ROBUST ENERGY-EFFICIENT DESIGN OF SERIES ELASTIC ACTUATORS

by

Edgar A. Bolívar Nieto

APPROVED BY SUPERVISORY COMMITTEE:

Robert D. Gregg, Chair

Tomoki Ohsawa

Mark Spong

Tyler Summers

A la memoria y berraquera de mi padre, quien admiro y extraño constantemente.
A mi madre, mi hermano y a la abuelita Elvia, ¡muchas gracias por tanto cariño!
&
A Parul Rai, quien me motiva a hacer esto una realidad.

ROBUST ENERGY-EFFICIENT DESIGN OF SERIES ELASTIC ACTUATORS

by

EDGAR A. BOLÍVAR NIETO, BS, MS

DISSERTATION

Presented to the Faculty of

The University of Texas at Dallas

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY IN

MECHANICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT DALLAS

August 2019

ACKNOWLEDGMENTS

In 2009, after four years of undergraduate studies, Dr. Jaime Castillo ignited my interest in the field of robotics. It took me four years as an undergrad to find in robotics my professional passion; the quality and energy of Jaime's class was a fundamental factor in this journey. The following semester, Dr. Jorge Sofrony taught the basics of robot control; the book of Dr. Spong became the standard, to the point that I considered its authors as professional stars. I never dreamed that I would have Dr. Spong as a Committee Member of my PhD dissertation; I did not even want to pursue a PhD I will be always in debt to La Universidad Nacional de Colombia, its professors, and amazing classmates, this mix created the perfect environment to fulfill my purpose without the financial burden of a private school.

In 2010, Dr. Yaoyu Li at The University of Wisconsin-Milwaukee opened the door to my first experience abroad. Dr. Li, your support was a vital portion of this process. I am extremely thankful for the opportunity to have joined your lab and also for your later connections with The University of Texas at Dallas (UTD). Personal reasons wanted me to stay in Colombia, but after three years I decided to pursue a PhD in the U.S. In early 2014, after my admission to UTD, I met Bobby for the first time through a Skype interview; a one-hour conversation that pinpointed the key ideas that would become the object of my research and the initial structure of this dissertation. Bobby, thanks to you and the LocoLab I am convinced that pursuing a PhD was the best decision I could ever make. Thanks to your efforts I felt that the only bottleneck in the process was me; all the mentorship, resources, and professional support were there for me to aim high and accomplish our goals. Siavash, your mentorship gave me the structure I needed to focus my efforts, your passion for science will be always an example for me. Dario, David Q., Ge, Leon, David A., Kyle, Toby, Saurav, Nikhil, Jon, Mark, Taha, Hasan, Anne, Ali, Akshay, Aaron, Jorge, Timothy, Kaveh, Asif, Lekan, Pablo, Gregory, and Jairo, thank you for all the meaningful conversations and the effort to make great science. Dr. Summers, Dr. Ohsawa, Dr. Spong, Dr. Rouse, Dr. Rollinson (Dave), Dr. Ruths, Dr. Yurkovich, Dr. Gans, Dr. Baughman, and Dr. Voit your classes, access to your labs or companies have shaped my vision for science and the content of this dissertation.

In October 2015, my role model, my father unexpectedly passed away. His attitude towards life and positive energy made me the man I am today. I am beyond words to express my gratitude for my family. It was a terrible time, no way I could have been prepared for that moment, but your love and support carried me through. I am proud and privileged for our accomplishments and strength, I know my father would have been proud of us too. Mi Piel

Canela, you complement me and gave me the happiness and support to accomplish this step in our life. ¡Gracias!

This dissertation has been made possible thanks to the funding of the National Science Foundation under award number: 1830360, "NRI: FND: COLLAB: A Robust Convex Optimization Framework for the Design of Ubiquitous Series Elastic Actuators", and the NIH National Institute of Child Health & Human Development of the NIH under Award Number DP2HD080349.

June 2019

ROBUST ENERGY-EFFICIENT DESIGN OF SERIES ELASTIC ACTUATORS

Edgar A. Bolívar Nieto, PhD
The University of Texas at Dallas, 2019

Supervising Professor: Robert D. Gregg, Chair

In the same manner that muscles benefit from the elasticity of tendons to be more energy efficient and powerful, electric motors can benefit from elastic elements to improve the efficiency and reliability of robots. For more than twenty years, roboticists have purposefully used springs to connect the motor and the moving joint aiming for efficient and reliable motion. However, there is no clear understanding of how to select the torque-elongation profile of the spring to minimize the motor energy consumption while satisfying motor speed-torque limitations and maximum elongation of the spring. Existing methods cannot guarantee a torque-elongation profile that is better than any other when analyzing arbitrary periodic motion of the joint, and cannot guarantee that the speed-torque of the motor and elongation of the spring are within safe limits when the parameters of the motor, tolerance in manufacturing of the spring, and definition of this periodic motion are uncertain. This dissertation addresses this issue by formulating the design of the torque-elongation spring profile as a robust convex optimization program. The resulting spring profile is guaranteed to be the best among all increasingly monotonic profiles, linear or nonlinear, and satisfies motor and spring constraints as long as the uncertainty stays within the uncertainty sets. As a case study, this dissertation applies the proposed formulation to the design of a series elastic actuator of a powered prosthetic ankle.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION AND PRELIMINARIES

Robots have become ubiquitous in controlled industrial environments, but struggle to interact in a less structured world. One of the biggest challenges in robotics is to design systems capable of interacting with humans, which requires robots to operate in uncertain environments (Yang et al., 2018). This challenge to design robots that cooperate with people (co-robots) defies traditional practices in software and hardware. Co-robots require unprecedented adaptability and performance in order to operate in uncertain environments where the safety of people and robots is of paramount importance (Bauer et al., 2008; Bicchi and Tonietti, 2004). In addition, collaboration in remote environments may require robots untethered from a power source. Thus, battery life becomes important, as it limits the time during which the robot is able to operate (Yang et al., 2018). From the hardware perspective, elasticity has the potential to improve battery life, safety, and reliability of co-robots.

In general, many components of the robot could be elastic, for example its links and joints; one popular strategy uses elasticity at the link to connect the load and a rigid actuator through an elastic element. This concept, pioneered by (Pratt and Williamson, 1995), is known as series elastic actuators (SEAs); Fig. 1.1 illustrates a diagram and a commercial product based on SEAs. In general, the rigid actuator could be an electric, hydraulic, or pneumatic actuator; in this dissertation, the analysis considers only electric motors. Examples of co-robots using SEAs include powered prosthetic legs (Hollander et al., 2006; Rouse et al., 2014), as well as humanoid robots (Hurst and Rizzi, 2008). In traditional SEAs, the motor is connected to a high-ratio linear transmission. Then an elastic element connects the transmission's output to the load (Paine et al., 2014). Designs with a low ratio transmission are less common, but still possible due to the increasing supply of high torque motors (Hirzinger et al., 2002; Schutz et al., 2016).

The architecture of SEAs offers important benefits to the actuation of co-robots. SEAs favor the implementation of control strategies that promote safe mechanical interaction. Impedance control is a renowned control framework for compliant robot interaction (Hogan, 1985), but it requires high-fidelity force control, which is challenging for highly geared, rigid actuators. To remain stable, these actuators often implement low gains in force control, resulting in poor force tracking, even at low frequencies, due to backlash and friction of the transmission (Robinson, 2000). In contrast, force control for SEAs can be executed by controlling the deflection of the elastic element (Robinson et al., 1999). This essentially

Figure 1.1. Illustration of an SEA (left). Commercial SEA, HEBI Robotics (right). The load connects to the motor through an elastic element; the motor position is different from the position of the load.

changes the force control problem (which typically must use low gains) to a position control problem with high gains, thereby reducing the force tracking error (Pratt and Williamson, 1995). Thus, improved force and impedance control fidelity can be obtained through the implementation of series elasticity at the cost of a limited reduction in force bandwidth (Pratt and Williamson, 1995).

In addition, springs can store elastic energy and release it with enormous power, which can be exploited in SEA-based designs to reduce speed-torque requirements and potentially the mass of the motor. The power per unit of mass of springs is orders of magnitude higher than current electromagnetic actuators (Hurst and Rizzi, 2008). Therefore, in the ideal case where a spring is sufficient to achieve the motion task, an SEA has the potential to be much lighter than a rigid actuator. Even when positive work is required by a motor, series elasticity can reduce the peak power requirements of the motor, allowing the use of a less powerful and hence lighter motor (Hollander et al., 2006). Researchers have used this principle to reduce actuator mass sevenfold in a powered ankle orthosis (Hollander et al., 2006). However, reducing the motor mass may not reduce total actuator mass due to the extra components required to include the elastic element. Therefore, decreasing total actuator mass depends on the mechanical implementation of the spring.

The elastic element in an SEA decouples the reflected inertia of the rigid actuator and the inertia of the load (Hurst and Rizzi, 2008), which improves safety during collisions. Safe collisions with the environment and co-workers are fundamental to the design of co-robots (Bicchi and Tonietti, 2004). Rigid inelastic collisions have large peak forces, which can be a safety hazard for co-workers and can reduce the operating life of some robot components,

e.g., transmission and load bearing elements (Bicchi and Tonietti, 2004). SEAs are a popular solution for reducing impulsive forces through elastic collisions, thereby protecting the actuators from impacts (Pratt and Williamson, 1995). For example, the SEA-based prosthetic ankle in (Au and Herr, 2008) reduced impact forces with the ground by more than 29%.

Series elasticity can increase battery life by reducing energy consumption of co-robots (Hurst and Rizzi, 2008; Rouse et al., 2014; Ham et al., 2009; Realmuto et al., 2015). Energy in a robotic application is either transferred to/from the load or dissipated. From the energy consumption perspective, an elastic element connected in series is passive and cannot reduce the energy required by the load, but it has the potential to reduce the energy dissipated for a given task (Bolívar et al., 2017). Reduction of dissipated energy is important, especially in tasks that are mainly dissipative such as level-ground locomotion. For example, in the case of the MIT cheetah robot (Seok et al., 2015), 76% of total energy consumption is dissipated by Joule heating of the motor windings, and the remaining 24% is dissipated by inelastic collisions and other mechanical losses. Another example is the ankle of a 75 kg human, which provides about 17 J per stride during normal walking, but a rigidly actuated prosthetic ankle consumes 33 J (Bolívar et al., 2017). The extra 16 J is dissipated through electrical and mechanical losses. The same motor connected in series with an elastic element requires about 25 J per stride (Bolívar et al., 2017), i.e., a 50% reduction in the energy dissipated. In the case of the second-generation UT-Dallas prosthetic leg (Elery et al., 2018), series elasticity at the ankle joint can extend its battery life by 33% (Bolívar et al., 2017).

To summarize, co-robots can benefit from SEAs in order to reduce the energy lost during impacts (Hurst and Rizzi, 2008), improve force control for human-robot interaction (Robinson et al., 1999), improve the safety of the human and robots (Bicchi and Tonietti, 2004), move loads with higher velocities (Braun et al., 2013), reduce energy consumption of the system (Vanderborght et al., 2006; Jafari et al., 2013; Ding and Park, 2017), and decrease peak motor power (Jafari et al., 2011, 2013; Hollander et al., 2006), so a smaller/lighter motor can be used. All these benefits are subject to the force-elongation profile of the SEA's elastic element.

Despite these advantages, the performance of an SEA system is sensitive to deviations from its nominal operating conditions. During unexpected situations, SEAs can consume more energy and/or draw higher peak power than rigid actuators (Grimmer and Seyfarth, 2011). The resulting overheating and battery drain is unacceptable for many applications, especially wearable robots. Unexpectedly demanding tasks can also cause the spring to reach its maximum elongation or bottom out, resulting in a loss of compliance and possible damage to the robot and/or an interacting human. During the development of the ATRIAS biped

robot (Hubicki et al., 2018), the most efficient choice of spring reached maximum deflection at fast walking speeds. To prevent mechanical damage, stiffer springs had to be used at the cost of efficiency. Conversely, the maximum deflection was the main consideration in the choice of spring stiffness for the SEAs of NASA's humanoid robot, Valkyrie (Paine et al., 2015). This approach guarantees safety but neglects the potential for SEAs to reduce energy consumption.

The potential of SEAs to reduce energy consumption and create safer co-robots depends on the design of the elastic element, specifically its force-elongation profile. As previously discussed, uncertainty in the operational conditions of the actuator impacts the actuator constraints, e.g., speed-torque requirements of the motor and maximum elongation of the spring, depending on the force-elongation profile of the spring. To this date, the design of the spring stiffness relies on very limited frequency analysis or pure optimization methods that lack the fundamental understanding of the benefits of elasticity. As a result, prior to this dissertation, no reliable method was available for the design of the force-elongation profile (linear or nonlinear) of series springs that is robust to task and actuator uncertainty.

Notation

The following notation applies to the mathematical expressions in this dissertation. The set of real numbers is denoted by $\mathbb{R}$, the non-negative real numbers as $\mathbb{R}_+$, and the positive numbers $\mathbb{R}_{++}$. Scalar numbers in $\mathbb{R}$ are denoted with italics, vectors in bold lower-case characters, and matrices with bold upper-case characters. For example, a, b are scalars; $\boldsymbol{a}, \boldsymbol{b}$ are vectors; and $\boldsymbol{A}, \boldsymbol{B}$ are matrices. The vector $\mathbf{1}$ is an m dimensional vector with ones along its entries. The dimension m is implied by context in the equations, unless explicitly stated as $\mathbf{1}_{m\times 1}$. Vector inequalities such as $\boldsymbol{a} \leq \mathbf{1}$ for $\boldsymbol{a} \in \mathbb{R}^n$ apply element-wise, i.e., $a_i \leq 1$ for $i = 1, \ldots, n$. The matrix inequality $\boldsymbol{A} \geq \mathbf{0}$ indicates that the square matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is positive semi-definite, i.e., $\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} \geq 0,\ \forall \boldsymbol{x} \in \mathbb{R}^n$.

1.1 Introduction to energy consumption of SEAs

SEAs are mechatronic devices that transduce electrical energy into mechanical and vice versa. From the energy perspective, they are similar to traditional electric motors; however, their capability to store and release elastic energy creates an additional opportunity to reduce the energy consumption and peak power of their electric motors. An SEA for mobile applications typically consists of a battery, electronic motor drives, electric motor, gear train, and an

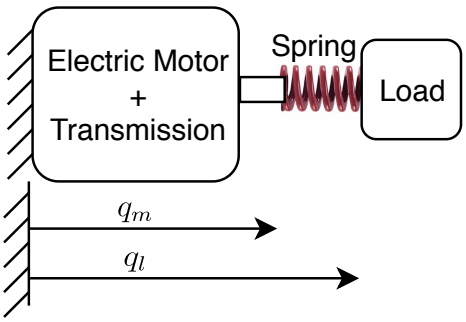

Figure 1.2. Diagram of an SEA. Equations (1.1)-(1.2) illustrate the system's equations of motion.

elastic element. In practice, each of these components is capable of dissipating energy. For example, the battery self-discharges, the motor drive produces Joule heat, friction in the transmission generates heat, and the elastic elements are not purely elastic (i.e., dissipate energy through their viscous behavior). This dissertation focuses on the energy consumed by the motor since it is the largest consumption in the system. In addition, the energy consumed by other elements of the SEA may be lumped in the expressions of energy of the motor. For instance, viscous friction at the transmission can be considered as additional motor viscous friction.

For a DC-motor, estimations of its energy consumption highly depend on the quality of its dynamic model. Some models neglect the effects of motor inertia, viscous friction, and the regeneration capabilities of modern electronic motor drives (Hollander et al., 2006; Grimmer and Seyfarth, 2011). Neglecting these terms may be sufficient for estimating energy consumption or motor power at constant speed or steady state, but it is inaccurate for more dynamic motions (Verstraten et al., 2015). This dissertation targets applications with significant changes in SEA rotational speed; thus, the estimation of energy consumption considers the torques due to inertia, viscous friction, and power flowing to and from the battery. The work of Verstraten et al. presents an experimental validation of the dynamic model used in this dissertation (Verstraten et al., 2015).

1.1.1 Equations of motion of an SEA

The Fig. 1.2 illustrates the configuration of an SEA. Using the Newton-Euler method, the corresponding balance of torques at the motor and load side provides the following equations

of motion

$$I_m\ddot{q}_m = -b_m\dot{q}_m + \tau_m + \frac{\tau_s}{\eta r}, \tag{1.1}$$

$$\tau_s = g(q_l, \dot{q}_l, \ddot{q}_l, \tau_e), \tag{1.2}$$

where $I_m \in \mathbb{R}_{++}$ is the inertia of the motor, $b_m \in \mathbb{R}_{++}$ its viscous friction coefficient, $\tau_s \in \mathbb{R}$ the torque produced by the spring, $r \in \mathbb{R}_{++}$ the transmission ratio, $\eta \in \{x \in \mathbb{R} : 0 < x \leq 1\}$ the efficiency of the transmission, τ_m is the motor torque, and $g : \mathbb{R}^4 \to \mathbb{R}$ defines the load dynamics mapping the load position $q_l \in \mathbb{R}$, load velocity $\dot{q}_l \in \mathbb{R}$, load acceleration $\ddot{q}_l \in \mathbb{R}$, and the external torque applied to the load $\tau_e \in \mathbb{R}$ to the torque produced by the spring. (Spong, 1987; Verstraten et al., 2015, 2017). For instance, in the case of an inertial load with viscous friction and an external torque, the load dynamics are defined by $g(q_l, \dot{q}_l, \ddot{q}_l, \tau_e) = -I_l\ddot{q}_l - b_l\dot{q}_l + \tau_e$, where I_l is the inertia of the load, and b_l its corresponding viscous friction coefficient. The variables $q_l, \dot{q}_l, \ddot{q}_l, \tau_e$ denote the *load trajectory* in this dissertation. The elongation of the elastic element is defined as $\delta = q_l - q_m/r$. As seen in (1.1)-(1.2), the elastic element is in series with the load and cannot modify the torque required to perform the motion, τ_s, but it decouples the position of the motor and the load such that inertial and viscous friction torques, $I_m\ddot{q}_m + b_m\dot{q}_m$, could reduce the torque of the motor, τ_m. This dynamic model can be expanded to include multiples degrees of freedom (DoF) as in (Spong, 1987), which will imply modifications to the function $g(\cdot)$ to include the coupling with other links in a kinematic chain. Without loss of generality, this dissertation considers only the single DoF formulation in (1.1)-(1.2), as the results extrapolate easily to multiple DoF with adequate definitions of the function $g(\cdot)$.

1.1.2 Energy flow in an SEA

This section introduces the energy flow of SEAs to have a clearer insight into the role of series elasticity in motor energy consumption. This dissertation focuses on SEAs powered by a battery (or any power source that can provide and receive power) and an electric motor, a typical scenario for portable devices such as wearable robots (Hurst and Rizzi, 2008). The Fig. 1.3 illustrates the corresponding energy flow and main components of this arrangement.

Energy flow in an electric motor occurs in two principal modes of operation: actuator and generator mode. As an actuator, the electric motor receives electrical energy from the battery through the driver and converts it into mechanical energy and heat. Traditional motor drives and mechanical transmissions are designed so that the electric motor can always operate in

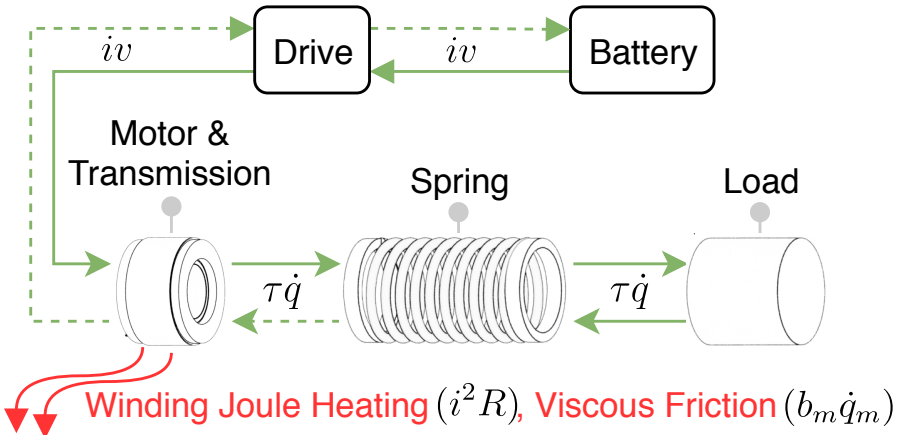

Figure 1.3. Energy flow of an SEA: Dashed lines indicate that the energy path may or may not exist depending on the construction of the device. For instance, energy flowing from the drive to the battery requires drivers capable of regeneration. Energy flowing from the load to the electric motor requires that the load is high enough to backdrive the motor-transmission system.

this mode. A more interesting scenario occurs when the motor can also work as a generator. For example, when the motor is decelerating a load, the kinetic energy of the load and elastic energy of the spring is transferred to the motor's rotor to store it as electrical energy in the battery.

However, traditional SEAs may not function in generator mode. These designs typically have linear transmissions with high reduction ratios. The reflected inertia of the motor after the transmission, which is proportional to the reduction ratio squared, is normally very high compared to the load. For example, three recent SEA designs reflect output inertia of 360 kg, 270 kg, and 294 kg for the UT-SEA (Paine et al., 2014), Valkyrie's SEA (Zhao et al., 2015), and THOR-SEA (Knabe et al., 2014) respectively, as indicated by (Schutz et al., 2016). As a consequence, the system requires a high load to backdrive. An additional limitation is the motor driver. In order to regenerate energy, motor drivers should be selected such that the electrical energy recovered from the motion of the rotor can flow back to charge the battery (Seok et al., 2015).

The results in this dissertation assume the SEA has been designed such that energy can flow from the load to the energy source and vice versa. In other words, the load is high enough to backdrive the motor, and adequate electronics allow energy to flow to and from

the battery. In this case, the energy consumption of the motor, $E_m \in \mathbb{R}$, is given by

$$E_m = \int_{t_0}^{t_f} \Big(\underbrace{\frac{\tau_m^2}{k_m^2}}_{\substack{\text{Winding} \\ \text{Joule} \\ \text{heating}}} + \underbrace{\tau_m \dot{q}_m}_{\substack{\text{Rotor} \\ \text{mechanical} \\ \text{power}}} \Big) dt, \tag{1.3}$$

where t_0 and t_f are the initial and final times of the trajectory respectively, and $k_m \in \mathbb{R}_{++}$ is the motor constant. Notice that the energy associated with Joule heating can be also written as $i_m^2 R$, since $\tau_m = i_m k_t$ and $k_m = k_t/\sqrt{R}$, where $i_m \in \mathbb{R}$ is the electric current flowing through the motor, $R \in \mathbb{R}_{++}$ the motor terminal resistance, and $k_t \in \mathbb{R}_{++}$ the motor torque constant (Verstraten et al., 2015).

Alternative Expressions of Motor Energy Consumption

Some authors express energy consumption as (Hollander et al., 2006; Grimmer and Seyfarth, 2011)

$$E_m = \int_{t_0}^{t_f} |\tau_m \dot{q}_m| dt. \tag{1.4}$$

This representation of energy consumption implies that the motor consumes energy even when the product of $\tau_m \dot{q}_m$ is negative. This situation is unrealistic because it assumes that energy is required from the power source when the motor absorbs energy from the load, i.e., acts as a generator. Even when the power source is not capable to absorb energy, such as in traditional power supplies, the energy regenerated from the motor is dissipated through Joule heating at the motor, motor drive, or a shunt regulator. In that case, the regenerated energy is dissipated, and the energy consumption of the motor is

$$E_m = \int_{t_0}^{t_f} \max\{\tau_m \dot{q}_m, 0\} dt. \tag{1.5}$$

Note that (1.4) and (1.5) only account for the mechanical energy at the rotor of the motor. A more accurate representation includes the winding Joule heating as in (1.3) (Rezazadeh and Hurst, 2014; Verstraten et al., 2015). As discussed previously, winding losses through Joule heating are significant for DC-Motors (Seok et al., 2015).

1.1.3 Effects of periodic motion on energy consumption

Periodic motion plays an important role in this dissertation; it will be fundamental to show convexity of the optimization program that defines the optimal force-elongation profile of the

spring. In this dissertation, periodic motion refers to load trajectories where $q_l, \dot{q}_l, \ddot{q}_l$, and τ_e are the same at the initial, t_0, and final time, t_f. This definition is flexible enough to include trajectories in lower limb biomechanics such as the load trajectories of the ankle during walking, stair climbing, sit-to-stand and stand-to-sit, etc. In fact, one can argue that most of the motion in co-robots can be decomposed in a series of periodic trajectories. Considering periodic motion simplifies the expression of energy consumption in (1.3) as follows. Using the expression of motor torque, τ_m, from (1.1) into the expressions of energy (1.3) leads to

$$\begin{aligned}
\int_{t_0}^{t_f} \tau_m \dot{q}_m dt &= \int_{t_0}^{t_f} \left(I_m \ddot{q}_m + b_m \dot{q}_m - \frac{\tau_s}{\eta r} \right) \dot{q}_m dt, \\
&= \int_{t_0}^{t_f} \left(b_m \dot{q}_m^2 - \frac{\tau_s \dot{q}_m}{\eta r} \right) dt + \cancelto{0}{\int_{\dot{q}_{m_0}}^{\dot{q}_{m_f}} I_m \dot{q}_m d\dot{q}_m}, \\
&= \int_{t_0}^{t_f} \left(b_m \dot{q}_m^2 - \frac{\tau_s}{\eta r} (\dot{q}_m - r\dot{q}_l + r\dot{q}_l) \right) dt, \\
&= \int_{t_0}^{t_f} \left(b_m \dot{q}_m^2 - \frac{\tau_s \dot{q}_l}{\eta} \right) dt + \cancelto{0}{\int_{\delta_0}^{\delta_f} \frac{\tau_s}{\eta} d\delta}, \\
&= \int_{t_0}^{t_f} \left(b_m \dot{q}_m^2 - \frac{\tau_s \dot{q}_l}{\eta} \right) dt, \qquad (1.6)
\end{aligned}$$

where $\dot{q}_{m_f} = \dot{q}_{m_0}$ and $\delta_f = \delta_0$ due to periodic motion. This simplification illustrates an important concept. For periodic motion, the kinetic energy due to inertia of the motor's rotor is equal to zero; as explained in the next chapter, eliminating this term will enable us to show convexity of the optimization program.

How Does Series Elasticity Reduce Energy Consumption in an SEA?

Periodic motion and its corresponding expression for energy consumption of the rotor (1.6) illustrates a fundamental principle for the energy consumption of SEAs: the mechanical energy provided to or absorbed from the motion of the load (i.e., $\int_{t_0}^{t_f} \frac{\tau_s \dot{q}_l}{\eta} dt$) is provided or absorbed by the electric motor regardless of the force-elongation profile of the elastic element. After all, the spring is an energetically passive element; it does not provide or dissipates energy at the end of a periodic cycle. Replacing (1.6) into (1.3) yields

$$E_m = \int_{t_0}^{t_f} \left(\frac{\tau_m^2}{k_m^2} + b_m \dot{q}_m^2 - \frac{\tau_s \dot{q}_l}{\eta} \right) dt,$$

meaning that the energy of an SEA categorizes in three terms: winding Joule heating, viscous friction, and the energy required from the load (which is independent of the force-elongation

profile of the spring). Thus, series elasticity only reduces the energy dissipated by the motor. This is relevant for tasks that are mainly dissipative such as level-ground locomotion (Seok et al., 2015). In practice, the dissipated energy through Joule heating at the motor winding is significantly higher than other losses in the motor (Seok et al., 2015; Verstraten et al., 2015). For a given motor, Joule heating is proportional to the square of motor torque. Unfortunately, the influence of series elasticity into motor torque may be insignificant, as it can only modify torque through the inertial and viscous friction torques at the shaft. The opportunity to reduce torque increases for systems with a high reduction ratio, where inertial and viscous friction torques from the motor could be significant. Thus, the reduction of energy consumption using a series elastic element highly depends on the selection of the motor, reduction ratio of the transmission, and the load trajectories.

1.2 Series elasticity to modify actuator constraints

Series elasticity not only modifies the energy consumption of the electric motor, it also modifies the motor speed and torque requirements to accomplish a specific tasks. This section discusses the effect of series elasticity into the actuator constraints, which is a key element in the elastic element design.

1.2.1 Actuator constraints for SEAs

In this dissertation, the performance of SEAs is limited by the following three constraints: maximum elongation of the spring, speed-torque relationship for DC-motors, and maximum torque for DC motors. The maximum elongation of the spring, $\delta_{\max}$, is fully defined by the torque of the load, $\tau_l = -\tau_s$, and the torque-elongation profile of the spring. The constraint due to maximum elongation of the spring states that $\delta \leq \delta_{\max}$ during the whole period of operation. Limitations in spring elongation due to mechanical implementation of the elastic element are typical; adequate selection of the torque-elongation profile is important to guarantee that the spring remains elastic and does not reach its maximum elongation or bottom out during operation. For example, the rotational spring in the open source prosthetic leg at University of Michigan has a maximum elongation of 15° (Azocar et al., 2018).

The constraint due to the speed-torque relationship of DC motors can be understood from its electrical model. The Fig. (1.4) illustrates a diagram of this model. The electric circuit behaves based on the following differential equation:

$$v_s = i_m R + L\frac{di_m}{dt} + v_{\text{emf}}, \tag{1.7}$$

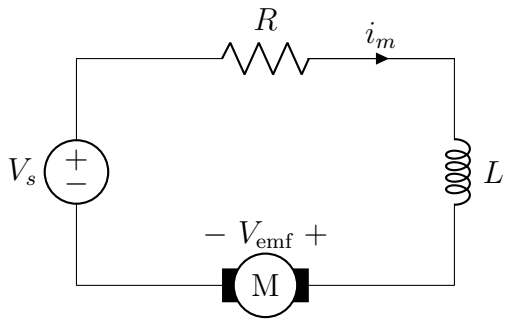

Figure 1.4. Electric diagram of a DC-Motor. The series resistance and inductance represent the electrical effects of the motor winding. The electromotive force voltage is proportional to the rotational speed, $v_{\text{emf}} = k_t \dot{q}_m$. Each phase of a permanent magnet brushless DC motor is also represented by this diagram.

where L is the inductance of the motor winding; v_{emf} is the electromotive force voltage, which is proportional to the rotational speed of the motor, $v_{\text{emf}} = k_t \dot{q}_m$; and v_s is the voltage of the power supply or battery. An advantage for the modeling of DC motors is that the motor torque and current are related by $\tau_m = k_t i_m$. Replacing this relationship and $v_{\text{emf}} = k_t \dot{q}_m$ into (1.7) yields

$$\tau_m R + L \frac{d\tau_m}{dt} = v_s k_t - k_t^2 \dot{q}_m.$$

Neglecting the inductive effects on the winding or assuming steady state operation at almost constant speed, i.e., v_{emf} is constant and therefore $\frac{di_m}{dt} \approx 0$, yields the following speed-torque relationship of DC-Motors:

$$\tau_m = v_s \frac{k_t}{R} - \dot{q}_m \frac{k_t^2}{R}. \tag{1.8}$$

For a given voltage input, this equation states that the maximum torque of the motor occurs at stall, when $\dot{q}_m = 0$. During that condition the electromotive voltage is zero and the current going through the winding is maximum. In other words, the electromotive voltage reduces the current that can flow through the winding for a given input voltage; this is due to the fact that during rotation the system simultaneously operates as a motor and as a generator. The equation also states that the maximum speed of the motor occurs when it rotates with no load, i.e., $\tau_m = 0$. Replacing the equality in (1.8) with "$\leq$" illustrate the feasible range of motor torques for a given rotational speed and voltage source. The affine relationship between torque and speed gives rise to the famous diamond shape that defines the feasible space in the speed-torque space, Fig.(1.5).

In practice, the maximum torque, τ_{mm}, is well below the stall torque in (1.8). As current increases, winding heat increases as well; therefore, the actual limit in torque is given by the

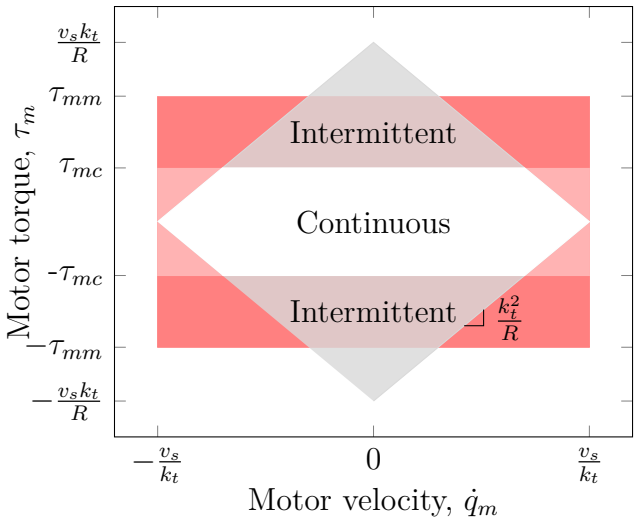

Figure 1.5. Speed-torque relationship of DC motors. The diamond shape represents the set of feasible speed and torque configurations from (1.8).The constant term $\frac{k_t^2}{R}$ defines the slope that connects the maximum theoretical speed and torque of the motor. The maximum torque, τ_{mm}, is defined as the maximum torque that can be achieved in intermittent use. This value of torque depends on the current motor winding temperature, time that the maximum torque will be executed, and temperature of the environment or thermal isolation of the motor. Manufacturers typically provide the maximum continuous torque, τ_{mc}, as the maximum torque that can be applied during continuous operation. The white region in the velocity-torque space defines the continuous operational range of the motor.

current that is required to raise the winding temperature at its maximum admissible level. Predicting the current limits according to winding temperature requires thermal models and preferably thermal sensors in the winding; however, manufacturers often prescribe maximum levels of current for continuous operation. This current defines the maximum continuous torque of the electric motor, τ_{mc}.

1.2.2 Effects of series elasticity on motor torque and speed

A very important and sometimes underrated application of series elasticity is its potential to modify speed-torque requirements for the electric motor. Reducing speed and torque requirements of the motor has been indirectly addressed in the literature when using series springs to reduce motor peak power, which is typically used to indicate motor performance. However, summarizing motor performance only by its power capabilities could be misleading; a feasible region in the speed-torque space defines more precisely the actual performance of DC-motors, as seen in Fig. (1.5). Thus, a more accurate design should select the spring

that guarantees that the motor speed and torque are within admissible levels. This could be imposed as a constraint in an optimization-based design, as motor speed and torque depend on the torque-elongation profile of the spring. Typically, the literature does not explore this potential benefit from series elasticity, although, satisfying speed-torque requirements of the motor could be even more important than reducing energy consumption or peak power. It defines whether or not a trajectory could be accomplished with a given actuator.

1.3 State of the art

Optimal design of series elastic elements has been an active subject of research since the early 2000s. There is a variety of design methods to date, they fit mostly into two categories: natural dynamics and parametric optimization. The following sections illustrate the principles of each methodology and finishes with a discussion on their implementation.

1.3.1 Natural dynamics

Natural dynamics is one of the first attempts to analytically find the energy-optimal force-elongation profile of the spring. If the load is considered to be inertial, the load-spring-motor system will oscillate with a natural frequency after the spring elongates and the motor remains stationary. Thus, it is possible to move the load while keeping the motor stationary. The main concept of natural dynamics is to match the natural oscillation of the load-spring-motor system as closely as possible with the required motion of the load. In this way, the spring minimizes the motion of the motor and therefore its energy consumption. For example, elastic components have been designed to match the natural dynamics of humanoid robots during walking (Vanderborght et al., 2006), cutting tool heads (Grioli et al., 2015), or sinusoidal motion of an inertial load (Verstraten et al., 2016).

Natural Dynamics Example: Sinusoidal Load Trajectory

The Fig. 1.6 illustrates a single robotic link actuated by an SEA. The goal of this example is to find the value of spring stiffness, k, that minimizes the motion of the motor. To achieve analytical results, the reference trajectory is limited to follow the sinusoidal motion: $q_l = A\sin(\omega t)$. Following the notation in (1.2) and (1.1), and neglecting the energy losses at the transmission, i.e., $\eta = 1$, the equations of motion are

$$g(q_l, \dot{q}_l, \ddot{q}_l) = -I_l\ddot{q}_l - \cos(q_l)mgl_{cm} - b_l\dot{q}_l \tag{1.9}$$

$$I_m\ddot{q}_m = \tau_m + \frac{g(q_l, \dot{q}_l, \ddot{q}_l)}{r} - b_m\dot{q}_m \tag{1.10}$$

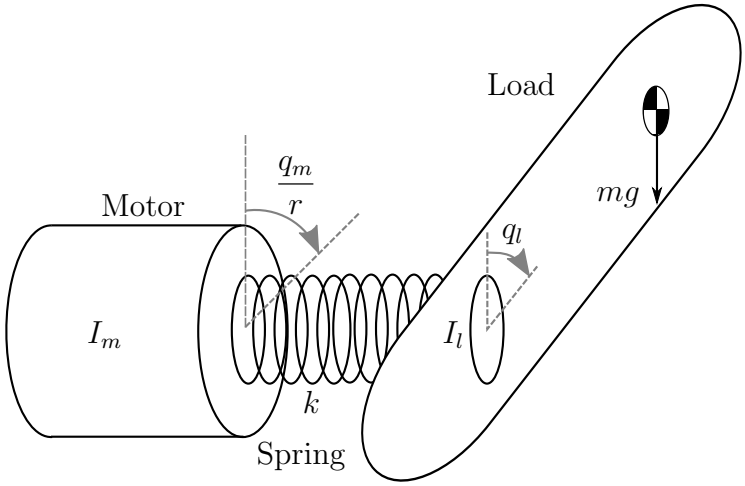

Figure 1.6. Single link actuated by an SEA. The position of the load is designed to follow a sinusoidal trajectory, i.e., $q_l = A\sin(\omega t)$.

where

$$g(q_l, \dot{q}_l, \ddot{q}_l) = k(q_l - \frac{q_m}{r}),$$

is the torque of the spring, I_l is the moment of inertia of the link with respect to its axis of rotation, b_l is the viscous friction coefficient of the load, q_l is the link angular position, m is mass of the link, l_{cm} distance from the axis of rotation to the center of mass of the link, and k is the stiffness of the elastic element. An initial approach to minimize motion is to reduce the velocity of the motor, which is proportional to the mechanical power at the rotor of the motor $p_m = \tau_m \dot{q}_m$. Solving for q_m from (1.9):

$$\begin{aligned}
q_m &= \left(q_l + \frac{I_l \ddot{q}_l + b_l \dot{q}_l - \cos(q_l) m g l_{cm}}{k} \right) r \\
\dot{q}_m &= \left(\dot{q}_l + \frac{I_l \dddot{q}_l + b_l \ddot{q}_l + \sin(q_l) m g l_{cm} \dot{q}_l}{k} \right) r \\
&= r \left(k - I_l \omega^2 + \sin(A\sin(\omega t)) m g l_{cm} \right) k^{-1} A\omega \cos(\omega t) - r b_l A \omega^2 \sin(\omega t) \\
&= \alpha \sin(\omega t + \phi), \qquad (1.11)
\end{aligned}$$

where

$$\begin{aligned}
\alpha &= r k^{-1} \sqrt{(b_l A \omega^2)^2 + (Aw(k - I_l w^2 + \sin(A\sin(\omega t)) m g l_{cm}))^2}, \qquad (1.12) \\
\phi &= \arctan(A\omega(k - I_l w^2 + \sin(A\sin(\omega t)) m g l_{cm} / (-b_l A \omega^2)).
\end{aligned}$$

The amplitude of the motor velocity in (1.12) can be minimized using standard methods from calculus; finding the values of k that guarantee $\frac{\partial \alpha}{\partial k} = 0$. Neglecting gravity terms for simplicity, allows the following optimal value of k:

$$k = \frac{b_l^2}{I_l} + I_l \omega^2. \tag{1.13}$$

We can obtain the value of the peak velocity by replacing the value of k from (1.13) into the equation related to the amplitude of the speed (1.12),

$$\max\{\dot{q}_m\} = \frac{A b_l \omega r}{\sqrt{b_l^2 + I_l^2 \omega^2}}. \tag{1.14}$$

To evaluate velocity reduction of the motor we can check the ratio, r_{vel}, between the peak amplitude of the motor velocity with and without elastic element,

$$r_{vel} = \frac{b_l}{\sqrt{b_l^2 + I_l^2 \omega^2}}. \tag{1.15}$$

This ratio is always less than one. This illustrates that the motor peak velocity using an SEA is always less than the motor peak velocity without elastic element. Torque of the motor for the rigid and elastic actuators is similar; thus peak power decreases when using an SEA. This example, adapted from (Grioli et al., 2015), shows the benefits of series elasticity.

1.3.2 Parametric optimization

Parametric optimization considers arbitrary reference trajectories and actuator constraints (Rouse et al., 2014; Realmuto et al., 2015). In this approach, the elastic element is defined over some parameters, e.g., stiffness of a linear spring, which become variables in a gradient-based optimization to find the elastic element that minimizes a cost function. Linear SEAs (Rouse et al., 2014; Hollander et al., 2006) have been designed using this approach and parallel nonlinear elastic elements as well (Realmuto et al., 2015).

1.3.3 Limitations of the state of the art

Natural dynamics provides the right intuition for the reduction of energy consumption, but it is limited to specific reference trajectories, and it is difficult to generalize its results for arbitrary periodic motion or include actuator constraints. The method normally reduces the motion of the electric motor, but this may not be the most energetically efficient solution. The motor at constant position dissipates energy by Joule heating due to the counteracting

torque required to keep the shaft in its position. A more complete solution will consider motion of the motor by finding the natural dynamics of the two-mass spring system resulting from the inertia of the motor and the load (Bolívar et al., 2017). Another limitation is due to the fact that natural dynamics for nonlinear springs may not have analytical solutions; this complicates the design of nonlinear series elasticity. As discussed in Section 1.2, series elasticity can modify actuator requirements to satisfy its constraints. However, natural dynamics does not provide a structured framework for the design of springs that satisfy actuator constraints.

Parametric optimization allows an explicit consideration of the constraints, but results are limited to the parameter's space and may not be global. For example, the parallel elastic element in (Realmuto et al., 2015) was parameterized by a polynomial function up to order 6 and optimized over its coefficients with respect to a multiobjective function of torque, peak power, and energy consumption. The favored solution was a cubic nonlinear spring, but this result is limited to the chosen parameterization. Moreover, if the problem is not known to be convex, there is no guarantee that a gradient-based optimizer will find a global optimum. Both natural dynamics and parametric optimization provide no guarantee that the optimization results are global.

Existing methods to design the stiffness that minimize energy consumption of SEAs, such as natural dynamics (Verstraten et al., 2016) and optimization formulations (Rouse et al., 2014; Bolívar et al., 2017) assume nominal reference kinematic and kinetic trajectories of the load. These nominal trajectories easily change during operation in human-robot interaction. For example, in the design of an SEA for a powered prosthetic leg, the reference kinematic and kinetic trajectories change as the subject walks with different speeds or wears different accessories, such as a backpack. When the load conditions deviate from their nominal values, the energy consumption and peak power of SEAs may not be optimal (Grimmer and Seyfarth, 2011). Additionally, the speed and torque requirements for the motor may be outside the motor's specifications, i.e., the task becomes infeasible. For instances when changing stiffness of the SEA makes the task feasible again, a possible solution is to replace SEAs with variable stiffness actuators (VSAs) (Grioli et al., 2015). However, VSAs require additional mechanisms to operate, increasing the mechanical complexity and potentially the mass of the actuator. Thus, it is interesting to know if a fixed-stiffness SEA could satisfy the actuator constraints despite uncertainty, and what trade-off such a robust design would have with energy consumption.

Acknowledging uncertainty in the reference trajectories leads to more realistic and robust designs. For example, in (Grimmer and Seyfarth, 2011), the optimal design of series stiffness

considered deviation from the nominal trajectories for the application of powered prosthetic legs. The SEA was optimized over both walking and running reference trajectories, but the design did not consider the wide range of other possible tasks and did not analyze the feasibility of the actuator. Uncertainty in the initial conditions, e.g., position and velocity of the load, during robot-environment interaction motivated a robust optimization design of compliance (Gasparri et al., 2016). The optimal compliance minimized interaction forces between two manipulators accomplishing a handover task. Brown and Ulsoy (Brown and Ulsoy, 2013) considered task uncertainty by defining the reference trajectory as a sample from a probability distribution of reference trajectories. Their stochastic approach to designing linear parallel elastic elements provided energy savings and constraint satisfaction for 90% of the reference trajectories, but worst-case scenarios would violate the strict safety requirements of a co-robot. More arbitrary reference motions resulted in stiffer optimal solutions, converging to a rigid actuator for totally arbitrary motion (Brown and Ulsoy, 2013). However, increasing stiffness may not be the solution when actuator constraints must be satisfied despite uncertainty, as demonstrated later in this dissertation. Thus, a robust formulation is required to guarantee feasibility of the actuator, i.e., the actuator satisfies the speed, torque, and elongation constraints, even in the worst case conditions that could manifest during operation.

1.4 Intellectual merit

The intellectual merit is founded in 1) a robust convex optimization framework for determining series elastic elements that globally minimize energy consumption and satisfy constraints despite uncertainty, which enables 2) a potentially transformative design methodology for SEAs that safely and efficiently perform a variety of tasks for ubiquitous interaction with humans. The results in this dissertation rely on the theory of robust optimization. Current theory on this field applies mainly to convex optimization problems, but the design of series elastic elements has not been proved to be convex before the work included in this dissertation.

Formulating energy consumption of a linear series elastic element as a convex function leads to the necessary conditions to understand when a series elastic element can save energy, and a non-parametric optimization determines the most energy that can possibly be saved by a nonlinear spring. In addition, solvers for convex problems are fast enough to find optimal solutions in real time; expressing the design of series elastic elements as a convex optimization problem benefits devices that could modify stiffness during operation (e.g.,

variable stiffness actuators (VSA)). These innovations in modeling, analysis, and design will enable more versatile, scalable SEAs to be deployed in co-robots that remain efficient and safe (e.g., never reach maximum spring deflection) over a wide range of interactions.

1.5 Organization of the dissertation

The fundamental equations that describe the dynamic behavior and energy consumption of SEAs have been introduced in this preliminary chapter. These dynamic equations are the building blocks to formulate the convex optimization programs for the design of linear and nonlinear torque-elongation spring profiles that minimize energy consumption and peak power in Chapter 2. Chapter 3 presents a robust formulation of these optimization programs after a short introduction to the main concepts in the field of robust optimization. The application of the complete design methodology into the design of a SEA-powered prosthetic ankle is presented in Chapter 4. This chapter also discusses the trade-off between minimization of peak power and energy consumption, along with suggestions for motor and reduction ratio selection for this application. The subsequent chapter concludes the dissertation. Appendix A describes the MATLAB toolbox that contains the programming code to solve the optimization programs from Chapters 2 and 3. It also has information on the open-source repository that describes this toolbox in Code Ocean. The results in Chapter 4 are based on simulation results; appendix B discusses the fabrication of nonlinear springs for an experimental implementation of the proposed strategies.

CHAPTER 2

DESIGN OF ENERGY-EFFICIENT SERIES ELASTICITY [1]

As discussed in Section 1.1.3, SEA-based co-robots can dissipate less energy than a rigid-actuator co-robot. A significant portion of the dissipated energy at the actuator is due to Joule heating, which is for the most part a linear function of actuator torque. Series elasticity can reduce this torque in two forms: 1) modifying motor inertial and viscous friction torques and 2) reducing motor velocity requirements, which will allow the use of a higher transmission ratio and therefore lower motor torque. This approach requires a simultaneous optimization of the spring stiffness and the reduction ratio. This dissertation focuses on the former assuming that the transmission ratio and motor has already been chosen. Specifically, this chapter explains the design of the torque-elongation profile for linear and nonlinear series elasticity to minimize motor energy consumption, i.e., minimize winding Joule heating and viscous friction losses.

This dissertation differentiates between a linear and a nonlinear spring and proposes a unique optimization program for the design of the torque-elongation profile of each. Although the optimal linear spring can be retrieved from the nonlinear methodology, using the appropriate constraints, it is useful to consider the linear case separately. For instance, the linear formulation shows the necessary conditions for the load trajectories and motor parameters to guarantee energy savings when using series elasticity. In addition, the simplicity of the linear formulation enables a faster computation of the optimal spring stiffness; this is convenient for mechanisms that modify stiffness during operation. For the linear and nonlinear cases the optimization program will be shown to be convex.

The chapter concludes with the definition of a convex optimization program for the design of elastic elements that minimize peak power. Minimizing peak power is a standard objective in the literature, but it may be deceiving depending on the design stage. Specifically, if the electric motor and transmission have been selected, minimizing peak power does not improve performance or reduce actuator mass. The last part of this chapter illustrates the advantages and disadvantages when designing elastic elements to reduce peak power.

---

[1] Parts of this chapter have been adapted from (Bolivar Nieto et al., 2019). © 2019 IEEE. Adapted, with permission, from Bolivar Nieto, E.A., S. Rezazadeh, and R. D. Gregg, Minimizing Energy Consumption and Peak Power of Series Elastic Actuators: a Convex Optimization Framework for Elastic Element Design. IEEE/ASME Transactions on Mechatronics. 24(3), 1334-1345.

2.1 Design of linear series elasticity

The torque-elongation profile of a linear spring is defined by its stiffness value, $k \in \mathbb{R}_{++}$. The goal of this section is to present a convex optimization program for the design of k that minimizes motor energy consumption. From the optimization point of view, the optimization variable k belongs to a one-dimensional space. With the speed of modern processors, finding an optimal point in a one-dimensional space is a simple problem, even if the optimization program is not convex. Thus, how useful is it to show convexity for this application? The convex formulation guarantees a global optimal point, which is not the case for arbitrary optimization programs. In addition, convexity is useful to obtain a computationally tractable solution when the formulation has uncertainty in the parameters (more details in Chapter 3). Showing convexity will also illustrate the necessary conditions for the load trajectory and motor configurations to reduce energy consumption and will provide a very intuitive representation of the role of stiffness in the reduction of motor energy consumption. The content of this section has been adapted from (Bolivar Nieto et al., 2019).

2.1.1 Energy consumption as a convex function of compliance

In the case of a linear spring, elongation and torque are related by $\tau_s = k(q_l - q_m/r)$, where k is the stiffness constant. Using this relationship, the position of the motor and corresponding time derivatives can be expressed as a function of the given load position and the spring torque τ_s as follows:

$$\begin{aligned} q_m &= (q_l - \tau_s/k)r, \\ \dot{q}_m &= (\dot{q}_l - \dot{\tau}_s/k)r, \\ \ddot{q}_m &= (\ddot{q}_l - \ddot{\tau}_s/k)r. \end{aligned} \tag{2.1}$$

Replacing these expressions into (1.1) and defining compliance as the inverse of stiffness, $\alpha := 1/k$, the expression of motor torque can be written as an affine function of compliance as follows:

$$\tau_m = \gamma_1 \alpha + \gamma_2, \tag{2.2}$$

where

$$\gamma_1 = -\left(I_m \ddot{\tau}_s r + b_m \dot{\tau}_s r\right), \tag{2.3}$$

$$\gamma_2 = I_m \ddot{q}_l r + b_m \dot{q}_l r - \frac{\tau_s}{\eta r}, \tag{2.4}$$

are known constants that depend on the reference trajectory. Using the definition of τ_m in (2.2) and assuming periodic motion (1.6), the expression of energy consumption of the motor is the following convex-quadratic function of compliance:

$$\begin{aligned} E_m &= \int_{t_0}^{t_f} \left(\frac{\tau_m^2}{k_m^2} + \tau_m \dot{q}_m \right) dt, \\ &= \int_{t_0}^{t_f} \left(\frac{\tau_m^2}{k_m^2} + b_m \dot{q}_m^2 - \frac{\tau_s \dot{q}_m}{\eta r} \right) dt + \int_{t_0}^{t_f} I_m \dot{q}_m d\dot{q}_m, \\ &= a\alpha^2 + b\alpha + c, \end{aligned} \tag{2.5}$$

where

$$\begin{aligned} a &= \int_{t_0}^{t_f} \left(\frac{\gamma_1^2}{k_m^2} + b_m r^2 \dot{\tau}_s^2 \right) dt, \\ b &= \int_{t_0}^{t_f} \left(\frac{2\gamma_1\gamma_2}{k_m^2} - 2b_m r^2 \dot{q}_l \dot{\tau}_s \right) dt, \\ c &= \int_{t_0}^{t_f} \left(\frac{\gamma_2^2}{k_m^2} + b_m \dot{q}_l^2 r^2 - \frac{\dot{q}_l \tau_s}{\eta} \right) dt. \end{aligned}$$

With this formulation the optimization variable that makes the problem convex is the spring compliance, α.

Properties of the convex-quadratic function of compliance:

The quadratic expression (2.5) has the following properties:

1. $d^2E_m/d\alpha^2 = 2a \geq 0$, which follows from the definition of a. Therefore (2.5) is a convex function of compliance (Boyd and Vandenberghe, 2004, p. 71).

2. Parameter c is the energy consumed by a rigid actuator performing the same task without an elastic element, i.e.,

$$\lim_{k \to \infty} E_m = c.$$

3. The optimal value of compliance that minimizes energy consumption for any periodic trajectory is $\alpha = -b/(2a)$, neglecting actuator constraints (Fig. 2.1). This optimal value can be computed in polynomial time. Note that the integrals in the definition of a and b can be approximated with discrete-time summations.

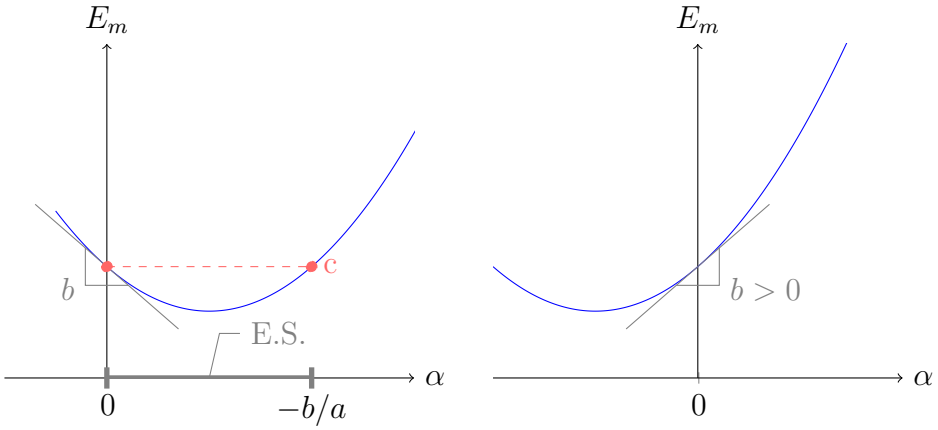

Figure 2.1. Left: Energy consumption as a function of compliance, α, where the energy savings (E.S.) region $0 \leq \alpha \leq -b/a$ provides E_m below the rigid level c. Right: Case of motor and load that would not benefit energetically from series elasticity.

2.1.2 Necessary conditions for reduction of energy consumption

The sign of b determines whether the reference trajectories and motor configuration will benefit from series elasticity in order to reduce energy consumption. The quadratic cost function (2.5) leads to two possible scenarios for the effect of compliance α on motor energy (Fig. 2.1). In the first case, $dE_m/d\alpha$ is negative at $\alpha = 0$; thus, series elasticity improves actuator efficiency over some range of compliance. In the second case, this slope is positive at $\alpha = 0$, so energy increases with compliance; i.e., there is no energetic benefit to linear series elasticity for the given task. Thus, using the modeling assumptions in (2.5), *the necessary and sufficient condition for a linear series spring to be energetically beneficial is that for the load trajectory and motor parameters satisfy* $b < 0$ in (2.5), i.e.,

$$\int_{t_0}^{t_f} \left(\frac{2\gamma_1\gamma_2}{k_m^2} - 2b_m r^2 \dot{q}_l \dot{\tau}_s \right) dt < 0. \tag{2.6}$$

2.1.3 Actuator constraints in the convex formulation

This section describes the actuator constraints in Section 1.2.1 as convex functions of the spring compliance, α. In practice, optimization programs are solved using numerical solvers such as Mosek or Gurobi (MOSEK-ApS, 2019; Gurobi Optimization, 2018). These solvers deal with optimization variables of finite dimension to express the objective and constraints functions. Consequently, we express continuous-time as discrete-time trajectories trajectories. Unless stated otherwise, all the variables previously defined are represented in discrete-

time using vector notation; e.g., $\boldsymbol{q_m} \in \mathbb{R}^n$ is the discrete-time representation of $q_m(t) \in \mathbb{R}$ using n equally spaced time samples.

Constraining maximum elongation of the spring

Limited elongation of the elastic element is typical in SEA applications. An elastic element reaching its maximum elongation could be dangerous for co-robots. When the spring bottoms out, the elastic collisions with the environment become inelastic, which may be harmful for the user and the robot itself. We express the elongation constraint as

$$\|\boldsymbol{\tau_s}\alpha\|_\infty \leq \delta_s,$$

where δ_s is the maximum elongation of the spring. This results in the constraint

$$\begin{aligned} \left\|m(\boldsymbol{\tau_{s/m}})\alpha\right\|_\infty &\leq \delta_s, \\ \begin{bmatrix} m(\boldsymbol{\tau_{s/m}}) \\ -m(\boldsymbol{\tau_{s/m}}) \end{bmatrix}^T \alpha &\leq \mathbf{1}\delta_s, \\ \boldsymbol{d_1}\alpha &\leq \boldsymbol{e_1}, \end{aligned} \tag{2.7}$$

where

$$\boldsymbol{d_1} = \begin{bmatrix} \boldsymbol{\tau_{s/m}} \\ -\boldsymbol{\tau_{s/m}} \end{bmatrix}, \ \boldsymbol{e_1} = \mathbf{1}\frac{\delta_s}{m},$$

m is a scalar factor, and $\boldsymbol{\tau_{s/m}}$ is a normalized expression of the spring torque per unit of m, i.e., $\boldsymbol{\tau_s} = m\boldsymbol{\tau_{s/m}}$. This normalized expression will be useful in the next chapter for the formulation of a robust solution of the constraints.

Maximum motor torque constraint

The limitations in peak torque of the motor are written as $\|\boldsymbol{\tau_m}\|_\infty \leq \tau_{max}$, where τ_{max} is the maximum peak value of torque. Recall that the torque of the motor can be written as an affine function of compliance (2.2), $\boldsymbol{\tau_m} = \boldsymbol{\gamma_1}\alpha + \boldsymbol{\gamma_2}$. Thus, constraining the peak torque is equivalent to the following affine constraint:

$$\begin{aligned} \|\boldsymbol{\gamma_1}\alpha + \boldsymbol{\gamma_2}\|_\infty &\leq \tau_{max}, \\ \begin{bmatrix} \boldsymbol{\gamma_1} \\ -\boldsymbol{\gamma_1} \end{bmatrix} \alpha &\leq \mathbf{1}\tau_{max} + \begin{bmatrix} -\boldsymbol{\gamma_2} \\ \boldsymbol{\gamma_2} \end{bmatrix}, \\ \boldsymbol{d_2}\alpha &\leq \boldsymbol{e_2}, \end{aligned} \tag{2.8}$$

where

$$\boldsymbol{d_2} = \begin{bmatrix} I_m \ddot{\boldsymbol{\tau}}_{\boldsymbol{s/m}} r + b_m \dot{\boldsymbol{\tau}}_{\boldsymbol{s/m}} r \\ -I_m \ddot{\boldsymbol{\tau}}_{\boldsymbol{s/m}} r - b_m \dot{\boldsymbol{\tau}}_{\boldsymbol{s/m}} r \end{bmatrix},$$

$$\boldsymbol{e_2} = \begin{bmatrix} \dfrac{\boldsymbol{\tau_{s/m}}}{\eta r} \\ \dfrac{-\boldsymbol{\tau_{s/m}}}{\eta r} \end{bmatrix} + \begin{bmatrix} -I_m r \ddot{\boldsymbol{q}}_{\boldsymbol{l}} - b_m r \dot{\boldsymbol{q}}_{\boldsymbol{l}} + \mathbf{1}\tau_{max} \\ I_m r \ddot{\boldsymbol{q}}_{\boldsymbol{l}} + b_m r \dot{\boldsymbol{q}}_{\boldsymbol{l}} + \mathbf{1}\tau_{max} \end{bmatrix} \frac{1}{m}.$$

Speed-torque relationship constraint

As an actuator, a DC motor simultaneously operates as an electric generator producing a back-emf voltage. This back-emf voltage, which is proportional to the motor's speed of rotation, limits the current that can flow through the motor's winding, which is proportional to the torque produced by the motor. As a consequence, the maximum torque that a DC motor generates is a function of the rotational speed (Section 1.2.1). This phenomenon is summarized by the equation $\tau_m(R/k_t) = v_s - k_t \dot{q}_m$. Then for a DC motor to be feasible $\tau_m(R/k_t) = v_s - k_t \dot{q}_m$ (Rezazadeh and Hurst, 2014). The same inequality applies for positive and negative values of speed and torque, therefore in total there are four inequalities to express the torque-velocity relationship constraints. The following affine constraint represents these inequalities:

$$\begin{aligned} \boldsymbol{\tau_m} &\leq \mathbf{1} v_s \frac{k_t}{R} - \frac{k_t^2}{R} \dot{\boldsymbol{q}}_{\boldsymbol{m}}, \\ \boldsymbol{\gamma_1}\alpha + \boldsymbol{\gamma_2} &\leq \mathbf{1} v_s \frac{k_t}{R} - \frac{k_t^2}{R} (\dot{\boldsymbol{q}}_{\boldsymbol{l}} - \dot{\boldsymbol{\tau}}_{\boldsymbol{l}} \alpha) r, \\ \boldsymbol{d_{3a}}\alpha &\leq \boldsymbol{e_{3a}}, \end{aligned} \tag{2.9}$$

where

$$\begin{aligned} \boldsymbol{d_{3a}} &= I_m \ddot{\boldsymbol{\tau}}_{\boldsymbol{s/m}} r + b_m \dot{\boldsymbol{\tau}}_{\boldsymbol{s/m}} r - \frac{k_t^2 r}{R} \dot{\boldsymbol{\tau}}_{\boldsymbol{s/m}}, \\ \boldsymbol{e_{3a}} &= \frac{\boldsymbol{\tau_{s/m}}}{\eta r} + \left(\mathbf{1} v_s \frac{k_t}{R} - I_m r \ddot{\boldsymbol{q}}_{\boldsymbol{l}} - b_m r \dot{\boldsymbol{q}}_{\boldsymbol{l}} - \frac{k_t^2 r}{R} \dot{\boldsymbol{q}}_{\boldsymbol{l}} \right) \frac{1}{m}. \end{aligned}$$

Using positive and negative values of torque and speed we can define three similar versions of the inequality (2.9), which we will denote using the vectors $\boldsymbol{d_{3_{b,c,d}}}$ and $\boldsymbol{e_{3_{b,c,d}}}$. In summary, the torque and velocity relationship constraints can be lumped into the single vector inequality constraint

$$\boldsymbol{d}_3 \alpha \leq \boldsymbol{e}_3, \tag{2.10}$$

where

$$\boldsymbol{d_3} = [\boldsymbol{d_{3a}^T}, \boldsymbol{d_{3b}^T}, \boldsymbol{d_{3c}^T}, \boldsymbol{d_{3d}^T}]^T, \ \boldsymbol{e_3} = [\boldsymbol{e_{3a}^T}, \boldsymbol{e_{3b}^T}, \boldsymbol{e_{3c}^T}, \boldsymbol{e_{3d}^T}]^T.$$

RMS torque and maximum speed

Long-term operation of an electric motor can generate excessive heat and can be harmful for the actuator. Constraining the RMS torque is a typical method to guarantee that long-term operation is safe for the device. In the proposed framework, the square of the RMS torque can be written as a convex-quadratic function of compliance and therefore can be included as a constraint. However, RMS torque also appears in the objective function (2.5). Therefore, it is redundant to include it as a constraint. The constraint (2.10) already considers the maximum speed of rotation of the motor, which is equivalent to $\tau_m(R/k_t) \leq v_s - k_t\dot{q}_m$ when the motor torque, τ_m, is zero.

Lumping the constraints

Peak motor torque, peak motor velocity, speed-torque relationship constraints, and maximum elongation of the spring can be represented as the following vector inequalities:

$$\boldsymbol{d}\alpha \leq \boldsymbol{e_l} \tag{2.11}$$

where

$$\boldsymbol{d} = [\boldsymbol{d_1}^T, \boldsymbol{d_2}^T, \boldsymbol{d_3}^T]^T,\ \boldsymbol{e_l} = [\boldsymbol{e_1}^T, \boldsymbol{e_2}^T, \boldsymbol{e_3}^T]^T. \tag{2.12}$$

2.1.4 The convex optimization program for energy efficient linear SEAs

Using the objective function (2.5) and the constraints (2.11), we can formulate the following optimization program that summarizes the design framework for energy efficient linear SEAs, under nominal operational conditions:

$$\begin{aligned} \underset{\alpha}{\text{minimize}} \quad & a\alpha^2 + b\alpha + c, \\ \text{subject to} \quad & \boldsymbol{d}\alpha \leq \boldsymbol{e_l} \end{aligned} \tag{2.13}$$

2.2 Design of nonlinear series elasticity

Design of the nonlinear series spring that minimizes energy consumption is similar to the linear formulation in Section 2.1. The methodology assumes that the load trajectory and parameters of the motor-transmission are known. The optimal torque-elongation profile should represent a conservative spring; to this end, we constrain the spring torque to be a monotonically increasing function of its elongation. As in Section 2.1, the resulting optimization program is shown to be convex.

The goal of the proposed method is to define the function $f : \mathbb{R} \to \mathbb{R}$ that relates torque and elongation of the spring, i.e.,

$$\tau_s = f(\delta).$$

The torque-elongation profile of the nonlinear spring is fully defined by a vector of spring compliance and the equilibrium length of the spring. The equilibrium length is a constant parameter that represents the preload on the spring and will not be considered as an optimization variable, as it could be arbitrarily defined. Thus, in this section, the optimization variable is the compliance vector, $\boldsymbol{\alpha} \in \mathbb{R}^n$, which is defined as

$$\begin{aligned} \alpha_i &= \frac{d\delta_i}{d\tau_{s_i}},\ i = 1, \ldots, n \\ &= \frac{\dot{\delta}_i}{\dot{\tau}_{s_i}},\ i = 1, \ldots, n \end{aligned} \tag{2.14}$$

where $\boldsymbol{\delta}$ and $\boldsymbol{\tau_s}$ are the elongation and torque of the spring. This definition applies for $\dot{\tau}_{s_i} \neq 0$. When $\dot{\tau}_{s_i} = 0$, the spring compliance is defined to be zero. In this case, the spring stiffness approaches infinity, the spring becomes a rigid link.

2.2.1 Energy consumption as a convex function of the compliance vector

The formulation of the optimization program starts with the definition of motor energy consumption as a function of compliance, the optimization variable. Energy requirements depend on the torque and speed of the motor, thus we start our derivation with the definition of motor torque and speed as a function of the compliance vector.

Motor velocity and torque as functions of $\boldsymbol{\alpha}$

Rearranging (2.14), we obtain

$$\begin{aligned} \dot{\tau}_{s_i}\alpha_i &= \dot{\delta}_i,\ i = 1, \ldots, n \\ &= \dot{q}_{l_i} - \dot{q}_{m_i}\frac{1}{r},\ i = 1, \ldots, n \\ \operatorname{diag}(\dot{\boldsymbol{\tau}}_s)\boldsymbol{\alpha} &= \dot{\boldsymbol{q}}_l - \dot{\boldsymbol{q}}_m\frac{1}{r}. \end{aligned}$$

Thus, motor velocity and acceleration can be written as the following affine functions of compliance:

$$\dot{\boldsymbol{q}}_m = \boldsymbol{a} + \boldsymbol{B}\boldsymbol{\alpha} \tag{2.15}$$

$$\ddot{\boldsymbol{q}}_m = \boldsymbol{c} + \boldsymbol{D}\boldsymbol{\alpha} \tag{2.16}$$

where

$$\begin{aligned}\boldsymbol{a} &= \dot{\boldsymbol{q}}_l r,\\ \boldsymbol{B} &= -\mathrm{diag}(\dot{\boldsymbol{\tau}}_s) r,\\ \boldsymbol{c} &= \ddot{\boldsymbol{q}}_l r,\\ \boldsymbol{D} &= -\mathrm{diag}(\ddot{\boldsymbol{\tau}}_s) r.\end{aligned}$$

Using (2.15) and (2.16) we can write motor torque as

$$\begin{aligned}\boldsymbol{\tau}_m &= I_m \ddot{\boldsymbol{q}}_m + b_m \dot{\boldsymbol{q}}_m - \frac{\boldsymbol{\tau}_s}{\eta r}\\ &= \boldsymbol{e} + \boldsymbol{F}\boldsymbol{\alpha}\end{aligned} \tag{2.17}$$

where

$$\begin{aligned}\boldsymbol{e} &= \left(I_m \boldsymbol{c} + b_m \boldsymbol{a} - \frac{\boldsymbol{\tau}_s}{\eta r}\right),\\ \boldsymbol{F} &= \left(I_m \boldsymbol{D} + b_m \boldsymbol{B}\right).\end{aligned}$$

Energy Consumption as a Function of α

With the definition of velocity (2.15) and torque (2.17) of the motor we define the motor energy consumption as follows:

$$\begin{aligned}E &= \int_{t_0}^{t_f} \left(\frac{\tau_m^2}{k_m^2} + \tau_m \dot{q}_m\right) dt\\ &\approx \left(\frac{\boldsymbol{\tau}_m^T \boldsymbol{\tau}_m}{k_m^2} + b_m \dot{\boldsymbol{q}}_m^T \dot{\boldsymbol{q}}_m - \frac{\boldsymbol{\tau}_s^T \dot{\boldsymbol{q}}_l}{\eta}\right) \Delta t\\ &= \boldsymbol{\alpha}^{\boldsymbol{T}} \boldsymbol{G} \boldsymbol{\alpha} + \boldsymbol{h}\boldsymbol{\alpha} + w\end{aligned} \tag{2.18}$$

where

$$\begin{aligned}\boldsymbol{G} &= \left(\frac{\boldsymbol{F}^T \boldsymbol{F}}{k_m^2} + b_m \boldsymbol{B}^T \boldsymbol{B}\right) \Delta t\\ \boldsymbol{h} &= \left(\frac{2\boldsymbol{e}^T \boldsymbol{F}}{k_m^2} + 2 b_m \boldsymbol{a}^T \boldsymbol{B}\right) \Delta t\\ w &= \left(\frac{\boldsymbol{e}^T \boldsymbol{e}}{k_m^2} + b_m \boldsymbol{a}^T \boldsymbol{a} - \frac{\boldsymbol{\tau}_s^T \dot{\boldsymbol{q}}_l}{\eta}\right) \Delta t.\end{aligned}$$

The Hessian of the cost function (2.18), $d^2E/d\boldsymbol{\alpha}^2 = 2\boldsymbol{G}$, is a positive semi-definite matrix; therefore, the function is convex with respect to $\boldsymbol{\alpha}$ (Boyd and Vandenberghe, 2004, p. 71).

Positive semi-definiteness can be shown from the definition of $\boldsymbol{G}$. The Gramian matrices of $\boldsymbol{F}$ and $\boldsymbol{B}$, i.e., $\boldsymbol{F}^T\boldsymbol{F}$ and $\boldsymbol{B}^T\boldsymbol{B}$, are positive semi-definite as shown by their singular value decomposition, $\boldsymbol{F} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$, and

$$\begin{aligned}\boldsymbol{F}^T\boldsymbol{F} &= (\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T)^T(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T) \\ &= \boldsymbol{V}\boldsymbol{\Sigma}^2\boldsymbol{V}^T\end{aligned}$$

where $\boldsymbol{F} \in \mathbb{R}^{n\times n}$ with $\mathbf{rank}\boldsymbol{F} = r$, $\boldsymbol{U} \in \mathbb{R}^{n\times r}$, $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}$, $\boldsymbol{V} \in \mathbb{R}^{n\times r}$, $\boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}$, and $\Sigma = \mathbf{diag}(\sigma_1, \ldots, \sigma_r)$, with $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$.

2.2.2 Constraints for a conservative elastic element

Solving the optimization program defined by the objective function (2.18) may lead to an elastic element that is not conservative, i.e., the elastic element stores or dissipates energy after a full displacement cycle,

$$\int_{\delta_0}^{\delta_f} \tau_s d\delta \neq 0$$

. The Fig. 2.2 illustrates the torque-displacement diagram of a nonconservative spring. In practice, elastic elements could dissipate energy (for instance, by viscous friction), but the energy lost is negligible depending on the material and design of the spring. The design methodology presented here assumes that the spring is conservative and has no energy losses. To this end, the optimization program constraints the compliance vector, $\boldsymbol{\alpha}$, to generate a torque-elongation relationship, f, that is monotonically increasing.

The optimization method includes the following constraint to represent a monotonically increasing function:

$$(\tau_{s_i} - \tau_{s_{i-1}}) > 0 \implies (\delta_i - \delta_{i-1}) > 0, \tag{2.19}$$

$$(\tau_{s_i} - \tau_{s_{i-1}}) < 0 \implies (\delta_i - \delta_{i-1}) < 0, \tag{2.20}$$

$$(\tau_{s_i} - \tau_{s_{i-1}}) = 0 \implies (\delta_i - \delta_{i-1}) = 0. \tag{2.21}$$

for $i = 2, \ldots, n$; when $i = 1$, then $i - 1 = n$ because the motion is periodic. This constraint is equivalent to

$$\frac{\Delta\tau_s}{\Delta\delta} > 0.$$

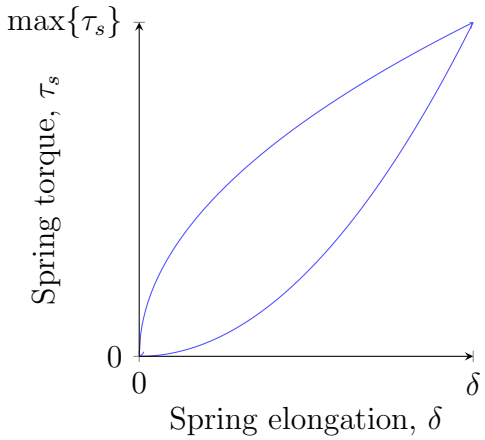

Figure 2.2. Nonconservative spring. At the end of a closed displacement cycle energy could be stored or dissipated by the spring.

Note that elongation of the spring, δ, is an affine function of the compliance vector, i.e,

$$\begin{aligned}
\delta &= q_l - \frac{q_m}{r} \\
&= \int_{t_0}^{t_f} \alpha \dot{\tau}_s dt + \delta_0 \\
&\approx \sum_{i=1}^{i=n} \frac{\alpha_i \dot{\tau}_{s_i} + \alpha_{i+1} \dot{\tau}_{s_{i+1}}}{2} \Delta t + \delta_0 \\
&= \left(\frac{\alpha_1 \dot{\tau}_{s_1}}{2} + \alpha_2 \dot{\tau}_{s_2} + \alpha_3 \dot{\tau}_{s_3} + \ldots + \alpha_{n-1} \dot{\tau}_{s_{n-1}} + \frac{\alpha_n \dot{\tau}_{s_n}}{2} \right) \Delta t + \delta_0;
\end{aligned}$$

the approximation is based on the trapezoidal method for numerical integration. In vector form this is equal to

$$\begin{aligned}
\boldsymbol{\delta} &\approx \Delta t \begin{bmatrix} 0 & \ldots & \ldots & \ldots & 0 \\ \frac{\dot{\tau}_{s_1}}{2} & \frac{\dot{\tau}_{s_2}}{2} & 0 & \ldots & 0 \\ \frac{\dot{\tau}_{s_1}}{2} & \dot{\tau}_{s_2} & \frac{\dot{\tau}_{s_3}}{2} & \ldots & 0 \\ \vdots & & & \ddots & 0 \\ \frac{\dot{\tau}_{s_1}}{2} & \dot{\tau}_{s_2} & \ldots & \dot{\tau}_{s_{n-1}} & \frac{\dot{\tau}_{s_n}}{2} \end{bmatrix} \boldsymbol{\alpha} + \mathbf{1}\delta_0 \\
&= \boldsymbol{L}\boldsymbol{\alpha} + \mathbf{1}\delta_0
\end{aligned} \tag{2.22}$$

The constraints (2.19) - (2.21) are a convex set defined by the a sum of affine functions of $\boldsymbol{\alpha}$.

2.2.3 Actuator constraints as a convex function of the compliance vector

In this section, we show convexity of the actuator constraints. Following a similar procedure as in Section 2.1.3, we consider the following actuator constraints: Maximum Elongation, Maximum Torque, and the Speed-torque relationship of electric motors.

Maximum elongation

Elongation is defined from (2.22) as $\boldsymbol{\delta} = \boldsymbol{L}\boldsymbol{\alpha} + \mathbf{1}\delta_0$. Then to limit the maximum absolute value of elongation, $\|\boldsymbol{\delta}\|_\infty \leq \delta_{\max}$, we impose the constraint

$$\begin{bmatrix} \boldsymbol{L} \\ -\boldsymbol{L} \end{bmatrix} \boldsymbol{\alpha} \leq \begin{bmatrix} \mathbf{1}(\delta_{\max} - \delta_0) \\ \mathbf{1}(\delta_{\max} + \delta_0) \end{bmatrix} \tag{2.23}$$

Maximum torque

From the definition of motor torque (2.17), we can limit its maximum absolute value, $\|\boldsymbol{\tau}_m\|_\infty$, using

$$\begin{bmatrix} \boldsymbol{F} \\ -\boldsymbol{F} \end{bmatrix} \boldsymbol{\alpha} \leq \begin{bmatrix} \mathbf{1}\tau_{\max} - \boldsymbol{e} \\ \mathbf{1}\tau_{\max} + \boldsymbol{e} \end{bmatrix} \tag{2.24}$$

Speed-torque relationship

As discussed in Section 1.2.1, for a four-quadrant motor, the speed and torque of the motor should satisfy the following inequality:

$$|\tau_m| \leq \frac{k_t}{R} V_{in} - \frac{k_t^2}{R} |\dot{q}_m|. \tag{2.25}$$

Using (2.17) and (2.15), we can write (2.25) as a function of α as follows:

$$\begin{bmatrix} \boldsymbol{F} + \frac{k_t^2}{R}\boldsymbol{B} \\ \boldsymbol{F} - \frac{k_t^2}{R}\boldsymbol{B} \\ -\boldsymbol{F} + \frac{k_t^2}{R}\boldsymbol{B} \\ -\boldsymbol{F} - \frac{k_t^2}{R}\boldsymbol{B} \end{bmatrix} \boldsymbol{\alpha} \leq \begin{bmatrix} \mathbf{1}\frac{k_t}{R}V_{in} - \boldsymbol{e} - \frac{k_t^2}{R}\boldsymbol{a} \\ \mathbf{1}\frac{k_t}{R}V_{in} - \boldsymbol{e} + \frac{k_t^2}{R}\boldsymbol{a} \\ \mathbf{1}\frac{k_t}{R}V_{in} + \boldsymbol{e} - \frac{k_t^2}{R}\boldsymbol{a} \\ \mathbf{1}\frac{k_t}{R}V_{in} + \boldsymbol{e} + \frac{k_t^2}{R}\boldsymbol{a} \end{bmatrix} \tag{2.26}$$

2.2.4 The convex optimization program for energy efficient nonlinear SEAs

Collecting the objective and constraints functions, we can write the optimization program as follows:

$$\begin{aligned} \underset{\boldsymbol{\alpha}}{\text{minimize}} \quad & \boldsymbol{\alpha}^{\boldsymbol{T}}\boldsymbol{G}\boldsymbol{\alpha} + \boldsymbol{h}\boldsymbol{\alpha} + w, \\ \text{subject to} \quad & \boldsymbol{M}\boldsymbol{\alpha} \leq \boldsymbol{p} \end{aligned} \tag{2.27}$$

where $\boldsymbol{M}$, $\boldsymbol{p}$ concatenate the constraints in (2.23), (2.24), and (2.26) in the following form:

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{L} \\ -\boldsymbol{L} \\ \boldsymbol{F} \\ -\boldsymbol{F} \\ \boldsymbol{F} + \frac{k_t^2}{R}\boldsymbol{B} \\ \boldsymbol{F} - \frac{k_t^2}{R}\boldsymbol{B} \\ -\boldsymbol{F} + \frac{k_t^2}{R}\boldsymbol{B} \\ -\boldsymbol{F} - \frac{k_t^2}{R}\boldsymbol{B} \end{bmatrix}, \; \boldsymbol{p} = \begin{bmatrix} \mathbf{1}(\delta_{\max} - \delta_0) \\ \mathbf{1}(\delta_{\max} + \delta_0) \\ \mathbf{1}\tau_{\max} - \boldsymbol{e} \\ \mathbf{1}\tau_{\max} + \boldsymbol{e} \\ \mathbf{1}\frac{k_t^2}{R}V_{in} - \boldsymbol{e} - \frac{k_t^2}{R}\boldsymbol{a} \\ \mathbf{1}\frac{k_t^2}{R}V_{in} - \boldsymbol{e} + \frac{k_t^2}{R}\boldsymbol{a} \\ \mathbf{1}\frac{k_t^2}{R}V_{in} + \boldsymbol{e} - \frac{k_t^2}{R}\boldsymbol{a} \\ \mathbf{1}\frac{k_t^2}{R}V_{in} + \boldsymbol{e} + \frac{k_t^2}{R}\boldsymbol{a} \end{bmatrix}$$

2.2.5 Example: natural oscillation of a nonlinear spring

In this example, we apply the convex optimization framework in (2.27) to design the SEA's elastic element for the reference trajectory corresponding to the natural oscillation of a nonlinear spring. The example compares the numerical results of the optimization with the well established analysis of a mass-spring system. This comparison is a good starting point to gain intuition into the design methodology. For instance, if the reference position of the SEA corresponds to the natural oscillation of a nonlinear spring and the cost function is the energy dissipated by viscous friction, the optimal elastic element should be the same as the nonlinear spring used to generate the trajectories. This section illustrates this case and compares it with the optimization results using the following cost functions: winding losses, and total energy consumption.

The Fig. 2.3-(a) describes a single mass-spring system with a nonlinear spring, $\tau_s = k_{\text{cubic}}q_l^3 + kq_l$, and corresponding equation of motion $\tau_s = -I_l\ddot{q}_l$, where $I_l = 125\,\text{g}\cdot\text{m}^2$ is the inertia of the load, $k_{\text{cubic}} = 40\,\text{N}\cdot\text{m/rad}^3$, and $k = 10\,\text{N}\cdot\text{m/rad}$. The main objective of this

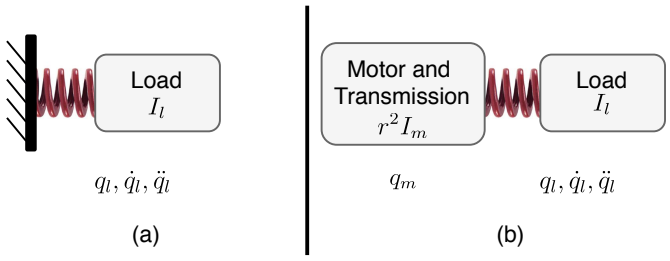

Figure 2.3. (a) Single-mass spring system. The elastic element describes the nonlinear spring with $\tau_s = k_{\text{cubic}} q_l^3 + k q_l$. (b) Double-mass single-spring system. The equilibrium position of the elastic element is $q_l = q_m/r$, elongation is defined as $\delta = q_l - q_m/r$. Motor and transmission are considered to be backdrivable.

case study is to validate the optimization results against the known natural-oscillation of a cubic spring. To simplify the analysis, we consider no actuator constraints and a no-loss mechanical transmission, i.e., $\eta = 1$. The Fig. 2.4 illustrates the motion of the load for an initial displacement, $q_l(0) = \pi/2\,\text{rad}$. This natural vibration is defined as our reference motion. The SEA, Fig. 2.3-(b), can generate this motion with the motor holding its initial position if the elastic element matches the nonlinear spring in Fig. 2.3-(a). However, this approach may not be energetically efficient. If the load is high enough to backdrive the system, the motor must apply a reactionary torque to hold its initial position. This torque requires a current that generates heat losses at the motor's winding due to Joule heating. In contrast, we can solve the optimization problem in (2.27) to find the elastic element that minimizes the total energy expenditure (i.e., winding losses and viscous friction). To evaluate the proposed methodology, we solved the optimization problem for each of the following cost functions: energy dissipated by winding Joule heating, energy dissipated by viscous friction, and total energy consumption. Each of these cost functions is formulated from appropriate modifications to the matrices $\boldsymbol{G}$ and $\boldsymbol{h}$ in (2.27). The resulting elastic elements, torques, and positions of the motor are illustrated in Fig. 2.5. Table 2.1 summarizes the energy expenditure for each case.

Minimizing viscous friction leads to the same elastic element as in Fig. 2.3-(a), which validates the numerical results with respect to first principles. The energy required to produce the motion is then 20.175 J, which is all dissipated in the motor's winding. In contrast, minimizing the total energy consumption results in a cost of 9.579 J, 52% less compared to the previous case. The elastic element is nonlinear but is not defined by $\tau_s = 40\delta^3 + 10\delta$, and the motor no longer remains stationary. Minimizing only the energy dissipated by the

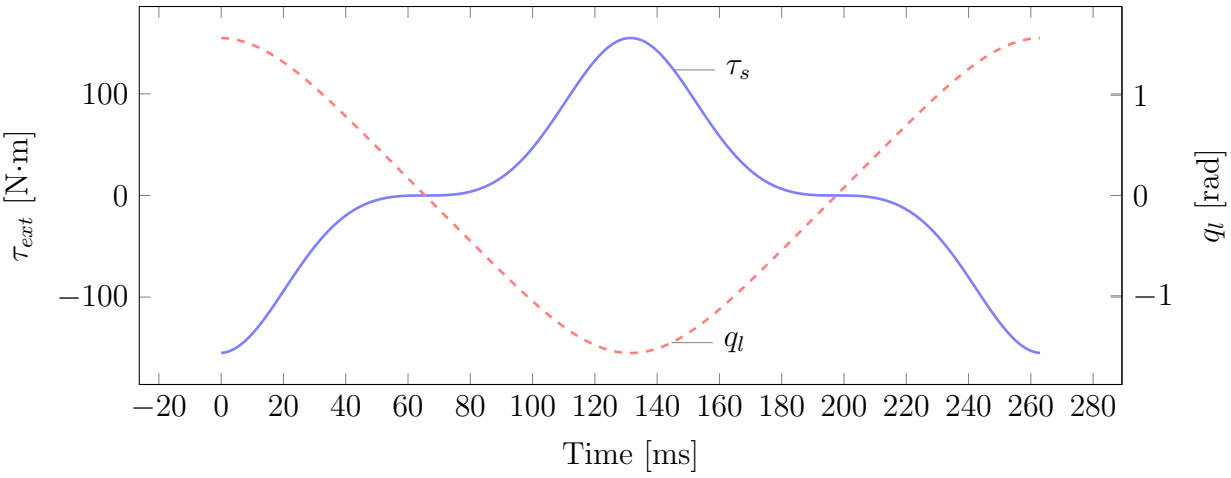

Figure 2.4. The reference trajectory of the load is defined by the natural oscillation of the single mass-spring system in Fig. 2.3-(a) with k_{cubic} =40 N·m/rad^3, k =10 N·m/rad, I_l =125 g·m^2, and $q_l(0) = \pi/2$ rad.

Table 2.1. Energy expenditure. Natural oscillation of cubic spring.

Cost function	Joule heating [J]	Viscous friction [J]	Total Energy [J]
Viscous friction	20.175	0.000	20.175
Winding losses	0.008	20.511	20.519
Total energy	4.972	4.607	9.579

motor's winding leads to an elastic element that minimizes as much motor-torque as possible, as seen in Fig. 2.5. This SEA spring approximates the natural dynamics of the double-mass single-spring system defined by the inertia of the load and the motor.

2.3 Minimizing peak power

Often, the series elastic element is designed to minimize the peak power of the motor (Hollander et al., 2006). Peak motor power is proportional to the mass of the motor; therefore, minimizing it is beneficial to minimize the mass of the robot. However, it is very important to recognize that minimizing peak power is only relevant when the motor and transmission have not been selected. If those elements are set in the design, minimizing peak power could be useless. Minimizing peak power may not minimize energy consumption and may not reduce the speed or torque requirements in regions that are outside the feasible region of the

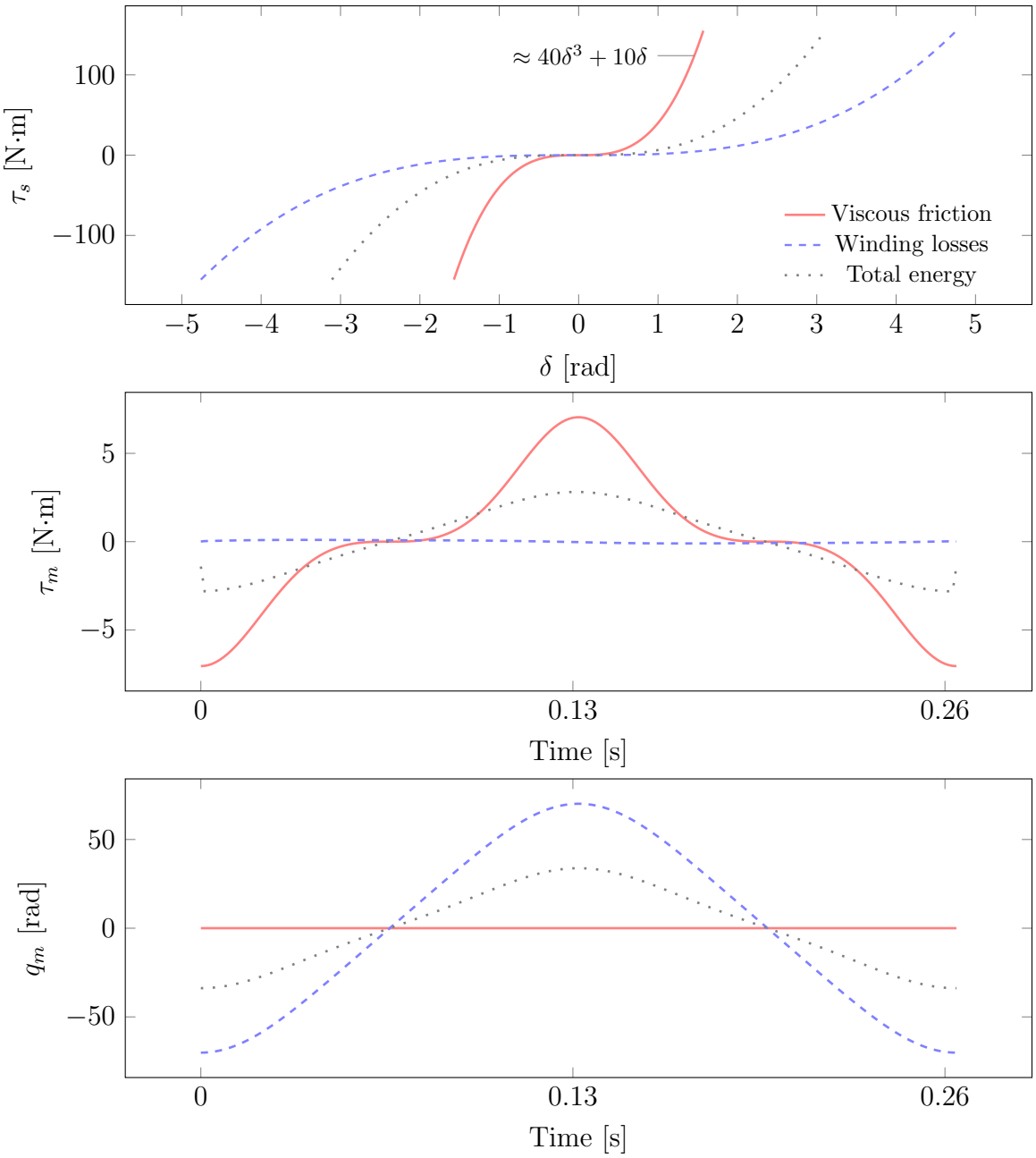

Figure 2.5. Optimization results considering natural oscillation of a nonlinear spring as the reference motion. The solid line corresponds to the elastic element that minimizes the energy consumption due to viscous friction. It matches $\tau_s = 40\delta^3 + 10\delta$, the nonlinear spring used in the single mass-spring system. The dotted line describes the elastic element that minimizes winding losses due to Joule heating. The dashed line describes the elastic element that minimizes both winding losses and viscous friction, i.e., total energy. The corresponding energy expenditure is shown in Table 2.1.

motor (Section 1.2.1). However, it is of interest to see if this objective function is convex in our formulation. This section presents a convex approximation to the reduction of peak power and provides insights on when to select peak power as the criteria for the design of series springs. The work presented in this section has been adapted from (Bolivar Nieto et al., 2019).

For the minimizing peak power we will use the motor position, $\boldsymbol{q_m}$, as the optimization variable. In order to formulate a convex optimization framework, we approximate the continuous time derivative with a discrete time representation in (1.1). We discretize time for one period of the reference motion into n points and approximate time derivatives using the matrix operation $\dot{\boldsymbol{q}}_m \approx D\boldsymbol{q}_m$, where $\boldsymbol{q}_m, \dot{\boldsymbol{q}}_m \in \mathbb{R}^n$ is the discrete-time representation of q_m and $\dot{q}_m$, $D \in \mathbb{R}^{n\times n}$ is

$$D = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & -1 \\ -1 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & & & & \vdots \\ 1 & \cdots & & & -1 & 0 \end{bmatrix} \frac{1}{2\Delta t}, \tag{2.28}$$

and Δt is the sample rate. $D\boldsymbol{q}_m$ is the discrete time derivative of $\boldsymbol{q}_m$ based on the central difference method. The first and last rows of D assume that $\boldsymbol{q}_m$ represents a periodic trajectory, i.e., $\boldsymbol{q}_{m_{(n+1)}} = \boldsymbol{q}_{m_{(1)}}$, where $\boldsymbol{q}_{m_{(i)}}$ is the ith element of the vector $\boldsymbol{q}_m$. Then the equations of motion, (1.1)-(1.2), can be approximated as

$$\boldsymbol{\tau}_m = (I_m D_2 + b_m D)\boldsymbol{q}_m - \boldsymbol{\tau}_s \frac{1}{\eta r}, \tag{2.29}$$

Design of the elastic element to minimize peak power follows the same principles as in the case of energy consumption. The analysis starts with the definition of mechanical power of the electric motor, $\boldsymbol{p}_m \in \mathbb{R}^n$, as a function of $\boldsymbol{q}_m$:

$$\begin{aligned} \boldsymbol{p}_{m(i)} &= \boldsymbol{\tau}_{m(i)}\dot{\boldsymbol{q}}_{m(i)}, \\ &= \left((I_m D_{2(i,*)} + b_m D_{(i,*)})\boldsymbol{q}_m - \boldsymbol{\tau}_{s(i)} \frac{1}{\eta r} \right) D_{(i,*)}\boldsymbol{q}_m, \\ &= \boldsymbol{q}_m^T \underbrace{\left(I_m D_{2(i,*)}^T D_{(i,*)} + b_m D_{(i,*)}^T D_{(i,*)} \right)}_{G_i} \boldsymbol{q}_m - \underbrace{\boldsymbol{\tau}_{s(i)} \frac{1}{\eta r} D_{(i,*)}}_{-H_i} \boldsymbol{q}_m, \qquad (2.30) \\ &= \boldsymbol{q}_m^T G_i \boldsymbol{q}_m + H_i \boldsymbol{q}_m, \qquad (2.31) \end{aligned}$$

for $i = 1, 2, ..., n$, where $D_{(i,*)}$ refers to the ith row of the matrix $D \in \mathbb{R}^{n\times n}$. In other words, every element of the power vector, $\boldsymbol{p}_m$, is a quadratic expression of the motor position.

In contrast to energy consumption, peak power, $\|\boldsymbol{p}_m\|_\infty$, is not a convex function of $\boldsymbol{q}_m$ because, in general, the infinity norm of a set of quadratic functions is not convex. In addition, every quadratic function in (2.31)may not be convex as shown by its Hessian, G_i, in (2.30). This matrix is non-definite because the matrix $I_m D_{2(i,*)}^T D_{(i,*)}$ may have positive and negative eigenvalues. To keep the advantages of a convex optimization problem, we propose a convex approximation to the expression of peak power. This approximation neglects the torque due to inertia, and maximum power is considered instead of peak power, i.e., we do not take the absolute value. Other approximations of peak power are reported in the literature for numerical optimization, for instance, pseudo-power has been defined in (Hong et al., 2017). The simulation results indicate that our convex approximation resembles the actual expression of peak power, and that minimizing the convex version minimizes the actual expression as well.

A convex approximation to peak power

Peak power can be approximated with maximum power to obtain a convex version of the cost function. The max function, $\max\{f_1, f_2, ..., f_n\}$, is convex when each function $f_1, f_2, ..., f_n$ is also convex (Boyd and Vandenberghe, 2004). This justifies the use of the max function instead of the infinity-norm. In our case, each function f_i corresponds to the quadratic expression (2.31). These expressions are convex if and only if G_i is positive semi-definite. From its definition (2.30), the matrix G_i is positive semi-definite if inertial torques are neglected. With this in mind, we define $\boldsymbol{p}_m^{cvx} \in \mathbb{R}^n$, a convex approximation of actual power $\boldsymbol{p}_m$, as follows:

$$\begin{aligned}\boldsymbol{p}_{m(i)}^{cvx} &:= \boldsymbol{q}_m^T \underbrace{\left(b_m D_{(i,*)}^T D_{(i,*)}\right)}_{G_i^{cvx}} \boldsymbol{q}_m - \boldsymbol{\tau}_{s(i)} \frac{1}{\eta r} D_{(i,*)} \boldsymbol{q}_m, \\ &= \boldsymbol{q}_m^T G_i^{cvx} \boldsymbol{q}_m + H_i \boldsymbol{q}_m.\end{aligned}$$

With this approximation the convex optimization problem is written as

$$\underset{\boldsymbol{q}_m}{\text{minimize}} \quad \max\{\boldsymbol{p}_m^{cvx}\}, \tag{2.32}$$

Convexity can be shown since every element in the vector $\boldsymbol{p}_m^{cvx}$ is a convex-quadratic function of $\boldsymbol{q}_m$, i.e., the matrix G_i^{cvx} is positive semi-definite for all i since it is the Gramian matrix of $D_{(i,*)}$. The maximum of a set of convex functions is also convex (Boyd and Vandenberghe, 2004, p. 80). All the constraints are affine with respect to $\boldsymbol{q}_m$, therefore the optimization problem is convex.

Regularization to avoid high accelerations

The solution to the optimization problem (2.32) may result in a position trajectory, $\boldsymbol{q}_m$, with very high accelerations. This has a great impact in the calculation of the actual peak power, especially for SEAs using a high reduction-ratio transmission. One way to solve this is to penalize solutions with high accelerations. This can be achieved by including acceleration of the motor in the cost function as follows:

$$\underset{\boldsymbol{q}_m}{\text{minimize}} \quad \max\{\boldsymbol{p}_m^{cvx}\} + \gamma_1 \left\| D_2 \boldsymbol{q}_m \right\|_\infty \tag{2.33}$$

where γ_1 is a scalar constant that controls the influence of the peak acceleration relative to the maximum of the convex expression of power. The optimization problem (2.33) remains convex. The term $\gamma_1 \left\| D_2 \boldsymbol{q}_m \right\|_\infty$ corresponds to the infinity-norm function, which is convex, composed by the linear expression $D_2 \boldsymbol{q}_m$. The composition of a convex function with an affine function results in a convex function (Boyd and Vandenberghe, 2004).

Energy consumption using motor position as optimization variable

This section describes an alternative version of (2.18) using motor position as the optimization variable. This version will be useful for the formulation of a multiobjective optimization problem between peak power and energy consumption. Using the discrete-time formulation of the dynamics (2.29) and the simplification of rotor mechanical power in (1.6), the energy required by the motor (1.3) can be approximated in discrete form as

$$\begin{aligned} E_m &\approx \sum_{i=1}^{n} \left(\frac{\boldsymbol{\tau}_{m_{(i)}}^2}{k_m^2} + \boldsymbol{\tau}_{m_{(i)}} \dot{\boldsymbol{q}}_{m_{(i)}} \right) \Delta t, \\ &= \left(\frac{\boldsymbol{\tau}_m^T \boldsymbol{\tau}_m}{k_m^2} + b_m \boldsymbol{q}_m^T D^T D \boldsymbol{q}_m - \frac{\boldsymbol{\tau}_s^T \dot{\boldsymbol{q}}_l}{\eta} \right) \Delta t. \end{aligned} \tag{2.34}$$

By defining $\boldsymbol{\tau}_m / k_m = F \boldsymbol{q}_m + \boldsymbol{c}$, where $F = (I_m D_2 + b_m D)/k_m$ and $\boldsymbol{c} = -\boldsymbol{\tau}_s/(\eta k_m r)$, we rewrite (2.34) as

$$\begin{aligned} E_m &= \left(\| F \boldsymbol{q}_m + \boldsymbol{c} \|_2^2 + b_m \boldsymbol{q}_m^T D^T D \boldsymbol{q}_m - \boldsymbol{\tau}_{ela}^T \dot{\boldsymbol{q}}_l / \eta \right) \Delta t, \\ &= \boldsymbol{q}_m^T Q_e \boldsymbol{q}_m + A_e \boldsymbol{q}_m + c_e, \end{aligned} \tag{2.35}$$

where

$$Q_e = \left(F^T F + b_m D^T D \right) \Delta t, \tag{2.36}$$

$$A_e = \left(2 \boldsymbol{c}^T F \right) \Delta t, \tag{2.37}$$

$$c_e = \left(\boldsymbol{c}^T \boldsymbol{c} - \boldsymbol{\tau}_{ela}^T \dot{\boldsymbol{q}}_l / \eta \right) \Delta t. \tag{2.38}$$

2.3.1 Multiobjective optimization: energy consumption and peak power

This section describes the optimization problem that simultaneously minimizes energy consumption and peak power. These two objectives are not always competing; however, when they do, a trade-off curve is useful to guide the design process. For example, a global minimum that reduces energy consumption can lead to higher peak powers compared to actuators without elastic elements. This motivates a multiobjective optimization framework where the designer can choose the appropriate trade-off based on the design specifications. The proposed methodology combines the optimization problems introduced in (2.35) and (2.32) to generate the following multiobjective program:

$$\begin{aligned} \underset{\boldsymbol{q}_m}{\text{minimize}} \quad & \theta\gamma_2 \left(\boldsymbol{q}_m^T Q_e \boldsymbol{q}_m + A_e \boldsymbol{q}_m + c_e\right) + (1-\theta)\left(\max\{\boldsymbol{p}_m^{cvx}\} + \gamma_1 \left\|D_2 \boldsymbol{q}_m\right\|_\infty\right), \\ \text{subject to} \quad & A_1 \boldsymbol{q}_m < \boldsymbol{b}_1, \\ & A_2 \boldsymbol{q}_m = \boldsymbol{b}_2, \\ & h(\boldsymbol{q}_m) \leq \boldsymbol{u}_{max}, \end{aligned} \tag{2.39}$$

where γ_2 is a the factor that scales the magnitude of energy consumption relative to peak power, and $\theta \in [0, 1]$ is the factor that controls the trade-off between energy consumption and peak power in the solution of the optimization problem. Using $\theta = 0$ indicates that only peak power will be minimized, and $\theta = 1$ minimizes only energy consumption. It can be shown that the solution to the optimization problem in (2.39) corresponds to a pareto optimal point (Boyd and Vandenberghe, 2004, p. 178). The optimization problem remains convex, since it is the positive sum of convex functions.

2.3.2 When to minimize peak power?

Peak power is a useful objective function when the motor and transmission ratio has not been chosen in the design. Minimizing peak power may lead to the selection of a smaller motor to achieve the same task. However, this practice may not be conclusive for the selection of the adequate motor and transmission. If the motor and transmission have not been selected, it is recommended to solve the feasibility problem in (2.27) for multiple choices of motor and transmission. This will certainly determine whether a smaller motor could achieve the task.

CHAPTER 3

ROBUST ENERGY-EFFICIENT DESIGN OF AN SEA [1]

Many of the benefits of series elasticity are lost when the task deviates from its nominal design conditions. In fact, one of the biggest reasons not to consider SEAs to reduce energy consumption is their vulnerability against uncertainty. For instance, a robotic ankle powered by an SEA may consume more energy than a robotic ankle without spring when the user walks faster than the nominal speed considered for the design of the optimal SEA (Grimmer and Seyfarth, 2011). In addition to higher energy consumption, uncertainty may require speed or torques of the load that are outside the feasible or safe region of the actuator.

Robots can adapt to uncertainty by changing their inherent mechanical behavior. For example, robots driven by VSAs adapt their inherent compliance according to the task, but require additional mechanisms to control their variable compliance. Usually the energy savings obtained from variable inherent compliance are small compared to the significant mechanical complexity of traditional VSAs, making these actuators unpopular in practice (Hurst and Rizzi, 2008). This unfortunate trade-off could be balanced by novel materials such as dielectric elastomers. These elastomers provide inherent variable compliance in a mechanically simple package (Allen et al., 2019; Bolivar et al., 2016). However, dielectric elastomers are still in the process of development and are not yet commercially available. Given the current mechanical complexity of VSA mechanisms, it is of interest to know if robots with fixed mechanical behavior could be robust against uncertainty with a justifiable trade-off in energy performance.

Section 1.3.3 introduces the state of the art of available methods for robust design of elastic actuators. These methods do not consider the worst possible realization of uncertainty for the feasibility of constraints; they provide a probabilistic confidence margin to satisfy them. Safety margins are also popular for a robust mechanical design. However, they tend to result in over-designed actuators because the safety factors lump without precision the different kinds of uncertainty during operation.

In this section we propose an optimization-based design framework that guarantees satisfaction of the constraints as long as the load trajectory and actuator parameters are within

[1] Parts of this chapter have been adapted from (Bolivar Nieto et al., 2019), © 2019 IEEE. Adapted, with permission, from Bolivar Nieto, E.A., S. Rezazadeh, and R. D. Gregg, Minimizing Energy Consumption and Peak Power of Series Elastic Actuators: a Convex Optimization Framework for Elastic Element Design. IEEE/ASME Transactions on Mechatronics. 24(3), 1334-1345.

the uncertainty sets. The design framework guarantees that the elongation, speed, and torque constraints of the actuator are satisfied for the worst possible realization of uncertainty. The design methodology leverages the convex formulations in Chapter 2 and could be used to understand the benefits of nonlinear springs on the robustness of SEAs.

3.1 Introduction to robust optimization

Robust optimization is a branch of optimization theory that deals with the solution of optimization programs with uncertainty in its parameters. It was originally developed to support operation research algorithms, but it is currently an active topic of research with a growing number of applications, such as robust control, statistics, and machine learning. The seminal work of (Ben-Tal et al., 2009) provides a rigorous and very general treatment of the field. In this section, we introduce the fundamental principles from robust optimization that apply to our SEA spring design.

The general idea of robust optimization is to solve the following optimization program:

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{minimize}} \quad & f(\boldsymbol{x}, \boldsymbol{\beta}) \\ \text{subject to} \quad & \boldsymbol{x} \in \mathcal{X}(\boldsymbol{\beta}) \\ & \boldsymbol{\beta} \in \mathcal{U}_\beta \end{aligned}$$

where f is the objective function that depends on the optimization variable $\boldsymbol{x}$ and the vector of parameters $\boldsymbol{\beta}$. The optimization variable is constrained to be within the set $\mathcal{X}(\boldsymbol{\beta})$ that depends as well on the vector of parameters, $\boldsymbol{\beta}$. The parameters $\boldsymbol{\beta}$ are unknown, but remain within the uncertainty set $\mathcal{U}_\beta$. A robust solution to this optimization program guarantees that $\boldsymbol{x} \in \mathcal{X}$ for all possible realizations of $\boldsymbol{\beta}$, and it minimizes the worst possible value of the objective function, i.e.,

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad \left[\sup_{\boldsymbol{\beta}} f(\boldsymbol{x}, \boldsymbol{\beta}) \right].$$

A robust feasible solution means that the constraints are satisfied as long as $\boldsymbol{\beta} \in \mathcal{U}_\beta$. In general, this optimization program can be solved for any definition of f, $\mathcal{X}$, and $\mathcal{U}$. However, the optimal solution may not be computationally tractable. In fact, most of the research efforts in robust optimization focus on the identification of optimization programs and uncertainty sets that lead to a computationally tractable solution (Ben-Tal et al., 2009). For example, for the case of uncertain conic programs, i.e.,

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{minimize}} \quad & \boldsymbol{c}^T \boldsymbol{x} \\ \text{subject to} \quad & \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x} \in \mathbf{K}, \ \forall \{\boldsymbol{c}, \boldsymbol{b}, \boldsymbol{A}\} \in \mathcal{U}_\beta \end{aligned}$$

where $\mathbf{K}$ is a convex cone, tractable solutions have been identified for the following conditions: 1) when the cone $\mathbf{K}$ is as "simple" as possible, i.e., a non-negative orthant, as in the case of linear optimization, or 2) when the uncertainty set $\mathcal{U}_\beta$ is also easily defined, i.e., a polytope defined by a finite amount of affine inequalities (Ben-Tal et al., 2009). As shown in this section, the optimization programs in (2.13) and (2.27) fit into the second category; the convex cone is a Lorentz-cone (which represents quadratic optimization programs) and the uncertainty sets are a "simple" polytope defined by a reasonable amount of affine inequalities.

In this dissertation a robust solution refers to a robust feasible solution of the following optimization program:

$$\begin{aligned} \underset{\boldsymbol{\alpha}}{\text{minimize}} \quad & \boldsymbol{\alpha}^{\boldsymbol{T}}\boldsymbol{G}\boldsymbol{\alpha} + \boldsymbol{h}\boldsymbol{\alpha} + w, \\ \text{subject to} \quad & \boldsymbol{M}\boldsymbol{\alpha} \leq \boldsymbol{p},\ \forall\{\boldsymbol{M}, \boldsymbol{p}\} \in \mathcal{U} \end{aligned} \tag{3.1}$$

Although the optimization program in (3.1) does not consider uncertainty in the objective function, this uncertainty can be included as uncertainty in the constraints using a slack variable as follows:

$$\begin{aligned} \underset{\boldsymbol{\alpha},t}{\text{minimize}} \quad & t \\ \text{subject to} \quad & \boldsymbol{\alpha}^{\boldsymbol{T}}\boldsymbol{G}\boldsymbol{\alpha} + \boldsymbol{h}\boldsymbol{\alpha} + w \leq t \\ & \boldsymbol{M}\boldsymbol{\alpha} \leq \boldsymbol{p},\ \forall\{\boldsymbol{G}, \boldsymbol{h}, w\boldsymbol{M}, \boldsymbol{p}\} \in \mathcal{U}. \end{aligned}$$

However, in this dissertation we will consider uncertainty only in the affine parameters $\boldsymbol{M}$ and $\boldsymbol{p}$, i.e., simultaneous uncertainty in the left and right hand side of the inequality constraints. It is traditional in the field to consider the uncertainty in the left and right hand side as separate cases. However, there are scenarios in which the uncertain parameters in $\boldsymbol{\beta}$ affect the left and right hand sides simultaneously. Thus, it is necessary to consider both sources of uncertainty at the same time.

The robust solution to the optimization program (3.1) depends on the definition of the uncertainty set. As discussed in Section 3.2, the uncertainty typical of the design of an SEA is well identified as a polytope. This specific form of the uncertainty set allows us to write the optimization program (3.1) as

$$\begin{aligned} \underset{\boldsymbol{\alpha}}{\text{minimize}} \quad & \boldsymbol{\alpha}^{\boldsymbol{T}}\boldsymbol{G}\boldsymbol{\alpha} + \boldsymbol{h}\boldsymbol{\alpha} + w, \\ \text{subject to} \quad & \boldsymbol{s}_i^T\boldsymbol{c}_{ui} + \boldsymbol{d}_{ci}\boldsymbol{c}_{ci} \leq 0,\ i = 1, \ldots, m.\ \forall \boldsymbol{s}_i \in \mathcal{U}_{s_i} \end{aligned} \tag{3.2}$$

where $\boldsymbol{s}_i$ is the vector with the uncertain and $\boldsymbol{d}_{ci}$ the certain coefficients in row i in $\boldsymbol{M}$ and $\boldsymbol{p}$, the vector $\boldsymbol{c}_{ui}$ includes all the optimization variables that multiply the uncertain coefficients, and $\boldsymbol{c}_{ci}$ is the vector of optimization variables that multiply coefficients that are certain. With

this notation, the subscripts ci and ui denote certain and uncertain coefficients, respectively. For example, if all the coefficients in row i are uncertain, then

$$\boldsymbol{s}_i^T = \left[\boldsymbol{m}_i^T,\ p_i\right],\ \boldsymbol{c}_{ui} = \begin{bmatrix}\boldsymbol{\alpha}\\ -\mathbf{1}\end{bmatrix},\ \boldsymbol{d}_{ci} = 0,\ \boldsymbol{c}_{ic} = 0.$$

The distinction between certain and uncertain portions of the constraints will be useful to avoid redundant constraints for the robust solution of (3.1). Redundant constraints may lead to unnecessary infeasibility of the optimization program. The uncertainty set is defined as

$$\mathcal{U}_{s_i} = \{\boldsymbol{s}_i | \boldsymbol{D}_{u_i}\boldsymbol{s}_i \leq \boldsymbol{d}_{u_i}\},\ \text{and}\ \mathcal{U}_s = \mathcal{U}_{s_1} \times \ldots \times \mathcal{U}_{s_m}.$$

Thus the optimization program in (3.2) is equivalent to:

$$\begin{aligned} \underset{\boldsymbol{\alpha}}{\text{minimize}} \quad & \boldsymbol{\alpha}^{\boldsymbol{T}}\boldsymbol{G}\boldsymbol{\alpha} + \boldsymbol{h}\boldsymbol{\alpha} + w, \\ \text{subject to} \quad & \begin{bmatrix}\underset{\boldsymbol{s}_i}{\text{maximize}} & \boldsymbol{s}_i^T\boldsymbol{c}_{ui} \\ \text{subject to} & \boldsymbol{D}_{u_i}\boldsymbol{s}_i \leq \boldsymbol{d}_i,\end{bmatrix} + \boldsymbol{d}_{ci}\boldsymbol{c}_{ci} \leq 0,\ i = 1, \ldots, m. \end{aligned} \tag{3.3}$$

which represents a min-max optimization program. The computationally tractable solution of (3.3) relies on a transformation from the min-max to a min-min program using the Lagrange dual of the inner maximization program. Note that maximizing an objective function is equivalent to minimizing the negative value of the function. With this in mind we formulate the Lagrange dual as follows

$$L_i(\boldsymbol{s}_i, \boldsymbol{\lambda}) = -\boldsymbol{c}_{ui}^T\boldsymbol{s}_i + \boldsymbol{\lambda}^T(\boldsymbol{D}_{u_i}\boldsymbol{s}_i - \boldsymbol{d}_i).$$

The dual function is then

$$\begin{aligned} g(\boldsymbol{\lambda}) &= \inf_{\boldsymbol{s}_i} L_i(\boldsymbol{s}_i, \boldsymbol{\lambda}) \\ &= \inf_{\boldsymbol{s}_i} \left[\left(-\boldsymbol{c}_{ui}^T + \boldsymbol{\lambda}^T\boldsymbol{D}_{u_i}\right)\boldsymbol{s}_i\right] - \boldsymbol{\lambda}^T\boldsymbol{d}_i, \end{aligned}$$

which is an affine function of $\boldsymbol{s}_i$. This function is unbounded below when

$$-\boldsymbol{c}_{ui}^T + \boldsymbol{\lambda}^T\boldsymbol{D}_{u_i} \neq \mathbf{0}.$$

Otherwise, the infimum of the function is equal to $-\boldsymbol{\lambda}^T\boldsymbol{d}_i$. Note that the Lagrange dual and the primal program are equivalent due to zero duality gap; the program is convex and satisfies

Slater's condition (Boyd and Vandenberghe, 2004, p. 226). Thus, using the Lagrange dual of the inner maximization program, we can rewrite (3.3) as

$$\begin{aligned} \underset{\boldsymbol{\alpha}}{\text{minimize}} \quad & \boldsymbol{\alpha}^{\boldsymbol{T}}\boldsymbol{G}\boldsymbol{\alpha} + \boldsymbol{h}\boldsymbol{\alpha} + w, \\ \text{subject to} \quad & \begin{bmatrix} \underset{\boldsymbol{\lambda}_i}{\text{minimize}} & \boldsymbol{d}_i^T\boldsymbol{\lambda}_i \\ \text{subject to} & \boldsymbol{\lambda}_i \geq 0 \\ & \boldsymbol{D}_{u_i}^T\boldsymbol{\lambda}_i = \boldsymbol{c}_{ui} \end{bmatrix} + \boldsymbol{d}_{ci}\boldsymbol{c}_{ci} \leq 0, \ i = 1, \ldots, m. \end{aligned}$$

which is equivalent to

$$\begin{aligned} \underset{\boldsymbol{\alpha},\boldsymbol{\lambda}_i}{\text{minimize}} \quad & \boldsymbol{\alpha}^{\boldsymbol{T}}\boldsymbol{G}\boldsymbol{\alpha} + \boldsymbol{h}\boldsymbol{\alpha} + w, \\ \text{subject to} \quad & \boldsymbol{d}_i^T\boldsymbol{\lambda}_i + \boldsymbol{d}_{ci}\boldsymbol{c}_{ci} \leq 0 \\ & \boldsymbol{\lambda}_i \geq 0 \\ & \boldsymbol{D}_{u_i}^T\boldsymbol{\lambda}_i = \boldsymbol{c}_{ui}, \ i = 1, \ldots, m. \end{aligned} \tag{3.4}$$

The programs are equivalent because the optimal point in (3.4) is also optimal and feasible in (3.3). The optimization program in (3.4) is also known as the robust counterpart of (3.3) and provides a computationally tractable solution for the original uncertain program. However, this robust counterpart could quickly increase the number of constraints, as it requires m Lagrange variables per row of the matrix $\boldsymbol{M}$, where m is the number of uncertain terms in $\boldsymbol{M}$ and $\boldsymbol{p}$.

Example

As an example of the previous result, we solve the following uncertain optimization program:

$$\begin{aligned} \underset{\boldsymbol{\alpha}}{\text{minimize}} \quad & \alpha_1^2 + \alpha_2^2, \\ \text{subject to} \quad & a\alpha_1 \geq 2, \ a \in [1, \ 10], \\ & \alpha_1 \leq b, \ b \in [3, \ 4]. \end{aligned} \tag{3.5}$$

We can get to the optimal solution to this program by inspection, which we will compare against the solution obtained with the formulation in (3.4). The unconstrained cost function has an optimal point at $\boldsymbol{\alpha} = \boldsymbol{0}$. Because the function is monotonically increasing, the optimal point of the constrained program is the closest feasible point to $\boldsymbol{\alpha} = \boldsymbol{0}$. Considering the worst possible case for the uncertain coefficients, i.e., $a = 1, \ b = 3$, the robust feasible solution is, by inspection, $\alpha_1 = 2, \ \alpha_2 = 0$ (Fig. 3.1). Let's formulate the robust counterpart to arrive

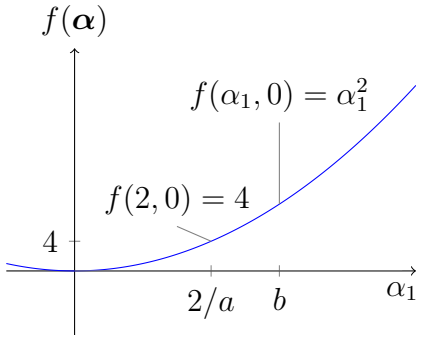

Figure 3.1. Uncertain quadratic optimization program. The coefficients, constraints, and cost function are defined in (3.5). The optimal value ($f(2,0) = 4$) is at the boundary of the feasible region ($2/a \leq \alpha_1 \leq b$).

at the same solution. First, we write the optimization program according to (3.3)

$$\begin{aligned}
&\underset{\boldsymbol{\alpha}}{\text{minimize}} \quad \boldsymbol{\alpha}^T \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{\alpha} \\
&\text{subject to} \quad \begin{bmatrix} \underset{a}{\text{maximize}} & -a\alpha_1 \\ \text{subject to} & \begin{bmatrix} 1 \\ -1 \end{bmatrix} - a \leq \begin{bmatrix} 1 \\ 10 \end{bmatrix} \end{bmatrix} + 2 \leq 0, \\
&\qquad\qquad \begin{bmatrix} \underset{b}{\text{maximize}} & -b \\ \text{subject to} & \begin{bmatrix} 1 \\ -1 \end{bmatrix} - b \leq \begin{bmatrix} -3 \\ 4 \end{bmatrix} \end{bmatrix} + \alpha_1 \leq 0,
\end{aligned} \tag{3.6}$$

Using the notation in (3.4), the robust counterpart is

$$\begin{aligned}
\underset{\boldsymbol{\alpha}, \boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2}{\text{minimize}} \quad & \boldsymbol{\alpha}^T \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{\alpha} \\
\text{subject to} \quad & \begin{bmatrix} -1 & 10 \end{bmatrix} \boldsymbol{\lambda}_1 + 2 \leq 0 \\
& \boldsymbol{\lambda}_1 \geq \mathbf{0} \\
& \begin{bmatrix} 1 & -1 \end{bmatrix} \boldsymbol{\lambda}_1 = \alpha_1 \\
& \begin{bmatrix} -3 & 4 \end{bmatrix} \boldsymbol{\lambda}_2 + \alpha_1 \leq 0 \\
& \boldsymbol{\lambda}_2 \geq \mathbf{0} \\
& \begin{bmatrix} 1 & -1 \end{bmatrix} \boldsymbol{\lambda}_2 = 1
\end{aligned}$$

This is a standard quadratic program and can be solved numerically using solvers like Gurobi or Mosek (Gurobi Optimization, 2018; MOSEK-ApS, 2019). The optimal point matches our results by inspection, $\alpha_1 = 2, \alpha_2 = 0$. Of course, the complexity of this particular optimization program did not need the reformulation in (3.4). However, the method will be extremely useful for more complex programs, which normally involve a higher number of uncertain constraints as in our optimization problem for the design of SEAs. The application of the robust counterpart formulation for the design of robust SEAs will be the topic of discussion in the following sections.

3.2 Sources of uncertainty for SEAs

In general, every coefficient in the equations of motion (1.1), (1.2) is subject to uncertainty; however, we will focus on the parameters that are more significant to the application of rehabilitation robotics, i.e., powered prosthetic legs and exoskeletons. The formulation in (3.4) is broad enough to accommodate sources of uncertainty different from the ones considered in this work. For the optimization programs in 2.27 and 2.13, we deal with the following five sources of uncertainty:

1. Manufacturing of the spring. Uncertainty due to manufacturing of the spring is typical given the limited manufacturing tolerance of suppliers. For instance, the University of Michigan (collaborators in the NSF grant that funded this dissertation) reported limited accuracy with the manufacturing of their rotational springs (Azocar et al., 2018). Their rotational springs follow a disk design that comprises 24 parallel cantilever springs organized radially (Fig. 3.2). Each disk has a (constant) stiffness $\approx 100\,\mathrm{N{\cdot}m/rad}$ and is 4.3 mm thick. However, manufactured springs have an actual stiffness with a standard deviation of about $\pm$ 10 % from the desired stiffness value (Azocar et al., 2018).

2. Unmodeled dynamics, i.e., uncertain torque in the motor dynamics. This torque corresponds to $\boldsymbol{\tau}_{\text{unc}}$, which appears now in the following equations of motion:

$$\begin{aligned}\boldsymbol{\tau}_m &= I_m\ddot{\boldsymbol{q}}_m + b_m\dot{\boldsymbol{q}}_m - \frac{\boldsymbol{\tau}_s}{\eta r} + \boldsymbol{\tau}_{\text{unc}} \\ &= \boldsymbol{e} + \boldsymbol{F}\boldsymbol{\alpha},\end{aligned}$$

where

$$\boldsymbol{e} = \left(I_m\boldsymbol{c} + b_m\boldsymbol{a} - \frac{\boldsymbol{\tau}_s}{\eta r} + \boldsymbol{\tau}_{\text{unc}}\right). \tag{3.7}$$

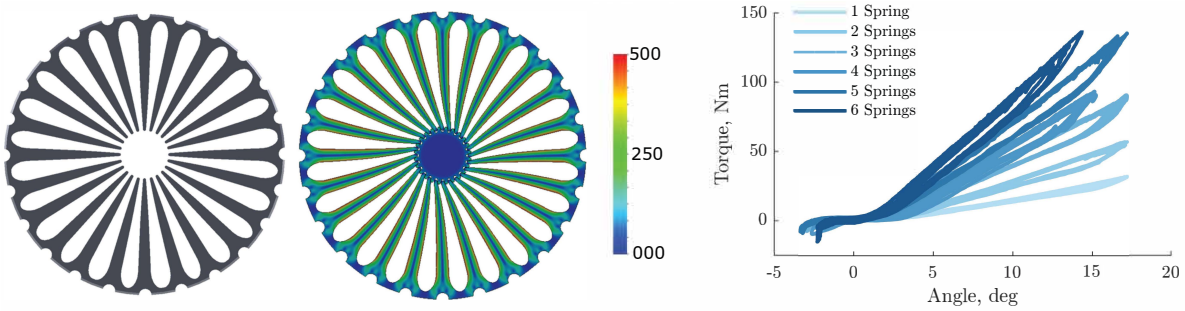

Figure 3.2. Linear spring disk design with FEA showing Von Misses stress (MPa) during loading

This new definition of $\boldsymbol{e}$ replaces the definition in (2.17) and will be used for the rest of this dissertation. The unmodeled dynamics torque, $\boldsymbol{\tau}_{\text{unc}}$, lumps unmodeled effects at the motor and load side such as cogging torque and friction.

3. Efficiency of the transmission. This efficiency depends on the specific kind of transmission, e.g., spur gears, bevel gears, harmonic drive, etc. In the case of spur gear pairs, a strong dependency in load torque exists, at low torques efficiency decreases quickly. Regarding speed, low speeds yield higher efficiency (Verstraten et al., 2015). In our model the efficiency is crudely modeled as a constant multiplicative factor, accounting for uncertainty in this factor is critical for a realistic design.

4. Kinematics of the load.

5. Kinetics of the load. In rehabilitation robotics one of the biggest sources of uncertainty is the definition of the motion tasks. Reference kinematics and kinetics vary considerably in practice as users change walking speed; inclination during locomotion; their mass, for instance, by carrying a backpack; or perform custom tasks that deviate from traditional operation, e.g., dancing, jumping, or adapt their natural gait due to irregular terrain. For instance, the joint position of a human ankle during level ground walking varies with a standard deviation of $\pm 5^{\circ}$ (Winter, 1983),

3.3 Robust design of energy-efficient linear SEAs

The previous sources of uncertainty impact the solution of the optimization program (2.13) for the design of linear SEAs. Specifically, we consider the effect of uncertainty in the feasibility of the constraints. The goal is to prescribe the right value of compliance that minimizes energy consumption and satisfies the constraints for every possible realization

of uncertainty within the uncertainty sets. Uncertainty in our formulation means that the reference kinematics and kinetics of the load, the manufacturing accuracy of the spring, the efficiency of the transmission, and the unmodeled dynamics are not precisely known but are restricted to belong to an uncertainty set, $\mathcal{U}$. In our formulation, $\mathcal{U}$ is defined as the Cartesian product

$$\mathcal{U} = \mathcal{U}_{q_l} \times \mathcal{U}_{\dot{q}_l} \times \mathcal{U}_{\ddot{q}_l} \times \mathcal{U}_m \times \mathcal{U}_\eta \times \mathcal{U}_{\tau_u} \times \mathcal{U}_d,$$

where the uncertainty sets $\mathcal{U}_{q_l}$, $\mathcal{U}_{\dot{q}_l}$, $\mathcal{U}_{\ddot{q}_l}$, $\mathcal{U}_m$, $\mathcal{U}_\eta$, $\mathcal{U}_{\tau_u}$, and $\mathcal{U}_d$ express all the possible realizations for the load position, velocity, and acceleration; the multiplicative factor of the load torque; the efficiency of the transmission; the unmodeled dynamics; and the manufacturing accuracy of the spring respectively.

For the position of the load, the uncertainty set is defined as follows:

$$\mathcal{U}_{q_l} = \{\boldsymbol{q}_l \in \mathbb{R}^n : \bar{\boldsymbol{q}}_l - \mathbf{1}\varepsilon_{q_l} \leq \boldsymbol{q}_l \leq \bar{\boldsymbol{q}}_l + \mathbf{1}\varepsilon_{q_l}\},$$

where $\bar{\boldsymbol{q}}_l \in \mathbb{R}^n$ and $\varepsilon_{q_l} \in \mathbb{R}$ represent the nominal load trajectory and uncertainty of the load position, respectively. In other words, the position of the load, $\boldsymbol{q}_l$, is within $\bar{\boldsymbol{q}}_l \pm \mathbf{1}\varepsilon_{q_l}$. Using the respective nominal and uncertainty values $(\dot{\bar{\boldsymbol{q}}}_l, \ddot{\bar{\boldsymbol{q}}}_l, \bar{\eta}, \bar{\tau}_u, \varepsilon_{\dot{q}_l}, \varepsilon_{\ddot{q}_l}, \varepsilon_\eta, \varepsilon_{\tau_u})$, we use the same definition for $\mathcal{U}_{\dot{q}_l}, \mathcal{U}_{\ddot{q}_l}, \mathcal{U}_\eta$, and $\mathcal{U}_{\tau_u}$. Uncertainty in the manufacturing of the spring is defined as the factor $(1 \pm \varepsilon_d)$ that multiplies the spring compliance. Because it is a multiplicative factor, uncertainty in the manufacturing of the spring is equivalent to uncertainty in the coefficient vector $\boldsymbol{d}$, in (2.11). Therefore the corresponding uncertainty set is defined by

$$\mathcal{U}_d = \{\boldsymbol{d} \in \mathbb{R}^p : \boldsymbol{d} - \varepsilon_d|\boldsymbol{d}| \leq \boldsymbol{d} \leq \boldsymbol{d} + \varepsilon_d|\boldsymbol{d}|\},$$

where p is the number of constraints. Inequalities and absolute values for vectors are element-wise. This uncertainty in the manufacturing of the spring implies that its stiffness is in the set

$$k \in \{k \in \mathbb{R} : [(1+\varepsilon_d)\alpha]^{-1} \leq k \leq [(1-\varepsilon_d)\alpha]^{-1}\}.$$

Uncertainty in the kinetic reference trajectories is defined by a nominal value and a uncertain multiplicative factor. Precisely, the reference torque of the load $\boldsymbol{\tau}_l$ is considered to be $\boldsymbol{\tau}_l = m(\boldsymbol{\tau}_{l/m})$, where $\boldsymbol{\tau}_{l/m}$ is a nominal value of $\boldsymbol{\tau}_l$ per unit of m. Our uncertain multiplicative factor, m, could be any element within the set

$$\mathcal{U}_m = \{m \in \mathbb{R} : 0 < \bar{m} - \varepsilon_m \leq m \leq \bar{m} + \varepsilon_m\},$$

where $\bar{m} \in \mathbb{R}$ is the nominal value of m and $\varepsilon_m \in \mathbb{R}$ is its corresponding uncertainty. In other words, $\boldsymbol{\tau}_l = (\bar{m} \pm \varepsilon_m)\boldsymbol{\tau}_{l/m}$.

The robust formulation of the constraints

A robust feasible design should satisfy the constraints (2.11) for all possible realizations of the uncertainty within the uncertainty set. Note that the uncertainty in the manufacturing of the spring manifests as uncertainty in the vector $\boldsymbol{d}$ in (2.11). Thus, a robust feasible design results in an optimal selection of α that satisfies

$$\boldsymbol{d}\alpha \leq \boldsymbol{e_l},\ \forall \boldsymbol{q}_l, \dot{\boldsymbol{q}}_l, \ddot{\boldsymbol{q}}_l, m, \eta, \tau_u, \boldsymbol{d} \in \mathcal{U}. \tag{3.8}$$

Because $\alpha > 0$, a robust feasible solution can be obtained as a special case of (3.4). When $\alpha > 0$, constraints of the form

$$a_i\alpha \leq b_i,$$

are robustly solved when choosing a_i and b_i such that

$$\frac{b_i}{a_i} \leq \frac{b_{i_u}}{a_{i_u}},\ \forall\ b_{i_u},\ a_{i_u} \in \mathcal{U}.$$

In our formulation, this is equivalent to

$$\bar{\boldsymbol{d}}\alpha \leq \underline{\boldsymbol{e_l}}, \tag{3.9}$$

where $\bar{\boldsymbol{d}}$ and $\underline{\boldsymbol{e_l}}$ are the vectors that represent the worst case representation of the uncertainty. These vectors are defined as follows:

$$\bar{\boldsymbol{d}} = \boldsymbol{d} + \varepsilon_d|\boldsymbol{d}|,\ \underline{\boldsymbol{e_l}} = [\underline{\boldsymbol{e_l}}_1^T, \underline{\boldsymbol{e_l}}_2^T, \underline{\boldsymbol{e_l}}_3^T]^T, \tag{3.10}$$

where

$$\begin{aligned}
\underline{\boldsymbol{e_l}}_1 &= \mathbf{1}\frac{\delta_s}{\bar{m} + \varepsilon_m},\\
\underline{\boldsymbol{e_l}}_2 &= \begin{bmatrix} \boldsymbol{\tau}_{l/m} \\ -\boldsymbol{\tau}_{l/m} \end{bmatrix} \frac{1}{(\bar{\eta} \pm \varepsilon_\eta)} r + \boldsymbol{f}\frac{1}{\bar{m} \pm \varepsilon_m},\\
\boldsymbol{f} &= \begin{bmatrix} -I_m r(\ddot{\boldsymbol{q}}_l + \boldsymbol{\varepsilon}_{\ddot{q}_l}) - b_m r(\dot{\boldsymbol{q}}_l + \boldsymbol{\varepsilon}_{\dot{q}_l}) + \mathbf{1}(\underline{\tau_u} + \tau_{max}) \\ I_m r(\ddot{\boldsymbol{q}}_l - \boldsymbol{\varepsilon}_{\ddot{q}_l}) + b_m r(\dot{\boldsymbol{q}}_l - \boldsymbol{\varepsilon}_{\dot{q}_l}) + \mathbf{1}(\underline{\tau_u} + \tau_{max}) \end{bmatrix},\\
\underline{\boldsymbol{e_l}}_3 &= [\underline{\boldsymbol{e}}_{3a}^T, \underline{\boldsymbol{e}}_{3b}^T, \underline{\boldsymbol{e}}_{3c}^T, \underline{\boldsymbol{e}}_{3d}^T]^T,\\
\underline{\boldsymbol{e}}_{3a} &= \frac{\boldsymbol{\tau}_{l/m}}{(\bar{\eta} \pm \varepsilon_\eta) r} + (-I_m(\ddot{\boldsymbol{q}}_l + \varepsilon_{\ddot{q}_l}) r - b_m(\dot{\boldsymbol{q}}_l + \varepsilon_{\dot{q}_l}) r + \mathbf{1}\left(v_{in}\frac{k_t}{R} + \underline{\tau_u}\right) - \dots\\
&\quad \dots \frac{k_t^2 r}{R}(\dot{\boldsymbol{q}}_l + \varepsilon_{\dot{q}_l}))\frac{1}{(\bar{m} \pm \varepsilon_m)},
\end{aligned}$$

and the values for $\underline{\boldsymbol{e}}_{3b}, \underline{\boldsymbol{e}}_{3c}$, and $\underline{\boldsymbol{e}}_{3d}$ are defined using positive and negative values of torque and speed in the definition of the torque-speed constraints. The sign of $1/(\bar{m} \pm \varepsilon_m)$ depends on the elements of the vector that it multiplies, as the multiplication applies element-wise. When the element of the vector is positive, then the multiplication factor becomes $1/(\bar{m} + \varepsilon_m)$; when the element is negative, $1/(\bar{m} - \varepsilon_m)$. The same idea applies to $1/(\bar{m} \pm \varepsilon_\eta)$, which describes the worst possible scenario to satisfy the inequality (3.9). In conclusion, we can find the stiffness constant that minimizes energy consumption and satisfies the actuator constraints despite uncertainty by solving the following quadratic program:

$$\begin{aligned} \underset{\alpha}{\text{minimize}} \quad & a\alpha^2 + b\alpha + c, \\ \text{subject to} \quad & \bar{\boldsymbol{d}}\alpha \leq \underline{\boldsymbol{e}_{\boldsymbol{l}}} \end{aligned} \tag{3.11}$$

The content of this section has been adapted from (Bolivar Nieto et al., 2019).

3.4 Robust design of energy-efficient nonlinear SEAs

For the robust design on nonlinear SEAs, we account for the same sources of uncertainty as described in Section 3.2. The first step for a robust formulation is to define the uncertainty sets. In particular, the additive uncertainty in the spring compliance manifests in the vector $\boldsymbol{p}$ as defined in (2.27), which is constrained to belong to the set

$$\mathcal{U}_{\alpha_n} = \{\boldsymbol{p} \in \mathbb{R}^m : \bar{\boldsymbol{p}} - \text{abs}(\boldsymbol{M}\boldsymbol{1})\varepsilon_{\alpha_a} \leq \boldsymbol{p} \leq \bar{\boldsymbol{p}} + \text{abs}(\boldsymbol{M}\boldsymbol{1})\varepsilon_{\alpha_a}\},$$

where $\bar{\boldsymbol{p}}$ is the nominal right hand side vector of the inequality constraint and $\varepsilon_{\alpha_a} \in \mathbb{R}$ the uncertain value in compliance. The abs() value applies element-wise. The unmodeled dynamics torque is defined as

$$\tau_{\text{unc}_i} \in \mathcal{U}_{\tau_{\text{unc}i}} = \{\tau_{\text{unc}_i} \in \mathbb{R} : \ -\varepsilon_{\tau_{\text{unc}i}} \leq \tau_{\text{unc}_i} \leq \varepsilon_{\tau_{\text{unc}i}}\}, \ i = 1, \ldots, n.$$

Where $\boldsymbol{\varepsilon}_{\tau_{\text{unc}}} \in \mathbb{R}^n$ is the vector that defines its uncertainty. For the case of the efficiency of the transmission:

$$\eta \in \mathcal{U}_{\eta_n} = \{\eta \in \mathbb{R}_+ : \bar{\eta}(1 - \varepsilon_{\eta_n}) \leq \eta \leq \bar{\eta}(1 + \varepsilon_{\eta_n}), \ 0 \leq \epsilon_{\eta_n} \leq 1\}.$$

In other words, it has multiplicative uncertainty,

$$\eta = \bar{\eta}(1 \pm \epsilon_\eta).$$

where $\bar{\eta}$ is the nominal value and ε_η the uncertain multiplicative factor.

The uncertainty sets for the kinematics and kinetics take a similar approach. Additive and multiplicative uncertainty in the position, velocity, acceleration of the load, and the torque of the load are considered as follows:

$$\begin{aligned}
\boldsymbol{q}_l &= \bar{\boldsymbol{q}}_l(1 \pm \varepsilon_{q_{l_m}}) \pm \boldsymbol{\varepsilon}_{\boldsymbol{q_{l_a}}} \\
\dot{\boldsymbol{q}}_l &= \dot{\bar{\boldsymbol{q}}}_l(1 \pm \varepsilon_{\dot{q}_{l_m}}) \pm \boldsymbol{\varepsilon}_{\boldsymbol{\dot{q}_{l_a}}} \\
\ddot{\boldsymbol{q}}_l &= \ddot{\bar{\boldsymbol{q}}}_l(1 \pm \varepsilon_{\ddot{q}_{l_m}}) \pm \boldsymbol{\varepsilon}_{\boldsymbol{\ddot{q}_{l_a}}} \\
\boldsymbol{\tau}_s &= \bar{\boldsymbol{\tau}}_s(1 \pm \varepsilon_{\tau_{s_m}}) \pm \boldsymbol{\varepsilon}_{\boldsymbol{\tau_{s_a}}}.
\end{aligned}$$

The notation follows the same principle as in the other sources of uncertainty. Nominal and uncertain values are defined with a bar and epsilon respectively, e.g., $\bar{\boldsymbol{q}}_l$ is the nominal value of load position, $\varepsilon_{\dot{q}_{l_m}}$ is the multiplicative uncertainty in load velocity, and $\boldsymbol{\varepsilon}_{\boldsymbol{\ddot{q}_{l_a}}}$ the additive uncertainty in load acceleration. For the linear and nonlinear design frameworks, we assume that the uncertainty in position is independent of the uncertainty in its derivatives and integrals. The same applies for the uncertainty in velocities and accelerations.

This particular selection of uncertainty sets allows us to lump all the uncertainty in the right hand side of the inequality, $\boldsymbol{M\alpha} \leq \boldsymbol{p}$, in (2.26). With all the uncertainty in the right hand side, the robust counterpart could be easily identified by replacing $\boldsymbol{p}$ with its lowest possible value due to uncertainty.

The robust formulation of the constraints

To include the uncertainty due to limited accuracy during manufacturing of the spring and uncertain unmodeled dynamics torque, we write $\boldsymbol{p_1}$ as follows:

$$\boldsymbol{p_1} = \boldsymbol{p} - \operatorname{abs}(\boldsymbol{M})\,\mathbf{1}\varepsilon_{\alpha_a} - \begin{bmatrix} \mathbf{0} \\ \mathbf{1}_{6n\times 1} \end{bmatrix} \varepsilon_{\tau_{\text{unc}}} \tag{3.12}$$

To account for the uncertainty in the efficiency of the transmission we modify the vector $\boldsymbol{e}$ in (3.7) to be

$$\boldsymbol{e} = \left(I_m \boldsymbol{c} + b_m \boldsymbol{a} - \frac{\boldsymbol{\tau}_s}{\bar{\eta}(1 \pm \operatorname{sign}(\boldsymbol{\tau}_s)\varepsilon_{\eta_n})r} + \boldsymbol{\tau}_{\text{unc}} \right) \tag{3.13}$$

where the division is element-wise and the sign of $\pm$ prescribes the lowest possible value of $\boldsymbol{e}$. To account for the uncertainty in the kinematics we define $\boldsymbol{p_2}$ as follows:

$$\boldsymbol{p_2} = \boldsymbol{p_1} - \begin{bmatrix} \mathbf{0} \\ b_m r(\varepsilon_{\dot{q}_{l_m}} \text{abs}(\dot{\tilde{\boldsymbol{q}}}_l) + \mathbf{1}_{n\times 1}\varepsilon_{\dot{q}_{l_a}}) + I_m r(\varepsilon_{\ddot{q}_{l_m}} \text{abs}(\ddot{\tilde{\boldsymbol{q}}}_l) + \mathbf{1}_{n\times 1}\varepsilon_{\ddot{q}_{l_a}}) \\ b_m r(\varepsilon_{\dot{q}_{l_m}} \text{abs}(\dot{\tilde{\boldsymbol{q}}}_l) + \mathbf{1}_{n\times 1}\varepsilon_{\dot{q}_{l_a}}) + I_m r(\varepsilon_{\ddot{q}_{l_m}} \text{abs}(\ddot{\tilde{\boldsymbol{q}}}_l) + \mathbf{1}_{n\times 1}\varepsilon_{\ddot{q}_{l_a}}) \\ (b_m r + \dfrac{k_t^2}{R} r)(\varepsilon_{\dot{q}_{l_m}} \text{abs}(\dot{\tilde{\boldsymbol{q}}}_l) + \mathbf{1}_{n\times 1}\varepsilon_{\dot{q}_{l_a}}) + I_m r(\varepsilon_{\ddot{q}_{l_m}} \text{abs}(\ddot{\tilde{\boldsymbol{q}}}_l) + \mathbf{1}_{n\times 1}\varepsilon_{\ddot{q}_{l_a}}) \\ \vdots \\ (b_m r + \dfrac{k_t^2}{R} r)(\varepsilon_{\dot{q}_{l_m}} \text{abs}(\dot{\tilde{\boldsymbol{q}}}_l) + \mathbf{1}_{n\times 1}\varepsilon_{\dot{q}_{l_a}}) + I_m r(\varepsilon_{\ddot{q}_{l_m}} \text{abs}(\ddot{\tilde{\boldsymbol{q}}}_l) + \mathbf{1}_{n\times 1}\varepsilon_{\ddot{q}_{l_a}}) \end{bmatrix} \tag{3.14}$$

Using n as the number of samples in the discrete-time load trajectory. Finally, we include the uncertainty in the kinetics to define the lowest possible value of $\boldsymbol{p}$ as $\underline{\boldsymbol{p}}$:

$$\underline{\boldsymbol{p}} = \boldsymbol{p_2} - \frac{1}{\bar{\eta}(1-\varepsilon_{\eta_n} r)} \begin{bmatrix} \mathbf{0} \\ \varepsilon_{\tau_{s_m}} \text{abs}(\tau_s) + \mathbf{1}_{n\times 1}\varepsilon_{\tau_{s_a}} \\ \vdots \\ \varepsilon_{\tau_{s_m}} \text{abs}(\tau_s) + \mathbf{1}_{n\times 1}\varepsilon_{\tau_{s_a}} \end{bmatrix} \tag{3.15}$$

With this definition, the robust counterpart to the optimization program (2.27) is

$$\begin{aligned} \underset{\boldsymbol{\alpha}}{\text{minimize}} \quad & \boldsymbol{\alpha}^{\boldsymbol{T}} \boldsymbol{G} \boldsymbol{\alpha} + \boldsymbol{h}\boldsymbol{\alpha} + w, \\ \text{subject to} \quad & \boldsymbol{M}\boldsymbol{\alpha} \leq \underline{\boldsymbol{p}} \end{aligned} \tag{3.16}$$

CHAPTER 4

IMPLICATIONS FOR A POWERED PROSTHETIC ANKLE [1]

This chapter applies the methodology from Chapters 2 and 3 into the series spring design for a powered prosthetic ankle. This application is of interest, as the muscle-tendon system that powers the human ankle resembles the architecture of an SEA. Most of the high energetic efficiency of this joint is due to the storage and return of elastic energy at the Achilles tendon, which connects in series with the calf muscles (Endo and Herr, 2014; Perry and Burnfield, 2010). Thus, using a spring as an Achilles tendon and a motor as the calf muscles is a biologically inspired strategy to power a prosthetic ankle. In addition, battery-powered rehabilitation robotics regard energy efficiency and robustness as important criteria for their design. The combination of all these factors makes a powered prosthetic ankle an appealing case study for this dissertation.

The architecture of an SEA offers important benefits to the actuation of robotic systems. The elastic element in an SEA decouples the reflected inertia of the rigid actuator and the inertia of the load (Hurst and Rizzi, 2008). In addition, the spring can store elastic energy and release it with enormous power. SEAs also work as a soft load cell, suitable for measuring and controlling force generation (Robinson et al., 1999). Robots using SEAs exploit these important characteristics in order to reduce the energy lost during impacts (Hurst and Rizzi, 2008), improve the safety of the human and robots (Bicchi and Tonietti, 2004), move loads with higher velocities (Braun et al., 2013), reduce energy consumption of the system (Vanderborght et al., 2006; Jafari et al., 2013; Ding and Park, 2017), and decrease peak motor power (Jafari et al., 2011, 2013; Hollander et al., 2006), so a smaller/lighter motor can be used. All these benefits are subject to the design of the SEA's elastic element and the motion task.

In terms of control, adding a series spring normally reduces the torque bandwidth of the entire actuator. However, this should be analyzed in more detail, torque bandwidth for rigid high-geared systems may be high, but the torque tracking performance is poor. Using high-gear ratios amplifies the effect of difficult to model dynamics such as backlash and friction, in the end, the control accuracy is far from ideal (Robinson, 2000). In addition,

[1] Parts of this chapter have been adapted from (Bolivar Nieto et al., 2019), © 2019 IEEE. Adapted, with permission, from Bolivar Nieto, E.A., S. Rezazadeh, and R. D. Gregg, Minimizing Energy Consumption and Peak Power of Series Elastic Actuators: a Convex Optimization Framework for Elastic Element Design. IEEE/ASME Transactions on Mechatronics. 24(3), 1334-1345.

motors operating at high velocities may be loud and noisy (Elery et al., 2018). Low-gear or direct drive systems alleviate these effects by reducing the effects of friction and backlash at the motor side, they offer outstanding torque control tracking, quiet operation because the motor operates at low speeds (Elery et al., 2018), but dissipate a significant amount of heat (Seok et al., 2015).

Despite the strong biological evidence and the known benefits of elasticity, there is no actual consensus in the robotics community about the use of springs in series to actuate the ankle joint. In general, the community acknowledges the benefits of series elasticity, but the extra mechanical complexity and mass of the spring plus attachment mechanism discourage the implementation of an SEA architecture. Prosthetic legs powered by SEAs normally justify their selection in terms of energy savings, mass reduction, improved force control, and reduction of impact loads (Rouse et al., 2014; Azocar et al., 2018; Hollander et al., 2006; Au et al., 2009). However, these designs tend to be heavier than their rigid counterparts; in addition, energy savings and satisfaction of motor speed-torque constraints are not guaranteed when the load requirements deviate from the nominal conditions. As a result, there are prosthetic leg designs that operate without a spring in series with the motor (Lenzi et al., 2018, 2019; Tran et al., 2019; Sup et al., 2008; Shultz et al., 2016; Sup et al., 2008).

In this chapter, we take advantage of the global optimal results from our methodology and compare the linear spring designs against the nonlinear ones. The chapter begins with an analysis of the speed-torque requirements for a healthy human ankle joint. These requirements become the reference load trajectories for the optimal spring design in Sections 4.3 and 4.4. Section 4.3 deals specifically with the optimal nonlinear spring neglecting uncertainty in the design. Section 4.4 incorporates uncertainty in the optimal selection of the spring. The trade-off between peak power and energy consumption is discussed in Section 4.5. The chapter concludes in Section 4.6 with a discussion on the selection of motor and reduction ratio in the context of an SEA powered ankle.

4.1 Actuator requirements for different locomotion tasks

In biomechanics, the gait cycle refers to the progression of the human lower limbs during walking. A gait cycle can be categorized into five main divisions: initial double limb stance, single limb stance, terminal double limb stance, swing, and double limb stance. In each division, the ankle-foot complex provides important functions for locomotion. Here, we summarize the critical functionality that an ideal powered prosthetic ankle should replace.

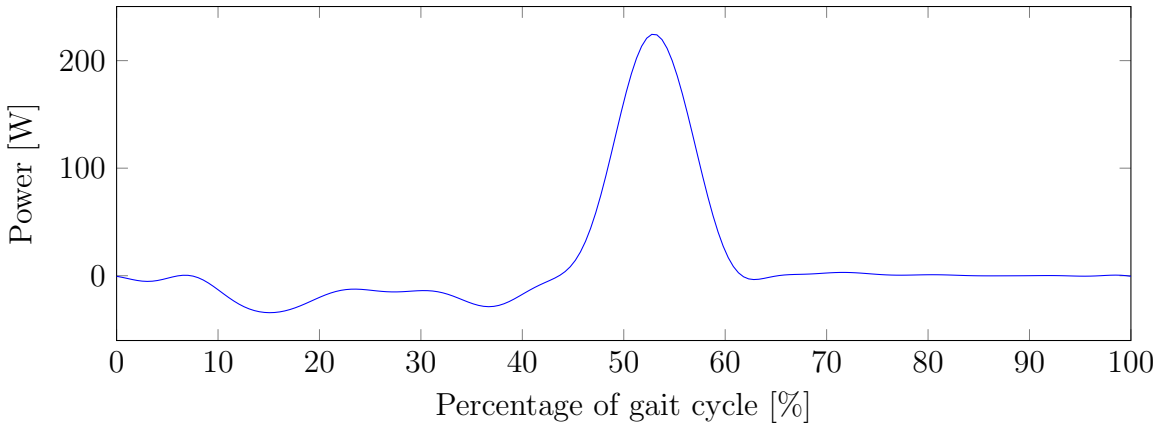

Figure 4.1. Ankle power during the gait cycle corresponding to a 69.1 kg subject, as reported in (Winter, 1983).

At initial contact or heel strike, the ankle-foot complex is positioned at 90° (neutral) to guarantee upward tilt of the forefoot. This position is critical for the subsequent loading response. During the loading phase the limb assumes rapid loading (60 % of body weight in just 2 % of the gait cycle). In this phase, the ankle works as a shock absorber. During mid-stance the main function of the ankle-foot complex is to act as a rocker for progression (Perry and Burnfield, 2010). During terminal stance, the ankle provides the highest power among all the joints and provides most of the energy required to project the body forward (45 % to 60 % of the gait cycle, Fig. 4.1). Most of that energy has been stored as elastic energy in the Achilles tendon during mid stance until heel rise. This functionality is key for adequate metabolic energy consumption during walking (Collins et al., 2015). The elastic component of the ankle has been also recognized in the literature by characterizing its torque-displacement behavior using the stiffness and quasi-stiffness of a spring (Rouse et al., 2013; Hansen et al., 2004).

A key factor for the design of a powered prosthetic ankle is its capability to resemble the position and torque profile of a healthy ankle. The Fig. 4.2 illustrates the reference position and torque trajectory for the actuator during level ground walking and running. Note that these trajectories represent the mean values for multiple individuals. Identification of the ideal reference positions and torques is still a challenge in the field, as the trajectories change over time even for the same subject (Embry et al., 2018). For example, Fig. 4.3 illustrates the effects of different walking speeds and user mass on the kinematics and kinetics of the average human ankle. These reference torques and positions are also inputs to our optimization methodology; thus, a design that is robust against these changes is of interest in practice.

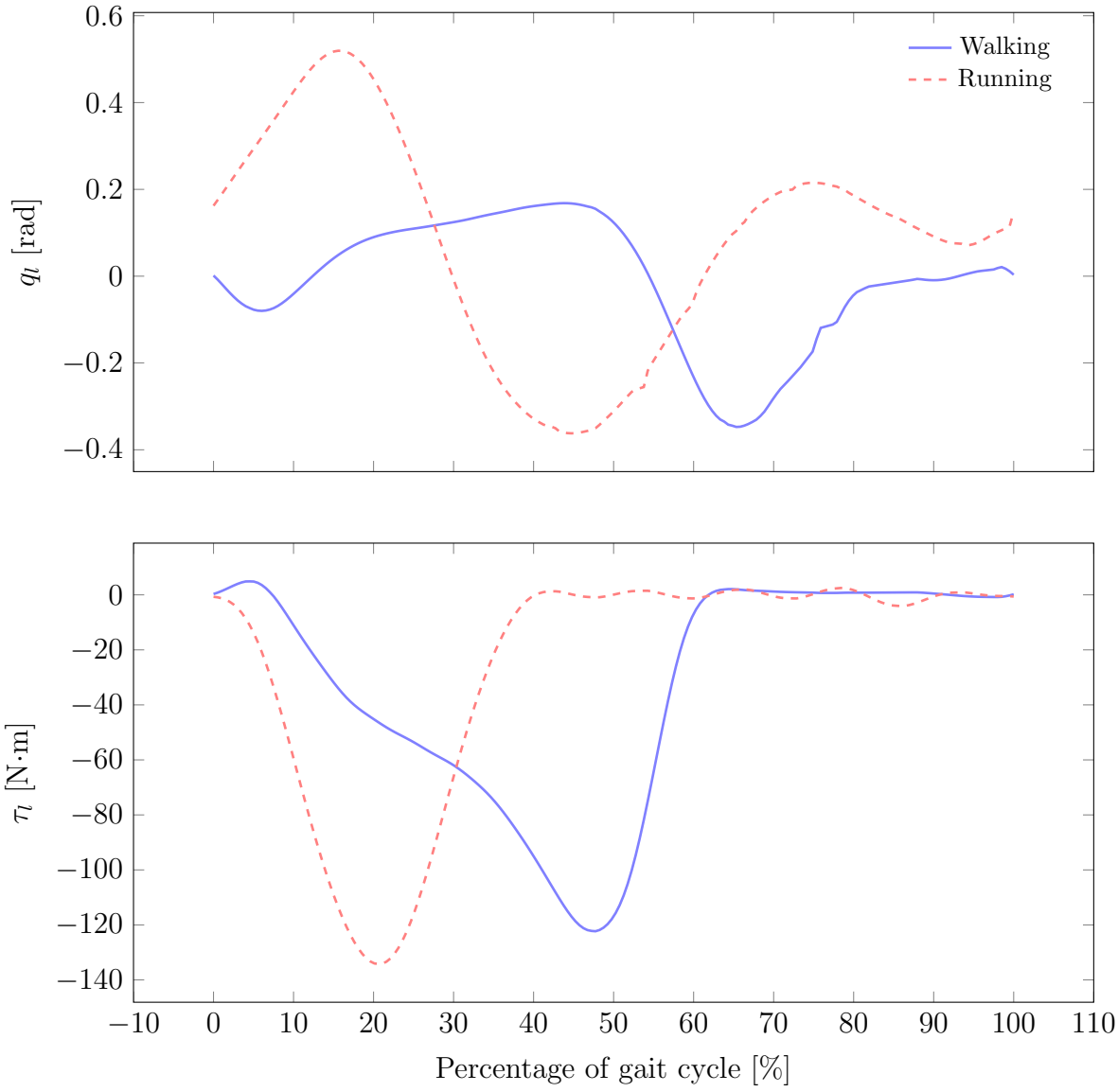

Figure 4.2. Motion of the human ankle during level ground walking (Winter, 1983) and running (Novacheck, 1998). The gait cycle begins with heel contact of one foot and finishes with the subsequent occurrence of the same foot.

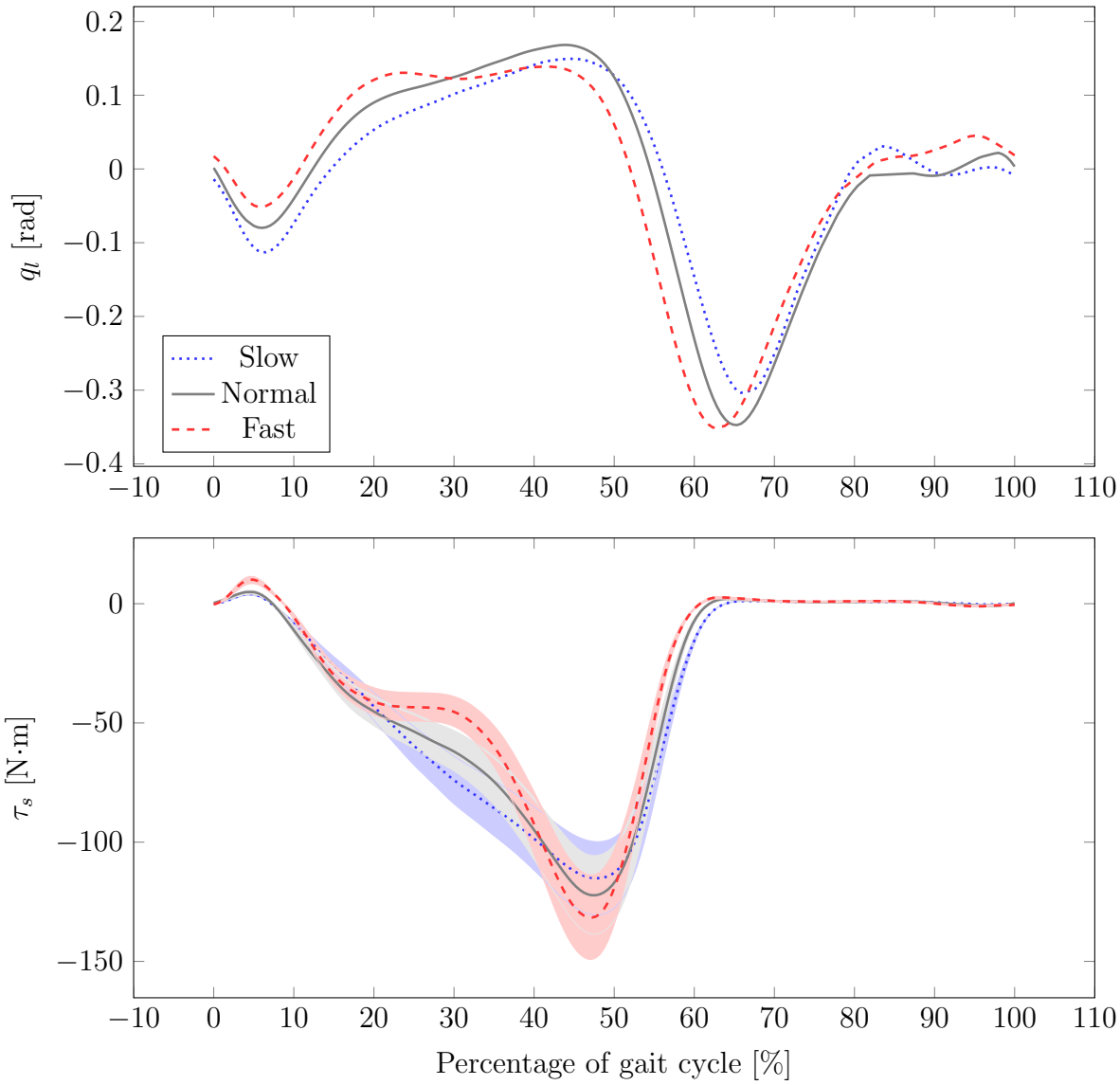

Figure 4.3. Motion of the human ankle during level ground walking as shown in (Winter, 1983). Slow, normal, and fast walking speeds are equivalent to cadences of 87, 105, and 123 steps per minute. In average, the ankle of a 75 kg subject walking at normal speed provides about 17 J during a single gait cycle. In the lower figure, translucent regions denote the minimum and maximum joint torques corresponding to 65 kg and 85 kg subjects.

4.2 Simulation setup

Each of the following case studies is based on simulation results using the following configuration. The trajectory of the load, i.e., $q_l, \dot{q}_l, \ddot{q}_l$, and τ_s in (1.2), is given and the optimization problem is numerically solved using CVX, a package for specifying and solving convex pro-

Table 4.1. Simulation parameters based on the motor ILM85x26 from RoboDrive and EC30 from Maxon motor.

Parameter	ILM85 x26	EC30	Units
Motor torque constant, k_t	0.24	0.0136	N·m/A
Motor terminal resistance, R	323	102	mΩ
Rotor inertia, I_{mr}	1.15	0.0333	kgcm^2
Rotor assembly, I_{ma}	0.131	0	kgcm^2
Motor inertia, $I_m = I_{mr} + I_{ma}$	1.246	0.0333	kgcm^2
Gear ratio, r	22	720	
Motor viscous friction, b_m	60	6.66	µN·m·s/rad
Max. motor torque, τ_{max}	8.3	0.337	N·m
Max. motor velocity, $\dot{q}_{max}$	1,500	20,000	rpm
Nominal power output,	@48 V, 410	@30 V, 300	W
Peak power output,	@(48 V, τ_{max}), 1,259	@(30 V, τ_{max}), 745	W

grams (Grant and Boyd, 2014, 2008). In all simulations, CVX executed the solver Mosek (MOSEK-ApS, 2019) with precision settings `cvx_precision best`. For the case studies, we used the parameters of a commercial high-torque frameless motor (Model: ILM 85x26, RoboDrive, Seefeld, Germany, Table 4.1) and low-torque high speed motor (Model: EC30, Maxon motor, Sachseln, Switzerland, Table 4.1), motivated by the motor selection of the first and second generation of the powered prosthetic legs at the University of Texas at Dallas (Quintero et al., 2018, 2016; Elery et al., 2018).

4.3 Minimizing energy consumption using a high reduction gearbox

As a first case study, we illustrate the design of a nonlinear SEA that minimizes energy consumption for a high-speed low-torque motor using a high reduction ratio. To facilitate the analysis, this case study does not consider uncertainty in the description of the optimization program; we will consider uncertainty in the following sections. The optimal spring is the optimal solution to the optimization program in (2.27), constraining only the maximum velocity and torque of the motor based on the parameters of the EC30 motor in Table 4.1.

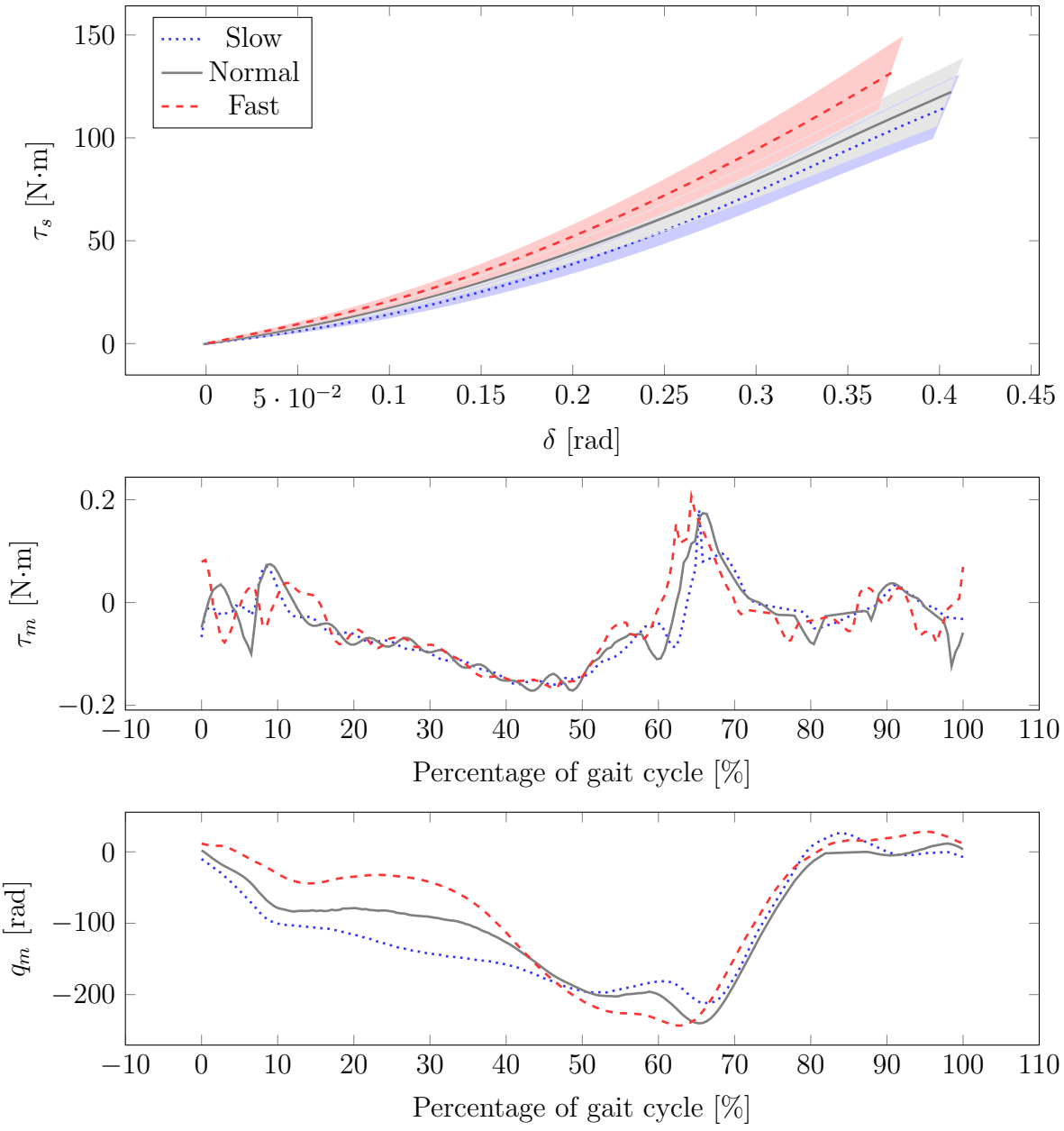

Figure 4.4. Optimization results considering motion of the ankle as the reference trajectory. Dotted, solid, and dashed lines indicate results for slow, normal, and fast level-ground walking speeds, respectively. Translucent regions denote upper and lower bounds corresponding to 85 kg and 65 kg subjects.

In particular, we analyze the ankle trajectories for slow, normal, and fast walking speeds and three subject weights: 65, 75, and 85 kg. These reference trajectories are shown in Fig. 4.3.

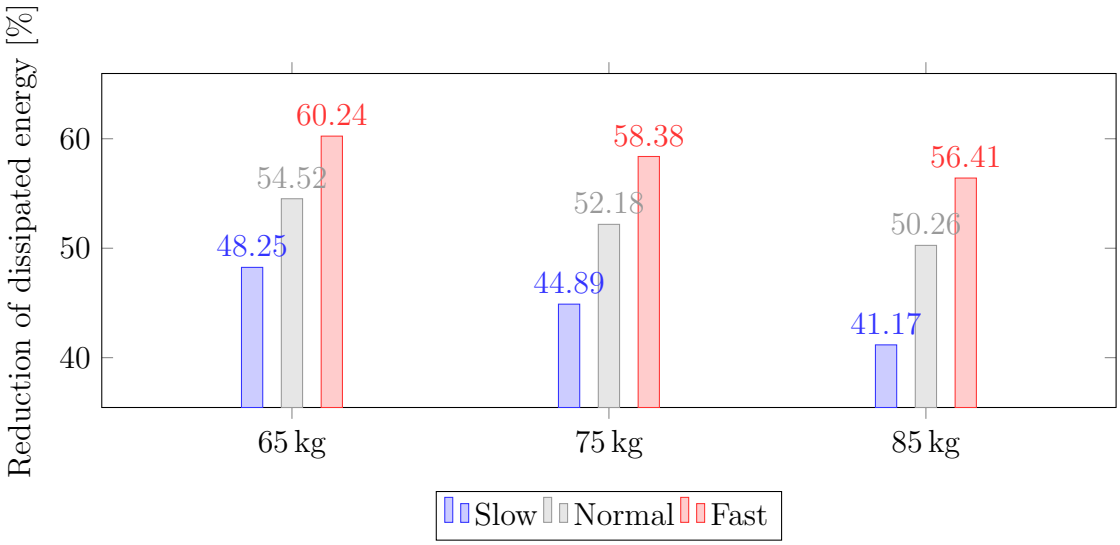

Figure 4.5. Energy savings for the ankle reference trajectory. Results for slow, normal, and fast level-ground walking for three different subject's weights.

The optimal elastic elements along with the torques and positions of the motor are illustrated in Fig. 4.4. As shown in Eqn. 1.6, the energy of the load (i.e., $\int_{t_0}^{t_f} \frac{\tau_{\text{ela}}\dot{q}_l}{\eta} dt$) is always provided by the motor regardless of the SEA's elastic element. Therefore, energy savings for an SEA will be considered as the reduction of dissipated energy between a rigid motor without elastic element and an SEA. For example, during normal speed the ankle of a 75 kg subject provides about 17 J per stride; however, the EC30 motor with the characteristics described in Table 4.1, without an elastic element, requires 33 J. The extra 16 J are dissipated in the motor's winding by Joule heating and viscous friction. In contrast, the same motor connected in series with our optimal elastic element requires about 25 J per stride. The energy dissipated will be 8 J, a reduction of about 50 % compared to a motor without an elastic element. A similar analysis for different walking speeds and subject weights is summarized in Fig. 4.5. Energy reduction is shown for all the cases considered. The optimal elastic element is nonlinear, as shown in Fig. 4.4. This indicates that, for the given electric motor and transmission, a quadratic elastic element would be the most efficient to generate the ankle motion.

4.4 Energy-efficient robust design

In this section, we apply our methods to the design of an SEA for a powered prosthetic ankle to minimize energy consumption while satisfying actuator constraints despite uncertainty.

As discussed in Section 4.1, actuator designs for powered prostheses normally use average kinetic and kinematic trajectories (Rouse et al., 2014; Au and Herr, 2008; Lenzi et al., 2018; Elery et al., 2018). However, load conditions during human locomotion vary significantly even for a single subject (Embry et al., 2016). Robust design is important in this application as human locomotion and manufacturing methods are inherently uncertain. For instance, the ankle joint position during human locomotion varies with a standard deviation of $\pm 5^{\circ}$ (Winter, 1983), and the stiffness of a manufactured spring has a standard deviation of about $\pm$ 10 % from the desired stiffness value (Azocar et al., 2018).

In our formulation, we take advantage of the connection between uncertainty in the kinematics and kinetics of the load and our definition of uncertainty sets in Section 3.3 to obtain a robust feasible design. The parameters $\varepsilon_{q_l}, \varepsilon_{\dot{q}_l}, \varepsilon_{\ddot{q}_l}$ define the uncertainty in the kinematics $\mathcal{U}_{\{q_l,\dot{q}_l,\ddot{q}_l\}}$. In this simulation case study, we define these parameters to be equal to the reported standard deviation of the joint kinematics in (Winter, 1983). Our formulation of uncertainty in the kinetics has a practical meaning in biomechanics. The reference torque of the ankle joint is traditionally normalized by the mass of the user (Winter, 2009). Because our definition of uncertainty in the kinetics is multiplicative, it is equivalent to uncertainty in the user's mass. As a result, it becomes relevant to rehabilitation and physical assistance robots where users can vary or a single user can wear additional accessories, such as backpacks. We select the uncertainty in the mass, ε_m, to be equal to the reported standard deviation of the subjects' mass in (Winter, 1983). The Fig. 4.6 illustrates the reference trajectories and corresponding bounds of uncertainty. Uncertainty in the manufacturing of the spring, ε_d, is equal to twice the standard deviation of the SEA spring stiffness of the open-source prosthetic leg at University of Michigan (Azocar et al., 2018). The uncertain torque, ε_{τ_u}, is equal to 10 % of the maximum continuous motor torque. Uncertainty in the efficiency of the transmission is based on experience, aiming for a realistic simulation case. Table 4.1 illustrates the parameters of the actuator and Table 4.2 the parameters of uncertainty.

4.4.1 Design using a linear SEA

This section describes the linear spring that solves the robust optimization in (2.13). To contextualize our results, we analyze three possible actuator designs: (a) a rigid actuator Maxon EC-30 without series elasticity, (b) an SEA using the same motor with optimal stiffness that satisfies constraints only for the nominal data, and (c) an SEA with the same motor that satisfies actuator constraints despite uncertainty using our robust formulation. Using (1.1) and (1.2) we compute the motor speed and torque trajectories considering the

Table 4.2. Uncertainty based on the variance reported in (Winter, 1983; Azocar et al., 2018).

Uncertainty in	Units
Mass, ε_m	$\pm 8.8\,\mathrm{kg}$
Reference position, ε_{q_l}	$\pm 5^\circ$
Reference velocity, $\varepsilon_{\dot{q}_l}$	$\pm 30\,\%$ rms average trajectory
Reference acceleration, $\varepsilon_{\ddot{q}_l}$	$\pm 30\,\%$ rms average trajectory
Transmission efficiency, ε_η	$\pm 20\,\%$
Unmodeled dynamics, ε_{τ_u}	$\pm 13.5\,\mathrm{mN{\cdot}m}$
Manufacturing of spring, ε_d	$\pm 20\,\%$

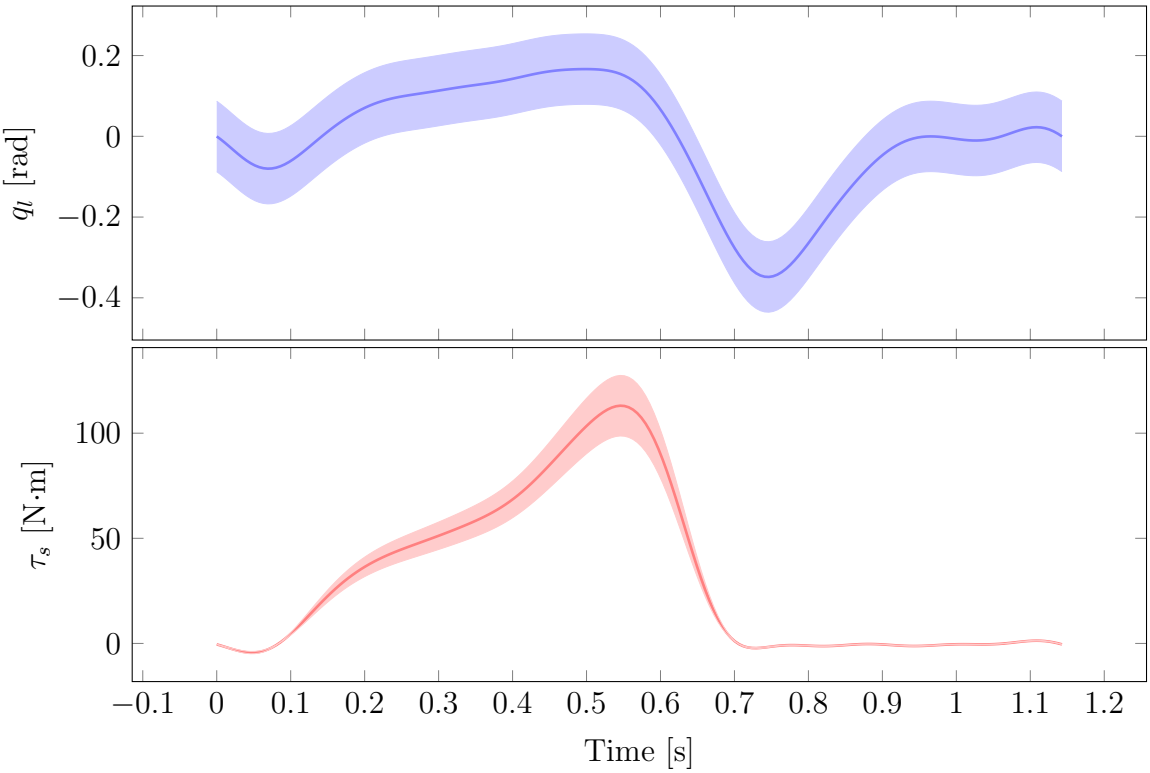

Figure 4.6. Position (top) and torque (bottom) of the human ankle during level ground walking (Winter, 1983). The solid line indicates the mean values for a 69.1 kg subject (Winter, 1983). The shaded region around the nominal trajectory illustrates the uncertainty in the position $\varepsilon_{q_l} = \pm 5^\circ$ and the mass of the subject $\varepsilon_m = \pm 8.8\,\mathrm{kg}$. This uncertainty corresponds to the standard deviation reported in (Winter, 1983).

Table 4.3. Optimization results that indicate weak trade-off between robustness and energy savings. Energy savings are relative to dissipated energy of the rigid actuator 11.7 J.

Design	Optimal Stiffness	Energy Savings
nominal (b)	217.4 N·m/rad	30.8%
robust feasible (c)	243.4 N·m/rad	30.45%

ankle kinematics and kinetics as the load. We used a Monte Carlo simulation with 10,000 trials to evaluate the effect of uncertainty using actuators (a), (b), and (c). The Fig. 4.7 illustrates the torque-speed plot for all the trials. For the actuator (a), the required speed and torque do not stay within the specifications of the motor and therefore the rigid actuator is infeasible. Including series elasticity, the design (b) makes the actuator feasible and dissipates 30.8% less energy compared to (a). This justifies the use of series elasticity, not only for the reduction of energy consumption, but also to maintain the requirements within the actuator specifications. The optimal stiffness of design (b) is 217.4 N·m/rad. However, this design becomes infeasible when the reference trajectory deviates within the uncertainty set, as shown in Fig. 4.7. Using our robust formulation, design (c) satisfies the constraints despite uncertainty using a spring stiffness of 243.4 N·m/rad. Design (c) reduces 30.45% of the dissipated energy compared to a 30.8% reduction in the case (b), where the reported energy savings are relative to the rigid case. The small trade-off in performance using the robust SEA is justified when feasibility of the constraints is satisfied. Table 4.3 summarizes the results.

4.4.2 Design using a nonlinear SEA

This section describes the nonlinear spring that solves the robust optimization program in (3.16) and compares the results between a linear and nonlinear robust design. Using the values of the EC30 motor in Table 4.1 and the values of uncertainty in Table 4.2, we obtained a quadratic nonlinear spring as shown in Fig. 4.8. The nominal and robust solutions are identical because the optimal solution is at the boundary of the monotonically increasing constraints (Section 2.2.2). These are the only constraints that assume nominal conditions even in the robust formulation because they may lead to equality constraints. In general, equality constraints with uncertain coefficients lead to an ill-posed optimization program. In terms of performance, the nonlinear SEA reduces 34.6 % of energy consumption compared to a rigid actuator, 4.15 % more savings than the robust linear solution. They also satisfy the actuator constraints as long as the uncertainty remains within the uncertainty set.

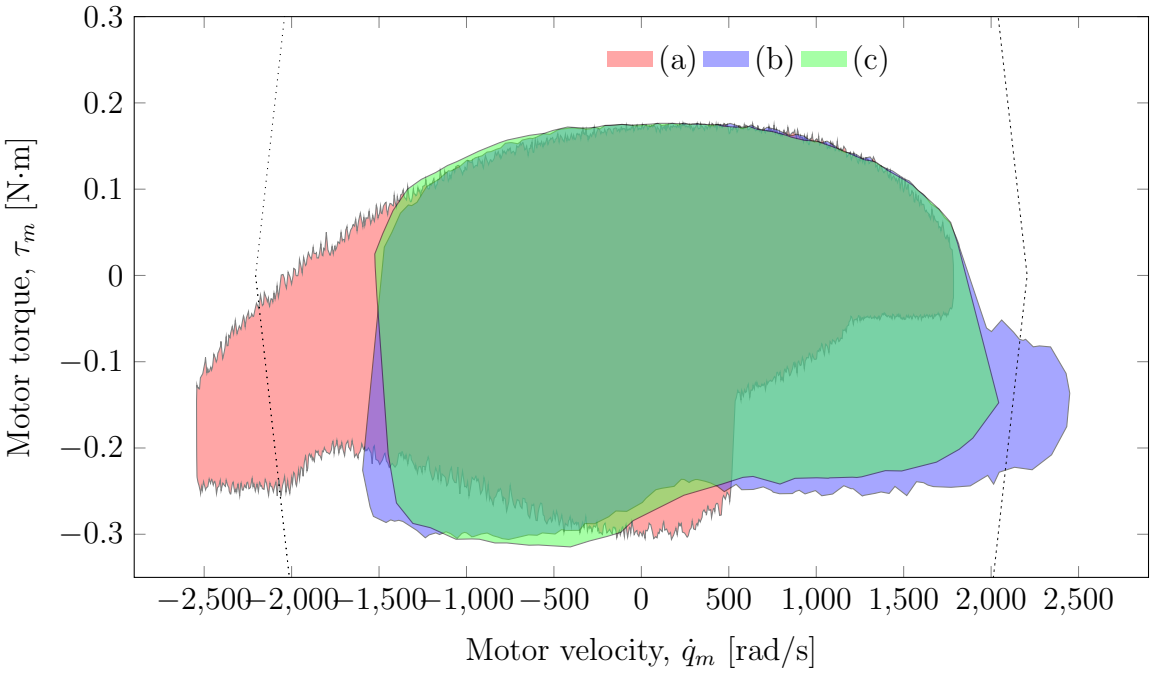

Figure 4.7. Speed and torque requirements of different actuators for a powered prosthetic ankle. The region enclosed by the dotted line describes the speeds and torques that satisfy the specifications of the motor, i.e., feasible region. The figure shows three possible actuator designs: (a) rigid actuator Maxon EC-30 without series elasticity, (b) SEA using the same motor with optimal stiffness that satisfies constraints only for the nominal data, and (c) SEA with the same motor that satisfies actuator constraints despite uncertainty using our formulation. The robust design (c) is the only actuator that satisfies the actuator constraints for all possible values of uncertainty.

Table 4.4. Uncertainty based on the variance in (Winter, 1983; Azocar et al., 2018).

Uncertainty in	Units
Mass, ε_m	$\pm 10\,\%$
Reference velocity, $\varepsilon_{\dot{q}_l}$	$\pm 5\,\%$ rms average trajectory
Reference acceleration, $\varepsilon_{\ddot{q}_l}$	$\pm 5\,\%$ rms average trajectory
Transmission efficiency, ε_η	$\pm 8\,\%$
Unmodeled dynamics, ε_{τ_u}	$\pm 5\,\%$ nominal motor torque 2.6 N·m
Manufacturing of spring, ε_d	$\pm 10\,\%$

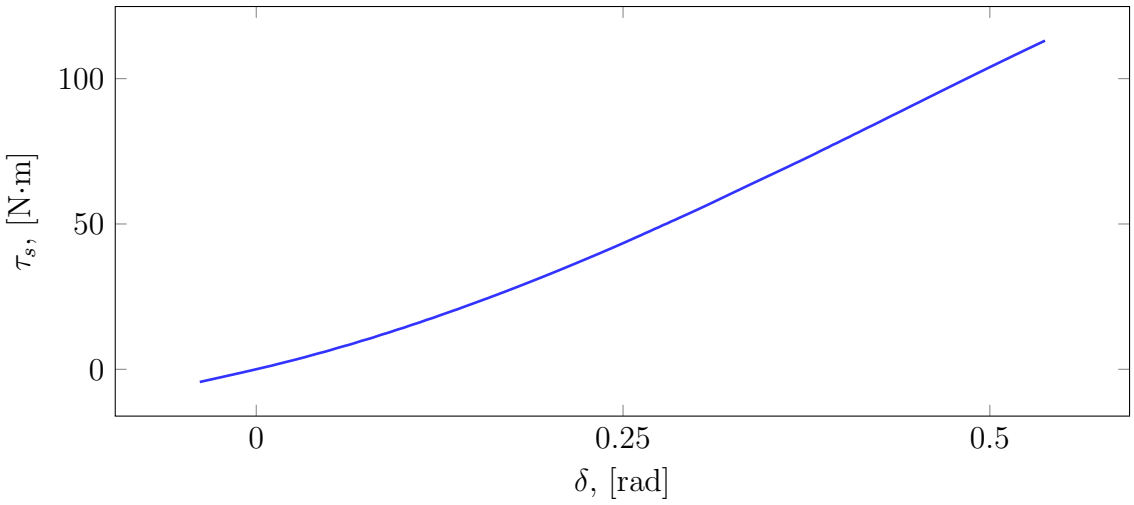

Figure 4.8. Optimal robust nonlinear spring. The nominal and robust solution are identical. The optimal solution is at the boundary of the monotonically increasing constraints (Section 2.2.2), which are not affected by uncertainty in our formulation.

As another case study, we consider the design of a robotic ankle using the ILM85x26 motor for level-ground running (Fig. 4.2) under the uncertainty defined by the parameters in Table 4.4. This case study shows the difference between the nominal and robust optimal springs. The Figs. 4.9 and 4.10 illustrate the optimal robust spring that satisfies the constraints as long as uncertainty is within the prescribed sets. The prescribed robust spring design in Fig. 4.9 is significantly stiffer than the nominal solution for elongations higher than 0.2 rad; however, during implementation, the robust spring could be softer than the nominal solution at elongations higher than 0.2 rad when considering the manufacturing uncertainty (ε_d $\pm 10\,\%$). In this case study, the uncertainty on the manufacturing of the spring modified the spring profile mainly at high elongations. As illustrated in Fig. 4.10, the nominal solution does not satisfy the speed constraints of the motor. This case study illustrates an unfeasible spring design when neglecting uncertainty and denotes the importance of a robust solution.

In our approach, the designer can isolate and analyze the effect of different kinds of uncertainty. As an example, using the EC-30 motor and the requirements for level-ground walking, we analyze the independent effect of uncertainty due to limited manufacturing of the spring and uncertain torque dynamics. In the case of uncertain manufacturing accuracy in the spring, we obtained the optimal robust springs and torque-velocity profiles for the motor as illustrated in Figs. 4.11 and 4.12 respectively. Our results indicate that the robust spring could have a constant uncertainty in the spring stiffness of up to $\pm 45\,\%$, experimental values are around $\pm 10\,\%$ (Azocar et al., 2018). Similarly, the maximum uncertain dynamics torque

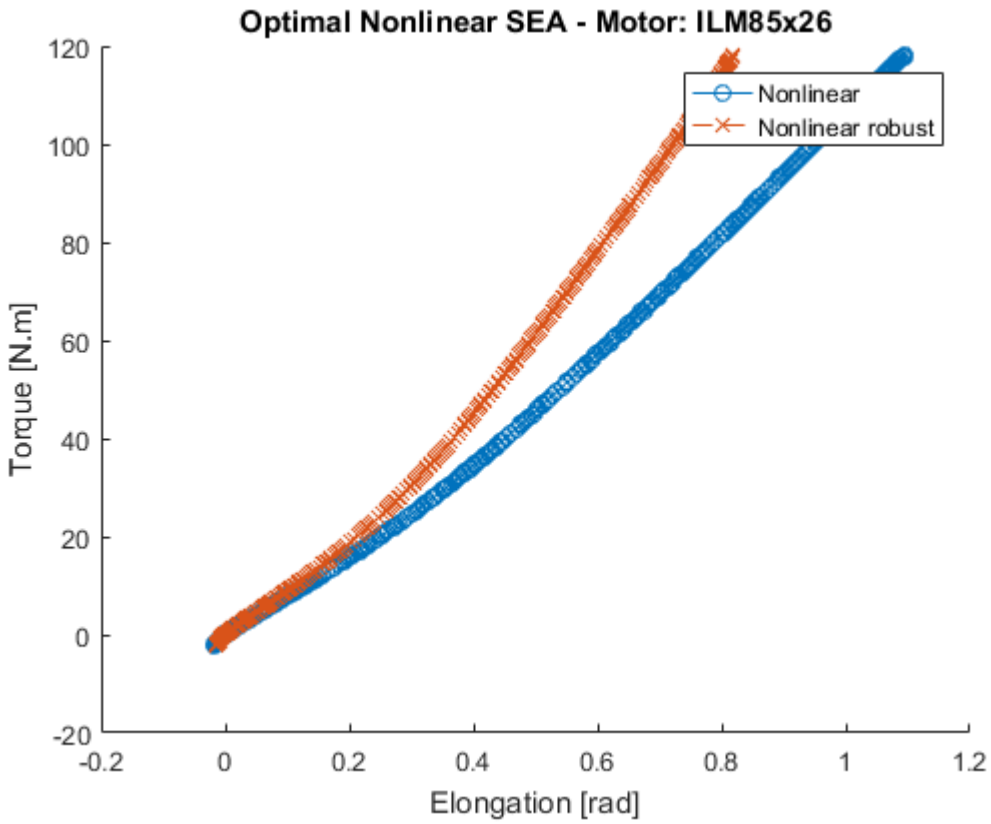

Figure 4.9. Optimal nonlinear spring for the robust and nominal solution.

is $\pm 150\,\text{mN}\cdot\text{m}$, which is about 45 % the peak torque of the motor. The optimal robust spring and velocity-torque requirements of the motor are in Figs 4.13 and 4.13. Interestingly, the nominal and robust optimal springs are similar because the active constraint is the maximum torque at the motor. The nominal elastic element already minimizes the rms value of torque when minimizing joule heating, thus the robust solution does not have more margin to reduce the motor torque.

When is it useful to consider a nonlinear SEA?

For the case studies analyzed in this section, a nonlinear SEA did not a show significant improvement in terms of energy reduction compared to a linear SEA. For the design of a powered prosthetic joint using any of the two motor-transmissions considered, a linear spring provides the adequate trade off between mechanical complexity and energy savings. However, different motor-transmission configurations or elongation constraints may justify a

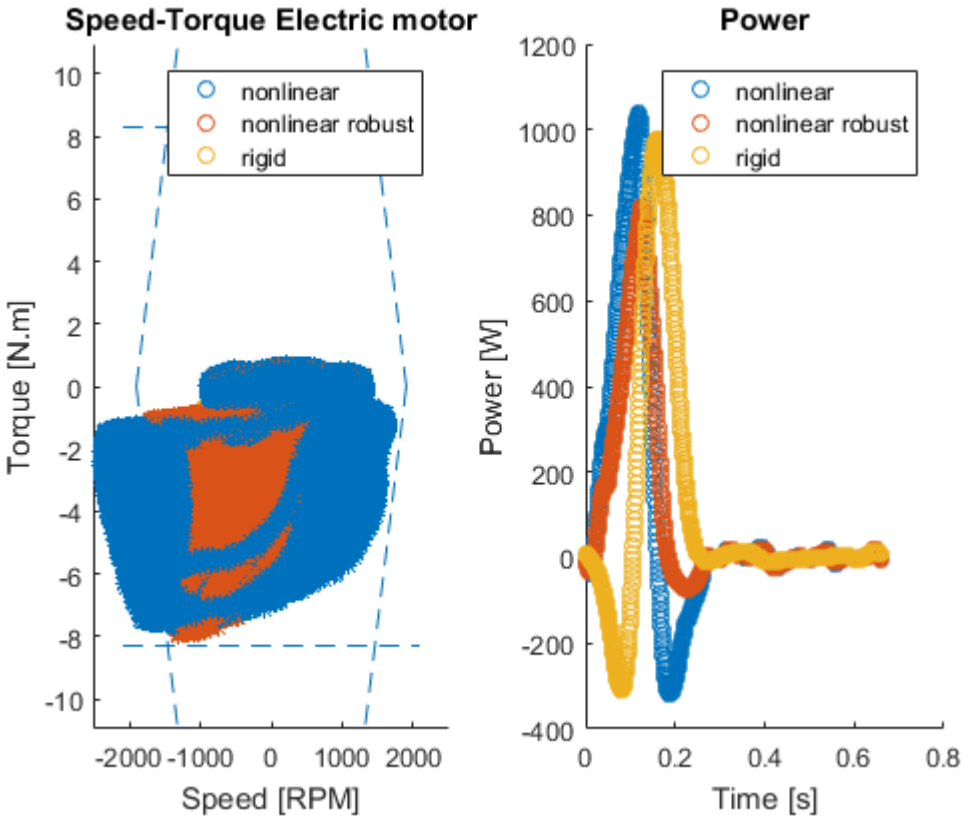

Figure 4.10. Speed-torque requirements for the optimal nonlinear spring for the robust and nominal solution.

nonlinear spring. For instance, the maximum elongation of the spring could be constrained such that the only feasible solution requires a nonlinear SEA, as shown in Section 4.4.2.

In general, the spring could be analyzed as a nonlinear dynamical system. Those systems characterize by having different frequency content between the input and the output. Considering displacement as the input and torque as the output, nonlinear SEA can show significant energy reduction when there is a difference in the frequency content of those two. This is shown in the example of Section 2.2.5 and Fig 4.15.

Robustness comparison

In this section, we design the series elastic element for the nominal case constraining its maximum elongation. This exemplifies the case where nonlinear series elasticity can achieve tasks that linear SEAs or rigid actuators cannot accomplish. In this case study, the task is defined by the kinematics and kinetics for the ankle joint of an 85 kg subject during

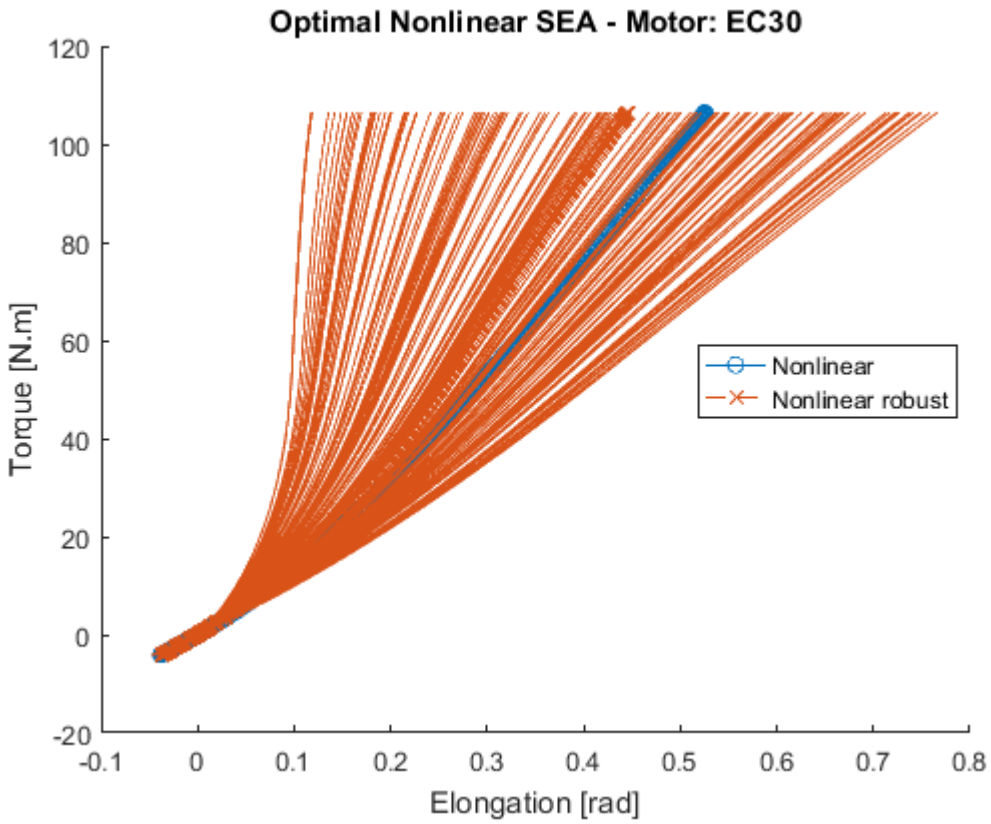

Figure 4.11. Optimal nonlinear robust springs assuming uncertainty of ±45 % in the spring stiffness. This could be interpreted as uncertainty in the modulus of the spring material or limited accuracy in the manufacturing of the spring. The blue line represents the spring profile for the nominal optimal solution.

running (Novacheck, 1998). In addition to the constraints in torque and velocity of the motor (less than 8.3 N·m and 1500 rpm respectively, ILM85x26 motor in Table 4.1.), we limit the maximum elongation of the spring to be less than 0.4 rad. This constraint may be imposed by the geometry of the mechanism or maximum elongation of the spring. For the rigid actuator, the task will require a peak torque of 9 N·m and maximum absolute speed of 1674 rpm, which is outside the motor's specifications using a 48 V power source. The solution of the linear spring approaches the characteristics of the rigid actuator as we constrain the elongation. For the given elongation constraint the linear spring SEA is not feasible. In contrast, the nonlinear spring in Fig. 4.16 elongates less than 0.4 rad, while the motor torque and velocity remain within specifications. The dissipated energy using the optimal nonlinear spring is 40.67 J per cycle and the peak power is 1161.96 W.

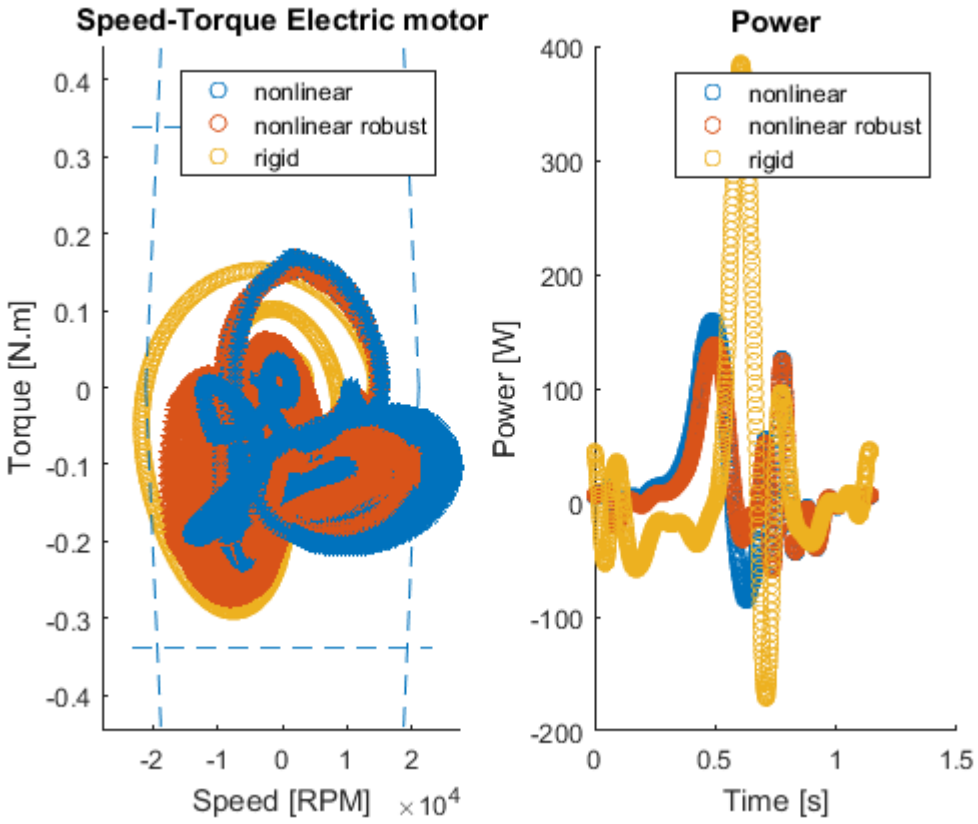

Figure 4.12. Speed-torque requirements for the optimal nonlinear spring for the robust and nominal solution with uncertainty of $\pm 45\,\%$ in the spring stiffness.

4.5 Minimize peak power and energy while performing multiple tasks

In this section, we focus on the design of the elastic element solving the optimization program in (2.39). As discussed in Section 2.3, minimizing peak power is useful only when the motor and transmission have not been selected. The ideal way to use this optimization program is to solve it for an initial motor and transmission configuration, then if the peak power is significantly lower one can iterate finding a solution with a lower power motor. The process iterates until the designer selects the lightest or lower power motor that solves the problem. In this case study, we will only perform the first step of the iteration, assuming as a starting point the characteristics of the motor and transmission defined in Table 4.1.

One of the advantages of the proposed methodology is the capability to analyze arbitrary periodic reference trajectories. Taking advantage of this flexibility, the design of the elastic element considers three different tasks for the prosthetic leg: level-ground walking or running as shown in Fig. 4.2, and a combination of walking and running. The walking and running

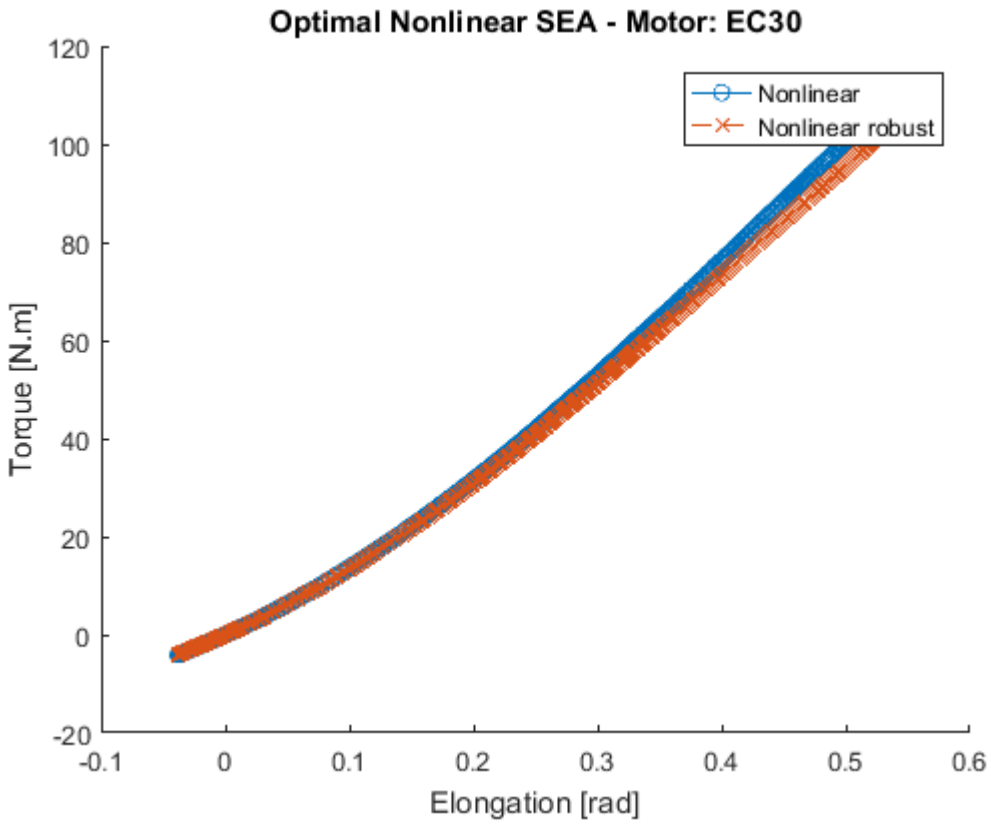

Figure 4.13. Optimal nonlinear robust spring with a uncertain dynamics torque of $\pm 150\,\text{mN}\cdot\text{m}$.

trajectory combines four steps of walking and one of running, which corresponds to the case where the user runs 20% of the time and walks during the remaining portion of operation. In this case, we only constrain the maximum absolute value of torque and velocity of the motor to be within the specifications of the datasheet (Table 4.1). The analysis assumes energy losses at the transmission, i.e., $\eta = 0.8$.

The multiobjective optimization involves analysis of the trade-off curves in Fig. 4.17. Results are reported relative to the peak power and dissipated energy of a rigid actuator performing the same task. The dissipated energy with a rigid actuator is 59.27 J and 33.75 J per gait cycle for walking and running tasks, respectively, while its peak power reaches 325.29 W and 1111.12 W for these tasks. Note that the dissipated energy values are per cycle, and because the walking period is longer than running (1.14 s walking (Winter, 1983) and 0.66 s for running (Novacheck, 1998)), it dissipates more energy, even though the running peak torques are higher (Fig. 4.2).

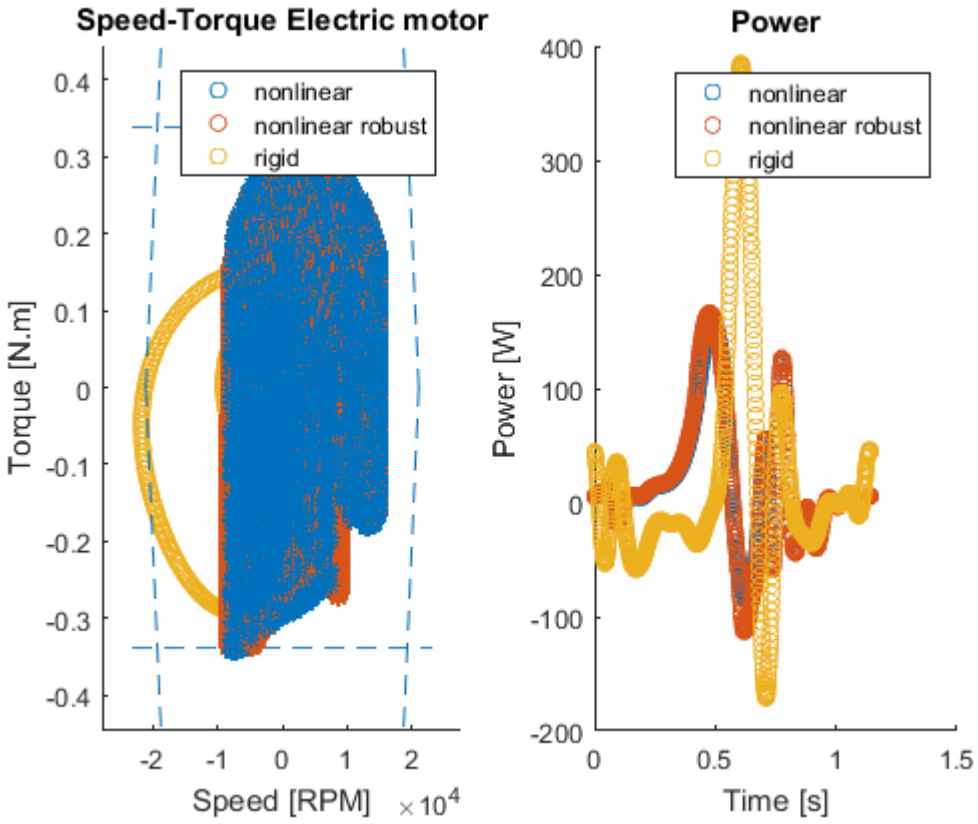

Figure 4.14. Speed-torque requirements for the optimal nonlinear spring for the robust and nominal solution with uncertain dynamics torque of ±150 mN·m.

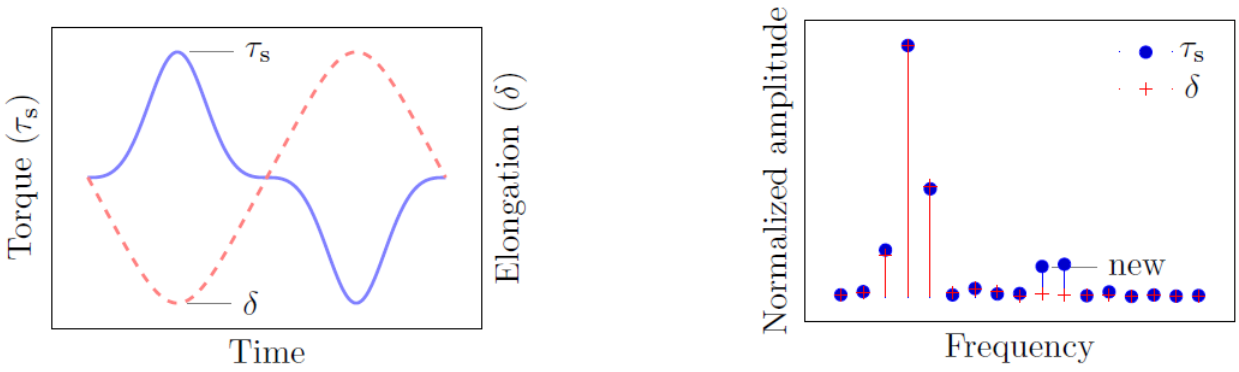

Figure 4.15. A nonlinear series spring could reduce further motor energy consumption compared to a linear spring when the load position and torque have different frequency content.

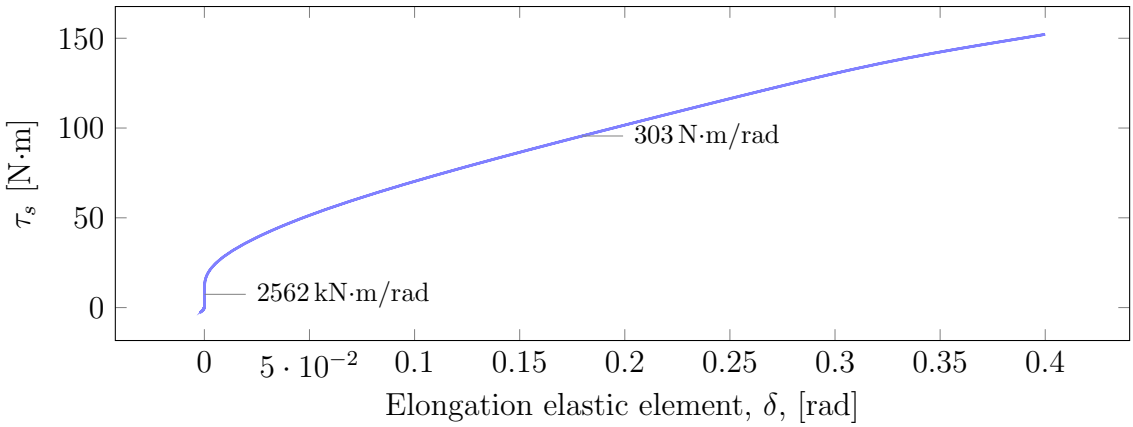

Figure 4.16. Optimal nonlinear elastic element subject to elongation and motor constraints. Local values of stiffness are reported in the graph. At small elongations the elastic element is almost rigid (2562 kN·m/rad). A parametric representation of this elastic element would involve a polynomial of high degree, which may be cumbersome for existing methods of design.

To generate the trade-off curves in Fig. 4.17, CVX solved the optimization problem (2.39) using 30 different values for $\theta \in [0, 1]$. The points between $\theta = 0$ and $\theta = 1$ were sampled from a sigmoid function to have an adequate distribution of points in the trade-off curves. In the proposed methodology, γ_1 and γ_2 in (2.39) control the relative magnitude of the two costs. γ_1 scales the maximum acceleration with respect to the convex simplification of peak power and γ_2 scales energy consumption relative to peak power. Comparing the relative magnitude of peak power and energy consumption, we defined $\gamma_1 = 0.02$ and $\gamma_2 = 300$.

4.5.1 The weak trade-off between peak power and energy consumption

For the given trajectories and motor configuration, there is a correlation between reduction of convex power and peak power, as shown in Fig. 4.17. In addition, the relationship between energy consumption and peak power represents a weak trade-off in the multiobjective optimization (Boyd and Vandenberghe, 2004, p. 182), i.e., a small increase in the dissipated energy will imply a significant reduction of peak power. For example, in the walking gait trade-off curve of Fig. 4.17, point (a) represents a reduction of 4.3% of dissipated energy and an increase of 187% of peak power when compared to a rigid actuator. A significant reduction of peak power is achieved by consuming a little more of energy, as seen in point (b) where dissipated energy and peak power reduce 1.61% and 65.91% respectively. The other two tasks use the same principle to define point (b), the solution of the multiobjective optimization problem. The selection of this point depends on the priorities of the designer.

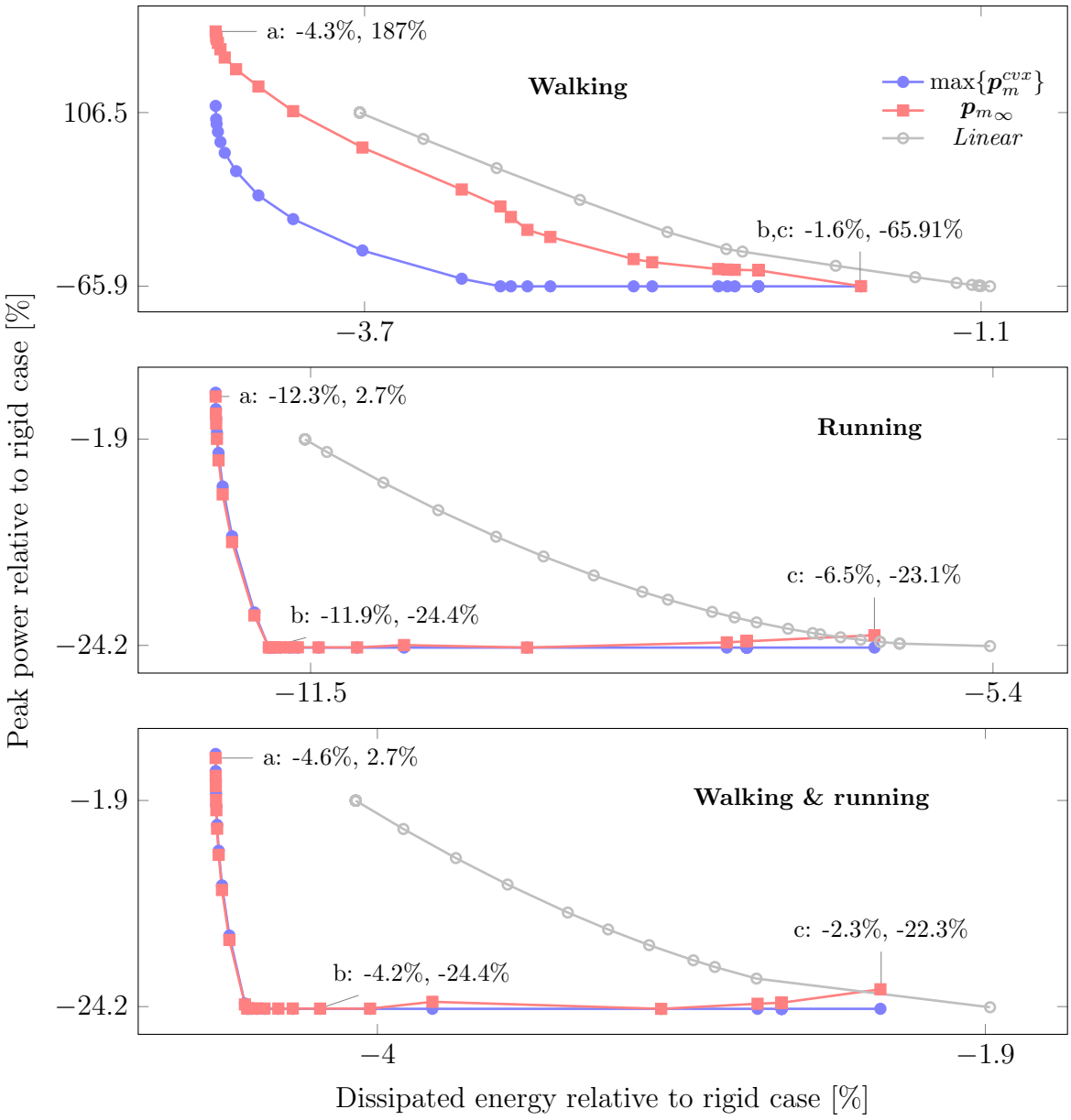

Figure 4.17. Trade-off curves for the tasks of level ground walking, running, and walking and running. Point (a) in each graph indicates the results when the cost function is only energy consumption, i.e., $\theta = 1$ in (2.39). Point (b) represents the optimal point based on the trade-off analysis. Point (c) represents the results when the cost function is only maximum convex power, i.e., $\theta = 0$ in (2.39). Relative percentage is computed as $100(x_{optim} - x_{rigid})/x_{rigid}$, where x_{optim} and x_{rigid} are the dissipated energy or peak power from the optimization algorithm and rigid case respectively. Peak power and its corresponding convex approximation are denoted by $\|\boldsymbol{p}_m\|_\infty$ and $\max\{\boldsymbol{p}_m^{cvx}\}$ respectively. As a reference, we include the energy and peak power savings using a linear series spring (legend: *Linear*).

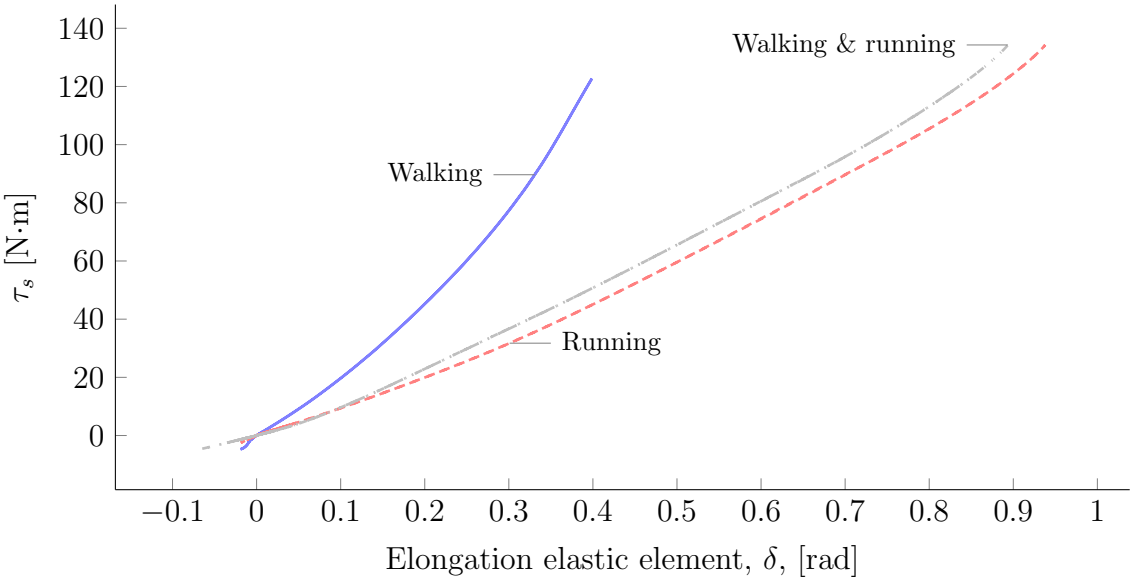

Figure 4.18. Each elastic element corresponds to the solution described by the optimal points (b) in Fig. 4.17, i.e., represents the optimal point based on our trade-off analysis for each trajectory. Each solution weights energy consumption and peak power differently as the value of θ is different in each case.

For example, if energy consumption is the main cost to minimize, then point (a) should be selected. For the case of peak power, point (c) may represent the best choice of the elastic element. Point (c) is defined with respect to the convex approximation of peak power, $\max\{\boldsymbol{p}_m^{cvx}\}$. Thus, it is possible to find optimal solutions with lower peak power and $\theta \neq 0$, e.g., point (b) for running and walking & running. The Fig. 4.18 illustrates the optimal elastic element for each of the optimal trade-off points (b). These conservative elastic elements are nonlinear and satisfy the torque and speed constraints of the motor.

The trade-off between peak power and energy consumption depends on the characteristics of the load, motor, and transmission. For instance, for the results in Section 4.5, minimizing peak power reduced both peak power and energy consumption for all the cases studied. However, minimizing energy alone increased peak power. Using the high-speed low-torque motor, with characteristics as in Table 4.1, has the opposite effect, where minimizing energy consumption also decreased peak power. In particular, the elastic element reduced the peak power of the Maxon motor from 450 W to 132 W when minimizing energy consumption. In both cases, the energy consumption obtained by minimizing energy consumption or peak power was similar. This indicates the weak trade-off in the multiobjective optimization. Thus, if the motor and transmission have not been designed, minimizing peak power is

preferred than minimizing energy consumption. Similar results are reported in (Grimmer and Seyfarth, 2011) using linear springs.

4.6 Guidelines for selecting a motor and reduction ratio

The potential reduction of energy consumption by using series elasticity depends on the effects of inertial and viscous torques at the motor end. For systems using transmissions with high reduction ratios, these effects become significant. Note that the elastic element can modify the position of the motor after the gearbox, thus it can modify its velocity and acceleration. High reduction ratios also amplify the effects of unmodeled dynamics such as static friction and backlash in the geartrain. This increases the error for torque tracking; including an elastic element converts the torque tracking into a position tracking problem, which is easier to solve with an electric motor using a high ratio transmission. In addition, the elastic element protects the gear train from impacts. In conclusion, it is recommended to use series springs for systems with a high transmission ratio.

For systems with low transmission ratio, normally driven by high torque motors, the decision is not that straightforward. The series elastic element does not modify much of the energy consumption. This is due to the fact that inertial and friction motor torques scale with the square of the reduction ratio after the transmission; thus a low reduction ratio decreases the effects of these torques with respect to the load torque. A low reduction ratio also reduces the relevance of unmodeled dynamics increasing significantly the torque control of direct drive systems without using elasticity. However, the series elastic element can modify the position of the motor such that its speed is reduced as well. Reducing speed is critical at certain portions of the trajectory as the motor is always limited in torque and speed. Thus the benefit of series springs for systems with low reduction ratios is not on the reduction of energy consumption but the modification of the motor speed profile. This modification may allow the low reduction ratio SEA to achieve a task that the same motor without elasticity could not satisfy by itself.

CHAPTER 5

CONCLUSIONS

In this dissertation, the selection of the torque-elongation spring profile that minimizes SEA's energy consumption and satisfies its constraints is formulated as a couple of convex quadratic programs. In the case of a nonlinear spring, this program is described by (2.27), and for the linear case by (2.13). These design methods could be extended to find a solution that satisfies actuator constraints despite uncertainty due to manufacturing of the spring, unmodeled dynamics, efficiency of the transmission, and the kinematics and kinetics of the load. The robust solution of these programs for the linear and nonlinear spring are defined by (3.11) and (3.16) respectively. Note that springs are passive systems. Series elasticity only reduces the energy dissipated at the motor, as shown in Section 1.1.2. Most of that energy is due to Joule heating, which is proportional to current and motor torque. Thus, to minimize energy consumption the series spring should be able to minimize torque.

The potential reduction of energy consumption by using series elasticity depends on the effects of inertial and viscous torques at the motor end. For systems using transmissions with high reduction ratios, these effects become significant. Thus, we observed significant energy savings for systems with high ratio transmissions. Taking into account the energy savings, the improvements in robustness, the potential reduction of impacts, and improvements in torque tracking, it is recommended to use series springs for systems with a high transmission ratio.

For systems with a low transmission ratio, normally driven by high torque motors, the decision is not that straightforward. The series spring does not modify much of the energy consumption. This is due to the fact that inertial and friction motor torques scale with the square of the reduction ratio after the transmission; thus, a low reduction ratio decreases the effects of these torques with respect to the load torque. A low reduction ratio also reduces the relevance of unmodeled dynamics increasing significantly the accuracy of torque control without using elasticity. However, the series elastic element can modify the position of the motor such that its speed is reduced as well. Reducing speed is critical at certain portions of the trajectory as the motor is always limited in torque and speed. Thus, the benefit of series springs for systems with low reduction ratios is not on the reduction of energy consumption, but the modification of the motor speed profile. This modification may allow the low reduction ratio SEA to achieve a task that the same motor without elasticity could not satisfy by itself.

As a case study, we analyzed the robust feasible design for a powered prosthetic ankle. Uncertainty from the recorded biomechanics naturally connected with the definition of the uncertainty sets. The results illustrate that a small trade off between robustness and energy consumption justifies a robust feasible design. It is important to note that the robust solution satisfies actuator constraints despite the uncertainty described in Table 4.2. Previous research (Brown and Ulsoy, 2013) did not consider a robust feasible solution of the optimization problem, however, they analyzed the effect of uncertainty in the energetic cost. Their results indicate that as the required motion of an SEA becomes more arbitrary, the optimal spring stiffness that minimizes power consumption approaches infinity, showing that the best design for a completely arbitrary task is a system without spring. In general, our results indicated a similar trend: the more arbitrary or the bigger the uncertainty sets, the stiffer the optimal design. However, when considering feasibility of the actuator, infinite spring stiffness may lead to an infeasible actuator. Thus, a robust feasible optimal solution cannot be obtained simply by increasing stiffness. Instead, it requires proper treatment of uncertainty as presented in our convex optimization method.

In addition to energy consumption, the proposed methodology can be extended to minimize a convex approximation of peak power. Minimization of peak power is of interest only when the motor and transmission have not been selected. Lowering requirements in peak power allows the designer to select a less powerful and lighter motor-transmission configuration. Minimizing peak power and energy consumption leads to a multi-objective optimization program. The trade-off between these two objectives depends on the characteristics of the load, motor, and transmission. For instance, for the results in Section 4.5, minimizing peak power produced a reduction in both peak power and energy consumption for all the cases studied. However, minimizing energy alone increased peak power. Using a high-speed low-torque motor had the opposite effect, where minimizing energy consumption also decreased peak power. Though results depend on the characteristics of the load, motor, and transmission, we observed a weak trade-off between peak power and energy consumption as shown in Fig. 4.17. Minimizing peak power lead to a reduction of energy consumption but not vice versa. Thus, for this multi-objective program, it is recommended to minimize peak power instead of energy consumption.

The convex approximation of peak power is close to the actual expression of power depending on the motor configuration. For all the cases considered in this article (Fig. 4.17), the approximation was accurate enough to provide a significant reduction of the actual expression of peak power. However, using a high transmission ratio increases the relevance of inertial torques in the definition of peak power. In this case the convex approximation

is less likely to yield an accurate estimation. The designer can evaluate the performance of the approximation offline to guide the design process. Using the convex approximation has significant advantages with respect to the quality of the solution and the time required to solve the optimization problem. In this case, the proposed convex QCQP program can be efficiently solved in polynomial-time (Nesterov and Nemirovskii, 1994), which is useful for an actuator that can modify its stiffness during operation, e.g., a VSA.

The convex-quadratic expressions of compliance in (2.5) and (2.27) are beneficial beyond our robust formulation. The convexity and simplicity of the expression allow optimization algorithms to find the optimal value of stiffness in polynomial time (Nesterov and Nemirovskii, 1994). This could be exploited by VSAs to calculate their reference stiffness values during operation. In the linear case without considering constraints, the proposed convex-quadratic expression has an analytical solution, which is useful to study the principles of series elasticity. For instance, (2.6) describes the necessary conditions for periodic trajectories so that series elasticity can reduce energy consumption. Future work should consider the optimal simultaneous design of series and parallel elasticity. Parallel elasticity can reduce torque of the motor regardless of the transmission ratio, and series elasticity reduces mostly the speed requirements. Thus, series and parallel elasticity play an important and complementary role on the robustness and energy consumption for a given task.

APPENDIX A

SEA MATLAB DESIGN TOOLBOX

This appendix describes the Matlab toolbox used to solve the optimization programs from Chapters 2 and 3. The toolbox runs in Matlab 2016b using Mosek (9.0.91) and CVX (Version 2.1, December 2018, Build 1127). Both CVX and Mosek requiere a free academic or commercial license.

The optimization toolbox consists mainly in the following four functions:

1. `CVX_linearSEA.m`
2. `CVX_linearSEA_robust.m`
3. `CVX_nonlinearSEA.m`
4. `CVX_nonlinearSEA_robust.m`

To provide the necessary inputs to the previous functions the user should configure the inputs to the script `linkParameters.m`. The toolbox provides the function `main_generatePeriodicTrajectories.m` to generate periodic trajectories using a Fourier decomposition of non-periodic trajectories.

A.1 Processing of input trajectory - `main_generatePeriodicTrajectories.m`

```
%% Script to generate periodic trajectories using a Fourier decomposition
clc, clearvars, close all

human.task = 'walking';          %-- Gait {running, walking, stair_ascent}
human.joint = 'ankle';           %-- Joint to analyze {knee, ankle}
human.mass = 1;                  %-- Mass of the user [Kg]

% fast_cadence, slow_cadence, normal_cadence
human.wSpeed = 'normal_cadence';
localFun_GenerateAndSavePeriodicTrajectories(human)


%% -- Local function library
function localFun_GenerateAndSavePeriodicTrajectories(human)
```

```
15 mass = human.mass;                    %-- Mass of the user [Kg]
16 joint = human.joint;                 %-- Joint to analyze {knee, ankle}
17 nPoints = 1000;
18
19
20 if strcmp(human.task, 'running')
21     load dataset_Novacheck
22     %--[2] T. F. Novacheck, The biomechanics of running,
23     %--Gait Posture, vol. 7, no. 1, Jan. 1998.
24     ciclePeriod = 1.3213 /2;
25
26     %-- Generate nPoints points to generate a grid
27     %-- Converting to [rad] for consistency with Winter data
28     ql = novacheck_running.(joint).position *pi /180;
29     %-- Multiplying by user weight and (-1) to get Dorsiflexion positive
30     %   torque as a positive number [Nm].
31     torque = -novacheck_running.(joint).torque*mass;
32     time = linspace(0, ciclePeriod, nPoints).';
33     textFile = sprintf('%s_%s_%dkg', ...
34     human.task, human.joint, human.mass);
35 elseif strcmp(human.task, 'walking')
36     load dataset_Winter
37     wSpeed = human.wSpeed;
38     %------------------------------WINTER'S DATA SCALING
39     if strcmp(wSpeed,'normal_cadence')
40         %--Cadence, Steps per minute. Winter page 12
41         spm = 105;
42         %--Sample time
43         sT = 60/spm*2/1001;
44     elseif strcmp(wSpeed,'fast_cadence')
45         %Cadence, Steps per minute. Winter page 12
46         spm = 123;
47         %--Sample time
48         sT = 60/spm*2/1001;
49     elseif strcmp(wSpeed,'slow_cadence')
50         %Cadence, Steps per minute. Winter page 12
51         spm = 87;
52         %--Sample time
53         sT = 60/spm*2/1001;
54     else
55         error('Please select a walking speed, e.g., normal, fast');
```

```
56      end
57      %-- Angle of the ankle joint,Positive Angle (deg) => Dorsiflexion.
58      %-- Negative => Plantarflexion
59      ql = level_walking.(wSpeed).(joint).position*pi/180;
60      %-- Torque of the ankle joint, Negative Torque => Dorsiflexion.
61      %-- Positive => Plantarflexion
62      torque = -level_walking.(wSpeed).(joint).torque*mass;
63      time = [0:sT:sT*(1000)]';
64      %-Name of the file
65      textFile = sprintf('%s_%s_%s_%dkg', ...
66      human.task, human.wSpeed, human.joint, human.mass);
67  elseif strcmp(human.task, 'stair_ascent')
68      load dataset_Riener
69      %-- Generate nPoints points to generate a grid
70      %-- Converting to [rad] for consistency with Winter's data
71      ql = riener.stairAscent.(joint).position *pi /180;
72      %-- Multiplying by user weight and (-1) to get Dorsiflexion positive
73      %   torque as a positive number [Nm].
74      torque = -riener.stairAscent.(joint).torque*mass;
75      time = riener.stairAscent.(joint).time;
76      textFile = sprintf('%s_%s_%dkg', ...
77      human.task, human.joint, human.mass);
78  else
79      error('Select adequate taks')
80  end
81
82  periodic_ql = fFourierDecomposition(time, ql, 200);
83  periodic_torque = fFourierDecomposition(time, torque, 10);
84
85  %-- Find the lowest frequency different from 0
86  freqs = periodic_ql.freq;
87  delFreq = min( freqs (freqs > 0 ) );
88
89  time = linspace(0, 1/delFreq, nPoints);
90  ql = periodic_ql.signalAnalytic;
91  torque = periodic_torque.signalAnalytic;
92
93  syms x
94
95  qld = diff(ql,x);
96  qldd = diff(qld,x);
```

```
97  torqueD = diff(torque,x);
98  torqueDD = diff(torqueD,x);
99
100 qlDisc = zeros(1,nPoints);
101 qldDisc = zeros(1,nPoints);
102 qlddDisc = zeros(1,nPoints);
103 torqueDisc = zeros(1,nPoints);
104 torquedDisc = zeros(1,nPoints);
105 torqueddDisc = zeros(1,nPoints);
106
107 tic
108 for i = 1:length(time)
109     qlDisc(i) = round(double(subs(ql,time(i))),10);
110     qldDisc(i) = round(double(subs(qld,time(i))),10);
111     qlddDisc(i) = round(double(subs(qldd,time(i))),10);
112     torqueDisc(i) = round(double(subs(torque,time(i))),10);
113     torquedDisc(i) = round(double(subs(torqueD,time(i))),10);
114     torqueddDisc(i) = round(double(subs(torqueDD,time(i))),10);
115 end
116 toc
117
118 save('Vikram', 'qlDisc', 'qldDisc', 'qlddDisc', 'torqueDisc',...
119     'torquedDisc', 'torqueddDisc', 'time');
120 end
```

A.2 Loading input parameters - `linkParameters.m`

```
1  function [robot,trajectory] = linkParameters(motor, varargin)
2  %LINKPARAMETERS Summary of this function goes here
3  %   Detailed explanation goes here
4
5  %Efficiency of the transmission
6  robot.eta = 0.8;
7  %Max elongation [rad]
8  robot.maxElong = 1.5;
9
10 if strcmp(motor,'EC45')
11     robot.kt = 36.9/1000;              %Torque constant motor [Nm/A]
```

```
12      robot.R = 0.608;                  %Terminal resistance phase to ...
            phase [ohms]
13      robot.iM = 181/1000/100^2;  %Inertia of the motor [Kg*m^2]
14      robot.r = 283;              %Reduction ratio of the transmission
15      robot.bm = robot.iM*2;      %Viscous damping of the motor [Nm*s]
16      robot.km = robot.kt/sqrt(robot.R);      %Motor constant
17      %-
18      robot.voltage = 36;
19      %--Constraints
20      robot.maxVelo = robot.voltage/robot.kt;     %Max motor velocity ...
            1500 RPM [rad/sec]
21      robot.RatedTorque = 0.128;                  %Peak torque [Nm]
22      robot.peakTorque = robot.RatedTorque*5;   %Peak torque [Nm]
23  elseif strcmp(motor,'EC30')
24      % 30V in leg 1, motor is rated for 24V
25      robot.voltage = 30;                     %Voltage power supply [Volts]
26      robot.kt = 13.6/1000;                   %Torque constant motor [Nm/A]
27      robot.R = 0.102;                        %Terminal resistance phase ...
            to phase [ohms]
28      robot.iM = 33.3/1000/100^2;             %Inertia of the motor [Kg*m^2]
29      %(%knee 360, ankle 720, 425)
30      robot.r = 600;                          %Reduction ratio of the ...
            transmission
31      robot.bm = robot.iM/2;                  %Viscous damping of the ...
            motor [Nm*s]
32      robot.km = robot.kt/sqrt(robot.R);      %Motor constant
33      %--Constraints
34      robot.maxVelo = robot.voltage/robot.kt; %Max motor velocity 1500 ...
            RPM [rad/sec]
35      robot.RatedTorque = 0.135;              %Peak torque [Nm]
36      robot.peakTorque = robot.RatedTorque*2.5;   %Peak torque [Nm]
37      robot.L = 16.3e-6;                          %Inductance [uH]
38  elseif strcmp(motor,'ILM85x26')
39      robot.kt = 0.24;                            %Torque constant ...
            motor [Nm/A]
40      robot.R = 323/1000;                         %Terminal resistance ...
            [ohms]
41      rotorInerRD = 1.15*(1e-2)^2;            %Inertia of the rotor ...
            (RoboDrive) [Kg*m^2]
42      rotorInerToby = 13098.7/1000*(1e-3)^2;  %Inertia of the rotor ...
            (Toby) [Kg*m^2]
```

```
43      robot.iM = rotorInerRD+rotorInerToby;   %Inertia of the motor [Kg*m^2]
44      robot.r = 22;
45      robot.bm = robot.iM/2;                  %Drag torque
46      robot.km = robot.kt/sqrt(robot.R);      %Motor constant
47      %--Constraints
48      robot.maxVelo = 1500*2*pi/60;       %Max motor velocity 1500 RPM ...
            [rad/sec]
49      robot.peakTorque = 8.3;             %Peak torque [Nm]
50      %     robot.peakTorque = 6.2;             %Peak torque [Nm]
51      robot.RatedTorque = 2.6;            %Peak torque [Nm]
52      robot.L = 920e-6;                   %Inductance [uH]
53      robot.voltage = 48;                 %Voltage power supply [Volts]
54  elseif strcmp(motor,'ILM70x18')
55      robot.kt = 0.18;                            %Torque constant ...
            motor [Nm/A]
56      robot.R = 655/1000;                         %Terminal resistance ...
            [ohms]
57      rotorInerRD = 0.34*(1e-2)^2;            %Inertia of the rotor ...
            (RoboDrive) [Kg*m^2]
58      rotorInerToby = 13098.7/1000*(1e-3)^2;  %Inertia of the rotor ...
            (Toby) [Kg*m^2]
59      robot.iM = rotorInerRD+rotorInerToby;   %Inertia of the motor [Kg*m^2]
60      robot.r = 38;
61      robot.bm = robot.iM/2;                  %Drag torque
62      robot.km = robot.kt/sqrt(robot.R);                  %Motor constant
63      %--Constraints
64      robot.maxVelo = 2100*2*pi/60;           %Max motor velocity 2100 ...
            RPM [rad/sec]
65      robot.peakTorque = 4;                   %Peak torque [Nm]
66      robot.RatedTorque = 1.25;               %Peak torque [Nm]
67      robot.L = 1350e-6;                       %Inductance [uH]
68      robot.voltage = 48;                     %Voltage power supply [Volts]
69  elseif strcmp(motor,'ILM85x04')
70      robot.kt    = 0.04;                        %Torque constant motor ...
            [Nm/A]
71      robot.R     = 138/1000;                    %Terminal resistance [ohms]
72      rotorInerRD     = 0.28*(1e-2)^2;           %Inertia of the rotor ...
            (RoboDrive) [Kg*m^2]
73      rotorInerToby   = 13098.7/1000*(1e-3)^2;   %Inertia of the rotor ...
            (Toby) [Kg*m^2]
```

```
74      robot.iM    = rotorInerRD+rotorInerToby;   %Inertia of the motor ...
            [Kg*m^2]
75      robot.r     = 160;
76      robot.bm    = robot.iM/2;                  %Drag torque
77      robot.km    = robot.kt/sqrt(robot.R);      %Motor constant
78      %--Constraints
79      robot.maxVelo       = 9000*2*pi/60;        %Max motor velocity 2100 ...
            RPM [rad/sec]
80      robot.peakTorque    = 1.2;                 %Peak torque [Nm]
81      robot.RatedTorque   = 0.43;                %Peak torque [Nm]
82      robot.L             = 120e-6;              %Inductance [uH]
83      robot.voltage       = 48;                  %Voltage power supply ...
            [Volts]
84  end
85
86  robot.noLoadSpeed = robot.voltage/robot.kt;
87  robot.stallTorque = robot.voltage*robot.kt/robot.R;
88
89  if nargin ≥ 1
90      task = varargin{1}.task;
91      trajectory.task         = task;
92      trajectory.constraints  = varargin{1}.constraints;
93      joint = varargin{1}.joint;
94      mass = varargin{1}.userMass;
95      trajectory.joint    = joint;
96      trajectory.mass     = mass;
97      if ( strcmp(task, 'walking') || strcmp(task, 'running')...
98              || strcmp(task, 'stair_ascent') )
99          % Uncertainty values for a robust solution
100         trajectory.mBar = mass;                        %Nominal mass of ...
                user [kg]
101         trajectory.mUnc = 8.8;                          %Uncertainty in ...
                mass [+- kg]
102         %-- Select walking speed from user or set to default
103         if (isfield(varargin{1}, 'wSpeed') && strcmp(task, 'walking'))
104             wSpeed = varargin{1}.wSpeed;
105             trajectory.wSpeed   = wSpeed;
106             eval( sprintf('load %s_%s_%s_1kg', task, wSpeed, joint) );
107         else
108             eval( sprintf('load %s_%s_1kg', task, joint) );
109         end
```

```
110         trajectory.time = time.';
111         trajectory.ql = qlDisc.';
112         trajectory.qld = qldDisc.';
113         trajectory.qldd = qlddDisc.';
114         trajectory.torque = torqueDisc.'*mass;
115         trajectory.torqued = torquedDisc.'*mass;
116         trajectory.torquedd = torqueddDisc.'*mass;
117     elseif ( strcmp(task, 'CubicSpring') )
118         eval( 'load CubicSpring5k')
119         %         eval( 'load CubicSpring' )
120         trajectory.time = time.';
121         trajectory.ql = qlDisc.';
122         trajectory.qld = qldDisc.';
123         trajectory.qldd = qlddDisc.';
124         trajectory.torque = torqueDisc.';
125         trajectory.torqued = torquedDisc.';
126         trajectory.torquedd = torqueddDisc.';
127     elseif ( strcmp(task, 'CubicLinearSpring') )
128         eval( 'load CubicLinearSpring')
129         trajectory.time = time.';
130         trajectory.ql = qlDisc.';
131         trajectory.qld = qldDisc.';
132         trajectory.qldd = qlddDisc.';
133         trajectory.torque = torqueDisc.';
134         trajectory.torqued = torquedDisc.';
135         trajectory.torquedd = torqueddDisc.';
136     elseif ( strcmp(task, 'VikramTest') )
137         eval( 'load Vikram')
138         trajectory.time = time.';
139         trajectory.ql = qlDisc.';
140         trajectory.qld = qldDisc.';
141         trajectory.qldd = qlddDisc.';
142         trajectory.torque = torqueDisc.';
143         trajectory.torqued = torquedDisc.';
144         trajectory.torquedd = torqueddDisc.';
145     else
146         error('Provide a proper reference task')
147     end
148
149 %     per     = 0.05;
150 %     %Uncertainty in efficiency (eta +- etaUnc)
```

```
151 %     robot.etaUnc = per;
152 %     %Multiplicative uncertainty in spring torque
153 %     trajectory.tauSUnc_m = 0.1;
154 %     %Additive uncertainty in spring torque
155 %     trajectory.tauSUnc_a = rms(trajectory.torque)*per;
156 %     %Uncertain dynamics - Uncertainty torque (+- tauUnc) [N.m]
157 %     robot.tauUnc = robot.RatedTorque*per;
158 %     %Uncertainty in position [rad]
159 %     trajectory.qlUnc = 0;
160 %     %Additive uncertainty in velocity [rad/s]
161 %     trajectory.qldUnc_a = rms(trajectory.qld)*per;    ...
      %rms(trajectory.qld)*0.3;
162 %     %Multiplicative uncertainty in velocity [rad/s]
163 %     trajectory.qldUnc_m = per;
164 %     %Uncertainty in acceleration [rad/s^2]
165 %     trajectory.qlddUnc_a = rms(trajectory.qldd)*per;   ...
      %rms(trajectory.qldd)*0.3;
166 %     %Multiplicative uncertainty in acceleration [rad/s^2]
167 %     trajectory.qlddUnc_m = per;
168 %     %Percentage of uncertainty compliance (1+-comp)
169 %     robot.comUnc = 0.018*0.1;   %10% of mean nominal solution
170 end
171 end
```

A.3 Implementation of QP programs - The four main functions of the toolbox

A.3.1 CVX_linearSEA.m

```
1  function [optiRobot] = CVX_linearSEA(robot, trajectory)
2  % CVX_linearSEA  Returns optimal linear SEA using nominal values of
3  % the trajectory and robot parameters.
4  %
5  %   OptimalSEA = CVX_linearSEA(robot, trajectory)
6  %
7  %   -- INPUTS:
8  %           robot:      [struct] Parameters of the electric motor and robot
9  %                                system.
10 %           trajectory: [struct] Desired load trajectory.
11 %
```

```
12 %   -- OUTPUTS:
13 %           optiRobot   [struct] Optimal SEA parameters and energy
14 %                                measurements
15 %
16 %   See also CVX_linearSEA_robust, CVX_nonlinearSEA, ...
       CVX_nonlinearSEA_robust.
17
18 %--Loading robot parameters
19 iM = robot.iM;
20 r = robot.r;
21 bm = robot.bm;
22 eta = robot.eta;
23 vIn = robot.voltage;
24 R = robot.R;
25 kt = robot.kt;
26 % km = robot.km;
27 % time = trajectory.time;
28
29 %-- Loading load trajectory
30 ql = trajectory.ql;
31 qld = trajectory.qld;
32 qldd = trajectory.qldd;
33 % time = trajectory.time;
34 torEla = -trajectory.torque;
35 torElad = -trajectory.torqued;
36 torEladd = -trajectory.torquedd;
37 steps = length(ql);
38
39 %-- Loading coefficients for the cost function
40 [Ene, gamma1, gamma2] = localFunCoeffOptim(robot, trajectory);
41 %-- Loading coefficients for the constraints
42 [aIneq, bIneq] = localFunConstraints(robot, trajectory, gamma1, gamma2);
43
44 %% CVX optimization formulation
45 cvx_begin quiet
46 % Mosek, Gurobi, SDPT3, sedumi
47 cvx_solver Mosek
48 % best, default
49 cvx_precision best
50
51 variable alpha1
```

```
52
53 Energy = Ene.a*alpha1^2 + Ene.b*alpha1 + Ene.c;
54
55 minimize (Energy)
56
57 %-- Lumped constraints
58 aIneq*alpha1 ≤ bIneq;
59
60 cvx_end
61
62 %-- Checking if the solution of cvx can be used
63 if ¬any(strcmp(cvx_status, {'Solved', 'Inaccurate/Solved',...
64         'Inaccurate/Unbounded', 'Inaccurate/Infeasible'}))
65     error(['CVX status:', cvx_status]);
66 elseif any(strcmp(cvx_status, {'Inaccurate/Solved',...
67         'Inaccurate/Unbounded', 'Inaccurate/Infeasible'}))
68     warning(['CVX status: ', cvx_status, '// Elastic element: Robust ...
        Linear']);
69 end
70 %% Post-processing
71 %--Motor performance
72 qm = (ql - torEla*alpha1)*r;
73 qmd = (qld - torElad*alpha1)*r;
74 qmdd = (qldd - torEladd*alpha1)*r;
75 tauM = iM*qmdd + bm*qmd - torEla./(r*eta);
76 elong = torEla*alpha1;
77
78 %-- CREATE POST PROCESSING FUNCTION
79 resultsCVX.qm = qm;
80 resultsCVX.qmd = qmd;
81 resultsCVX.qmdd = qmdd;
82 resultsCVX.elong = elong;
83 resultsCVX.tauM = tauM;
84 resultsCVX.solverStatus = cvx_status;
85 resultsCVX.torqueElastic = torEla;
86
87 %-- Getting results
88 optiRobot = fOptimizationPostprocessing(robot, trajectory, resultsCVX);
89 optiRobot.ConElongInf = norm(elong,inf) - robot.maxElong;
90 optiRobot.ConQmdInf = norm(qmd,inf) - robot.maxVelo;
91 optiRobot.ConTauM = norm(tauM,inf) - robot.peakTorque;
```

```
92  optiRobot.ConQ1 = tauM - (vIn*kt/R - kt^2/R*qmd);
93  optiRobot.ConQ1Max = max(optiRobot.ConQ1);
94  optiRobot.ConQ2 = -tauM - (vIn*kt/R - kt^2/R*qmd);
95  optiRobot.ConQ2Max = max(optiRobot.ConQ2);
96  optiRobot.ConQ3 = tauM - (vIn*kt/R + kt^2/R*qmd);
97  optiRobot.ConQ3Max = max(optiRobot.ConQ3);
98  optiRobot.ConQ4 = -tauM - (vIn*kt/R + kt^2/R*qmd);
99  optiRobot.ConQ4Max = max(optiRobot.ConQ4);
100 optiRobot.k = 1/alpha1;
101
102 %-- Check which constraint is at the boundary
103 %-- Checking that the max. value of the LHS of inequalities is less ...
       than 0
104 const = max([optiRobot.ConElongInf; optiRobot.ConQmdInf; ...
       optiRobot.ConTauM;...
105      optiRobot.ConQ1; optiRobot.ConQ2; optiRobot.ConQ3; optiRobot.ConQ4]);
106 if const ≤ 0
107      optiRobot.feasible = true;
108 else
109      optiRobot.feasible = false;
110 end
111
112 end
113
114 function [Ene, gamma1, gamma2] = localFunCoeffOptim(robot, trajectory)
115
116 %--Loading robot parameters
117 iM = robot.iM;
118 r = robot.r;
119 bm = robot.bm;
120 eta = robot.eta;
121 km = robot.km;
122 kt = robot.kt;
123 vIn = robot.voltage;
124 R = robot.R;
125
126 %-- Loading trajectory
127 % ql = trajectory.ql;
128 qld = trajectory.qld;
129 qldd = trajectory.qldd;
130 time = trajectory.time;
```

```
131 torqueEla = -trajectory.torque;
132 torqueElad = -trajectory.torqued;
133 torqueEladd = -trajectory.torquedd;
134 % This function has access to the workspace of the parent function
135 % Therefore, the scope of x is the function to which this workspace
136 % belongs, and all functions nested to any level within that function.
137
138 %-- Defining parameters
139 gamma1 = -(iM*torqueEladd*r + bm*torqueElad*r);
140 gamma2 = iM*qldd*r + bm*qld*r -  torqueEla/(eta*r);
141
142 Ene.a = trapz(time, (gamma1.^2/km^2 + bm*r^2*torqueElad.^2));
143 Ene.b = trapz(time, (2*gamma1.*gamma2/km^2 - 2*bm*r^2*qld.*torqueElad));
144 Ene.c = trapz(time, (gamma2.^2/km^2 + bm*qld.^2*r^2 - qld.*torqueEla/eta));
145 Ene.gamma1 = gamma1;
146 Ene.gamma2 = gamma2;
147
148 % Power.a = bm*r^2.*torqueElad.^2;
149 % Power.b = torqueEla.*torqueElad/eta - 2*bm*qld.*torqueElad*r^2;
150 % Power.c = bm*r^2*qld.^2 - torqueEla.*qld / eta;
151 end
152
153 function [aIneq, bIneq] = localFunConstraints(robot, trajectory, ...
       gamma1, gamma2)
154
155 constraints     = trajectory.constraints;
156
157 if strcmp(constraints, 'nominal')
158     %--Loading robot parameters
159     iM = robot.iM;
160     r = robot.r;
161     bm = robot.bm;
162     eta = robot.eta;
163     vIn = robot.voltage;
164     R = robot.R;
165     kt = robot.kt;
166     % km = robot.km;
167     % time = trajectory.time;
168
169     qmdMax = robot.maxVelo;
170     elongMax = robot.maxElong;
```

```
171     torqueMax = robot.peakTorque;
172
173     %-- Loading trajectory
174     ql = trajectory.ql;
175     qld = trajectory.qld;
176     qldd = trajectory.qldd;
177     % time = trajectory.time;
178     torEla = -trajectory.torque;
179     torElad = -trajectory.torqued;
180     torEladd = -trajectory.torquedd;
181
182     %-- Infinity norm of elongation less than max. elong
183     a1 = [torEla; -torEla];
184     b1 = [ones(length(torEla)*2, 1)*elongMax];
185
186     %-- Infinity norm of torque to be less than max. torque
187     a2 = [gamma1; -gamma1];
188     b2 = [torqueMax - gamma2; torqueMax + gamma2];
189
190     %-- Velocity Torque - Relationship
191     a41 = gamma1 - kt^2/R*r*torElad;
192     b41 = vIn*kt/R*ones(length(gamma1), 1)- kt^2/R*r*qld - gamma2;
193
194     a42 = gamma1 + kt^2/R*r*torElad;
195     b42 = vIn*kt/R*ones(length(gamma1), 1)+ kt^2/R*r*qld - gamma2;
196
197     a43 = -gamma1 - kt^2/R*r*torElad;
198     b43 = vIn*kt/R*ones(length(gamma1), 1)- kt^2/R*r*qld + gamma2;
199
200     a44 = -gamma1 + kt^2/R*r*torElad;
201     b44 = vIn*kt/R*ones(length(gamma1), 1)+ kt^2/R*r*qld + gamma2;
202
203     a4 = [a41; a42; a43; a44];
204     b4 = [b41; b42; b43; b44];
205     %-- Collect all the constraints
206     aIneq = [a1; a2; a4];
207     bIneq = [b1; b2; b4];
208 elseif strcmp(constraints, 'none')
209     aIneq = [];
210     bIneq = [];
211 end
```

```
212
213
214 % %-- Collect all the constraints
215 % aIneq = [a1; a2];
216 % bIneq = [b1; b2];
217
218 % aIneq = [];
219 % bIneq = [];
220 end
```

A.3.2 CVX_linearSEA_robust.m

```
1  function [optiRobot] = CVX_linearSEA_robust(robot, trajectory)
2  % CVX_linearSEA_robust  Returns optimal linear SEA using nominal and
3  % uncertain values of the trajectory and robot parameters.
4  %
5  %   OptimalSEA = CVX_linearSEA_robust(robot, trajectory)
6  %
7  %   -- INPUTS:
8  %           robot:      [struct] Parameters of the electric motor and robot
9  %                                system.
10 %           trajectory: [struct] Desired load trajectory.
11 %
12 %   -- OUTPUTS:
13 %           optiRobot   [struct] Optimal SEA parameters and energy
14 %                                measurements
15 %
16 %   See also CVX_linearSEA, CVX_nonlinearSEA, CVX_linearSEA.
17
18 %--Loading robot parameters
19 iM = robot.iM;
20 r = robot.r;
21 bm = robot.bm;
22 eta = robot.eta;
23 vIn = robot.voltage;
24 R = robot.R;
25 kt = robot.kt;
26 mBar = trajectory.mBar;     %Nominal mass [Kg]
27
```

```
28
29 %-- Nominal trajectory
30 ql = trajectory.ql;
31 qld = trajectory.qld;
32 qldd = trajectory.qldd;
33 torEla = -trajectory.torque;
34 torElad = -trajectory.torqued;
35 torEladd = -trajectory.torquedd;
36
37 %-- Loading coefficients for the cost function
38 Ene = localFunCoeffOptim(robot, trajectory);
39 %-- Loading robust coefficients for the affine contraints
40 [d, e] = robust_constraints(trajectory, robot);
41
42 %% CVX optimization formulation
43 cvx_begin quiet
44 % Mosek, Gurobi, SDPT3, sedumi
45 cvx_solver Mosek
46 % best, default
47 cvx_precision best
48
49 variables alpha1
50
51 minimize (Ene.a*alpha1^2 + Ene.b*alpha1 + Ene.c)
52
53 subject to
54
55     d.*alpha1 <= e;
56
57 cvx_end
58
59 %-- Checking if the solution of cvx can be used
60 if ~any(strcmp(cvx_status, {'Solved', 'Inaccurate/Solved',...
61         'Inaccurate/Unbounded', 'Inaccurate/Infeasible'}))
62     error(['CVX status:', cvx_status]);
63 elseif any(strcmp(cvx_status, {'Inaccurate/Solved',...
64         'Inaccurate/Unbounded', 'Inaccurate/Infeasible'}))
65     warning(['CVX status: ', cvx_status, '// Elastic element: Robust ...
        Linear']);
66 end
67 %% Post-processing
```

```
68 %-- Running expressions again to avoid mistakes
69 qm = (ql - torEla*alpha1)*r;
70 qmd = (qld - torElad*alpha1)*r;
71 qmdd = (qldd - torEladd*alpha1)*r;
72 tauM = iM*qmdd + bm*qmd - torEla./(r*eta);
73 elong = torEla*alpha1;
74
75 %-- CREATE POST PROCESSING FUNCTION - Walking
76 resultsCVX.qm = qm;
77 resultsCVX.qmd = qmd;
78 resultsCVX.qmdd = qmdd;
79 resultsCVX.elong = elong;
80 resultsCVX.tauM = tauM;
81 resultsCVX.solverStatus = cvx_status;
82 resultsCVX.torqueElastic = torEla;
83
84 %-- Getting results
85 optiRobot = fOptimizationPostprocessing(robot, trajectory, resultsCVX);
86 optiRobot.ConElongInf = norm(elong,inf) - robot.maxElong;
87 optiRobot.ConQmdInf = norm(qmd,inf) - robot.maxVelo;
88 optiRobot.ConTauM = norm(tauM,inf) - robot.peakTorque;
89 optiRobot.ConQ1 = tauM - (vIn*kt/R - kt^2/R*qmd);
90 optiRobot.ConQ1Max = max(optiRobot.ConQ1);
91 optiRobot.ConQ2 = -tauM - (vIn*kt/R - kt^2/R*qmd);
92 optiRobot.ConQ2Max = max(optiRobot.ConQ2);
93 optiRobot.ConQ3 = tauM - (vIn*kt/R + kt^2/R*qmd);
94 optiRobot.ConQ3Max = max(optiRobot.ConQ3);
95 optiRobot.ConQ4 = -tauM - (vIn*kt/R + kt^2/R*qmd);
96 optiRobot.ConQ4Max = max(optiRobot.ConQ4);
97 optiRobot.k = 1/alpha1;
98
99 %-- Are all the constraints satisfied?
100 %-- Checking that the max. value of the LHS of inequalities is less ...
       than 0
101 const = max([optiRobot.ConElongInf; optiRobot.ConQmdInf; ...
       optiRobot.ConTauM;...
102     optiRobot.ConQ1; optiRobot.ConQ2; optiRobot.ConQ3; optiRobot.ConQ4]);
103 if const ≤ 0
104     optiRobot.feasible = true;
105 else
106     optiRobot.feasible = false;
```

```
107 end
108
109 end
110
111 function [Ene] = localFunCoeffOptim(robot, trajectory)
112
113 %--Loading robot parameters
114 iM = robot.iM;
115 r = robot.r;
116 bm = robot.bm;
117 eta = robot.eta;
118 km = robot.km;
119 kt = robot.kt;
120 vIn = robot.voltage;
121 R = robot.R;
122
123 %-- Loading trajectory
124 % ql = trajectory.ql;
125 qld = trajectory.qld;
126 qldd = trajectory.qldd;
127 time = trajectory.time;
128 torqueEla = -trajectory.torque;
129 torqueElad = -trajectory.torqued;
130 torqueEladd = -trajectory.torquedd;
131 % This function has access to the workspace of the parent function
132 % Therefore, the scope of x is the function to which this workspace
133 % belongs, and all functions nested to any level within that function.
134
135 %-- Defining parameters
136 gamma1 = -(iM*torqueEladd*r + bm*torqueElad*r);
137 gamma2 = iM*qldd*r + bm*qld*r -  torqueEla/(eta*r);
138
139 Ene.a = trapz(time, (gamma1.^2/km^2 + bm*r^2*torqueElad.^2));
140 Ene.b = trapz(time, (2*gamma1.*gamma2/km^2 - 2*bm*r^2*qld.*torqueElad));
141 Ene.c = trapz(time, (gamma2.^2/km^2 + bm*qld.^2*r^2 - qld.*torqueEla/eta));
142 Ene.gamma1 = gamma1;
143 Ene.gamma2 = gamma2;
144
145 % Power.a = bm*r^2.*torqueElad.^2;
146 % Power.b = torqueEla.*torqueElad/eta - 2*bm*qld.*torqueElad*r^2;
147 % Power.c = bm*r^2*qld.^2 - torqueEla.*qld / eta;
```

```
148 end
149
150 function [aSqrtRo, bSqrtRo, cSqrtRo] = ...
       localFunRobustQuadraCons(trajectories,...
151     robot)
152
153 %--Loading robot parameters
154 iM = robot.iM;
155 r = robot.r;
156 bm = robot.bm;
157 eta = robot.eta;
158 % vIn = robot.voltage;
159 % R = robot.R;
160 % kt = robot.kt;
161 % km = robot.km;
162 % time = trajectory.time;
163
164 trajNam = fieldnames(trajectories);
165
166 aSqrtRo = zeros(numel(trajNam), 1);
167 bSqrtRo = zeros(numel(trajNam), 1);
168 cSqrtRo = zeros(numel(trajNam), 1);
169
170 for i = 1:numel(trajNam)
171     time = trajectories.(trajNam{i}).time;
172     %-- Loading trajectory
173 %     ql = trajectories.(trajNam{i}).ql;
174     qld = trajectories.(trajNam{i}).qld;
175     qldd = trajectories.(trajNam{i}).qldd;
176     % time = trajectory.time;
177     torqueEla = trajectories.(trajNam{i}).torque;
178     torqueElad = trajectories.(trajNam{i}).torqued;
179     torqueEladd = trajectories.(trajNam{i}).torquedd;
180
181     %-- Defining parameters
182     gamma1 = -(iM*torqueEladd*r + bm*torqueElad*r);
183     gamma2 = iM*qldd*r + bm*qld*r -  torqueEla/(eta*r);
184     ΔT = time(end) - time(1);
185
186     aSqrtRo(i) = ΔT*trapz(time, gamma1.^2);
187     bSqrtRo(i) = ΔT*trapz(time, 2*gamma1.*gamma2);
```

```
188     cSqrtRo(i) = ΔT*trapz(time, gamma2.^2);
189 end
190 end
```

A.3.3 CVX_nonlinearSEA.m

```
1  function [optiRobot] = CVX_nonlinearSEA(robot, trajectory)
2  % CVX_nonlinearSEA  Returns optimal nonlinear SEA using nominal values of
3  % the trajectory and robot parameters.
4  %
5  %   OptimalSEA = CVX_nonlinearSEA(robot, trajectory)
6  %
7  %   -- INPUTS:
8  %           robot:      [struct] Parameters of the electric motor and robot
9  %                                system.
10 %           trajectory: [struct] Desired load trajectory.
11 %
12 %   -- OUTPUTS:
13 %           optiRobot   [struct] Optimal SEA parameters and energy
14 %                                measurements
15 %
16 %   See also CVX_linearSEA_robust, CVX_linearSEA, CVX_nonlinearSEA_robust.
17
18 [cost] = nonlinear_SEA_cost(robot, trajectory);
19
20 %--Constraints
21 [Aineq, bineq] = nonlinear_SEA_constraints(robot, trajectory, cost);
22
23 %-- Computing the sqrt of matrix Q
24 % % http://cvxr.com/cvx/doc/advanced.html?highlight=square_pos
25 % [Qsqrt, ¬, S] = chol( Q0_energy );
26 % Qsqrt = Qsqrt*S;
27
28 cvx_begin quiet
29     %-Mosek, Gurobi, SDPT3, sedumi
30     cvx_solver Mosek
31
32     %-best, default
33     cvx_precision best
```

```
34
35      variable alpha1(length(cost.h),1)
36
37      minimize((alpha1.')*cost.G*alpha1+cost.h*alpha1+cost.k)
38  %       minimize (square_pos(norm(Qsqrt*qm)) + C_energy*qm)
39      subject to
40          Aineq*alpha1 ≤ bineq;
41  cvx_end
42
43  %-- Checking if the solution of cvx can be used
44  if ¬any(strcmp(cvx_status, {'Solved', 'Inaccurate/Solved',...
45          'Inaccurate/Unbounded', 'Inaccurate/Infeasible'}))
46      error(['CVX status:', cvx_status])
47  elseif any(strcmp(cvx_status, {'Inaccurate/Solved',...
48          'Inaccurate/Unbounded', 'Inaccurate/Infeasible'}))
49      warning(['CVX status: ', cvx_status])
50  end
51
52  %- Loading parameters
53  r       = robot.r;
54  time    = trajectory.time;
55  % ΔT  = time(2) - time(1);
56
57  %- Loading trajectories
58  ql      = trajectory.ql(1:end-1);
59  % qld      = trajectory.qld(1:end-1);
60
61  %--Torque done by the spring on the load
62  torque      = -trajectory.torque(1:end-1);
63  torqued     = -trajectory.torqued(1:end-1);
64
65  %-Index when torque is the closest to zero
66  minTorqueIndx = find(abs(torque) == min(abs(torque)));
67
68  %-Defining elongation and setting elongation equal to zero when torque ...
        is close
69  %to zero.
70  elong = cumtrapz(time(1:end-1), torqued.*alpha1);
71  elong = elong - elong(minTorqueIndx);
72
73  qm   = (ql - elong)*r;
```

```
74 qmd  = cost.a+cost.B*alpha1;
75 qmdd = cost.c+cost.D*alpha1;
76
77 qm(end + 1)      = qm(1);
78 qmd(end + 1)     = qmd(1);
79 qmdd(end + 1)    = qmdd(1);
80
81 %-- Post processing the results
82 optiRobot = fPostProcessing(qm, qmd, qmdd, cvx_status, robot, trajectory);
83 optiRobot.compliance = alpha1;
84 %-Periodic motion
85 optiRobot.compliance(end+1) = optiRobot.compliance(1);
86
87 % fprintf('\n--------\n')
88 % fprintf('Nonlinear Nominal:\n')
89 % fprintf('Optimization ...
      Cost[J]:\t%3.2f\tEnergy[J]:\t%3.2f\n',cvx_optval,...
90 %     optiRobot.energy_total)
91
92 end
```

A.3.4 CVX_nonlinearSEA_robust.m

```
1  function [optiRobot] = CVX_nonlinearSEA_robust(robot, trajectory)
2  % CVX_nonlinearSEA_robust  Returns optimal nonlinear SEA using nominal and
3  % uncertain values of the trajectory and robot parameters.
4  %
5  %   OptimalSEA = CVX_nonlinearSEA_robust(robot, trajectory)
6  %
7  %   -- INPUTS:
8  %           robot:      [struct] Parameters of the electric motor and robot
9  %                                system.
10 %           trajectory: [struct] Desired load trajectory.
11 %
12 %   -- OUTPUTS:
13 %           optiRobot   [struct] Optimal SEA parameters and energy
14 %                                measurements
15 %
16 %   See also CVX_linearSEA_robust, CVX_nonlinearSEA, CVX_linearSEA.
```

```
17
18 %--Energy optimization
19 [cons] = nonlinear_SEA_cost(robot, trajectory);
20 %--Constraints
21 [Aineq, bineq] = ...
22     nonlinear_SEA_constraints_robust(robot, trajectory, cons);
23
24 %-- Computing the sqrt of matrix Q
25 % % http://cvxr.com/cvx/doc/advanced.html?highlight=square_pos
26 % [Qsqrt, ¬, S] = chol( Q0_energy );
27 % Qsqrt = Qsqrt*S;
28
29 cvx_begin quiet
30     %-Mosek, Gurobi, SDPT3, sedumi
31     cvx_solver Mosek
32
33     %-best, default
34     cvx_precision best
35
36     variable alpha1(length(cons.h),1)
37
38     minimize((alpha1.')*cons.G*alpha1+cons.h*alpha1+cons.k)
39 %     minimize (square_pos(norm(Qsqrt*qm)) + C_energy*qm)
40
41     subject to
42         Aineq*alpha1 ≤ bineq;
43 cvx_end
44
45 %-- Checking if the solution of cvx can be used
46 if ¬any(strcmp(cvx_status, {'Solved', 'Inaccurate/Solved',...
47         'Inaccurate/Unbounded', 'Inaccurate/Infeasible'}))
48     error(['CVX status:', cvx_status])
49 elseif any(strcmp(cvx_status, {'Inaccurate/Solved',...
50         'Inaccurate/Unbounded', 'Inaccurate/Infeasible'}))
51     warning(['CVX status: ', cvx_status])
52 end
53
54 %- Loading parameters
55 r       = robot.r;
56 time    = trajectory.time;
57 % ΔT  = time(2) - time(1);
```

```
58
59 %- Loading trajectories
60 ql      = trajectory.ql(1:end-1);
61 % qld      = trajectory.qld(1:end-1);
62 %--Torque done by the spring on the load
63 torque      = -trajectory.torque(1:end-1);
64 torqued     = -trajectory.torqued(1:end-1);
65
66 %-Index when torque is the closest to zero
67 minTorqueIndx = find(abs(torque) == min(abs(torque)));
68
69 %-Defining elongation and setting elongation equal to zero when torque ...
      is close
70 %to zero.
71 elong = cumtrapz(time(1:end-1), torqued.*alpha1);
72 elong = elong - elong(minTorqueIndx);
73
74 qm   = (ql - elong)*r;
75 qmd  = cons.a+cons.B*alpha1;
76 qmdd = cons.c+cons.D*alpha1;
77
78 qm(end + 1)     = qm(1);
79 qmd(end + 1)    = qmd(1);
80 qmdd(end + 1)   = qmdd(1);
81
82 %-- Post processing the results
83 optiRobot = fPostProcessing(qm, qmd, qmdd, cvx_status, robot, trajectory);
84 optiRobot.compliance = alpha1;
85 %Periodic
86 optiRobot.compliance(end+1) = optiRobot.compliance(1);
87
88 % fprintf('\n--------\n')
89 % fprintf('Robust:\n')
90 % fprintf('Optimization Cost:\t%3.2f\tEnergy[J]:\t%3.2f\n',cvx_optval,...
91 %     optiRobot.energy_total)
92
93 end
```

Open source software - CodeOcean

A working implementation of the design toolbox has been uploaded to CodeOcean. This CodeOcean repository is accessible in *this link*.

APPENDIX B

IMPLEMENTATION OF NONLINEAR SPRINGS

This appendix provides some references for the implementation of custom nonlinear springs. The manufacturing of springs with custom torque-elongation profile presents practical challenges. For instance, manufacturing the spring prescribed in Fig. 4.16 requires additional mechanical design as off-the-shelf springs normally have a linear torque-elongation relationship. However, manufacturing of nonlinear springs has been already explored in robotic applications (Hawkes and Cutkosky, 2018). Jutte and Kota (Jutte and Kota, 2008) introduced a generalized nonlinear spring synthesis methodology for any prescribed nonlinear load-displacement function. Custom springs can also be designed using topology optimization, especially for hyperelastic structures (Chen et al., 2017). Vanderborght et al. (Vanderborght et al., 2013) summarized available techniques to produce nonlinear springs; cam, hypocycloid, and double-slider mechanisms coupled with linear springs are some examples of these techniques (Realmuto et al., 2015; Thorson and Caldwell, 2011; Park et al., 2009). It is important to note that some of these techniques may introduce energy losses, such as dissipated heat due to friction in the mechanisms, and result in elastic elements that may not be conservative. Materials with inherent nonlinear elasticity, such as polymers, may exhibit viscoelastic behavior with its respective energy losses (Bolivar et al., 2016). To account for these losses, the cost functions in the proposed formulation can be extended to include viscous friction losses in the spring, and the optimization problem remains convex. However, the proposed methodology does not include these losses explicitly as elastic elements are very efficient in practice (Hubicki et al., 2016).

Another approach to achieve nonlinearity is to modify linear torsion spring to have nonlinear capabilities by utilizing multiple spring disks engaging at staggered equilibrium angles (Fig. B.1). This approach has been proposed by our collaborators at U. Michigan. Through this approach, we can create piecewise linear approximations of nonlinear spring profiles.

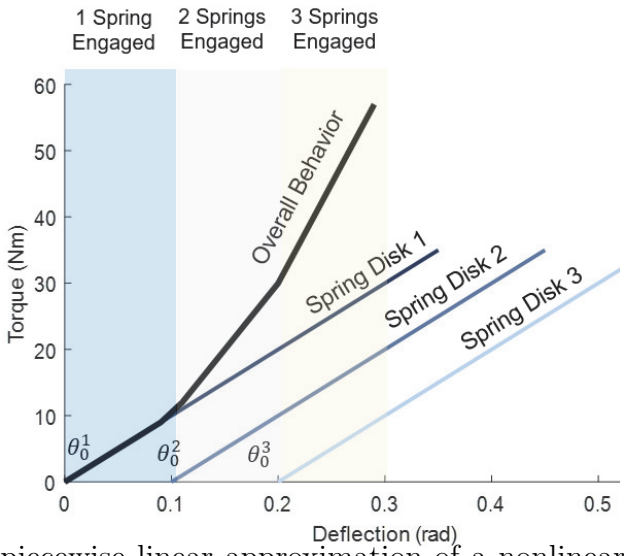

Figure B.1. Example piecewise linear approximation of a nonlinear spring profile that can be achieved using our stackable spring disks, engaging linear springs at different equilibrium positions. More stackable disks in parallel, engaging at different equilibrium positions can be used to achieve a better approximation of a nonlinear spring profile.

REFERENCES

Allen, D., E. Bolívar, S. Farmer, W. Voit, and R. D. Gregg (2019). Mechanical Simplification of Variable-Stiffness Actuators Using Dielectric Elastomer Transducers. *Actuators 8*(2), 44.

Au, S. K. and H. M. Herr (2008). Powered Ankle-Foot Prosthesis. *IEEE Robot. Autom. Mag. 15*(3), 52–59.

Au, S. K., J. Weber, and H. Herr (2009). Powered Ankle-Foot Prosthesis Improves Walking Metabolic Economy. *IEEE Trans. Robot. 25*, 51–66.

Azocar, A. F., L. M. Mooney, L. J. Hargrove, and E. J. Rouse (2018). Design and Characterization of an Open-Source Robotic Leg Prosthesis. In *2018 7th IEEE Int. Conf. Biomed. Robot. Biomechatronics*, pp. 111–118.

Bauer, A., D. Wollherr, and M. Buss (2008). Human-Robot Collaboration: A Survey. *Int. J. Humanoid Robot. 05*(01), 47–66.

Ben-Tal, A., L. El Ghaoui, and A. Nemirovski (2009, October). *Robust Optimization*. Princeton Series in Applied Mathematics. Princeton University Press.

Bicchi, A. and G. Tonietti (2004). Fast and "Soft-Arm" Tactics. *IEEE Robot. Autom. Mag. 11*(2), 22–33.

Bolivar, E., D. Allen, G. Ellson, J. Cossio, W. Voit, and R. Gregg (2016, Aug). Towards a series elastic actuator with electrically modulated stiffness for powered ankle-foot orthoses. In *IEEE International Conference on Automation Science and Engineering (CASE)*, pp. 1086–1093.

Bolívar, E., S. Rezazadeh, and R. D. Gregg (2017). A General Framework for Minimizing Energy Consumption of Series Elastic Actuators With Regeneration. In *ASME Dynamic Systems and Control Conference*, pp. V001T36A005.

Bolivar Nieto, E. A., S. Rezazadeh, and R. Gregg (2019). Minimizing energy consumption and peak power of series elastic actuators: a convex optimization framework for elastic element design. *IEEE/ASME Transactions on Mechatronics 24*(3), 1334–1345.

Bolivar Nieto, E. A., S. Rezazadeh, T. Summers, and R. D. Gregg (2019, Aug). Robust optimal design of energy efficient series elastic actuators: Application to a powered prosthetic ankle. In *IEEE International Conference on Rehabilitation Robotics (ICORR)*.

Boyd, S. P. and L. Vandenberghe (2004). *Convex Optimization*. New York, NY: Cambridge University Press.

Braun, D. J. et al. (2013). Robots Driven by Compliant Actuators: Optimal Control Under Actuation Constraints. *IEEE Trans. Robot. 29*, 1085–1101.

Brown, W. R. and A. G. Ulsoy (2013). A Maneuver Based Design of a Passive-Assist Device for Augmenting Active Joints. *J. Mechanisms Robotics 5*(3), 031003.

Chen, F., Y. Wang, M. Y. Wang, and Y. Zhang (2017). Topology optimization of hyperelastic structures using a level set method. *J. Comput. Phys. 351*, 437–454.

Collins, S., M. Wiggin, and G. Sawicki (2015). Reducing the energy cost of human walking using an unpowered exoskeleton. *Nature 522*(7555), 212–215.

Ding, Y. and H.-w. Park (2017). Design and Experimental Implementation of a Quasi-Direct-Drive Leg for Optimized Jumping. In *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*

Elery, T., S. Rezazadeh, C. Nesler, J. Doan, H. Zhu, and R. D. Gregg (2018). Design and Benchtop Validation of a Powered Knee-Ankle Prosthesis with High-Torque, Low-Impedance Actuators. In *IEEE Int. Conf. Robotics & Automation*.

Embry, K. R., D. J. Villarreal, and R. D. Gregg (2016). A unified parameterization of human gait across ambulation modes. In *IEEE Eng. Med. Bio. Conf.*, pp. 2179–2183.

Embry, K. R., D. J. Villarreal, R. L. Macaluso, and R. D. Gregg (2018, Dec). Modeling the kinematics of human locomotion over continuously varying speeds and inclines. *IEEE Transactions on Neural Systems and Rehabilitation Engineering 26*(12), 2342–2350.

Endo, K. and H. Herr (2014). A Model of Muscle–Tendon Function in Human Walking at Self-Selected Speed. *IEEE Trans. Neural Syst. Rehabil. Eng. 22*.

Gasparri, G. M., F. Fabiani, M. Garabini, L. Pallottino, M. Catalano, G. Grioli, R. Persichin, and A. Bicchi (2016). Robust optimization of system compliance for physical interaction in uncertain scenarios. In *IEEE-RAS 16th Int. Conf. Humanoid Robot.*, pp. 911–918. IEEE.

Grant, M. and S. Boyd (2014). CVX: Matlab Software for Disciplined Convex Programming, version 2.1.

Grant, M. C. and S. P. Boyd (2008). Graph Implementations for Nonsmooth Convex Programs. In *Recent Adv. Learn. Control*, Volume 371, pp. 95–110. London: Springer London.

Grimmer, M. and A. Seyfarth (2011). Stiffness adjustment of a Series Elastic Actuator in an ankle-foot prosthesis for walking and running: The trade-off between energy and peak power optimization. In *IEEE Int. Conf. Robot. Autom.*, pp. 1439–1444. IEEE.

Grioli, G. et al. (2015). Variable stiffness actuators: The user's point of view. *Int. J. Rob. Res. 34*(6), 727–743.

Gurobi Optimization, L. (2018). Gurobi optimizer reference manual.

Ham, R. et al. (2009). Compliant actuator designs. *IEEE Robot. Autom. Mag. 16*(3), 81–94.

Hansen, A. H., D. S. Childress, S. C. Miff, S. a. Gard, and K. P. Mesplay (2004). The human ankle during walking: Implications for design of biomimetic ankle prostheses. *J. Biomech. 37*, 1467–1474.

Hawkes, E. W. and M. R. Cutkosky (2018). Design of Materials and Mechanisms for Responsive Robots. *Annu. Rev. Control. Robot. Auton. Syst. 1*(1), 359–384.

Hirzinger, G. et al. (2002). DLR's torque-controlled light weight robot III - are we reaching the technological limits now? In *IEEE Int. Conf. Robot. Autom.*, Volume 2, pp. 1710–1716. IEEE.

Hogan, N. (1985). Impedance control: An approach to manipulation: Part II-Implementation. *Journal of dynamic systems, measurement, and control 107*(1), 8–16.

Hollander, K. W. et al. (2006). An Efficient Robotic Tendon for Gait Assistance. *J. Biomech. Eng. 128*(5), 788.

Hong, Y., J. Kim, and F. C. Park (2017). Comparative analysis of energy-based criteria for dynamics-based robot motion optimization. In *2017 IEEE Conf. Control Technol. Appl.*, pp. 175–180. IEEE.

Hubicki, C. et al. (2016). ATRIAS: Design and validation of a tether-free 3D-capable spring-mass bipedal robot. *Int. J. Rob. Res. 35*(12), 1497–1521.

Hubicki, C., A. Abate, P. Clary, S. Rezazadeh, et al. (2018). Walking and Running with Passive Compliance: Lessons from Engineering a Live Demonstration of the ATRIAS Biped. *IEEE Robotics and Automation Magazine*. in press.

Hurst, J. and A. Rizzi (2008). Series compliance for an efficient running gait. *IEEE Robot. Autom. Mag. 15*(3), 42–51.

Jafari, A., N. G. Tsagarakis, and D. G. Caldwell (2011). Exploiting natural dynamics for energy minimization using an Actuator with Adjustable Stiffness (AwAS). In *2011 IEEE Int. Conf. Robot. Autom.*, pp. 4632–4637. IEEE.

Jafari, A., N. G. Tsagarakis, and D. G. Caldwell (2013). A novel intrinsically energy efficient actuator with adjustable stiffness (AwAS). *IEEE/ASME Trans. Mechatronics 18*(1), 355–365.

Jutte, C. V. and S. Kota (2008). Design of Nonlinear Springs for Prescribed Load-Displacement Functions. *J. Mech. Des. 130*(8), 081403.

Knabe, C., B. Lee, V. Orekhov, and D. Hong (2014). Design of a Compact, Lightweight, Electromechanical Linear Series Elastic Actuator. In *Vol. 5B 38th Mech. Robot. Conf.* ASME.

Lenzi, T., M. Cempini, L. Hargrove, and T. Kuiken (2018, jul). Design, development, and testing of a lightweight hybrid robotic knee prosthesis. *Int. J. Rob. Res. in press*.

Lenzi, T., M. Cempini, L. J. Hargrove, and T. A. Kuiken (2019, apr). Design, Development, and Validation of a Lightweight Nonbackdrivable Robotic Ankle Prosthesis. *IEEE/ASME Trans. Mechatronics 24*(2), 471–482.

MOSEK-ApS (2019). *The MOSEK optimization toolbox for MATLAB manual. Version 9.0.*

Nesterov, Y. and A. Nemirovskii (1994). *Interior-Point Polynomial Algorithms in Convex Programming*. Society for Industrial and Applied Mathematics.

Novacheck, T. F. (1998). The biomechanics of running. *Gait Posture 7*(1), 77–95.

Paine, N. et al. (2015). Actuator Control for the NASA-JSC Valkyrie Humanoid Robot: A Decoupled Dynamics Approach for Torque Control of Series Elastic Robots. *J. Field Robotics 32*(3), 378–396.

Paine, N., S. Oh, and L. Sentis (2014). Design and Control Considerations for High-Performance Series Elastic Actuators. *IEEE/ASME Trans. Mechatronics 19*(3), 1080–1091.

Park, J., H. Kim, and J. Song (2009). Safe robot arm with safe joint mechanism using nonlinear spring system for collision safety. In *2009 IEEE Int. Conf. Robot. Autom.*, pp. 3371–3376. IEEE.

Perry, J. and J. Burnfield (2010). *Gait Analysis: Normal and Pathological Function*. Slack.

Pratt, G. and M. Williamson (1995). Series Elastic Actuators. In *Proc. 1995 IEEE/RSJ Int. Conf. Intell. Robot. Syst. Hum. Robot Interact. Coop. Robot.*, Volume 1, pp. 399–406. IEEE Comput. Soc. Press.

Quintero, D., D. Villarreal, D. Lambert, S. Kapp, and R. D. Gregg (2018). Continuous-phase control of a powered knee-ankle prosthesis: Amputee experiments across speeds and inclines. *IEEE Trans. Robotics PP*(99).

Quintero, D., D. J. Villarreal, and R. D. Gregg (2016). Preliminary experimental results of a unified controller for a powered knee-ankle prosthetic leg over various walking speeds. In *IEEE Int. Conf. Intelligent Robots Systems*.

Realmuto, J., G. Klute, and S. Devasia (2015). Nonlinear Passive Cam-Based Springs for Powered Ankle Prostheses. *J. Med. Device. 9*(1), 011007.

Rezazadeh, S. and J. Hurst (2014). On the optimal selection of motors and transmissions for electromechanical and robotic systems. In *IEEE International Conference on Intelligent Robots and Systems*, Chicago, IL, USA, pp. 4605–4611.

Robinson, D., J. Pratt, D. Paluska, and G. Pratt (1999). Series elastic actuator development for a biomimetic walking robot. In *IEEE/ASME Int. Conf. Adv. Intell. Mechatronics*, pp. 561–568. IEEE.

Robinson, D. W. (2000). *Design and Analysis of Series Elasticity in Closed-loop Actuator Force Control*. Ph. D. thesis, Massachusetts Institute of Technology.

Rouse, E. J., R. D. Gregg, L. Hargrove, and J. Sensinger (2013). The difference between stiffness and quasi-stiffness in the context of biomechanical modeling. *IEEE Trans. Biomed. Eng. 60*(2), 562–568.

Rouse, E. J., L. M. Mooney, and H. M. Herr (2014). Clutchable series-elastic actuator: Implications for prosthetic knee design. *Int. J. Rob. Res. 33*(13), 1611–1625.

Schutz, S., K. Mianowski, C. Kotting, A. Nejadfard, M. Reichardt, and K. Berns (2016). RRLAB SEA — A highly integrated compliant actuator with minimised reflected inertia. In *IEEE Int. Conf. Adv. Intell. Mechatronics*, pp. 252–257. IEEE.

Seok, S. et al. (2015). Design Principles for Energy-Efficient Legged Locomotion and Implementation on the MIT Cheetah Robot. *IEEE/ASME Trans. Mechatronics 20*(3), 1117–1129.

Shultz, A. H., B. E. Lawson, and M. Goldfarb (2016, apr). Variable Cadence Walking and Ground Adaptive Standing With a Powered Ankle Prosthesis. *IEEE Trans. Neural Syst. Rehabil. Eng. 24*(4), 495–505.

Spong, M. W. (1987). Modeling and Control of Elastic Joint Robots. *J. Dyn. Syst. Meas. Control 109*(4), 310.

Sup, F., A. Bohara, and M. Goldfarb (2008). Design and Control of a Powered Transfemoral Prosthesis. *Int. J. Rob. Res. 27*(2), 263–273.

Thorson, I. and D. Caldwell (2011). A Nonlinear Series Elastic Actuator for Highly Dynamic Motions. In *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pp. 390–394. IEEE.

Tran, M., L. Gabert, M. Cempini, and T. Lenzi (2019, apr). A Lightweight, Efficient Fully Powered Knee Prosthesis With Actively Variable Transmission. *IEEE Robot. Autom. Lett. 4*(2), 1186–1193.

Vanderborght, B. et al. (2006). Exploiting Natural Dynamics to Reduce Energy Consumption by Controlling the Compliance of Soft Actuators. *Int. J. Rob. Res. 25*(4), 343–358.

Vanderborght, B. et al. (2013). Variable impedance actuators: A review. *Rob. Auton. Syst. 61*(12), 1601–1614.

Verstraten, T. et al. (2015). Modeling and design of geared DC motors for energy efficiency: Comparison between theory and experiments. *Mechatronics 30*, 198–213.

Verstraten, T. et al. (2016). Series and Parallel Elastic Actuation: Impact of natural dynamics on power and energy consumption. *Mech. Mach. Theory 102*, 232–246.

Verstraten, T. et al. (2017). On the Importance of a Motor Model for the Optimization of SEA-driven Prosthetic Ankles. In *Wearable Robotics: Challenges and Trends. Biosystems & Biorobotics*, Volume 16, pp. 403–407. Springer.

Winter, D. A. (1983, dec). Biomechanical Motor Patterns in Normal Walking. *J. Mot. Behav. 15*(4), 302–330.

Winter, D. A. (2009). *Biomechanics and Motor Control of Human Movement* (2 ed.). New York, NY: Wiley.

Yang, G. et al. (2018). The grand challenges of Science Robotics. *Science Robotics*.

Zhao, Y., N. Paine, K. S. Kim, and L. Sentis (2015). Stability and Performance Limits of Latency-Prone Distributed Feedback Controllers. *IEEE Trans. Ind. Electron. 62*(11), 7151–7162.

BIOGRAPHICAL SKETCH

Edgar A. Bolívar Nieto received his BS degree in Mechatronics Engineering from the Universidad Nacional de Colombia (2011). Currently, he is a PhD candidate in Mechanical Engineering at The University of Texas at Dallas. He is with the Locomotor Control Systems Lab (LoCoLab) at The University of Texas at Dallas. His goal at the LoCoLab is to provide wearable robots with actuators and control systems that allow synergy between human and machine. Prior to joining the LoCoLab, he was an undergraduate research assistant at The University of Wisconsin-Milwaukee and research engineer at The Universidad Nacional de Colombia.

CURRICULUM VITAE

Edgar A. **Bolívar Nieto**

Graduate Research Assistant, Locomotor Control Systems Laboratory
The University of Texas at Dallas

ebolivar@ieee.org | sites.google.com/view/edgarbolivar | @ebolivarn | ebolivarn

Research Interests

Optimization	Optimization and its application to mechanical and control design, trajectory generation, and system identification
Wearable Robots	Rehabilitation exoskeletons and powered prosthetic legs
Robotic Actuators	Design, modeling, and control of electromechanical systems such as Series Elastic Actuators

Education

The University of Texas at Dallas — *Richardson, TX, USA*

Ph.D. Mechanical Engineering — *Expected Aug. 2019*

Thesis: "*A Robust Convex Optimization Framework for the Design of Series Elastic Actuators*"
Advisor: Dr. Robert D. Gregg

The University of Texas at Dallas — *Richardson, TX, USA*

M.S. Mechanical Engineering — *Dec. 2017*

Advisor: Dr. Robert D. Gregg

Universidad Nacional de Colombia — *Bogotá, Colombia*

B.S. Mechatronics Engineering — *Aug. 2011*

Advisors: Dr. Jorge Sofrony and Dr. Yaoyu Li

Positions

HEBI Robotics — *Pittsburgh, PA, USA*

Optimal Control Engineer Intern - Supervisor: Dr. David Rollinson — *May 2019 - PRESENT*

- Development of optimal trajectory generator
- Implementation of gain free effort control

The University of Texas at Dallas — *Richardson, TX, USA*

Graduate Research Assistant - Supervisor: Dr. Robert D. Gregg — *Aug. 2014 - May 2019*

- Developed convex optimization framework for the design of series elastic actuators
- Formulated optimization frameworks for the robust design of compliant actuators
- Designed and manufactured dielectric-elastomer variable-stiffness actuators
- Assisted on the writing of NSF and NIH grant proposals
- Mentored graduate and undergraduate students

Universidad Nacional de Colombia — *Bogotá, Colombia*

Research Engineer - Supervisor: Dr. Jorge Sofrony — *Aug. 2011 - Aug. 2013*

- Modeled and controlled a parallel 6-DOF Stewart Gough manufacturing robot
- Designed hardware and real-time control using PC-104 industrial computer under the xPC target environment
- Developed drivers for analog and digital I/O with xPC-target

University of Wisconsin-Milwaukee — *Milwaukee, WI, USA*

Undergraduate Research Assistant - Supervisor: Dr. Yaoyu Li — *Aug. 2010 - Jun. 2011*

- Developed nonlinear robust position and force control for wind turbine single-rod hydraulic pitching cylinders
- Implemented real-time control algorithms in Compact-Rio

Publications

PEER-REVIEWED JOURNAL ARTICLES

1. **E. Bolívar**, S. Rezazadeh, and R. D. Gregg, "Minimizing Energy Consumption and Peak Power of Series Elastic Actuators: A Convex Optimization Framework for Elastic Element Design," *IEEE/ASME Trans. Mechatronics*, to be published in the next issue of TMECH
2. D. Allen, **E. Bolívar**, S. Farmer, W. Voit, and R. D. Gregg, "Mechanical Simplification of Variable Stiffness Actuators Using Dielectric Elastomer Transducers," *Actuators*, In Press

PEER-REVIEWED CONFERENCE PROCEEDINGS

1. **E. Bolívar**, S. Rezazadeh, T. Summers, and R. D. Gregg, "Robust Optimal Design of Energy Efficient Series Elastic Actuators: Application to a Powered Prosthetic Ankle," *IEEE Int. Conf. Rehab. Robotics*, (Toronto, Canada), 2019,
2. **E. Bolívar**, S. Rezazadeh, and R. D. Gregg, "A General Framework for Minimizing Energy Consumption of Series Elastic Actuators With Regeneration," in *ASME Dynamic Systems and Control Conference*, (Tysons, VA, USA), p. V001T36A005, 2017. **Best Student Robotics Paper Award, Best Student Paper Finalist**
3. **E. Bolivar***, D. Allen*, G. Ellson, J. Cossio, W. Voit, and R. D. Gregg, "Towards a series elastic actuator with electrically modulated stiffness for powered ankle-foot orthoses," in *IEEE Int. Conf. Automation Science and Engineering*, (Fort Worth, TX, USA), pp. 1086–1093, 2016. * Both authors contributed equally to this manuscript

CONFERENCE ABSTRACTS

1. **E. Bolívar**, S. Rezazadeh, and R. D. Gregg, "How Much Energy Can Really Be Saved Using Series Elastic Actuators?," in *IEEE/RSJ Int. Conf. Intelligent Robots & Systems*, (Vancouver, BC, Canada), 2017

Patents

1. R. D. Gregg, D. P. Allen, **E. Bolívar**, and W. Voit, "Variable stiffness actuator with electrically modulated stiffness," 2018. U.S. Utility Patent Application, Patent application number 16/111,072.

Honors & Awards

2017	**Best Student Robotics Paper Award,** ASME Dynamic Systems and Control Conference	*Tysons, VA, USA*
2017	**Best Student Paper Finalist,** ASME Dynamic Systems and Control Conference	*Tysons, VA, USA*
2014	**Mechanical Engineering Department Graduate Student Scholarship,** The University of Texas at Dallas	*Richardson, TX, USA*
2014	**Doctoral Student Scholarship,** COLCIENCIAS - Administrative Department of Science, Technology, and Innovation	*Bogotá, Colombia*
2010	**Young Researcher and Innovator,** COLCIENCIAS - Administrative Department of Science, Technology, and Innovation	*Bogotá, Colombia*
2009	**Member of the Cohort of Best Engineering Students,** Universidad Nacional de Colombia	*Bogotá, Colombia*

Research Funding Experience

NSF National Robotics Initiative 2.0: Ubiquitous Collaborative Robots *USA*

NRI:FND:COLLAB: Optimal Design of Robust Compliant Actuators for Ubiquitous Co-Robots, $750,000, 09/15/18–08/31/21, Award Number: 1830360 *2018*

- Contributed to the planning and writing of the grant proposal
- Developed the preliminary work and aims for the proposal

National Institutes of Health/NICHD (1R01HD094772) *USA*

Controlling Locomotion over Continuously Varying Activities for Agile Powered Prosthetic Legs, *2018*

$2,227,090, 09/01/18—05/31/2023

- Contributed to the preliminary work section of the grant proposal

Teaching Experience

Southern Methodist University *Dallas, TX, USA*

EE 7360 Analog and Digital Control Systems. **Invited lecturer:** "Sneak peak to optimal control" *Fall 2017*

The University of Texas at Dallas *Richardson, TX, USA*

MECH 6314: Engineering Systems Modeling and Simulation. Teaching Assistant *Spring 2017, Spring 2016*

- **Invited lecturer:** "System Identification Using Least Squares" (Spring 2017)
- Graded homework, quizzes, and exams
- Developed grading metrics
- Formulated assignments and exams
- Complied with ABET evaluation metrics
- Developed, evaluated, and reported ABET - Course Learning Outcomes

Universidad Nacional de Colombia *Bogotá, Colombia*

Robotics. Teaching Assistant *Spring 2010*

Invited Talks

2016 **Towards unified control of prosthetic legs with a human-inspired phase variable**, IEEE CASE 2016 Workshop on Automation for Assistive Healthcare (in substitution for Dr. Robert D. Gregg) *Fort Worth, TX, USA*

Scholarly Review

Conference Reviewer

1. IEEE International Conference Robotics and Automation
2. American Control Conference
3. ASME Dynamic Systems and Control Conference
4. International Conference on Rehabilitation Robotics

Advising and Mentoring

Students

2018	**Vikram Louis Michael**, M.Sc. Student Mechanical Engineering, UT Dallas	*USA*
2018	**Abhishek Chakladar**, Undergraduate Student Computer Science, UT Dallas	*USA*
2016	**Vinnicius Rocha de Lima**, Undergraduate Student Mechanical Engineering, Federal University of Pernambuco	*Brazil*
2016	**Alexandre Leboucher**, Undergraduate Student Mechanical Engineering, McGill University	*Canada*
2015	**Jorge Cossio**, Undergraduate Student Bioengineering, UT Dallas	*USA*

Undergraduate Senior Design Team

2018	**Real-Time Optimization on an Exoskeleton Using Variable Stiffness Actuators,** Senior Design, UT Dallas	*USA*

Languages

- **English,** fluent. TOEFL: 108/120
- **Spanish,** native language

Training and Certifications

2018	**ACADEME Fellow,** NSF-sponsored two-weeks workshop: Teaching and Research Skills for Academic Careers	*Houston, TX, USA*
2015	**Responsible Conduct of Research,** The University of Texas at Dallas	*Richardson, TX, USA*
2015	**Certified LabVIEW Associate Developer,** National Instruments	*Richardson, TX, USA*

Professional Memberships

- **Student Member,** Institute of Electrical and Electronics Engineers (IEEE)
  - Control Systems Society
  - Robotics & Automation Society

References

1. **Dr. Robert D. Gregg**, The University of Texas at Dallas
   Assistant Professor, Eugene McDermott Professor
   Department of Bioengineering, Department of Mechanical Engineering
   Email: rgregg@utdallas.edu, Phone: (+1) 972 883 4657
2. **Dr. Tyler Summers**, The University of Texas at Dallas
   Assistant Professor
   Department of Mechanical Engineering, Department of Electrical Engineering
   Email: tyler.summers@utdallas.edu, Phone: (+1) 972 883 4554
3. **Dr. Elliot Rouse**, The University of Michigan
   Assistant Professor, Core Faculty Robotics Institute
   Department of Mechanical Engineering
   Email: ejrouse@umich.edu, Phone: (+1) 734 763 3629
4. **Dr. Siavash Rezazadeh**, The University of Texas at Dallas
   Research Scientist
   Department of Bioengineering
   Email: siavash.rezazadeh@utdallas.edu