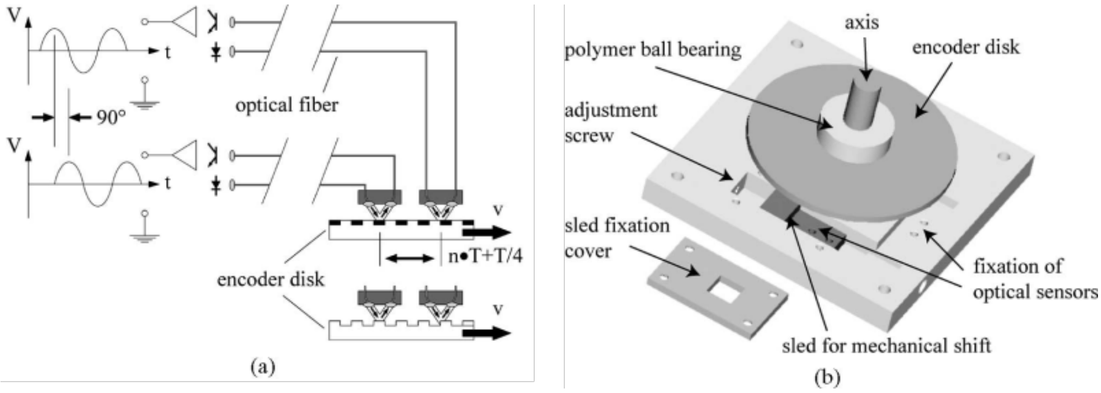
Encoders are devices that measures position in a discrete form through marks. These marks can be reflexive or opaque and disposed in a linear strip or in a wheel, for measuring linear displacement and angles respectively.

In general, encoders have a light source and a light intensity sensor, typically an infrared light emitting diode (LED) and a photo-transistor. To avoid using electric devices near the risk area, one solution is to isolate the LED and photo-transistor and transmit the light through optical fibers.

This device was used by Krieger et al. (2007) to measure angle and translation of a MRI-compatible needle placement device. Optical fibers were opposing each other and, in between the optical fibers, a linear and a wheel with opaque marks was used to transmit or block the light between fibers.

Another example is the angular encoder used by Gassert et al. (2006), illustrated in fig. 25.

Figure 25: Angular reflexive optical fiber encoders.



Source: From (GASSERT et al., 2006).

The device illustrated in fig. 25 is based on the reflection of the light on the markers of the encoder disk. A reflexive material or a difference in height can be used to differentiate the marks in the encoder disk.

The same principle can be used to measure force and torque by attaching the markers in an elastic material. The displacement of the markers can be related to the force or torque applied.

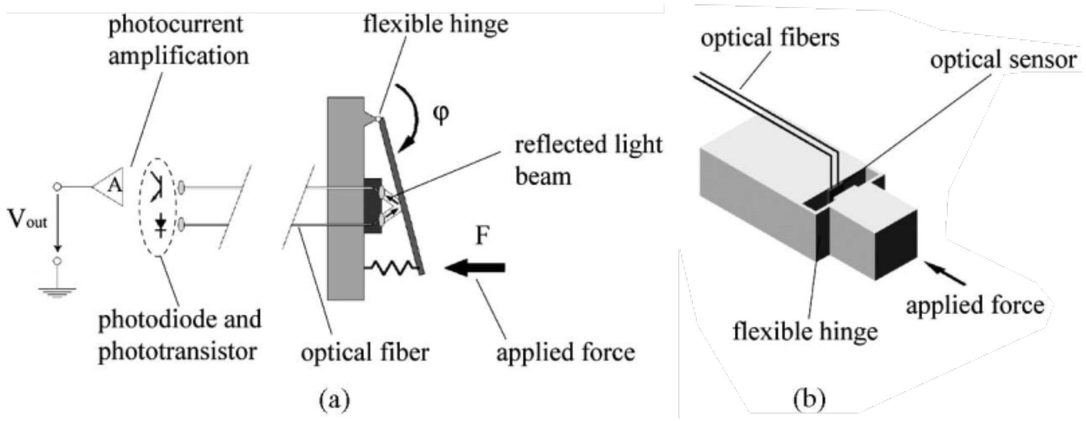
5.2.2 Reflexive target force sensor

Gassert et al. (2006) describes a force sensor that uses a reflexive target that alters the light intensity transmitted through the sensor fiber. The proposed devices are illustrated in fig. 26.

In fig. 26(a) the reflexive target angle is used to alter the light intensity transmitted to the sensor fiber, while in fig. 26(b) the distance between the optical fibers and the reflexive target is used as a way of changing the light intensity.

The applied force can then be calibrated with the light intensity received in the sensor. However, the light transmitted through the optical fiber can suffer

Figure 26: Light reflection based force sensors.



Source: From (GASSERT et al., 2006).

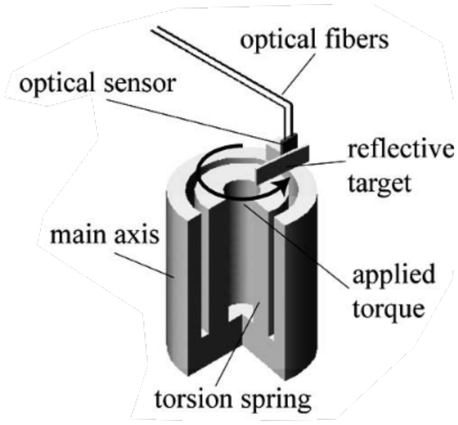
intensity losses during movement of bending of the fibers (GASSERT et al., 2006). Because of this effect, the placement of the sensor in movable parts have to be avoided or measurement errors can occur.

Chapuis et al. (2004) also use a reflexive element in a similar manner for a force and torque sensor.

5.2.3 Reflexive target torque sensor

By moving a reflexive target, torque can also be measured using optical fibers. Gassert et al. (2006) proposed the model illustrated in fig. 27.

Figure 27: Reflexive target based optical fiber torque sensor.



Source: From (GASSERT et al., 2006).

Two optical fibers are used to transmit light to the device and back to the sensor. The fibers are fixed in a wall and the reflexive target is fixed at a movable wall, attached to a torsion spring. When torque is applied to the torsion spring, the target moves away from the reflexive target, which dims the light at the sensor fiber.

As the measurement is based on light intensity, bending or moving the fiber can cause errors in the measurement.

5.2.4 Fiber Bragg Grating

Fiber Bragg Grating (FBG) is a periodic difference of the refractive index along the optical fiber length (HILL; MELTZ, 1997). These refractive index patterns along the fiber can be used as sensors. Because of the properties of the FBG, multiple sensors can be embedded in a single fiber.

FGBs can be made by the interference pattern of ultra-violet (UV) light. The optical fiber core is photosensitive to UV light, opposed to the fiber cladding which is transparent to UV light (HILL; MELTZ, 1997). So, when UV light radiates the optical fiber, the refractive index of the fiber's core is changed permanently (HILL; MELTZ, 1997). These periodic difference of refractive index caused by the UV light causes part of the incident light on the fiber to be reflected. The strongest light wavelength that is reflected by the FBG is called the Bragg wavelength λ_B given by

$$\lambda_B = 2n_{eff}\Lambda \tag{5.1}$$

where n_{eff} is the modal index and Λ the grating period (HILL; MELTZ, 1997).

Therefore, anything that changes the modal index or the grating period will cause a shift in the wavelength of the reflected light.

Interrogators are used to analyse the sensor's characteristics. When there are multiple sensors in the fiber, two methods are commonly used: time division multiplexing (TDM) and wavelength division multiplexing (WDM) (ZHOU et al., 2003).

With TDM, sensors are built with the same Bragg wavelength λ_B. Each sensor has low reflectivity so that other sensors can be illuminated. When the fiber is submitted to stimuli, such as strain, the time in which each FBG reflects light differs (ZHOU et al., 2003). This time shift in the FBG is measured and the magnitude of the stimuli is calculated.

The most common method, however, is the WDM (ZHOU et al., 2003). In this case, sensors have unique wavelength and the wavelength shift is measured.

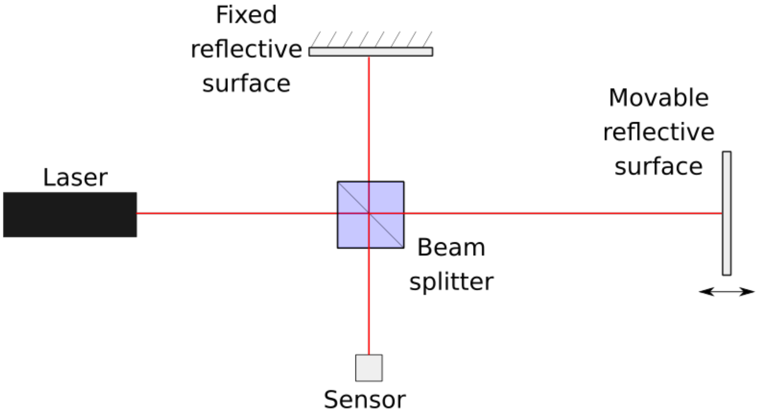
5.2.5 Optical fiber interferometers

Interferometers are devices that can accurately measure physical properties using the interference of coherent light. Typically, a laser is used as light source and the optical path difference causes an interference fringe, which is measured by detectors.

One example is the Michelson interferometer (fig. 28). This interferometer splits the light source in two different paths. One path is the reference distance and the other is sensing path. If both the reference path and the sensing path is exactly equal, then there is constructive interference and a bright spot will be detected in the sensor. However, if there is a difference between the sensing and reference paths, a destructive interference can occur, resulting in a dark spot.

The optical path length Γ can be calculated by

Figure 28: Michelson interferometer schematics.



Source: Own authorship.

$$\Gamma = dn \tag{5.2}$$

where d is the geometric distance that the light travels and n is the refractive index of the medium in which the light is traveling.

The difference in the optical path length $\Delta\Gamma$ for a Michelson interferometer is

$$\Delta\Gamma = \Gamma_s - \Gamma_{ref} = d_s n_s - d_{ref} n_{ref} \tag{5.3}$$

where Γ_s is the sensing arm optical path length and Γ_{ref} is the reference arm optical path length.

If the difference in optical path length corresponds to half wavelength of the light source λ, then there is a phase shift $\Delta\phi$ of π radians, resulting in total destructive interference. In general, there will be total destructive interference when

$$\Delta\Gamma_d = (2m+1)\frac{\lambda}{2} \quad \text{for } m \in Z \tag{5.4}$$

or

$$\Delta\phi_d = (2m+1)\pi \quad \text{for } m \in Z \tag{5.5}$$

and total constructive interference when

$$\Delta\Gamma_c = m\lambda \quad \text{for } m \in Z \tag{5.6}$$

or

$$\Delta\phi_c = 2m\pi \quad \text{for } m \in Z \tag{5.7}$$

When the movable reflexive surface moves, the sensor will detect bright and dark spots corresponding to 5.4-5.7. By counting how many bright and dark spots, it is possible to calculate the displacement of the reflexive surface. There is, however, a problem to determine the direction of the displacement.

There are some techniques to disambiguate the direction. One strategy proposed by Wang et al. (2001) is to work between a bright and a dark spot, in a nearly linear region. In this case, the displacement is determined by the intensity of the light.

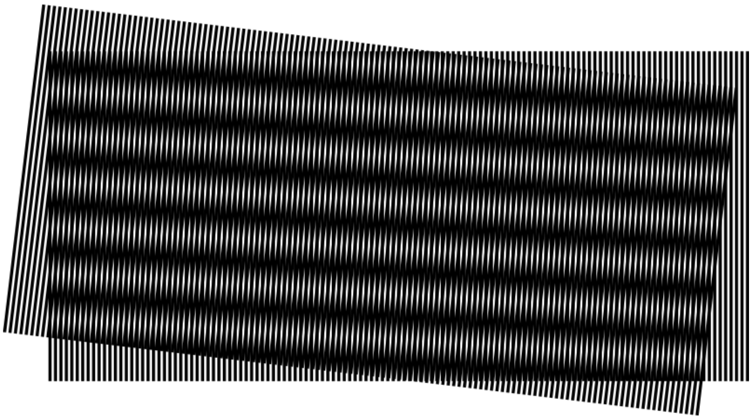
5.3 Image based measurements

Another way of measuring is by images or video cameras. Several effects can be exploited for measuring physical properties of the system. The following sections will address some of the techniques for measuring different physical properties.

5.3.1 Moiré fringes

Moiré fringes can be used to measure many different physical properties, such as displacement, strain and object's geometry. Moiré fringes occur when superposing two or more patterns with slightly different positions or shapes, as is illustrated in fig. 29.

Figure 29: Moiré fringes formed by two linear patterns superposed in angle.



Source: Own authorship.

One advantage of using moiré fringes is the ability to measure fine displacements without using very high-resolution cameras. They can be used for monitoring the position of the manipulator's arms, applied forces, torques and pressure.

These physical properties are measured indirectly, using a flexible material or displacing the patterns to change the moiré fringes.

5.3.2 Motion capture

Motion capture is commonly used by animators and movie productions. Two main techniques are used: with targets and without targets. Motion capture without targets is challenging and currently is a research topic. Using targets is easier for processing the image and generally produces more robust results.

These targets are used as reference points and can be used to calculate many physical properties. Current technologies support keeping track of the position of many reference points at the same time, even in three dimensions (3D).

Motion capture typically uses a model and assumptions to work properly. This is because there may be points that can be occluded in certain poses or, when using multiple cameras, to correspond points between images.

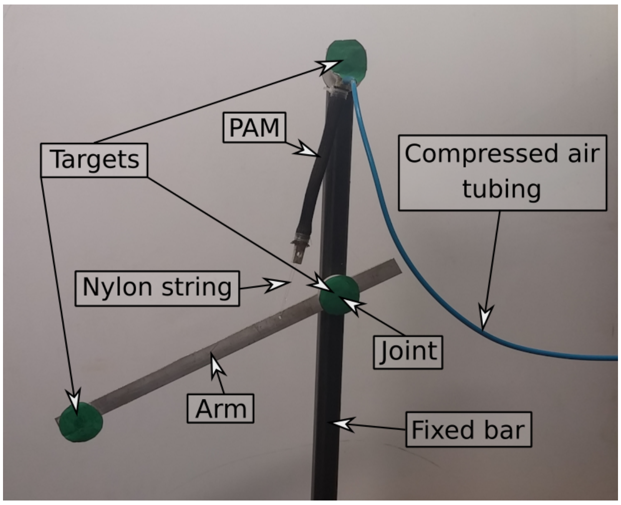
One advantage is that the sensing system can be placed far from the actuation device, and thus, provide insulation means.

This technique is actually simple to implement and can be used to test the control system.

5.3.2.1 Implementation

Motion capture is implemented on a test bench, for measuring the angle of a 1-DoF arm powered by a pneumatic artificial muscle. The test bench is illustrated in fig. 30.

Figure 30: Test bench for a 1-DoF arm actuated by a McKibben type PAM.

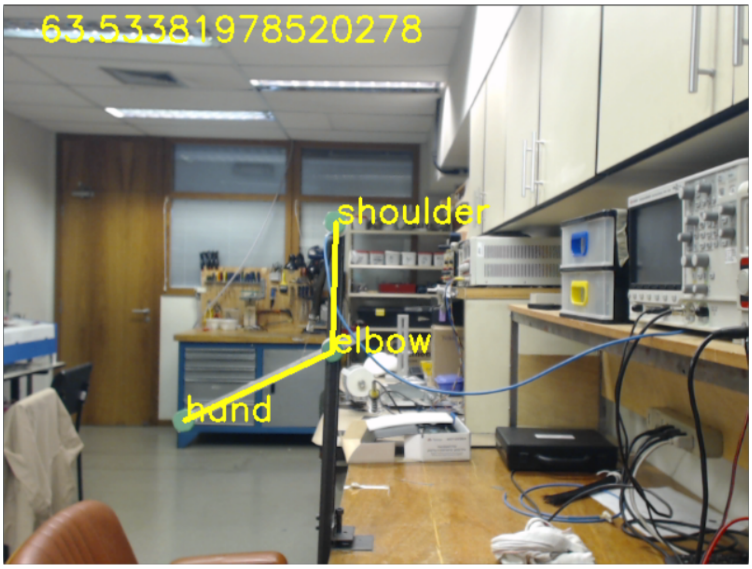


Source: Own authorship.

It is used a Logitech c270 720p 30fps webcam for capturing the image, and the whole image processing is written in Python3.6 using the OpenCV library.

The green targets are used as the reference positions for the shoulder, elbow and hand. A heuristic is used to identify which of the points corresponds to the shoulder, elbow and hand. After the identification, the angle of the arm can be calculated. The position tracking system identification is illustrated in fig. 31.

Figure 31: Position tracking system arm identification and angle determination.



Source: Own authorship.

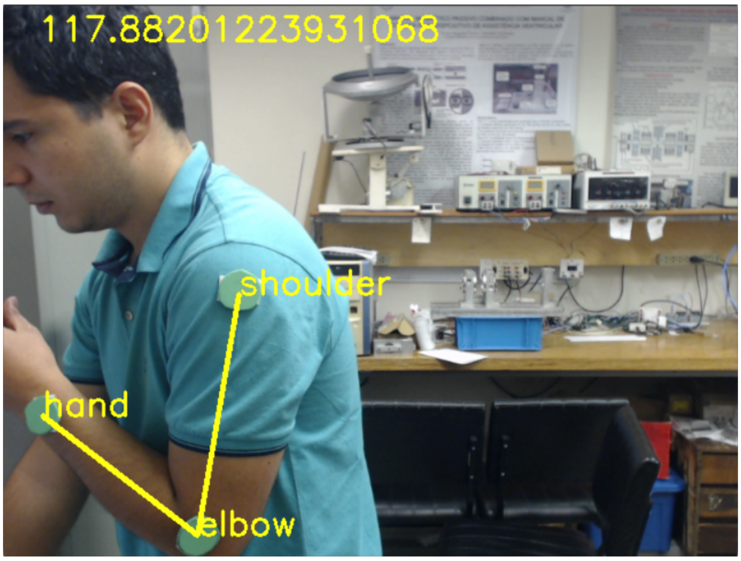
This position tracking system is flexible and can be used for industrial robots or for rehabilitation purposes for controlling an exoskeleton. An example of human usage is illustrated in fig. 32.

The average time for identifying the targets and calculating the angle is approximately 13 milliseconds. The accuracy of the system depends on the camera resolution, camera positioning and distance to the object. Higher resolution images can be more computationally expensive and, thus, will take longer to process.

5.4 Chapter conclusions

Compatible sensors for electromagnetic risk applications were reviewed. Depending on the application and the physical constraints of the environment, dif-

Figure 32: Arm detection in humans using the position tracking system.



Source: Own authorship.

ferent sensing techniques can be used. The accuracy is also a factor for choosing one technique over another.

Fortunately, there are sensing methods which provide high accuracy and fast response, such as FBGs. But they are generally costly to implement. An easy and more cost effective alternative are the image based methods, which can measure many physical properties, provided the targets are visible at a safe distance.

An image based measuring technique is implemented to test the proposed control system.

6 CONTROLLER TUNNING WITH SIMULATED ANNEALING

In this chapter, the hypothesis that Simulated Annealing is a viable solution to tune a controller for the muscle is tested. For that, a model of a test system is developed based on a real bench. The model from (SCAFF, 2015) is used to simulate the positioning system. Because the resulting model is nonlinear, a numeric method is described to simulate the system and test its performance.

6.1 Muscle model

The muscle model used is based on equation 3.10. This is a reasonable force model to be used in a first attempt for the simulations and control. It's not a very accurate force model, as discussed in 3.2, but it's a good first approximation to the muscle general behavior.

Additionally, the braid is also considered in the model. When the muscle is stretched, not only the internal chamber pressure resists the pulling force, but the braid also, and thus it have to be considered. This is the case when the braid fibers are touching each other, at the saturation angle. This saturation angle occurs because of the finite width of the braid fibers. If the muscle is stretched beyond this saturation angle, braid fiber elastic deformation starts to be significant.

For the sake of simplicity, it is considered that the muscle force beyond the rest length l_{rest} is

$$F_{m_{stretch}} = k_{braid}(l - l_{rest}) + (P_{int} - P_{ext})\frac{(L^2 - 3l^2)}{4\pi n^2} \tag{6.1}$$

where k_{braid} is the braid's axial elastic constant.

The internal pressure P_{int} is calculated using the ideal gas law

$$P_{int} = \frac{m\Re T}{V} \tag{6.2}$$

where m is the mass of the gas inside the muscle's chamber, $\Re$ the universal gas constant for the gas, T the gas temperature and V the chamber's internal volume.

Considering that the muscle volume is a perfect cylinder, 3.8 can be substituted in 6.2, resulting in

$$P_{int} = \frac{4\pi m\Re T n^2}{l\,(L^2 - l^2)} \tag{6.3}$$

The gas mass m inside the chamber can be calculated by integrating the valve's mass flow rate $\dot{m}$, given by

$$m = m_0 + \int_0^\tau \dot{m}\,\mathrm{d}t \tag{6.4}$$

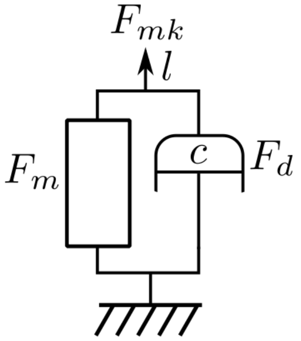
where τ is the total execution time of the system and m_0 the initial gas mass inside the muscle's chamber. If the muscle pressure at the beginning is the atmospheric pressure, then m_0 can be calculated by

$$m_0 = \frac{P_{atm} l_0 (L^2 - l_0^2)}{4\pi\Re T n^2} \tag{6.5}$$

where l_0 is the muscle's initial length.

For the dynamic model of the muscle, a dashpot can be included in parallel with the muscle as in fig. 33.

Figure 33: Dynamic muscle model schematics.



Source: Own authorship.

The damping coefficient c is determined experimentally. The total muscle force for the McKibben based muscle F_{mk} is, then, the sum of the muscle static force F_m and the dashpot force F_d.

6.2 Valve model

Depending on the valve used, such as proportional valves, a model for the valve can be avoided. These valves controls the air flow according to an analog input signal. The controller can be designed to generate the input signal directly to these valves. This is a simpler way of controlling the muscles, but generally they make the system expensive (XIE et al., 2018).

Alternatively, fast switching On-Off valves can be used. Using pulse width modulation (PWM), it is possible to control the mass flow rate through the valve. The mass flow rate depends on many factors and some are intrinsic to the valve model. Because of that, an empirical model is adopted.

Because the mass flow rate seems to decay nearly exponentially with the pressure difference at the inlet and outlet, the following empirical model for the mass flow rate is used

$$\dot{m} = m_{max}\left(1 - e^{\left(\frac{P_{out}-P_{in}}{P_{in}-P_{atm}}\right)s+f}\right) \tag{6.6}$$

where m_{max}, s and f are parameters to be identified experimentally, P_{in} is the pressure at the inlet of the valve and P_{out} is the pressure at the outlet of the valve.

6.3 PWM model

Pulse width modulation is used to produce the signals to activate the fast switching valves. The PWM has two main parameters: modulation frequency and duty cycle. The modulation frequency is the frequency f_{mod} in which the signal is repeated and the duty cycle d_c corresponds to the fraction of time that the signal is on, given by

$$d_c = t_{on} f_{mod} \tag{6.7}$$

where t_{on} is the time in which the signal is on and the modulation frequency f_{mod} is given by

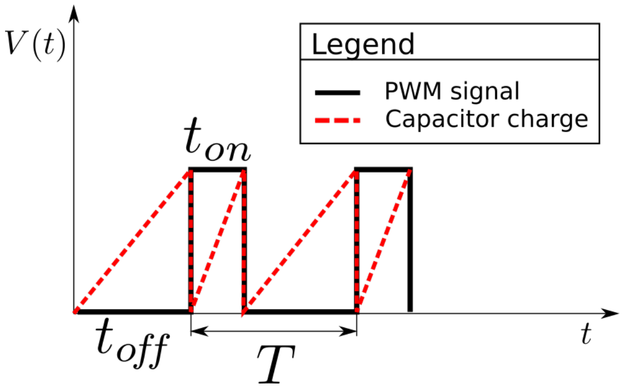
$$f_{mod} = \frac{1}{T} \tag{6.8}$$

where T is the period of the PWM signal.

The PWM can be modeled as a charging capacitor, where the rate $\dot{q}$ in which it charges is proportional to the duty cycle. When the charge in the capacitor is full, it changes the state of the output signal $V(t)$, as fig. 34 illustrates.

The t_{on} charging rate $\dot{q}_{on}$ is given by

Figure 34: PWM charging capacitor model.



Source: Own authorship.

$$\dot{q}_{on} = \frac{f_{mod}}{d_c} \tag{6.9}$$

and the t_{off} charging rate $\dot{q}_{off}$ is given by

$$\dot{q}_{off} = \frac{f_{mod}}{1 - d_c} \tag{6.10}$$

6.4 Muscle-mass-spring model

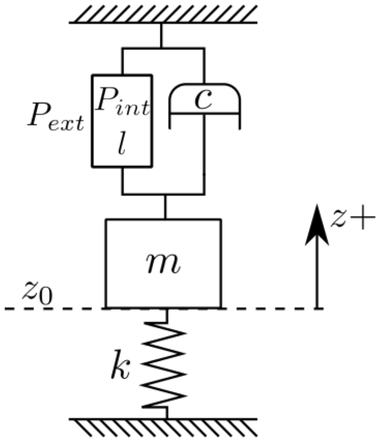
For the 1-DoF muscle-mass-spring system, the model of the muscle can be simply combined with the spring and mass element, as in fig. 35.

The spring elastic constant k is identified experimentally for an extension spring. The mass element of mass m is pulled by the muscle upwards, while the spring pulls the mass downwards. The height position z is controlled by opening and closing the valve, which changes the internal pressure P_{int} of the muscle.

The differential equation of this system is given by

$$-F_m - c\dot{z} - mg - kz = m\ddot{z} \tag{6.11}$$

Figure 35: Muscle-mass-spring system dynamic model schematics.



Source: Own authorship.

which is a nonlinear differential equation that needs to be solved numerically.

6.5 PID model

The PID controller, or three term controller, is based on the proportional, derivative and integrative terms. These terms are calculated with respect to the system error $e(t)$, which is measured by the control system.

The control signal $u_{PID}(t)$ is given by

$$u_{PID}(t) = K_p e(t) + K_i \int_0^t e(\tau) \mathrm{d}\tau + K_d \frac{d}{\mathrm{d}t} e(t) \tag{6.12}$$

where K_p, K_i and K_d are the proportional, integrative and derivative gains and τ is the integration variable corresponding to all times from 0 up to the current time t.

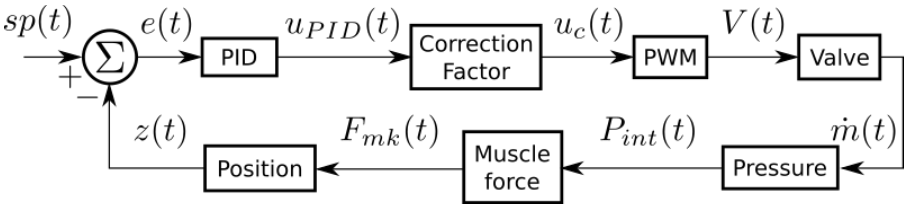
The system error is the difference between the current position $z(t)$ to the setpoint position $sp(t)$, given by

$$e(t) = sp(t) - z(t) \tag{6.13}$$

6.6 Control system model

The control system model of the muscle-mass-spring system is summarized by the block diagram in fig. 36.

Figure 36: Muscle-mass-spring control system block diagram.



Source: Own authorship.

It is of note that a correction factor is used after the PID controller because of the valve's response. This correction factor is given by

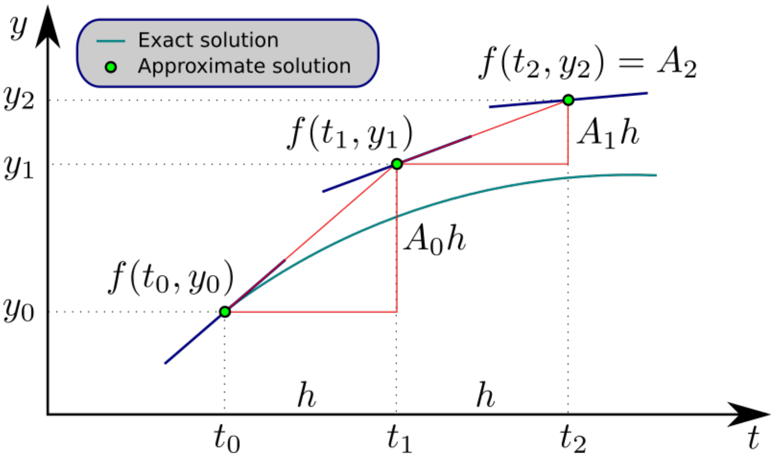
$$u_c(t) = e^{u_{PID}(t)} \tag{6.14}$$

6.7 Control system simulation

The control system described in 6.6 is solved numerically using the Euler's method, also known as Runge-Kutta 1 or RK-1. The Euler's method is a simple numerical method used to solve differential equations with initial values. These initial values are used to progress the solution gradually by steps, as illustrated in fig. 37.

If the initial conditions are known, such as the value $y(t_0) = y_0$ and the derivative of the function at that point $y'(t_0) = f(t_0, y_0) = A_0$, the next value of the solution can be approximated by linearly extrapolating the solution in the direction of the derivative $y_1 = y_0 + A_0h$. Using the approximated value y_1, the derivative $y'(t_1) = f(t_1, y_1) = A_1$ can be determined using the differential equation. The following values are determined similarly, by using the derivative

Figure 37: Euler's method for solving a differential equation with an initial value.



Source: Own authorship.

of the last point and the differential equation to calculate the next derivative.

The Euler's method and all of the control system models are implemented using the Cython programming language.

6.8 Controller tuning methods

Many controllers may need tuning or the selection of parameters. In the case of the PID controller, the parameters to be selected are the proportional, integrative and derivative gains. Because of the nonlinearity of the system, classical methods cannot be used directly to select the gains. Nonlinear control theory was applied by many researches before, as previously commented in chapter 4.1. In this chapter alternative methods are proposed and discussed.

The tuning can be based in two main ways: simulation based and experimentation methods, or no model methods.

6.8.1 Simulation based methods

When depending on the simulation, the model have to be accurate for the tuning to be effective. The model can be identified or physically based (first principles). With PAMs, identified models are more accurate than first principles models. However, experiments have to take place before the design of the controller and are accurate to the particular muscle, or control system, used. First principles based models enables the study and design of the controller before the system is even constructed, with the drawback that many factors may be ignored and its accuracy compromised.

Independent of the model used, the tuning method can be used in the same way. Simulation based methods will use the models and simulation method described in the previous sections for testing the efficacy of the proposed tuning method.

6.8.1.1 Simulated Annealing

The simulated annealing optimization algorithm can be used to tune the controller by properly designing a cost function, or objective function. This objective function represents the controller desired performance by penalizing undesired behavior. The algorithm will try to minimize this cost function and optimize the system behavior according to the penalization criterion of the objective function.

The SA algorithm was chosen because of the capacity of finding the global minimum value of very complex cost functions, even with many local minima. This might be the case for complex systems and should be a robust method for finding the global minima, instead of the gradient descent that would stop at the first local minima.

The SA algorithm was coded in Cython based on the pseudo code in fig. 22.

The global condition used was the number of accepted solutions greater than zero and the local condition was based on the max number of accepted solutions and iterations per temperature. Also, the Boltzmann constant k_B was not considered.

The objective function value $F(\vec{x})$ and the perturbed objective function value $F(\vec{x}^*)$ are computed using the designed objective function over the simulation result using the PID gains vector $\vec{x}$.

For all tests, the simulated annealing algorithm was executed in four simultaneous processes, using an Intel 64-bit Core^{TM} I3-2120 CPU @ 3.30GHz with 8GB of RAM on an Arch Linux, kernel version 4.16.11-1, compiled with Cython 0.25.2 and run with Python3.6.

The first step of this method is to design the objective function to be optimized, according to the application's requirements. As an example, the requirement can be the fastest time response without overshoot. For that requirement, two simple procedures for penalizing the system behavior can be used: direct penalization and indirect penalization.

The following methods will use the muscle-mass-system control system with simulation parameters given in table 1.

6.8.1.1.1 Direct penalization In the direct penalization, unwanted behaviors are measured and penalized. For example, in the case of fastest response time without overshoot, the raise time and overshoot can be measured and a penalization factor can be established for each behavior.

In a simplified way, one can stop the simulation if the position of the mass passes through the setpoint and set a high penalization for this behavior. In this way, the simulation time can be drastically reduced. However, a flat region in the objective function's hypersurface will be produced by this strategy, causing

Table 1: Muscle-mass-spring system simulation parameters.

Device	Parameter	Symbol	Value	Unit
Spring	Rest length	x_{rest}	0.10	m
Spring	Spring constant	k	300	Nm^{-1}
Spring	Initial length	x_0	0.10	m
Mass	Mass	m	0.5	kg
Mass	Gravitational acceleration	g	9.8	$\mathrm{ms}^{-}2$
Mass	Initial position	z_0	0.000	m
Muscle	Rest length	l_{rest}	0.127	m
Muscle	Initial length	l_0	0.127	m
Muscle	Fiber length	L	0.145	m
Muscle	Fiber turns	n	1.6	
Muscle	Atmospheric pressure	P_{atm}	101.325	kPa
Muscle	Initial pressure	P_{int_0}	101.325	kPa
Muscle	Braid elastic constant	k_b	3000000	Nm^{-1}
Gas	Temperature	T	293.15	K
Gas	Specific gas constant	$\Re$	287.058	$\mathrm{Jkg}^{-1}\mathrm{K}^{-1}$

Source: Own authorship.

difficulty for the SA algorithm to converge. Because of that, this actually can make the algorithm take longer to produce a good result instead of finishing faster.

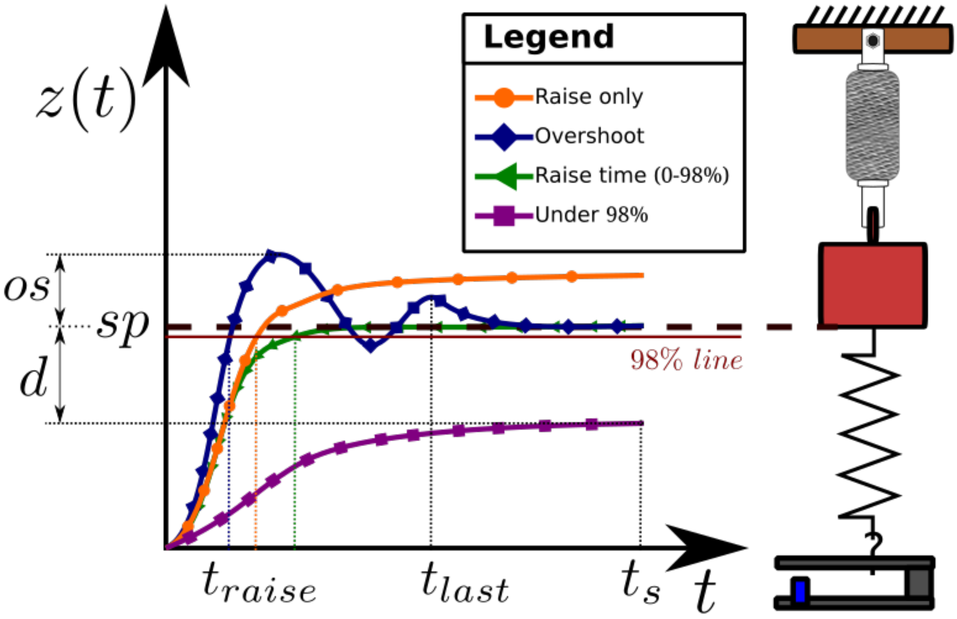
For the direct penalization criterion, it will be considered four penalization situations: under 98% of the setpoint sp, 0 → 98% raise time, overshoot and raise only. These four situations are illustrated in fig. 38.

The under 98% of the setpoint situation, corresponds to low proportional and integrative gains which can't drive the mass to the setpoint during the simulation time t_s. For that situation the penalization ρ_{under} can be based on the distance to the setpoint d, resulting in

$$\rho_{under} = t_s + d = t_s + sp - z(t_s) \tag{6.15}$$

The $0 \rightarrow 98\%$ of the setpoint penalization $\rho_{0\rightarrow 98\%}$ can be the raise time itself. The lower the raise time, the better is the response. Also, the worst case of the

Figure 38: Direct penalization criterion situations.



Source: Own authorship.

ρ_{under} is t_s, which is the worse raise time. The penalization for this situation is given by

$$\rho_{0\rightarrow 98\%} = t_{raise} \tag{6.16}$$

In the case of overshoot, the distance from the setpoint to the maximum of the overshoot can be used. Because overshoot is not desired, a constant factor can be used to increase its penalization. When there is overshoot, it is possible that the position oscillates, overshooting again. This is a worse solution and must be penalized. To account for this situation, the time in which the last overshoot occurred can be used as the penalization factor. The total penalization for the overshoot case ρ_{oShoot} is

$$\rho_{os} = t_s + 350os + t_{last} \tag{6.17}$$

Table 2: Simulated annealing with direct penalization results.

Test number	P	I	D	Score
1	778.1981	33.0148	16.2071	0.1501
2	817.6187	251.7287	18.8097	0.1492
3	808.1250	113.9967	17.6768	0.1512
4	770.8543	136.3697	16.4484	0.1505

Source: Own authorship.

For the raise only situation, the gains are too high and is the worst case of all. A higher penalization must be applied, based on how much it passed through the setpoint. The penalization of the raise only situation ρ_{ro} is given by

$$\rho_{ro} = t_s + 600\,(z(t_s) - sp) \tag{6.18}$$

The best solutions for the overshoot and raise only situations are equal to t_s, which is the best case of the under 98% of the setpoint and the worst raise time. This helps to avoid discontinuations on the objective function and avoids attributing different values to pratically identical cases (boundaries).

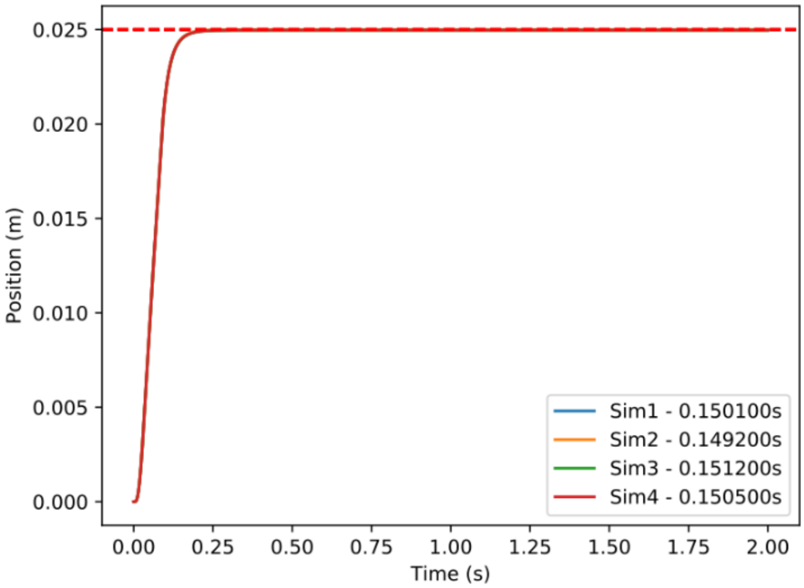
To conclude, the score is given by the value of the corresponding penalization. Therefore, after the test is done, the signal is analysed and categorized into the above penalization categories, and the score is calculated accordingly.

Using this penalization method, the SA algorithm was executed and the PID gains were optimized. The results are given in table 2.

The PID gains of the 4 executions were close to each other, except for the integrative gain where the variation was higher. The final scores are also close, which corresponds to the raise time. These four PID gains were used to simulate the system responses, which are illustrated in fig. 39.

The SA algorithm execution was manually interrupted with approximately 7 hours of execution because, at the end, the simulation was producing responses

Figure 39: System step responses for the PID gains obtained with the SA algorithm and the direct penalization method.



Source: Own authorship.

with equal scores, which makes the algorithm keep accepting these solutions and never terminate. This is caused by the fix step size adopted for simulating the system of 0.0001, producing plain regions. This can be avoided by another appending another global condition for the while loop, terminating the execution if the same response persists for a maximum number of permitted equal solutions.

The step responses produced with the obtained PID gains resulted in low raise time and without overshoots. Thus, this is a viable method for successfully finding good PID gains.

6.8.1.1.2 Indirect method As an example of indirect method, one can use the integral time-weighted absolute error (ITAE) criterion. This is a common criterion for tuning controllers. Another variation can be used, which is the integral of the absolute error (IAE) criterion, that will be used to tune the PID.

The IAE and ITAE criterion measures indirectly the raise time and overshoot, producing higher scores when the system have high raise time and over-

Table 3: Simulated annealing results with the IAE criterion.

Test number	P	I	D	Score (10^{-4})	Duration
1	737.3473	371.0319	16.3310	16.3775	12.29h
2	1690.7083	1.9676	14.4064	18.2092	7.44h
3	384.6905	264.2752	2.1301	16.1116	7.60h
4	2049.9084	147.7134	19.9065	18.0224	5.51h

Source: Own authorship.

shoot. These criteria produces more continuous scores and might be easier for the optimization algorithms to converge.

The IAE criterion cost function ρ_{IAE} is given by

$$\rho_{IAE} = \int_{0}^{t_s} |e(t)| \, \mathrm{d}t \tag{6.19}$$

and the ITAE criterion cost function ρ_{ITAE} is given by

$$\rho_{ITAE} = \int_{o}^{t_s} t \, |e(t)| \, \mathrm{d}t \tag{6.20}$$

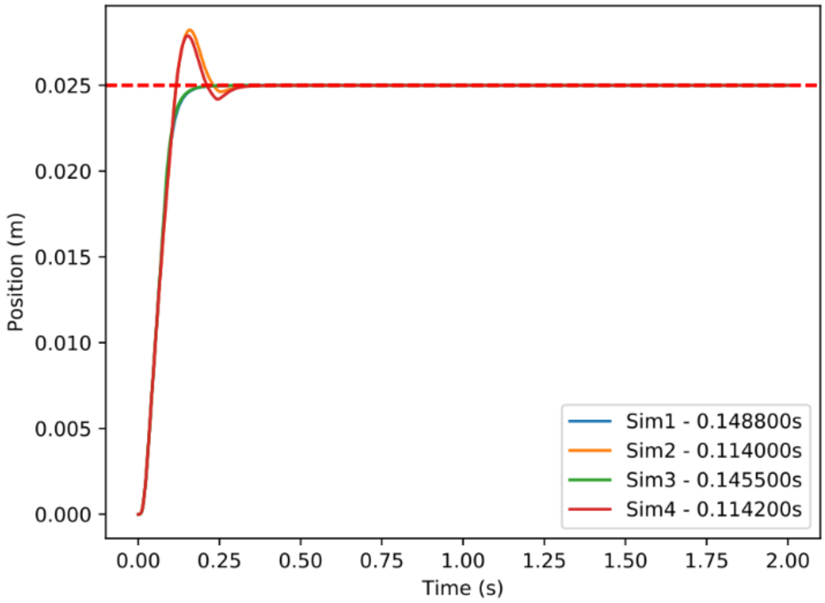
Using the IAE criterion, the simulated annealing was executed. Results are available in table 3.

There are 2 solutions that achieved lower scores and 2 solutions that obtained higher scores. Duration of execution of the SA algorithm also varies between the tests. Solutions have significantly different PID gains among each other, which means that there are multiple local minima in the hypersurface of the objective function.

The simulated step responses with the PID gains obtained by the IAE criterion are illustrated in fig. 40.

Two bad solutions, where there is overshoot, were selected as optimum. However, the ones that had no overshoot have better performance than all of the

Figure 40: System step responses for the PID gains obtained with the SA algorithm and the IAE criterion.



Source: Own authorship.

solutions found with the direct method. While the IAE method can produce better solutions, it has a more difficult convergence than the direct method.

This is an example of how the design of the objective function is critical for producing good solutions. And that some objective functions may produce challenging hypersurfaces for the algorithm to find an optimal solution.

The SA algorithm was able to find good solutions, even though it got stuck in local minima two times. Although the SA is a robust optimization method, its a stochastic optimization process that may need multiple executions before the optimal solution can be found.

Because the two penalization methods produced different optimal solutions, its not clear weather there are even better solutions for this system with this penalization method. Unfortunately, the only way to compare the performance of the SA algorithm with the true minima for this objective function is by brute force, which can be unfeasible for some applications.

6.8.1.2 Brute force

Because this is a relatively fast system to simulate using the models described in the previous section, the brute force method is actually feasible, at least for studying the objective function and the control system.

To compare the performance of the proposed SA algorithm against pure brute force, and to be able to analyse how the control system behaves for different PID gains, the IAE criterion was plotted for various different PID gains.

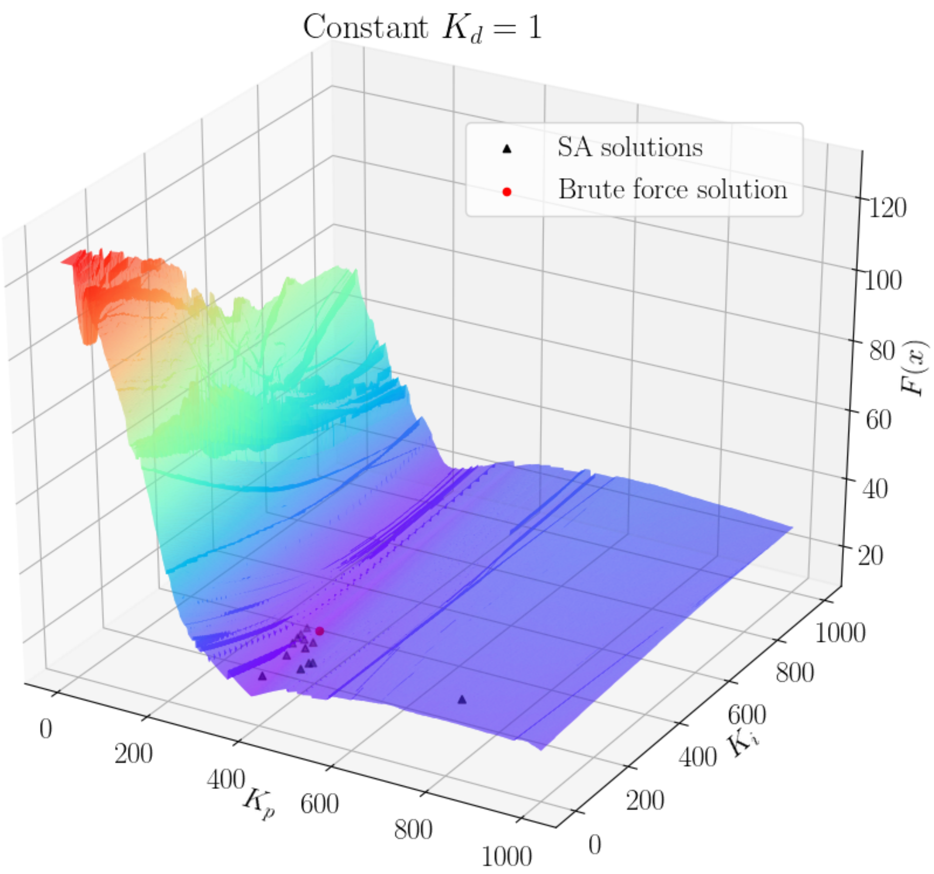
It was stipulated gains ranging from 0 to 1000 for the K_p and K_i gains and a constant value for the K_d of 1, for plotting purposes. The result is illustrated in fig. 41.

Each simulation takes about 1 second to finish. The variation gain step is 1, and so, one million simulations were executed to produce the plot in fig. 41, with a total of approximately 11.6 days of processing in single core, excluding storage operations and other operations like computing the IAE score.

For that, the Águia (Eagle) supercomputer from the Superintendence of Information Technology of the University of São Paulo was used. It is composed of 128 servers with 20 cores and 512GB of RAM each. The processor is an Intel Xeon CPU E7- 2870 @ 2.40GHz. It runs on a custom build GNU/Linux system. The Águia supercomputer is shared among many users, and so, only a fraction of this resource was used for producing the brute force result. This fraction depends on the current load, but about 300 processors were used and the whole brute force took about 1 hour to finish.

It should be noted that this brute force solution is only valid for $k_d = 1$, and the resolution of the answer is the step size used for the gains of 1, while the exploration area of the SA algorithm was $0 \to 5000$ for K_p, $0 \to 4000$ for K_i and $0 \to 20$ for K_d.

Figure 41: Brute force results for the IAE criterion.



Source: Own authorship.

For the exploration area used by the SA algorithm, the brute force would take more than 16 days to finish using the Águia supercomputer with 300 processors and with a solution resolution of 1. While SA would take about 8.2h each 4 trials in a common computer with 4 threads, like most of the computers today.

In terms of efficacy, fig. 41 also compares the brute force solution (red circle) with the SA annealing solutions (black triangles). SA solutions are close to the brute force solution in the majority of cases, which demonstrates that the SA is successfully finding good solutions to the control problem.

Because the system is nonlinear, a gain scheduling approach would be necessary to maintain a good performance among the whole actuator contraction range. This would result in a table with PID parameters for specific contraction ranges that can be switched online during the position control. This table can be as complete as necessary, attending contraction and extension situations for many ranges. This table can be made offline and, if the physical model is accurate enough, before the muscle is even built. Brute force solutions would be impractical for a common computer and, depending on the completeness of the table, even for a supercomputer like Águia.

The objective function and the simulation can be modified to account for sensor noise and external disturbances for more robust controller parameters or even to fit a curve for de P, I and D control parameters, instead of populating a table.

6.8.2 No model methods

No model methods are based on the real control system. This is a great advantage because the optimization will occur for that specific control system without any simplification or modeling limitations. However, it is challenging because of time constraints, risk of system failure and the objective function hypersurface can be even more complex for the optimization algorithms.

A test bench was constructed to test these algorithms, previously comented in section 5.3.2.1 and illustrated in fig. 30.

The control system is composed of two FESTO fast acting solenoid valves (MHJ10-S-2.5-QS-4-LF), a BeagleBone Black Rev. C single board computer as the controller, a Logitech c270 720p 30fps webcam for the position tracking, a desktop computer for image processing and a driver for commanding the valves.

The PID algorithm is implemented in Python in the BeagleBone Black. All of the image processing and position calculation is done in the desktop computer and send via ethernet to the BeagleBone Black.

The controller runs at 30Hz, because the bottleneck of this system is the camera frame rate.

6.8.2.1 Simulated Annealing

The SA took about 8.2h to find the solution for the simulation base method, where each simulation took about 1 second to finish, for a 5 seconds simulation time. Using the SA algorithm in the real bench would take about 7 seconds for each test, including 5 seconds experiment and 2 seconds for the reset time. If the hypersurface convergence difficulty is comparable to the simulation objective function, then the expected execution time would be 57.4h.

To test weather the SA can be used in a real system, the test window was reduced to 2 seconds, with a total of 4 seconds for each evaluation.

The convergence for the real system is not as good as with the simulation, and the SA was trapped in a totally random, nearly 100% acceptance ratio regime during all execution, terminated in a bad solution and, eventually, in a good solution.

One hypothesis for this bad performance is that the system had a lot of noise in the objective function. There is fluctuation in the objective function value for the same PID parameters, that makes the ΔE calculation and acceptance decision compromised.

One possible solution is to stipulate a dead zone δ in which the system cannot conclude if ΔE is positive or negative, and more evaluations are executed to statistically determine ΔE.

However, this can result in even more evaluations and make this optimization method impractical.

6.8.2.2 Brute force

Even though SA can be impractical because of the convergence rate and possibly find a bad solution after that, brute force can be an option for some systems.

For example, using a range of 0 → 5 for K_p, 0 → 5 for K_i and a constant K_d, varying the parameters with steps of 0.1 for K_p and 1 for K_i, a brute force solution for a 4 seconds evaluation would take about 20.4 minutes. Further refinement can be made at the minimum region.

This approach was used to produce the fig. 42, using the IAE criterion to optimize the parameters for positioning the arm from $83^o \rightarrow 105^o$.

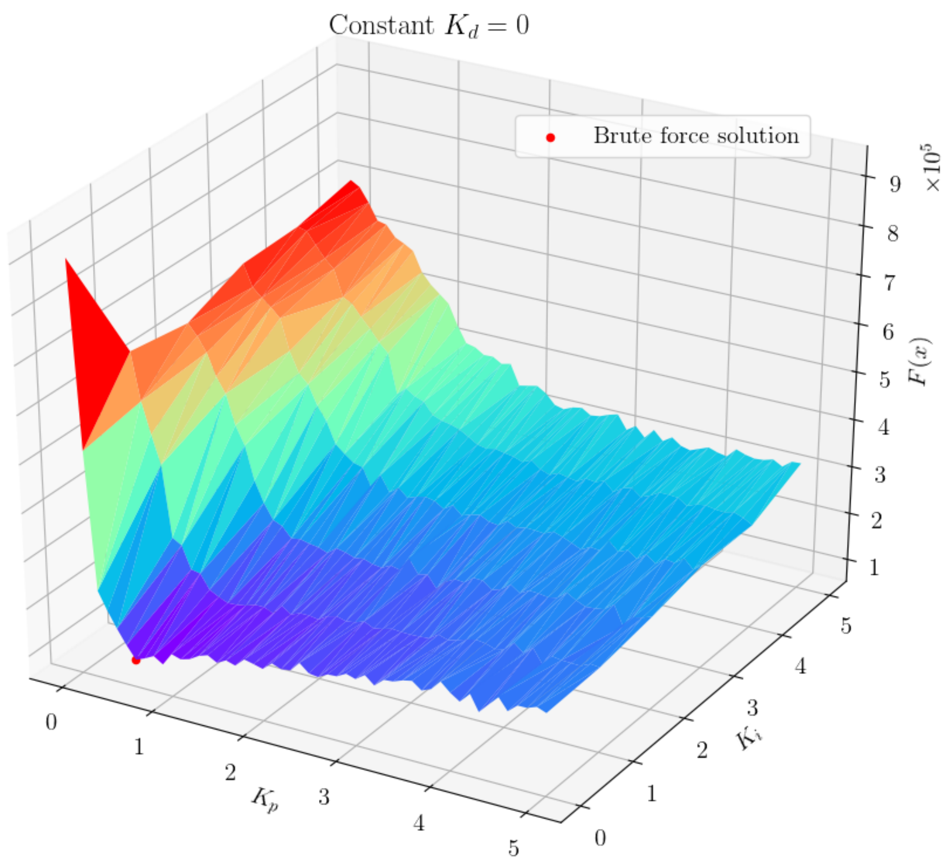
Of course, finding a good solution in this case was possible because of the previous knowledge of the system and the region of interest. Without this knowledge, the exploration area would be bigger and the required time for finding the solution would be nearly impractical. Furthermore, many tests would be necessary to account for the changing parameters due to the contraction range and populate a table for gain scheduling.

From the brute force results in fig. 42, the surface of the objective function doesn't look so complex to optimize. There are some fluctuations due to sensor noise, friction, vibrations, PWM timing, and so on, but the obtained surface doesn't have many sudden changes and local minima if only P and I gains are considered.

6.9 Chapter conclusions

Tuning a controller for a nonlinear system is a challenging task. To tackle this problem, two methods were proposed: model base methods and no model

Figure 42: Brute force results for the IAE criterion on the 1-DoF arm bench.



Source: Own authorship.

methods.

These methods use an objective function as a metric to be optimized. In the case of the model based methods, a simulation is executed to compute the value of the objective function. In contrast, the no model method use a real system to compute the value of the objective function.

The objective function design is covered, proposing two principles: direct and indirect. An example objective function is designed using each principle.

Using these methods is possible to find the controller's parameters before the control system is even built (model based methods) or find the parameters directly in the real system, identifying the model and generating the table of gains or without modeling at all (no model methods).

Both methods are implemented and tested, showing system responses compatible with the objective function criterion for the simulation method. Using the bench to generate the objective function score made the SA unable to converge to good solutions, probably because of the noisy position measurements. To accomplish an optimization using the bench, a more robust to noise optimization method must be used.

7 OPTIMIZATION OF NOISY OBJECTIVE FUNCTIONS

In the previous chapter, SA and brute force were used to tune the gains of a PID controller. SA found good solutions to the control problem that were very close to the brute force solution produced with a supercomputer. Although SA is capable of finding good gains for the controller on the simulated system, it fails to converge on the real bench. One hypothesis of this failure is that the bench system produces noisy data, which produces false positives in the energy test and, therefore, accept solutions due to noise instead of quality.

In chapter 4, it was reviewed some popular optimization algorithms, but they are mainly focused on the derivative, or the gradient, of the function, which are very sensitive to noise. In this chapter, optimization algorithms that are robust to noise will be described and compared in benchmark functions and the best performing algorithms will be used to find the controller's gains of the positioning system on the bench. Additionally, one algorithm called Shrinking Window Optimization Algorithm will be proposed and compared with some state-of-the-art algorithms.

7.1 Genetic Algorithms

The Genetic Algorithms, or GA, are a class of computational models inspired by the theory of evolution applied to solve optimization problems (WHITLEY, 1994). The idea that new species arise from a common ancestor and that natural selection promotes the prosperity of species that adapts to the environment, is exploited to solve complex optimization problems. The process that generates new species that are optimal to the environment can be mimicked to generate new solutions that are optimal to some criteria.

The "environment" is translated into an objective function and the species, or solutions, are subjected to the survival of the fittest. In nature, however, the fittest is not always the one that survives, but the one with higher probability of surviving. In an optimization perspective, this uncertainty of which solution is going to "survive" is good in the sense that the algorithm is not trapped by a local optimum. Therefore, this "nature selection" must have some kind of randomness.

Sexual reproduction promotes more genetic diversity than asexual reproduction, and is dominant among more advanced and complex organisms. For the optimization analogy, combining two good solutions can produce an even better solution.

Mutation is also a factor that promotes diversity and the rise of new species, or a new class of solutions. In an optimization perspective, mutation can be used to explore the solution space and eventually find the global optimum.

Each individual have a DNA, or genetic code, that is combined due to reproduction or changed due to mutation. In a solution, the genetic code is formed by a sequential list of numbers, or symbols, that represents the parameters to be optimized, e.g. the coordinates of point in a fitness hypersurface.

A group of solutions are created to "compete" with each other, and this group is known as a population, which represent a generation. The "natural selection" occurs in this generation, according to the "fitness value" of the solutions and some randomness. The selected solutions are used to populate the next generation, and so on.

In summary, genetic algorithms have these main steps:

- **creation**: create the population of the first generation randomly across the solution space or using some other method;
- **selection**: attribute a fitness value for each solution and select a subgroup with a given probability, which is proportional with the fitness value;
- **reproduction**: copy or combine the solutions in the selected subgroup to form the population of the next generation;
- **mutation**: randomly change the parameters of a solution with a given probability;
- **populate**: if not all solutions are "children" of the selected subgroup, generate the rest of the population;
- **repeat until stop criteria**: go to selection or stop the algorithm.

Some techniques are used to improve exploration and make the selection more robust to objective function values. In the latter case, instead of using the value of the objective function to calculate the probability, this value is only used to rank the solutions. The probability of selection in this case depends on the position of the solution in the rank. As an example, solutions are ranked in from 1 to N, 1 being the best solution according the $J(x_i)$ and N the population size, representing the worse solution. So, the probability P can be determined by

$$P(r) = \left\{ P_c\text{if r} = 1(1-P_c)^{r-1}P_c\text{if } r > 1 < N(1-P_c)^{r-1} \right. \tag{7.1}$$

here P_c is a probability constant to be chosen and r the position in the rank of the solution. In the former case, a diversity metric can be used in the selection.

In general, good solutions that are different may produce better solutions when combined.

Despite the powerful concept, GA has many degrees of freedom for the implementation. Each of the steps mentioned above can be implemented differently, using heuristics to guide the process. There is an effort to "tweak" the algorithm to produce solutions, avoid trapping the generation in local optima and be able to explore the whole solution space.

7.2 Bayesian Optimization

Bayesian Optimization, or BO, is often used to optimize functions that are expensive to evaluate and that are subject to noise. It works by building a probabilistic model of the function and calculate an acquisition function to determine the next sample region.

The probabilistic model is often based on the Gaussian Process (GP), which is used as a prior, and as data is collected from evaluating the function, the prior is updated to form a posterior distribution. The posterior distribution is used to calculate the acquisition function and determine the next sample coordinates.

It is based o the Bayes' theorem, given by

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{7.2}$$

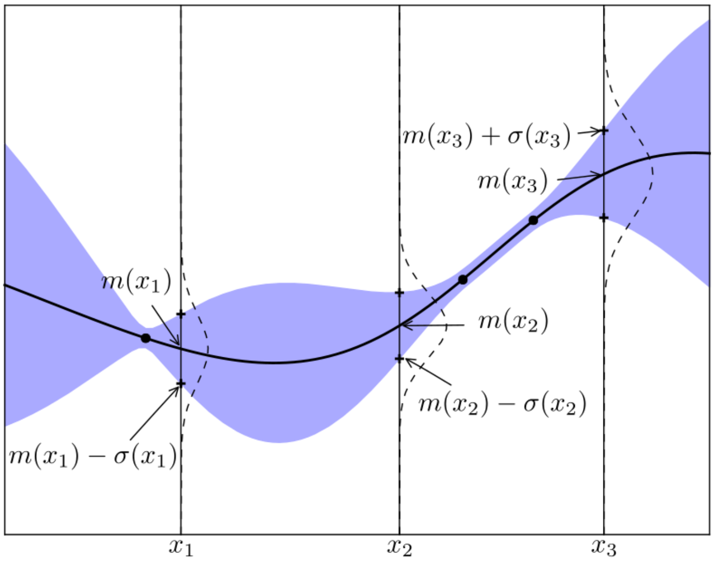
The term $P(B)$, a normalization term, is often ignored in BO because the goal is not to calculate a specific conditional probability, but to have a metric to optimize a quantity. And so, Bayes' Theorem can be applied to an optimization problem as

$$P(M|E) \propto P(E|M)P(M) \tag{7.3}$$

in which the posterior probability of a model M, given evidence data E is proportional to the likelihood of E given M times the prior probability of M (BROCHU; CORA; FREITAS, 2010).

GP is used to create a surrogate model of the objective function. The surrogate model $f(x)$ is analogous to a function composed of a mean curve m and variance of a normal distribution over the possible values of f at x (BROCHU; CORA; FREITAS, 2010). One example of a surrogate model is illustrated in fig. 43.

Figure 43: Surrogate model of an objective function as a Gaussian Process.



Source: Adapted from (BROCHU; CORA; FREITAS, 2010).

A surrogate model $f(x)$ is specified by

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')) \tag{7.4}$$

where k is the covariance as a function of the values x and x'. One example of a covariance function k is the squared exponential function, given by

$$k(x, x') = e^{-\frac{1}{2}||x-x'||^2} \tag{7.5}$$

More sophisticated covariance functions are used to tune the smoothness of the function according to its dimension with a hyperparameter. Therefore, some parameters may have different levels of smoothness in the surrogate model.

From the example in fig. 43, a good guess for the next sampling region would be the far right of the graph, where the mean $m(x_n)$ plus the standard deviation $\sigma(x_n)$ is likely to return the highest value of $f(x)$ in this interval. Another possible strategy would be to explore the hypersurface of $f(x)$ by sampling regions of high variance. This states a trade-off between exploration and exploitation which guides the algorithm towards knowing more about f(x) or improving the solution about a maximum. This trade-off is expressed as an acquisition function and the sampling coordinate x_n is determined by optimizing the acquisition function.

In summary, the algorithm for Bayesian Optimization is as follows:

- determine x_n by optimizing the acquisition function;
- sample the, possibly noisy, objective function $y_n = f(x_n) + \epsilon_n$, where ϵ_n represents the noise of the function evaluation at x_n;
- update the GP with the new evidence data x_n, y_n;
- repeat until the stop criterion.

Although the creation of a surrogate model makes the selection of the sampling coordinate more reasonable, updating the surrogate model, calculating the acquisition function and optimizing it can introduce significant overhead that increases over the iterations. Because of that, BO is not advised for optimizing functions that are fast or cheap to evaluate. Nevertheless, it can be used for comparison purposes to determine the performance of optimization algorithms in a benchmark.

When using a test bench to evaluate the objective function, the experiment setup can take much longer than the experiment itself and the number of iterations can be limited. In this case, BO can be a valid option because the computational overhead might be comparable with the experiment setup time for few iterations (e.g. < 500 iterations).

7.3 Shrinking Window Optimization Algorithm

The Shrinking Window Optimization Algorithm (SWOA) was inspired by the Simulated Annealing algorithm and by some ideas from the genetic algorithms, considering the drawbacks of the SA when optimizing the control system in the bench. It is a sequential black-box optimization algorithm, designed to optimize objective functions subjected to noise. Even though it is sequential, it is easy to improve the learning time through parallelization on the function evaluation.

The algorithm is mainly consisted of a convergence factor, a movable search area (Window), a sampler and a chooser. The convergence factor is responsible for tuning the algorithm for exploration or exploitation, by shrinking the search window or changing the probability of choosing a point. The search window is the domain in which the sampler will acquire data. The chooser attributes a probability to each data point and randomly selects one point.

The algorithm works by repeating the process of calculating the convergence factor, adjusting the window size, sampling and choosing a data point until the stop criterion is reached.

By attributing a non-zero probability to the sampled data points, the algorithm has the ability to escape from local minima.

7.3.1 Implementation

Let J be a real-value cost function in $\Re^n$, ξ be the convergence vector, ε be the search domain size vector, Ω be the complete search domain set and N the number of samples per iteration. The objective of the algorithm is to find an element x in the domain Ω which minimizes, or nearly minimizes, the value of $J(x)$.

Initially, N samples are taken from Ω, which can be made in parallel if possible. Then, a probability is attributed to each sample x_i according to their value $J(x_i)$. Lower values are given higher probability of being chosen than lower values. The convergence of the algorithm is critically dependent on this probability attribution. An example is given below:

$$P(x_i|x_{best}, x_{max}) = \left| \frac{x_i - x_{max}}{x_i} \right| \frac{1}{e^{x_i - x_{best}}} \tag{7.6}$$

Where $P(x_i|x_{best}, x_{max})$ represents the probability of the solution x_i being chosen given x_{best} and x_{max}. The variable x_{max} is the value of the worst solution from the N samples and the variable x_{best} is the current best solution.

To normalize the samples' probabilities such that $\sum_{i=1}^{N} P_i = 1$, a range from $0 \rightarrow 1$ is filled with sub-regions for each sampled data point. The size of each region is proportional to the probability of that point being chosen. Then a random number from $0 \rightarrow 1$ is generated, which determines the selection of the point.

In every following iteration, the window size ε is updated according to the convergence vector ξ:

$$\varepsilon_{i+1} = \varepsilon_i \odot \xi_i \tag{7.7}$$

Where $\odot$ is the Hadamard product, or element wise product. The convergence vector can also be updated according to some criterion, e.g. considering the variance of the samples or, for simplicity, each element can be constant for a geometrical window shrinking.

The next window is centered at the current solution x_n, and the process of sampling, probability attribution and choosing a point is repeated until the stop criterion.

The convergence of this algorithm to a, hopefully, good solution is a stochastic process of solution selections. Initially, the window size is large and solution from many different locations can be evaluated. The probability of choosing a good solution is proportionally higher than a worse solution (according to eq. 7.6), which guides the algorithm towards a good subsection of the hypersurface. Also, the best solution is always considered, which also helps to guide the window towards the location of the best solutions when transitioning from the exploration phase to the exploitation phase.

The idea is that the best solution should be the most probable of being found, which makes sense because on every window, the best solution has always the highest probability. Also, because the best solution is also considered, very bad solutions are increasingly more improbable of being selected. So, it is to expect, that if this process proceeds several times, the probability of finding a good solution increases, as the chain of probabilities will favor the best solution over the others. So, the exploration of the hypersurface will occur when the window size is large, the window will be guided towards the location of good solutions and exploitation will occur when the window size shrinks and good solutions that were found along the way will causes bad solutions to be more difficult to be selected.

7.4 Optimization on noisy benchmark functions

In this section, some benchmark noisy functions will be used to compare and analyse the performance of the mentioned optimization algorithms in this chapter. Additionally, a pure random search is used as a comparison basis.

The implementation of the algorithms are from the author of this thesis, based on the theory of this chapter, with exception of the Bayesian optimization in which the python library scikit-optimize implementation was used.

Two benchmark functions will be used, namely J_1 and J_2. The former is given by

$$J_1(a, b) = a^2 + b^2 + 4\sin(a) + 10\sin(b) + \chi(0, 5) \tag{7.8}$$

and the latter is given by

$$J_2(a, b, c, d) = a^2 + b^2 + c^2 + d^2 + 4\left(\sin(a) + \sin(c)\right) + 10\left(\sin(b) + \sin(d)\right) + 2\chi(0, 5) \tag{7.9}$$

where $\chi(0, 5)$ represents a Gaussian noise with mean value of 0 and standard deviation of 5. The objective is to find the parameters a, b or a, b, c, d that minimizes the function J_1 and J_2, respectively.

On the optimization tests, the time of execution, mean objective function value, standard deviation and best solution are recorded for comparison. Results for the optimization of two parameters on J_1 are shown in table 4 and the

Table 4: Optimization results for the J_1 benchmark function.

Algorithm	$\bar{min}(J_1)$	σ	Evaluations	Repetition	J_{1best}	Time of execution
Random	-12.918	3.149	100	100	-21.790	0.20s
BO	-21.819	2.192	100	100	-30.034	8671.80s
GA	-12.427	4.731	100	100	-25.825	0.54s
SWOA	-19.650	2.971	100	100	-31.622	0.22s
Random	-13.791	4.225	500	10	-20.948	0.01s
BO	-25.763	1.834	500	10	-28.896	31279.0s
GA	-17.659	4.468	500	10	-24.434	0.22s
SWOA	-24.584	2.073	500	10	-29.200	0.11s

Source: Own authorship.

Table 5: Optimization results for the J_2 benchmark function.

Algorithm	$\bar{min}(J_2)$	σ	Evaluations	Repetition	J_{2best}	Time of execution
Random	-21.929	3.469	500	10	-28.972	0.12s
BO	-15.640	8.523	500	10	-25.200	41118.97s
GA	-24.174	4.560	500	10	-34.079	0.24s
SWOA	-43.235	8.112	500	10	-62.344	0.41s

Source: Own authorship.

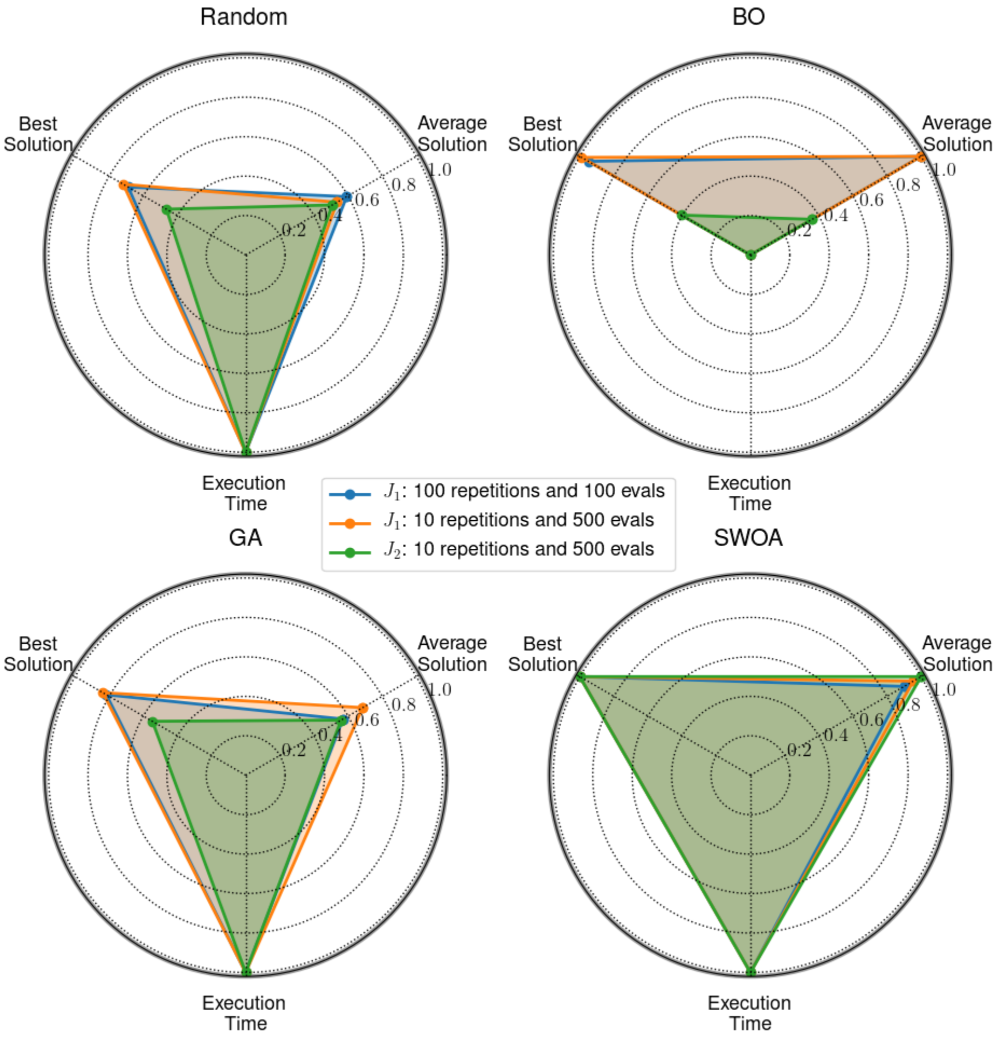
optimization of four parameters on J_2 are shown in table 5.

Note that the repetition for the 500 evaluation optimization tests is 10. This is because BO takes too long to finish and to keep the comparison fair with the other algorithms. The comparison of the algorithms is illustrated in fig. 44. The results are normalized and higher values represent better performance.

According to the results in table 4, when dealing with very few iterations (100 evaluations test), BO finds a better solution on average and with less deviation, but it is much more computationally expensive. GA was the worst performing algorithm for few iterations, even losing for the random approach. Although SWOA lost to BO on average, it found the best solution overall.

When more evaluations are taken, GA starts to be more effective and surpasses the random algorithm. BO consistently finds the best solution on average, but SWOA kept on finding the best solution overall, in a fraction of the time of

Figure 44: Algorithms comparison on the benchmark tests.



Source: Own authorship.

BO.

As the number of the iterations grows, it is expected that GA would be more effective. However, for optimizing the system in the real world, the number of iterations can rapidly make this approach unfeasible.

The results in table 5 elucidates the performance of the algorithms when the number of parameters increases. For instance, BO was the worst performer, even losing for the random approach. The complexity of the surrogate model increased to the point that even 500 evaluations was not sufficient for it to find a good solution, and also increased the time need to compute the next try. GA was the second best algorithm, finding the second best solution on average and the second best solution overall. The most effective algorithm for this function was the SWOA, finding not only the best solution, but with a considerable difference in quality of the solution in comparison with the other algorithms.

In summary, it was found that BO is a great algorithm to be used in situations where there are few parameters to optimize and the time for the evaluation is long. In this case, the calculation of the surrogate model is compensated by the fewer iterations needed to find a good solution. GA is a better algorithm when the number of parameters is higher and the time for the function evaluation is not high. This way the GA can benefit from more evaluations and find better solutions than BO. SWOA showed to good in both cases, finding good solutions in few iterations with few parameters and even the best one when dealing the increased parameters, also benefiting from more evaluations.

Now these algorithms have to be tested in the real system optimization. Because the PID controller has only 3 parameters to be optimized and finding the best solution in the minimum ammount of iterations is desired, the top 2 algorithms for this task would be BO and SWOA.

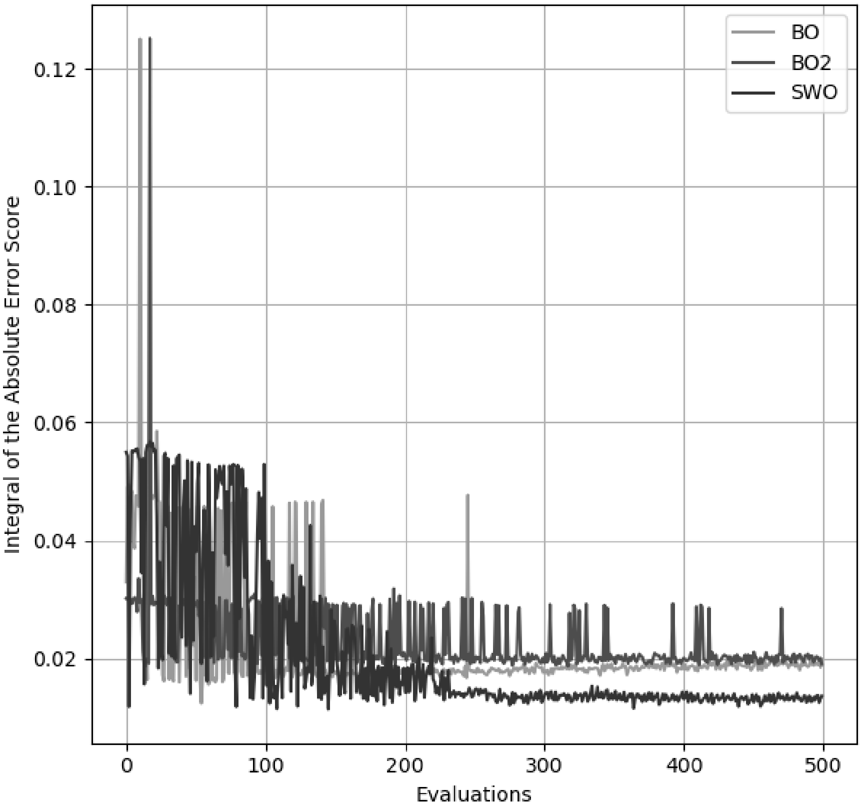
7.5 Optimizing control parameters in the physical bench

As discussed in previous sections, there are some algorithms that can optimize noisy functions. This is very important because bench tests are subject to noise from several sources, such as from sensors, vibrations, ambient conditions, and so on. To test whether the discussed algorithms are capable of optimizing the controller's parameters in a reasonable time, the following experiment was used.

A pneumatic muscle is fixed at the top end to a dynamometer and the bottom end to a mass, which is attached to a spring in a metal beam. This beam can deform elastically and the displacement is measured by a capacitive sensor. The sensor signal is used to control the position of the mass, using a PID controller which parameters are going to be optimized using the top two algorithms that were described in this chapter: BO and SWOA.

The objective of the control system is to position the mass as fast as possible without overshoot. The IAE criterion was used to optimize the PID parameters. The learning process is illustrated in fig. 45.

Figure 45: PID parameters learning and performance comparison.



Source: Own authorship.

This experiment showed that optimizing the controller's parameters in the bench is a viable option when using BO and SWOA. All trials could find a good solution in the 500 evaluations limit.

While BO took around 2.7h to complete the 500 evaluations, SWOA completed in around 1.4h, being 48,15

It's clear from the plot in fig. 45 that while both tries of BO couldn't improve the solution after the first 50 evaluations, SWOA kept improving the solution until the end of the test.

The plot in fig. 46 depicts the SWOA convergence to the solution during the test.

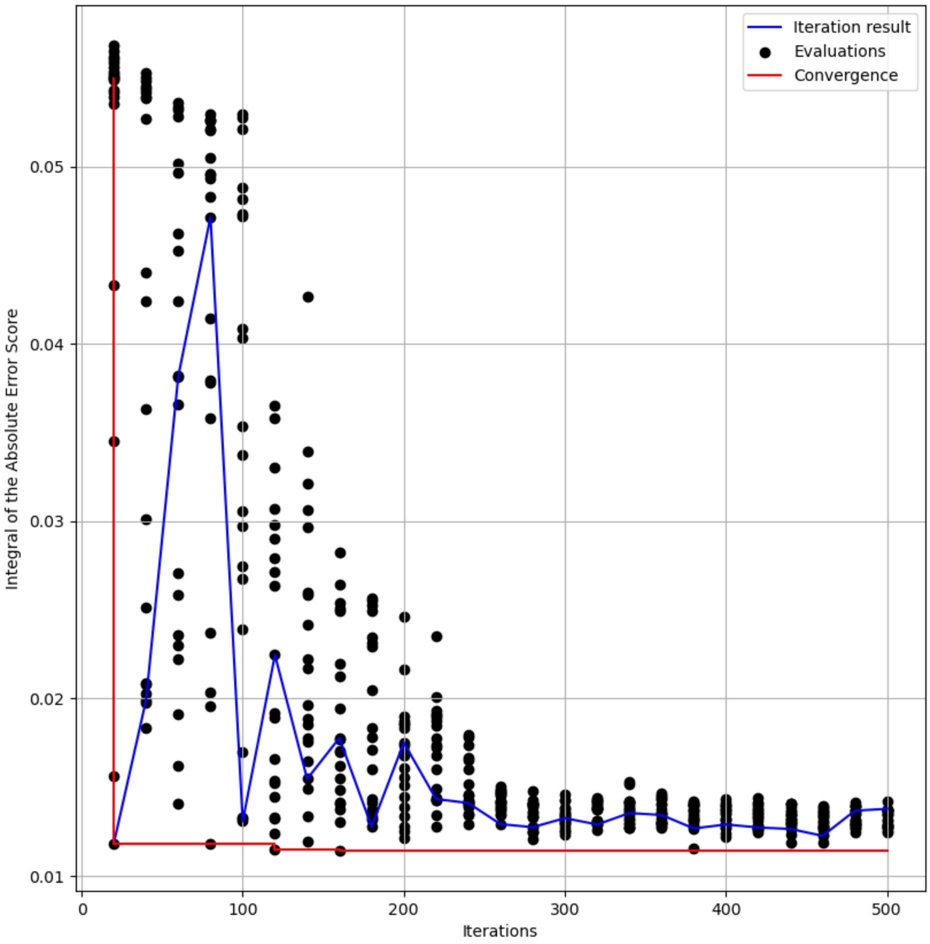
On every iteration, 20 evaluations are made (repesented by the black points). On every iteration, a candidate solution is chosen, the window is shrunk and another iteration begins, which the window is centered on the last iteration's result. The blue line represents the selected solution of each iteration and the red line represents the convergence of the best solution. Note that the iteration result is not always the best solution to promote exploration.

7.6 Chapter conclusions

Some optimization algorithms known to be robust to noise were described and evaluated. Also, a novel algorithm called Shrinking Window Optimization Algorithm was proposed and tested against the described algorithms.

On synthetic functions it was noted that BO is good when very few parameters needs to be optimized, finding on average the best solution, but needs orders of magnitude more time to execute. Genetic Algorithms are good when the number of evaluation is high. SWOA found the best solution in all cases, can be used for few or many parameters, few iterations or many iterations and the time of execution is comparable to the fastest algorithm.

Figure 46: SWOA learning process during the bench test.



Source: Own authorship.

On the real bench, both BO and SWOA could find good solutions to the control problem, but SWOA found the best solution, being 8

Also, SWOA scales better with increased number of parameters, compared to BO, being able to find better solutions earlier, which makes it suitable to learn more complex control responses.

8 CONCLUSIONS

It was reviewed some potential applications for pneumatic artificial muscles as non conductive actuators for biomedial and rehabilitation applications. There are several other applications for these actuators, and the development of tools for designing, fabricating, sensing and controlling systems powered by PAMs can spread their use in many areas in the future.

Many different muscle constructions were reviewed. Some design characteristics can be used to produce an actuator according to the application's requirements. For example, some muscle constructions have lower hysteresis than others, work with lower pressures, have higher durability, and so on.

Unfortunately, just a few have first principle models and they are not very accurate. Therefore, models have to be developed or strategies for working with them without first principle models are necessary.

The control problem was reviewed and, although there were many researches related to PAMs control, the following drawbacks were identified:

- inaccurate physically based models which affect the controller performance and limit its practical use in designing model based controllers;
- usage of identified models that requires prior construction and experimentation before designing the controller;
- complex control strategies that require highly qualified personnel for imple-

menting and tuning the controller;

- usage of high cost valves and sensors;
- usage of conductive and ferromagnetic apparatus, which cannot be used in electromagnetic risk applications;
- trial and error based tunning;
- controllers designed for specific apparatus and lack of flexibility for interchangeable parts and
- not flexible for designing controllers with different performance requirements and constraints.

A control strategy was proposed which is to use a simple and flexible controller and tune it with an optimization algorithm, according to the desired performance criterion translated into a cost function.

For the sensing problem, it was reviewed some techniques used in the literature. An image based sensor was implemented and used for controller validation purposes.

Two approaches were proposed for tuning the controller: model based and no model methods. Model based methods can be used when the mathematical model of the PAM is available. For situations where no accurate models are available, it was proposed optimization methods that are based in the real system evaluation.

Tests were made with both approaches and results showed that it is possible to tune the controller according to the desired performance criterion, simulating the nonlinear system and using an optimization algorithm to find the controller's parameters, or by optimizing the parameters in real time in the bench.

When learning the control in the bench, the proposed method SWOA, had the best results, outperforming state-of-the-art algorithm BO in 8

The hypothesis that the control of a very nonlinear system such as the PAM actuated positioning system can be learned in real time in the bench is plausible and the experiments showed good results, positioning the mass rapidly and without overshoot as the criterion sugested.

The method showed to be flexible, because the proposed algorithms are capable of finding good solutions on a variety of loss functions. Therefore, custom loss functions can be tailored to each application's requirements.

Also, the methods can be combined when a reasonable model is available, be first optimizing the simulated system and be fine-tuned in the real system.

9 FUTURE WORK

The proposed algorithm (SWOA) has some parameters that can be further studied to improve the algorithm performance. The probability attribution, shrink scheduling, etc, can be further investigated to provide more robustness on the optimization process.

It was used a controller as a base to be optimized. However, as the SWOA showed to scale well with more parameters, the entire control law can be learned instead of using the logic of a fixed controller such as the PID. This way the gain scheduling approach can be simplified to a generic function or a small Neural Network.

Another approach would be to learn the system dynamics and to use the model predictive control approach. A generic function can be used as base and be optimized in real time to predict the next state, given the controller's input. Then the best control signal can be optimized for the next step, and so on.

The design of objective functions can also be studied. They are responsible for the quality and convergence to a good solution. Combining different objective functions can make the optimization faster and with higher quality solutions, with fewer optimization trials.

REFERENCES

AGUIAR, C. *Avaliação de Acidente Vascular Cerebral em Tomografia Computadorizada Utilizando Algoritmo de Otimização de Formigas*. Dissertação (Mestrado), 2017.

ANDROWIS, G. J.; NOLAN, K. J. Evaluation of a robotic exoskeleton for gait training in acute stroke: A case study. In: *Wearable Robotics: Challenges and Trends*. [S.l.]: Springer, 2017. p. 9–13.

ANH, H. P. H. Online tuning gain scheduling mimo neural pid control of the 2-axes pneumatic artificial muscle (pam) robot arm. *Expert systems with applications*, Elsevier, v. 37, n. 9, p. 6547–6560, 2010.

ÅSTRÖM, K. J.; HÄGGLUND, T. The future of pid control. *Control engineering practice*, Elsevier, v. 9, n. 11, p. 1163–1175, 2001.

BALDWIN, H. A. Realizable models of muscle function. In: *Biomechanics*. [S.l.]: Springer, 1969. p. 139–147.

BARGAR, W. L.; BAUER, A.; BÖRNER, M. Primary and revision total hip replacement using the robodoc (r) system. *Clinical orthopaedics and related research*, LWW, v. 354, p. 82–91, 1998.

BEULLENS, T. *Hydraulic or pneumatic drive device*. [S.l.]: Google Patents, 1989. US Patent 4,841,845.

BRENNER, D. J.; HALL, E. J. Computed tomography - an increasing source of radiation exposure. *New England Journal of Medicine*, Mass Medical Soc, v. 357, n. 22, p. 2277–2284, 2007.

BROCHU, E.; CORA, V. M.; FREITAS, N. D. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.

BUSHONG, S. C.; CLARKE, G. *Magnetic resonance imaging: physical and biological principles*. [S.l.]: Elsevier Health Sciences, 2014.

CAO, J.; XIE, S. Q.; DAS, R. Mimo sliding mode controller for gait exoskeleton driven by pneumatic muscles. *IEEE Transactions on Control Systems Technology*, IEEE, v. 26, n. 1, p. 274–281, 2018.

CARBONELL, P.; JIANG, Z.; REPPERGER, D. Nonlinear control of a pneumatic muscle actuator: backstepping vs. sliding-mode. In: IEEE. *Control Applications, 2001.(CCA'01). Proceedings of the 2001 IEEE International Conference on*. [S.l.], 2001. p. 167–172.

CHAN, S.; LILLY, J. H.; REPPERGER, D. W.; BERLIN, J. E. Fuzzy pd+ i learning control for a pneumatic muscle. In: IEEE. *Fuzzy Systems, 2003. FUZZ'03. The 12th IEEE International Conference on.* [S.l.], 2003. v. 1, p. 278–283.

CHAPUIS, D.; GASSERT, R.; SACHE, L.; BURDET, E.; BLEULER, H. Design of a simple mri/fmri compatible force/torque sensor. In: IEEE. *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on.* [S.l.], 2004. v. 3, p. 2593–2599.

CHIANG, C.-J.; CHEN, Y.-C. Neural network fuzzy sliding mode control of pneumatic muscle actuators. *Engineering Applications of Artificial Intelligence*, Elsevier, v. 65, p. 68–86, 2017.

CHOU, C.-P.; HANNAFORD, B. Measurement and modeling of mckibben pneumatic artificial muscles. *IEEE Transactions on robotics and automation*, IEEE, v. 12, n. 1, p. 90–102, 1996.

COMBER, D.; BARTH, E. J. Precision position tracking of mr-compatible pneumatic piston-cylinder using sliding mode control. In: AMERICAN SOCIETY OF MECHANICAL ENGINEERS. *ASME 2011 Dynamic Systems and Control Conference and Bath/ASME Symposium on Fluid Power and Motion Control.* [S.l.], 2011. p. 45–51.

CSIKÓS, S.; SZ, B.; SÁROSI, J. Fuzzy control of an antagonistic system driven by pneumatic muscle actuators. In: *4th International Conference and Workshop Mechatronics in Practice and Education-MECHEDU.* [S.l.: s.n.], 2017. p. 89–92.

DAERDEN, F. Conception and realization of pleated pneumatic artificial muscles and their use as compliant actuation elements. *Vrije Universiteit Brussel*, p. 176, 1999.

DAERDEN, F.; LEFEBER, D. Pneumatic artificial muscles: actuators for robotics and automation. *European journal of mechanical and environmental engineering*, v. 47, n. 1, p. 11–21, 2002.

DAERDEN, F.; LEFEBER, D.; VERRELST, B.; HAM, R. V. Pleated pneumatic artificial muscles: actuators for automation and robotics. In: IEEE. *Advanced Intelligent Mechatronics, 2001. Proceedings. 2001 IEEE/ASME International Conference on.* [S.l.], 2001. v. 2, p. 738–743.

DE, H. H. *Tensioning device for producing a linear pull.* [S.l.]: Google Patents, 1949. US Patent 2,483,088.

DECHARMS, R. C. Applications of real-time fmri. *Nature Reviews Neuroscience*, Nature Publishing Group, v. 9, n. 9, p. 720, 2008.

DESIKAN, A.; CRICHTON, S.; HOANG, U.; BARRATT, B.; BEEVERS, S. D.; KELLY, F. J.; WOLFE, C. D. Effect of exhaust-and nonexhaust-related components of particulate matter on long-term survival after stroke. *Stroke*, Am Heart Assoc, v. 47, n. 12, p. 2916–2922, 2016.

DONGARE, A.; KHARDE, R.; KACHARE, A. D. et al. Introduction to artificial neural network. *International Journal of Engineering and Innovative Technology (IJEIT)*, Citeseer, v. 2, n. 1, p. 189–194, 2012.

FEDERSPIL, P. A.; GEISTHOFF, U. W.; HENRICH, D.; PLINKERT, P. K. Development of the first force-controlled robot for otoneurosurgery. *The Laryngoscope*, Wiley Online Library, v. 113, n. 3, p. 465–471, 2003.

FISCHER, G. S.; IORDACHITA, I.; CSOMA, C.; TOKUDA, J.; DIMAIO, S. P.; TEMPANY, C. M.; HATA, N.; FICHTINGER, G. Mri-compatible pneumatic robot for transperineal prostate needle placement. *IEEE/ASME transactions on mechatronics*, IEEE, v. 13, n. 3, p. 295–305, 2008.

GASSERT, R.; MOSER, R.; BURDET, E.; BLEULER, H. Mri/fmri-compatible robotic system with force feedback for interaction with human motion. *IEEE/ASME transactions on mechatronics*, IEEE, v. 11, n. 2, p. 216–224, 2006.

GAYLORD, R. H. *Fluid actuated motor system and stroking device.* [S.l.]: Google Patents, 1958. US Patent 2,844,126.

GENDREAU, M.; POTVIN, J.-Y. Metaheuristics in combinatorial optimization. *Annals of Operations Research*, Springer, v. 140, n. 1, p. 189–213, 2005.

GUO, D.; WANG, W.; ZHANG, Y.; YAN, Q.; CAI, J. Angle tracking robust learning control for pneumatic artificial muscle systems. *IEEE Access*, IEEE, v. 9, p. 142232–142238, 2021.

HAIK, J.; DANIEL, S.; TESSONE, A.; ORENSTEIN, A.; WINKLER, E. Mri induced fourth-degree burn in an extremity, leading to amputation. *burns*, Elsevier, v. 35, n. 2, p. 294–296, 2009.

HALL, W. A. The safety and efficacy of stereotactic biopsy for intracranial lesions. *Cancer*, Wiley Online Library, v. 82, n. 9, p. 1749–1755, 1998.

HENRI, M. A. *Elastic diaphragm.* [S.l.]: Google Patents, 1953. US Patent 2,642,091.

HILL, K. O.; MELTZ, G. Fiber bragg grating technology fundamentals and overview. *Journal of lightwave technology*, IEEE, v. 15, n. 8, p. 1263–1276, 1997.

HOCKSTEIN, N. G.; GOURIN, C.; FAUST, R.; TERRIS, D. J. A history of robots: from science fiction to surgical robotics. *Journal of robotic surgery*, Springer, v. 1, n. 2, p. 113–118, 2007.

HUANG, V. S.; KRAKAUER, J. W. Robotic neurorehabilitation: a computational motor learning perspective. *Journal of neuroengineering and rehabilitation*, BioMed Central, v. 6, n. 1, p. 5, 2009.

IMMEGA, G.; KUKOLJ, M. *Axially contractable actuator.* [S.l.]: Google Patents, 1990. US Patent 4,939,982.

JAKOPEC, M.; BAENA, F. R. y; HARRIS, S. J.; GOMES, P.; COBB, J.; DAVIES, B. L. The hands-on orthopaedic robot" acrobot": Early clinical trials of total knee replacement surgery. *IEEE Transactions on Robotics and Automation*, IEEE, v. 19, n. 5, p. 902–911, 2003.

KHANICHEH, A.; MUTO, A.; TRIANTAFYLLOU, C.; WEINBERG, B.; ASTRAKAS, L.; TZIKA, A.; MAVROIDIS, C. fmri-compatible rehabilitation hand device. *Journal of neuroengineering and rehabilitation*, BioMed Central, v. 3, n. 1, p. 24, 2006.

KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. Optimization by simulated annealing. *science*, American Association for the Advancement of Science, v. 220, n. 4598, p. 671–680, 1983.

KITAGO, T.; GOLDSMITH, J.; HARRAN, M.; KANE, L.; BERARD, J.; HUANG, S.; RYAN, S. L.; MAZZONI, P.; KRAKAUER, J. W.; HUANG, V. S. Robotic therapy for chronic stroke: general recovery of impairment or improved task-specific skill? *Journal of Neurophysiology*, Am Physiological Soc, v. 114, n. 3, p. 1885–1894, 2015.

KLEINWACHTER, H.; GEERK, J. *Device with a pressurizable variable capacity chamber for transforming a fluid pressure into a motion.* [S.l.]: Google Patents, 1972. US Patent 3,638,536.

KLUTE, G. K.; HANNAFORD, B. Accounting for elastic energy storage in mckibben artificial muscle actuators. *Journal of dynamic systems, measurement, and control*, American Society of Mechanical Engineers, v. 122, n. 2, p. 386–388, 2000.

KOTHERA, C. S.; PHILEN, M.; TONDU, B. Modelling of the mckibben artificial muscle: A review. *Journal of Intelligent Material Systems and Structures*, Sage Publications Sage UK: London, England, v. 23, n. 3, p. 225–253, 2012.

KRAKAUER, J. W.; CARMICHAEL, S. T.; CORBETT, D.; WITTENBERG, G. F. Getting neurorehabilitation right: what can be learned from animal models? *Neurorehabilitation and neural repair*, Sage Publications Sage CA: Los Angeles, CA, v. 26, n. 8, p. 923–931, 2012.

KRIEGER, A.; CSOMA, C.; IORDACHITA, I. I.; GUION, P.; SINGH, A. K.; FICHTINGER, G.; WHITCOMB, L. L. Design and preliminary accuracy studies of an mri-guided transrectal prostate intervention system. In: AYACHE, N.; OURSELIN, S.; MAEDER, A. (Ed.). *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2007*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. p. 59–67. ISBN 978-3-540-75759-7.

KUKOLJ, M. *Axially contractable actuator.* [S.l.]: Google Patents, 1988. US Patent 4,733,603.

LE, Q. V.; NGIAM, J.; COATES, A.; LAHIRI, A.; PROCHNOW, B.; NG, A. Y. On optimization methods for deep learning. In: OMNIPRESS. *Proceedings of the 28th International Conference on International Conference on Machine Learning*. [S.l.], 2011. p. 265–272.

LILLY, J. H.; YANG, L. Sliding mode tracking for pneumatic muscle actuators in opposing pair configuration. *IEEE transactions on control systems technology*, IEEE, v. 13, n. 4, p. 550–558, 2005.

LIN, L.-H.; YEN, J.-Y.; WANG, F.-C. Robust control for a pneumatic muscle actuator system. *Transactions of the Canadian Society for Mechanical Engineering*, v. 37, n. 3, p. 581–590, 2013.

LIU, J.; DAI, T.; ELSTER, T.; SAHGAL, V.; BROWN, R.; YUE, G. Simultaneous measurement of human joint force, surface electromyograms, and functional mri-measured brain activation. *Journal of Neuroscience Methods*, Elsevier, v. 101, n. 1, p. 49–57, 2000.

MADDEN, J. D.; VANDESTEEG, N. A.; ANQUETIL, P. A.; MADDEN, P. G.; TAKSHI, A.; PYTEL, R. Z.; LAFONTAINE, S. R.; WIERINGA, P. A.; HUNTER, I. W. Artificial muscle technology: physical principles and naval prospects. *IEEE Journal of oceanic engineering*, IEEE, v. 29, n. 3, p. 706–728, 2004.

MAJIDI, C. Soft robotics: a perspective?current trends and prospects for the future. *Soft Robotics*, Mary Ann Liebert, Inc. 140 Huguenot Street, 3rd Floor New Rochelle, NY 10801 USA, v. 1, n. 1, p. 5–11, 2014.

MEDRANO-CERDA, G. A.; BOWLER, C. J.; CALDWELL, D. G. Adaptive position control of antagonistic pneumatic muscle actuators. In: IEEE. *Intelligent Robots and Systems 95.'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*. [S.l.], 1995. v. 1, p. 378–383.

MORONE, G.; PAOLUCCI, S.; CHERUBINI, A.; ANGELIS, D. D.; VENTURIERO, V.; COIRO, P.; IOSA, M. Robot-assisted gait training for stroke patients: current state of the art and perspectives of robotics. *Neuropsychiatric disease and treatment*, Dove Press, v. 13, p. 1303, 2017.

NG, K. L.; GIBSON, E. M.; HUBBARD, R.; YANG, J.; CAFFO, B.; O?BRIEN, R. J.; KRAKAUER, J. W.; ZEILER, S. R. Fluoxetine maintains a state of heightened responsiveness to motor training early after stroke in a mouse model. *Stroke*, Am Heart Assoc, v. 46, n. 10, p. 2951–2960, 2015.

PAYNTER, H. M. *High pressure fluid-driven tension actuators and method for constructing them*. [S.l.]: Google Patents, 1988. US Patent 4,751,869.

PAYNTER, H. M. *Hyperboloid of revolution fluid-driven tension actuators and method of making*. [S.l.]: Google Patents, 1988. US Patent 4,721,030.

REPPERGER, D.; JOHNSON, K.; PHILIPS, C. Nonlinear feedback controller design of a pneumatic muscle actuator system. In: IEEE. *American Control Conference, 1999. Proceedings of the 1999.* [S.l.], 1999. v. 3, p. 1525–1529.

ROTH, G. A.; FOROUZANFAR, M. H.; MORAN, A. E.; BARBER, R.; NGUYEN, G.; FEIGIN, V. L.; NAGHAVI, M.; MENSAH, G. A.; MURRAY, C. J. Demographic and epidemiologic drivers of global cardiovascular mortality. *New England Journal of Medicine*, Mass Medical Soc, v. 372, n. 14, p. 1333–1341, 2015.

SAKTHIVELU, V.; CHONG, S. H. Motion control of a 1-dof pneumatic muscle actuator positioning system. 2015.

SCAFF, W. *Precisão em posicionamento de manipulador não condutor acionado por músculos artificiais pneumáticos.* 2015.

SCHMID, P. J. Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, Cambridge University Press, v. 656, p. 5–28, 2010.

SCHRÖDER, J.; KAWAMURA, K.; GOCKEL, T.; DILLMANN, R. Improved control of a humanoid arm driven by pneumatic actuators. *Proceedings of Humanoids 2003*, 2003.

SCHULTE, H. J.,"the characteristics of the mckibben artificial muscle," in [the application of external power in prosthetics and orthotics], washington, dc: Nat. *Acad. Sci.-Nat. Res. Council*, 1961.

SENTHILKUMAR, D. Design of robust fuzzy controllers for uncertain nonlinear systems. *Indian Institute of Technology Guwahati*, 2010.

SHEN, X. Nonlinear model-based control of pneumatic artificial muscle servo systems. *Control Engineering Practice*, Elsevier, v. 18, n. 3, p. 311–317, 2010.

SITUM, Z.; HERCEG, S. Design and control of a manipulator arm driven by pneumatic muscle actuators. In: IEEE. *Control and Automation, 2008 16th Mediterranean Conference on.* [S.l.], 2008. p. 926–931.

STEWART, A. M.; PRETTY, C. G.; ADAMS, M.; CHEN, X. Hybrid exoskeletons for upper limb stroke rehabilitation. In: *Handbook of Research on Biomimetics and Biomedical Robotics.* [S.l.]: IGI Global, 2018. p. 76–98.

STOIANOVICI, D.; KIM, C.; SRIMATHVEERAVALLI, G.; SEBRECHT, P.; PETRISOR, D.; COLEMAN, J.; SOLOMON, S. B.; HRICAK, H. Mri-safe robot for endorectal prostate biopsy. *IEEE/ASME Transactions on Mechatronics*, IEEE, v. 19, n. 4, p. 1289–1299, 2014.

SU, H.; IORDACHITA, I. I.; TOKUDA, J.; HATA, N.; LIU, X.; SEIFABADI, R.; XU, S.; WOOD, B.; FISCHER, G. S. Fiber-optic force sensors for mri-guided interventions and rehabilitation: A review. *IEEE sensors journal*, IEEE, v. 17, n. 7, p. 1952–1963, 2017.

TADANO, K.; AKAI, M.; KADOTA, K.; KAWASHIMA, K. Development of grip amplified glove using bi-articular mechanism with pneumatic artificial rubber muscle. In: IEEE. *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. [S.l.], 2010. p. 2363–2368.

TAKAGI, T.; SAKAGUCHI, Y. *Pneumatic actuator for manipulator*. [S.l.]: Google Patents, 1986. US Patent 4,615,260.

TAKOSOGLU, J. E.; LASKI, P. A.; BLASIAK, S.; BRACHA, G.; PIETRALA, D. Determining the static characteristics of pneumatic muscles. *Measurement and Control*, SAGE Publications Sage UK: London, England, v. 49, n. 2, p. 62–71, 2016.

TAN, M.; CHONG, S.; TANG, T.; SHUKOR, A.; SAKTHIVELU, V.; MA, T.; NAWAWI, M. M. Tracking control of vertical pneumatic artificial muscle system using pid. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, v. 8, n. 7, p. 129–132, 2016.

TANG, T.; CHONG, S.; CHAN, C.; SAKTHIVELU, V. Point-to-point positioning control of a pneumatic muscle actuated system using improved-pid control. In: IEEE. *Automatic Control and Intelligent Systems (I2CACIS), IEEE International Conference on*. [S.l.], 2016. p. 45–50.

THANH, T. D. C.; AHN, K. K. Nonlinear pid control to improve the control performance of 2 axes pneumatic artificial muscle manipulator using neural network. *Mechatronics*, Elsevier, v. 16, n. 9, p. 577–587, 2006.

TONDU, B. What is an artificial muscle? a systemic approach. *Actuators*, v. 4, n. 4, p. 336–352, 2015. ISSN 2076-0825. Disponível em: <http://www.mdpi.com/2076-0825/4/4/336>.

TONDU, B.; EMIRKHANIAN, R.; MATHÉ, S.; RICARD, A. A ph-activated artificial muscle using the mckibben-type braided structure. *Sensors and Actuators A: Physical*, Elsevier, v. 150, n. 1, p. 124–130, 2009.

TONDU, B.; LOPEZ, P. The mckibben muscle and its use in actuating robot-arms showing similarities with human arm behaviour. *Industrial Robot: An International Journal*, MCB UP Ltd, v. 24, n. 6, p. 432–439, 1997.

TONDU, B.; LOPEZ, P. Modeling and control of mckibben artificial muscle robot actuators. *IEEE control systems*, IEEE, v. 20, n. 2, p. 15–38, 2000.

WALCZAK, P.; WOJTKIEWICZ, J.; NOWAKOWSKI, A.; HABICH, A.; HOLAK, P.; XU, J.; ADAMIAK, Z.; CHEHADE, M.; PEARL, M. S.; GAILLOUD, P. et al. Real-time mri for precise and predictable intra-arterial stem cell delivery to the central nervous system. *Journal of Cerebral Blood Flow & Metabolism*, SAGE Publications Sage UK: London, England, v. 37, n. 7, p. 2346–2358, 2017.

WANG, A.; XIAO, H.; WANG, J.; WANG, Z.; ZHAO, W.; MAY, R. G. Self-calibrated interferometric-intensity-based optical fiber sensors. *Journal of lightwave technology*, IEEE, v. 19, n. 10, p. 1495, 2001.

WANG, G.; WERELEY, N. M.; PILLSBURY, T. Non-linear quasi-static model of pneumatic artificial muscle actuators. *Journal of Intelligent Material Systems and Structures*, SAGE Publications Sage UK: London, England, v. 26, n. 5, p. 541–553, 2015.

WHITLEY, D. A genetic algorithm tutorial. *Statistics and computing*, Springer, v. 4, n. 2, p. 65–85, 1994.

WIDMER, M.; HELD, J. P.; WITTMANN, F.; LAMBERCY, O.; LUTZ, K.; LUFT, A. R. Does motivation matter in upper-limb rehabilitation after stroke? armeosenso-reward: study protocol for a randomized controlled trial. *Trials*, BioMed Central, v. 18, n. 1, p. 580, 2017.

WU, J.; HUANG, J.; WANG, Y.; XING, K.; XU, Q. Fuzzy pid control of a wearable rehabilitation robotic hand driven by pneumatic muscles. In: IEEE. *Micro-NanoMechatronics and Human Science, 2009. MHS 2009. International Symposium on.* [S.l.], 2009. p. 408–413.

WU, Z.; MITTAL, S.; KISH, K.; YU, Y.; HU, J.; HAACKE, E. M. Identification of calcification with mri using susceptibility-weighted imaging: a case study. *Journal of Magnetic Resonance Imaging*, Wiley Online Library, v. 29, n. 1, p. 177–182, 2009.

XIE, S.; MEI, J.; LIU, H.; WANG, Y. Hysteresis modeling and trajectory tracking control of the pneumatic muscle actuator using modified prandtl–ishlinskii model. *Mechanism and Machine Theory*, Elsevier, v. 120, p. 213–224, 2018.

YARLOTT, J. M. *Fluid actuator.* [S.l.]: Google Patents, 1972. US Patent 3,645,173.

YPMA, T. J. Historical development of the newton–raphson method. *SIAM review*, SIAM, v. 37, n. 4, p. 531–551, 1995.

ZHOU, Z.; GRAVER, T. W.; HSU, L.; OU, J.-p. Techniques of advanced fbg sensors: Fabrication, demodulation, encapsulation, and their application in the structural health monitoring of bridges. *Pacific Science Review*, v. 5, n. 1, p. 116–121, 2003.