Investigation of Extremum Seeking Control for Adaptive Exercise Machines

Brahm T. Powell
Cleveland State University

Follow this and additional works at: http://engagedscholarship.csuohio.edu/etdarchive
Part of the Mechanical Engineering Commons
How does access to this work benefit you? Let us know!

Recommended Citation
http://engagedscholarship.csuohio.edu/etdarchive/1015

This Thesis is brought to you for free and open access by EngagedScholarship@CSU. It has been accepted for inclusion in ETD Archive by an authorized administrator of EngagedScholarship@CSU. For more information, please contact library.es@csuohio.edu.
INVESTIGATION OF EXTREMUM SEEKING CONTROL FOR ADAPTIVE EXERCISE MACHINES

Brahm T. Powell

Bachelors of Science in Mechanical Engineering
Cleveland State University
May 2016

submitted in partial fulfillment of requirements for the degree
MASTERS OF SCIENCE IN MECHANICAL ENGINEERING
at the
CLEVELAND STATE UNIVERSITY
AUGUST 2017
We hereby approve this thesis for

Brahm T. Powell

Candidate for the Master of Science in Mechanical Engineering degree for the
Department of Mechanical Engineering
and the CLEVELAND STATE UNIVERSITY
College of Graduate Studies

Thesis Chairperson, Hanz Richter, Ph.D.

Department & Date

Thesis Committee Member, Antonie van den Bogert, Ph.D.

Department & Date

Thesis Committee Member, Daniel Simon, Ph.D.

Department & Date

Student’s Date of Defense: August 11, 2017
ACKNOWLEDGMENTS

Many thanks go to my primary research advisor, Dr. Richter, for developing my research abilities and stretching my limits. I would also like to thank Dr. Halloran for introducing me to my first research opportunity and guiding me along that path. Working alongside my fellow research students has also been immensely encouraging. I would also like to thank my family, who put up with my odd hours of work and helped me through whatever they could. Finally, I praise the Lord for the strength and wisdom He has given me to get through everything I have been through.

This study was supported by the National Science Foundation through grant 1544702 from the Cyber-Physical Systems Program.
INVESTIGATION OF EXTREMUM SEEKING CONTROL FOR
ADAPTIVE EXERCISE MACHINES

Brahm T. Powell

ABSTRACT

Many muscle rehabilitation regimens are non-adaptive and recommended subjectively by physicians. While there are advantages to having the feedback of a qualified physician, utilizing real-time muscle performance feedback could be beneficial. An extremum seeking control design is proposed to fulfill the need for an automated, load-varying exercise machine that can optimize muscle performance.

Several steps are outlined to contribute to the realization of this goal. First, the extremum seeking control scheme is discussed. Second, the Hill muscle model will be described. Theoretical muscle effort extrema will be derived for selected optimization cases, namely maximizing average squared power by varying load stiffness. Thirdly, a muscle-actuated linkage framework will be developed for simulation. This framework allows for automated creation of a linkage with an arbitrary number of links and muscles with easily customizable parameters. Finally, the controller will be simulated against the linkage to demonstrate the feasibility of the proposed control design. Successful completion of these steps is crucial to the development of an adaptive exercise machine. Although feasibility is not shown for every type of load or performance measure, the proposed framework is streamlined enough to allow for - and encourage - future research and customization.
TABLE OF CONTENTS

ABSTRACT iv

LIST OF TABLES viii

LIST OF FIGURES ix

I INTRODUCTION 1
   1.1 Motivation ........................................ 2
   1.2 Literature Review ................................. 5
   1.3 Thesis Contributions and Organization ............ 6

II EXTRENUM SEEKING CONTROL 8
   2.1 Overview and Motivation for Extremum Seeking Control . 9
   2.2 Setup and Proof of Stability ....................... 11
      2.2.1 Intuitive Convergence ....................... 13
      2.2.2 Formal Proof of Stability ................... 15
   2.3 Miscellaneous Variants ............................ 17
      2.3.1 Deterministic vs. Stochastic Perturbations . 17
      2.3.2 Other Variants ............................... 19
   2.4 Example Problems ................................. 20
      2.4.1 Scalar Input (Single Degree of Freedom) ..... 21
      2.4.2 Vector Inputs ............................... 22
   2.5 Extremum Seeking Control Applications ............ 23
   2.6 Discussion ....................................... 25
### III SINGLE-MUSCLE MODEL

3.1 Hill Muscle Model ........................................ 27  
3.2 Parameter Selection, Extrema Existence ............... 31  
  3.2.1 Output Measures ...................................... 31  
  3.2.2 Existence of Extrema ................................. 36  
3.3 Controlled Single-Muscle System ......................... 47  
  3.3.1 Sliding Mode Control of a Single Muscle .......... 48  
  3.3.2 Trajectory Tracking Simulation Example .......... 50  
3.4 Muscle Extrema Validation ............................... 52  
  3.4.1 Ensuring Assumptions are Met ...................... 52  
  3.4.2 Simulation Results ................................. 55  
3.5 Discussion .............................................. 64  

### IV MULTI-MUSCLE LINKAGE MODEL

4.1 Linkage Control Design ................................. 67  
  4.1.1 Synthetic Control: Inverse Dynamics .......... 69  
  4.1.2 Synthetic Control: Impedance Control ........ 70  
  4.1.3 Backstepping Controller .......................... 72  
  4.1.4 Composite Controller Stability .................. 74  
4.2 Algorithmic Adaptations for Simulation ............... 76  
  4.2.1 Transformation Matrices ......................... 77  
  4.2.2 Gravity Vector ..................................... 78  
  4.2.3 Coriolis Matrix .................................... 79  
  4.2.4 Mass Matrix Derivative ......................... 79  
  4.2.5 Velocity Jacobian Derivative ................. 80
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.3 Simulation Example</td>
<td>81</td>
</tr>
<tr>
<td>4.3.1 Impedance Control</td>
<td>83</td>
</tr>
<tr>
<td>4.3.2 Inverse Dynamics</td>
<td>85</td>
</tr>
<tr>
<td>4.4 Discussion</td>
<td>87</td>
</tr>
<tr>
<td>V SIMULATION OF FULL CONTROLLED LINKAGE MODEL</td>
<td>89</td>
</tr>
<tr>
<td>5.1 General Load Setup</td>
<td>90</td>
</tr>
<tr>
<td>5.2 Constant Load Parameters</td>
<td>91</td>
</tr>
<tr>
<td>5.2.1 Single-Link Manipulator</td>
<td>91</td>
</tr>
<tr>
<td>5.2.2 Two-Link Manipulator</td>
<td>101</td>
</tr>
<tr>
<td>5.3 Extremum Seeking Simulations</td>
<td>106</td>
</tr>
<tr>
<td>5.3.1 Single-Link Manipulator</td>
<td>107</td>
</tr>
<tr>
<td>5.3.2 Two-Link Manipulator</td>
<td>113</td>
</tr>
<tr>
<td>5.3.3 Interpretation of Results</td>
<td>117</td>
</tr>
<tr>
<td>5.4 Discussion</td>
<td>120</td>
</tr>
<tr>
<td>VI CONCLUSIONS AND FUTURE WORK</td>
<td>122</td>
</tr>
<tr>
<td>BIBLIOGRAPHY</td>
<td>127</td>
</tr>
<tr>
<td>APPENDIX</td>
<td>138</td>
</tr>
<tr>
<td>A. CODE REPOSITORY</td>
<td>139</td>
</tr>
</tbody>
</table>
## LIST OF TABLES

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Muscle example parameters</td>
<td>50</td>
</tr>
<tr>
<td>II</td>
<td>Muscle example parameters - power optimization</td>
<td>56</td>
</tr>
<tr>
<td>III</td>
<td>Muscle example parameters - squared power optimization</td>
<td>61</td>
</tr>
<tr>
<td>IV</td>
<td>Two-link manipulator example muscle placement</td>
<td>82</td>
</tr>
<tr>
<td>V</td>
<td>Two-link manipulator example parameters</td>
<td>83</td>
</tr>
<tr>
<td>VI</td>
<td>One-link manipulator example muscle placement</td>
<td>92</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Extremum seeking controller setup</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>Extremum seeking controller setup, plant visualization</td>
<td>14</td>
</tr>
<tr>
<td>3</td>
<td>Example ESC inputs and outputs - 1 DOF</td>
<td>21</td>
</tr>
<tr>
<td>4</td>
<td>Example ESC inputs and outputs - 2 DOF</td>
<td>23</td>
</tr>
<tr>
<td>5</td>
<td>Single Hill muscle</td>
<td>27</td>
</tr>
<tr>
<td>6</td>
<td>Hill tendon force-length relationship</td>
<td>29</td>
</tr>
<tr>
<td>7</td>
<td>Hill $PE$ force-length relationship</td>
<td>30</td>
</tr>
<tr>
<td>8</td>
<td>Double-frequency sinusoid example</td>
<td>41</td>
</tr>
<tr>
<td>9</td>
<td>Hill muscle, SMC tracking</td>
<td>51</td>
</tr>
<tr>
<td>10</td>
<td>Hill muscle, SMC controls</td>
<td>51</td>
</tr>
<tr>
<td>11</td>
<td>ESC single muscle $P_{CE,avg}$ optimization - tracking</td>
<td>57</td>
</tr>
<tr>
<td>12</td>
<td>ESC single muscle $P_{CE,avg}$ optimization - muscle lengths and rates</td>
<td>58</td>
</tr>
<tr>
<td>13</td>
<td>ESC single muscle $P_{CE,avg}$ optimization - load stiffness</td>
<td>60</td>
</tr>
<tr>
<td>14</td>
<td>ESC single muscle $P_{CE,avg}$ optimization - $P_{CE,avg}$</td>
<td>60</td>
</tr>
<tr>
<td>15</td>
<td>ESC single muscle $P_{CE,avg}^2$ optimization - tracking</td>
<td>62</td>
</tr>
<tr>
<td>16</td>
<td>ESC single muscle $P_{CE,avg}^2$ optimization - muscle lengths and rates</td>
<td>63</td>
</tr>
<tr>
<td>Page</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>ESC single muscle $P_{CE,avg}^2$ optimization - load stiffness</td>
<td>64</td>
</tr>
<tr>
<td>18</td>
<td>ESC single muscle $P_{CE,avg}^2$ optimization - $P_{CE,avg}^2$</td>
<td>64</td>
</tr>
<tr>
<td>19</td>
<td>Two-link setup</td>
<td>82</td>
</tr>
<tr>
<td>20</td>
<td>Two-link manipulator under impedance control - tracking</td>
<td>84</td>
</tr>
<tr>
<td>21</td>
<td>Two-link manipulator under impedance control - controls</td>
<td>84</td>
</tr>
<tr>
<td>22</td>
<td>Two-link manipulator under impedance control - lengths and rates</td>
<td>85</td>
</tr>
<tr>
<td>23</td>
<td>Two-link manipulator under inverse dynamics control - tracking</td>
<td>86</td>
</tr>
<tr>
<td>24</td>
<td>Two-link manipulator under inverse dynamics control - controls</td>
<td>86</td>
</tr>
<tr>
<td>25</td>
<td>Two-link manipulator under inverse dynamics control - lengths and rates</td>
<td>87</td>
</tr>
<tr>
<td>26</td>
<td>One-link setup</td>
<td>92</td>
</tr>
<tr>
<td>27</td>
<td>One link, inverse dynamics (setpoint), constant $c$ - tracking</td>
<td>93</td>
</tr>
<tr>
<td>28</td>
<td>One link, inverse dynamics (setpoint), constant $c - F$</td>
<td>94</td>
</tr>
<tr>
<td>29</td>
<td>One link, inverse dynamics (setpoint), constant $c$ - controls</td>
<td>94</td>
</tr>
<tr>
<td>30</td>
<td>One link, impedance control (setpoint), constant $c$ - tracking</td>
<td>95</td>
</tr>
<tr>
<td>31</td>
<td>One link, impedance control (setpoint), constant $c - F$</td>
<td>96</td>
</tr>
<tr>
<td>32</td>
<td>One link, impedance control (setpoint), constant $c$ - controls</td>
<td>96</td>
</tr>
<tr>
<td>33</td>
<td>One link, inverse dynamics (trajectory), constant $c$ - tracking</td>
<td>97</td>
</tr>
<tr>
<td>34</td>
<td>One link, inverse dynamics (trajectory), constant $c - F$</td>
<td>98</td>
</tr>
<tr>
<td>35</td>
<td>One link, inverse dynamics (trajectory), constant $c$ - controls</td>
<td>98</td>
</tr>
<tr>
<td>36</td>
<td>One link, impedance control (trajectory), constant $c$ - tracking</td>
<td>99</td>
</tr>
<tr>
<td>37</td>
<td>One link, impedance control (trajectory), constant $c - F$</td>
<td>100</td>
</tr>
<tr>
<td>38</td>
<td>One link, impedance control (trajectory), constant $c$ - controls</td>
<td>100</td>
</tr>
</tbody>
</table>
CHAPTER I

INTRODUCTION

Muscle deterioration and injury is a common problem arising from a variety of situations. For example, approximately 7 million people in the United States have experienced a stroke [18], and many of these individuals experience or will experience some muscular difficulties. Astronauts need to be able to combat muscle deterioration and bone density loss in microgravity [8] [10]. Competitive sports boast a high number of athlete injuries; roughly 1 million injuries were reported in NCAA (National Collegiate Athletic Association) athletes within a recent period of five academic years [29]. Elderly adults tend to suffer from rapid muscle tissue loss due to inactivity, a phenomenon that accelerates with age [14]. All of these point to the need for rehabilitative or strengthening therapies, which could arise in the form of individualized exercise machines.
1.1 Motivation

Muscle injury and deterioration is a very inhibitive problem for many people, and there is a drought of user-tailored adaptive exercise machines to assist in the rehabilitation or strength-rebuilding process. Many exercise machines are non-adaptive. That is, the load parameters are set initially by a physician or user and are held constant. Though some can be modified manually, most machines do not adapt to the user’s specific physical needs. Some exercise machines such as described in [12] employ feedback control to vary load parameters; however, the controller is designed for joint trajectory tracking or joint torque tracking instead of muscle effort optimization.

One area of need for adaptive exercise machines is stroke patient rehabilitation. Approximately 7 million Americans over the age of 20 have experienced a stroke [18], and stroke patients commonly experience post-stroke muscular problems. Stroke recovery has been shown to be more effective when resistive exercises are incorporated into patients’ exercise regimens [55]. There are several robotic devices [49] [25] [22] that fall under the umbrella of non-adaptive stroke rehabilitation machines. The majority of stroke rehabilitation machines are not focused on exercise and strengthening, however; most focus on regaining mobility, generally by using machine learning and pattern recognition to associate some bodily signal with controlled assisted output motion [7]. Though the brain and body interface designs proposed in [7] and [45] must learn a specific user’s brain activity or other bodily signal patterns, they do not aim to modify load parameters to maximize muscle effort.
There are some non-adaptive exercise machines designed for counteracting the effects of microgravity [8]. Significant bone density loss has been a problematic byproduct of extended microgravity exposure, as well as muscle tissue loss, especially in gravity-counteracting muscles. Most zero-gravity exercise machines focus on lower-body resistive force application such as the device produced in [10].

Most machines have been designed with user generalization in mind instead of focusing on real-time customization. That is, the goal of most studies has been to improve the performance of a particular device or controller without regard to user-specific needs (other than necessary kinematic and physical constraints). Many rehabilitation devices put a focus on trajectory tracking and torque control instead of muscle effort maximization. That is, the user is supposed to follow a certain motion profile, and the system will assist in trajectory tracking. Load parameters are not optimized to maximize muscle effort. For example, in [4], a device is proposed that has increased mobility over other ankle rehabilitation robots. However, the purpose was not to optimize load conditions for muscle effort exertion, but rather to more properly accommodate the natural motion of a human ankle and allow for at-home usability. The device itself was non-adaptive. A knee rehabilitation robot was proposed in [2], but again, its purpose was to enable motion recording and trajectory tracking. Load (torque) parameters were non-adaptive.

Other rehabilitation and exercise devices do not necessarily aim for trajectory optimization. The knee rehabilitation device proposed in [39] is one such case. This device does measure muscle activations, but does not aim to
optimize muscle effort by modifying load parameters.

Some machines in practice and under development are aimed more towards sports training and are controlled with the intent of mimicking actual forces experienced during certain activities such as rowing [19]. In these cases, there is certainly a potential for load optimization with the intention of strengthening certain muscles of interest; however, such an adaptive approach is still a novel concept.

It is interesting that, considering the wealth of work in the area of human-machine system control, there is only one documented instance of design of an adaptive exercise machine [11]. Other human-machine systems involving interaction control generally focus on load-sharing mechanisms for rehabilitation, such as the ones mentioned above and in [46]. Some recent developments in the area of exoskeletons have also contributed greatly to the field of human-machine load sharing control [1].

One of the more interesting human-machine control system cases is intelligent user control of prostheses. In [40], several aspects of this problem were discussed and analyzed: neural signal analysis, prosthesis design, computer vision for external object identification, training of necessary learning algorithms for associating inputs with prosthesis actions, and more. However, such an approach focuses on prosthesis control instead of strengthening existing muscles.
1.2 Literature Review

Although human-machine systems utilizing interaction control have been at the forefront of both medical and commercial automation [47] technology for some time, there has been little research in the area of exercise/rehabilitation machines that adapt to user specific needs in real-time. The medical example closest to the intent of this paper is the research conducted in [11]. In [11], an extremum seeking control design scheme was proposed for maximization of user power output. The proposed machine would modulate load torque to optimize user power output. However, there are several assumptions, system simplifications, and application-specific design considerations that are made that make further research and optimization desirable.

First of all, the proposed machine and controller in [11] assumes a single-degree-of-freedom system. This is rarely the case in a physical workout, where a person consists of multiple degrees-of-freedom with several muscle inputs. It would be much more beneficial to have a generalized human model (primarily for simulation purposes, but also for experimental generalization) and exercise machine model to accommodate for more complex exercises.

Second, the controller is designed with user trajectory control in mind. Though strict enforcement of a target trajectory may be a goal of a particular exercise regimen, it may be desirable to design an extremum seeking controller that uses feedback from the user without strictly enforcing a trajectory (or being streamlined in such a way that would allow for additional control goals such as user trajectory enforcement). Though it would generally be logical
for the user to intend to follow some reference trajectory, it may be the case that a new ideal trajectory will be discovered through reaction to the load parameters generated by the extremum seeking controller.

Third, only one type of feedback from the user is assumed. It would be helpful to know whether different performance measures can be used for different applications and research. It may even be desirable to allow for weighted average feedback from multiple sources, so that certain muscles’ effort output can be optimized more intensely than others’.

Fourth, it is assumed in [11] that the user’s muscle power output is directly related to the user’s speed of motion during the workout. This is not necessarily the case. Because of the redundancy of muscle placement in the human body and the elasticity of muscles, the power output of a particular muscle or set of muscles is dependent on more than just limb trajectory.

Fifth, generalization of load parameters is not considered. If the controller were designed to allow for an arbitrary load setup, any number of exercise machines could be accommodated. It would be very simple to adapt the controller for use in other machines.

1.3 Thesis Contributions and Organization

A basic framework for development of an adaptive exercise machine employing extremum seeking control is proposed. The exercise machine aims to optimize some user output such as muscle power by modifying load parameters. Although several metrics could be selected, this work focuses primarily on cases in which the average squared muscle power over one period of mo-
tion is taken to be the performance metric. Specifically, the “muscle power” refers to the power output of the contractile element of the muscle (defined in Chapter III). Average squared muscle activations will also be discussed as a possible performance measure, but it will be shown that this approach does not yield maxima. Included is the design of a generalized muscle-actuated human linkage model as well as design considerations for the extremum seeking controller. The muscle-actuated linkage model developed is a framework that allows the user to generate a linkage with an arbitrary number of links and muscles. The linkage generation framework employs Hill-type muscles for joint actuation. Integration of example models with an extremum seeking controller is analyzed to verify the existence of extrema. Extrema are shown to exist in some cases, but not all.

Chapter II will provide an overview and background of extremum seeking control. Chapter III will present the single-muscle model used for simulation against the extremum seeking controller. This muscle model will be used in the development of a robotics-motivated muscle-actuated linkage in Chapter IV that will be used to model a human acting against the proposed exercise machine. This linkage will then be simulated against the extremum seeking controller, and results will be discussed in Chapter V. Finally, Chapter VI will close with a conclusion and a discussion of proposed future work.
CHAPTER II

EXTRENUM SEEKING CONTROL

The primary advantage of the proposed exercise machine design is its ability to adapt to the user’s needs. It will vary load parameters such as virtual stiffness to maximize some measure of muscle effort. In order to seek this maximum, an extremum seeking controller is proposed.

In this chapter, a basic extremum seeking control (ESC) algorithm is outlined and discussed. Section 2.1 supplies background information on the algorithm and Section 2.2 summarizes a proof of stability of the controller. Section 2.3 describes different classes of the algorithm. Two simple examples to visualize the behavior and convergence of ESC are presented in Section 2.4, while Section 2.5 discusses applications of Extremum Seeking Control. The chapter is concluded with a discussion in Section 2.6.
2.1 Overview and Motivation for Extremum Seeking Control

Extremum seeking control (ESC) is not new, but first appeared in the 1920’s [3] and was widely used in practice since the 1950’s [44]. Extremum seeking control is widely considered the first major adaptive control scheme [3]. What is widely accepted as the first documented form of ESC was an optimizing controller for power transfer between a train and overhead powering wires [34]. The control design underwent major development in the 1940’s in several countries, including the USSR [3], but the first known English publication did not surface until 1951. This was the year in which [13] was published, which details the design of an ESC system for optimizing internal combustion engine power output by finding an optimal ignition timing schedule. The 1960’s were especially fertile in yielding research and development in the area of ESC [52], but the following three decades exhibited a shift away from this method of control in mainstream applications [52] [3]. Although progress was still made, there was a heavier focus on newer, developing adaptive controllers, including designs such as model reference adaptive control [3] that utilized more analytically appealing methods such as Lyapunov analysis. Emphasis was put on stabilization instead of optimization, which made emerging adaptive controllers desirable. It was not until 2000 that interest in ESC was revived, when a thorough analysis of the stability of ESC was put forth [32]. Since then, many applications have been explored, and variants of the generic ESC scheme multiplied rapidly. Several modern applications are discussed in
Section 2.5.

A distinction does need to be made between ESC and modern forms of adaptive control. Whilst most modern adaptive controllers aim to regulate a system to a *known* trajectory or setpoint, ESC attempts to locate and enforce an *unknown* optimal output. In ESC, input parameters are varied to optimize some performance measure of a given system by continuous measurement evaluation [60] [32]. Another difference between ESC and most adaptive control schemes is that ESC is not model-based - that is, the controller has no knowledge of the plant (although there are variants that include some knowledge of plant parameters). The plant is some unknown or uncertain nonlinear static mapping (or can be approximated as such) that does not have a straightforward analytical solution. An extremum seeking controller can be used to modulate the input parameter set until the desired extremum is approximately located. ESC has been shown to converge to some neighborhood of the optimal parameter value that produces an extremum [32].

In order for an extremum to be located, some scalar performance indicator must be readily available by measurement or calculation. This performance measure $y(\theta)$ must be a static input-output mapping dependent upon the input parameter vector $\theta$. An input parameter could be any tunable parameter that impacts the value of the performance function [20]. The controller varies the input parameter either deterministically or stochastically and then measures the change in the performance indicator, which is used to continuously update the estimate $\hat{\theta}$ of the parameters until the controller reaches a neighborhood of the optimal parameter set $\theta^*$. 
In some cases, a system is partially known with some parameter uncertainty, as in [20] and [42]. In this case, the plant does not need to be treated as a black box, as some knowledge is available about the plant.

It is also possible to modify the algorithm to search for the global extrema [53]. However, most flavors of extremum seeking control (including the scheme utilized by this work) are only designed to seek local optima.

ESC was selected for this study due to its ability to locate maxima (or minima) of an unknown function. Analytical derivation of a human’s point of maximum muscle effort exertion with respect to load conditions is virtually impossible. This can be attributed not only to the significant physical variation between subjects, but also to the complexity of the setup of the human body in general and the effects of time on the human body. Although an optimal load setup can be approximately derived for a single idealized muscle, such a setup does not adequately describe the behavior of a real human. Thus, numerical optimization via a process such as ESC becomes necessary.

2.2 Setup and Proof of Stability

Figure 1 illustrates the basic setup of the controller.
In Figure 1, $\hat{\theta}$ is the current estimate of the optimal parameter set; $\theta$ is the modulated input that is passed to the plant; $y$ is the performance measure of the plant; $\eta$ is the DC component of the signal $y$; and $\xi$ is the DC component of the parameter estimation.

The controller begins with an initial parameter estimate $\hat{\theta}$, which is perturbed by some modulation parameter. For this study, a sinusoidal perturbation is chosen (for reasons discussed in Section 2.3). This modulated $\theta$ is then passed to the plant and the performance indicator $y$ is measured. This performance measure $y$ is then passed through a high-pass filter to obtain the high-frequency component (denoted as $y - \eta$) of the output $y$. This high-frequency signal component is multiplied by the modulation signal, and the DC component $\xi$ of this signal is extracted via a low-pass filter. The signal $\xi$ is then passed to an integrator that updates the estimate $\hat{\theta}$ of the parameters.

We can intuitively infer convergence of $\theta$ to a (local) optimum, as well as prove formally that such a system is stable. The following proofs are adapted from [32].
2.2.1 Intuitive Convergence

Let it be assumed that the plant consists of some general state-space dynamic system $\dot{x} = f(x, u)$ cascaded with some output performance measure function $y = h(x)$. If the input $u$ is defined as some function of the state variable $x$ and some tunable system parameter set $\theta$ so $u = \alpha(x, \theta)$, then the state equation can be rewritten as $\dot{x} = f(x, \alpha(x, \theta))$. For the sake of visualization and conceptual understanding, let $\theta$ be taken as a scalar value, even though this approach can be extended to the case when $\theta$ is a vector of parameters. If $\theta$ is a vector of parameters, then it may be necessary to form a state-space representation of the high- and low-pass filters for scalability (as state-space simulation blocks allow for easier multi-input, multi-output system definitions), especially if it is desired to vary the filtering frequencies between parameters.

Assuming that the input $\theta$ is varied slowly enough by the extremum seeking controller such that $x = l(\theta)$ is a smooth function of $\theta$ alone, then the plant dynamics can be modeled as follows:

$$\dot{x} = f(l(\theta), \alpha(l(\theta), \theta)) \quad (2.1)$$
$$y(\theta) = h(l(\theta)) \quad (2.2)$$

As can be seen, the state equation as well as the output equation become functions of $\theta$ alone.

Let it now be assumed that $y$ has a local maximum at $\theta^*$. That is, $y'(\theta^*) = 0$ and $y''(\theta^*) < 0$. This allows us to re-imagine Figure 1 as shown in Figure 2, if $\theta$ is treated as a scalar value:
When a perturbation $P(t)$ is applied to the current estimate $\hat{\theta}$ and passed into the plant, some output $y$ is calculated. If this output can be linearized at a point, then the high-frequency component of the signal can be approximated as

$$y - \eta \approx y'(\theta)P(t)$$

which is essentially the component of $y$ that results due to the perturbation. This is then multiplied by the perturbation signal to yield the following:

$$(y - \eta)P(t) \approx y'(\theta)P(t)^2$$

For the sake of an intuitive study of stability, the sign of this value (dependent solely on the sign of $y'(\theta)$) is more important than the magnitude. Note that, because $\theta^*$ is a local maximum, $y'(\theta) > 0$ for $\theta < \theta^*$ and $y'(\theta) < 0$ for $\theta > \theta^*$ when $\theta$ is close enough to the local optimum $\theta^*$. Therefore, when $\theta < \theta^*$, a positive $\xi$ will be extracted by the low-pass filter in the ESC and passed to the integrator, which will increase the estimate $\hat{\theta}$, thus moving it closer to $\theta^*$. When $\theta > \theta^*$, a negative $\xi$ will be extracted by the low-pass filter and passed.
to the integrator, which will decrease the estimate $\hat{\theta}$, thus moving it closer to $\theta^*$. Thus, it is intuitive that ESC should approximately converge to the local maximum $\theta^*$.

2.2.2 Formal Proof of Stability

Until recently, there has not existed a formal proof of stability of the extremum-seeking control scheme. However, within the last couple decades, a few proofs of stability have arisen, which have even allowed researchers to analyze the stability of particular classes of ESC. Here a general proof put forth by [32] will be briefly summarized. Note that this proof assumes a scalar parameter set $\theta$, although the proof can easily accommodate a vector of parameters. Also, this proof assumes a sinusoidal (deterministic) perturbation.

We must first form a preliminary state-space summary of ESC:

$$
\begin{bmatrix}
\dot{x} \\
\dot{\hat{\theta}} \\
\dot{\xi} \\
\dot{\eta}
\end{bmatrix} =
\begin{bmatrix}
f(x, \alpha(x, \hat{\theta} + a \sin(\omega t))) \\
K \xi \\
\omega_l ((y - \eta) a \sin(\omega t) - \xi) \\
\omega_h (y - \eta)
\end{bmatrix}
$$

(2.5)

Take the parameter estimation error to be $\tilde{\theta} = \hat{\theta} - \theta^*$ and the DC error to be $\tilde{\eta} = \eta - h(l(\theta^*))$. We will now break up the constants $\omega_h$, $\omega_l$, and $k$ into products of the positive constants $\omega$ and $\delta$ and some other constant ($\omega'_H$, $\omega'_L$, or $K'$, for $\omega_h$, $\omega_l$, and $k$, respectively).

$$
\omega_h = \omega \omega_H = \omega \delta \omega'_H
$$

(2.6)

$$
\omega_l = \omega \omega_L = \omega \delta \omega'_L
$$

(2.7)

$$
k = \omega K = \omega \delta K'
$$

(2.8)
This breakdown allows the system to be rewritten using a new time scale, \( \tau = \omega t \). From here, the extremum seeking control system can be modeled in state-space form as seen in Equations 2.9 and 2.10.

\[
\omega \frac{dx}{d\tau} = f(x, \alpha(x, \tilde{\theta} + \theta^* + a \sin \tau)) \tag{2.9}
\]

\[
\frac{d}{d\tau} \begin{bmatrix} \tilde{\theta} \\ \xi \\ \tilde{\eta} \end{bmatrix} = \delta \begin{bmatrix} K' \xi \\ \omega'_L((h(x) - h(l(\theta^*)) - \tilde{\eta})a \sin \tau - \xi) \\ \omega'_H(h(x) - h(l(\theta^*)) - \tilde{\eta}) \end{bmatrix} \tag{2.10}
\]

From here, an averaging analysis can be applied. Holding \( x \) at its pseudo-equilibrium such that \( x = l(\theta^* + \tilde{\theta} + a \sin \tau) \) and substituting into Equation 2.10 produces a reduced system, denoted by the reduced states \([\tilde{\theta}_r, \xi_r, \tilde{\eta}_r]^T\):

\[
\frac{d}{d\tau} \begin{bmatrix} \tilde{\theta}_r \\ \xi_r \\ \tilde{\eta}_r \end{bmatrix} = \delta \begin{bmatrix} K' \xi_r \\ \omega'_L((v - \tilde{\eta}_r)a \sin \tau - \xi_r) \\ \omega'_H(v - \tilde{\eta}_r) \end{bmatrix} \tag{2.11}
\]

Here, \( v \) is a function of \((\tilde{\theta}_r + a \sin \tau)\) such that the following is true:

\[
v = v(\tilde{\theta}_r + a \sin \tau) = h(l(\theta^* + \tilde{\theta}_r + a \sin \tau)) - h(l(\theta^*)) \tag{2.12}
\]

The average model, described by \([\tilde{\theta}_r^a, \xi_r^a, \tilde{\eta}_r^a]^T\), can now be written as

\[
\frac{d}{d\tau} \begin{bmatrix} \tilde{\theta}_r^a \\ \xi_r^a \\ \tilde{\eta}_r^a \end{bmatrix} = \delta \begin{bmatrix} K' \xi_r^a \\ \omega'_L(-\xi_r^a + \frac{a}{2\pi} \int_0^{2\pi} v_a \sin \sigma d\sigma) \\ \omega'_H(-\tilde{\eta}_r^a + \frac{1}{2\pi} \int_0^{2\pi} v_a d\sigma) \end{bmatrix} \tag{2.13}
\]

where \( v_a = v(\tilde{\theta}_r^a + a \sin \sigma) \).

The average equilibrium of this system is reached when:

\[
\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \int_0^{2\pi} v(\tilde{\theta}_r^{a,e} + a \sin \sigma) \sin \sigma d\sigma \\ \frac{1}{2\pi} \int_0^{2\pi} v(\tilde{\theta}_r^{a,e} + a \sin \sigma) d\sigma - \tilde{\eta}_r^{a,e} \end{bmatrix} \tag{2.14}
\]

In Equation 2.14, the superscript \( ^e \) denotes the equilibrium value of the state variable. From here, we assume that \( \tilde{\theta}_r^{a,e} \) can be approximated as a polynomial of the form \( \tilde{\theta}_r^{a,e} = b_1 a + b_2 \sigma^2 + H \) where \( H \) is a collection of higher-order terms.
with respect to $a$. Applying the techniques described in [32], this yields the following solution:

$$\begin{bmatrix}
\tilde{\theta}_r^{a,e} \\
\xi_r^{a,e} \\
\tilde{\eta}_r^{a,e}
\end{bmatrix} = \begin{bmatrix}
-v^\prime(0) a^2 + H \\
0 \\
\frac{v''(0)}{4} a^2 + H
\end{bmatrix} \tag{2.15}
$$

Evaluating the eigenvalues of the Jacobian $J_r^a$ of the averaged system in Equation 2.13 at the equilibrium point will tell us the system is only stable if $\int_0^{2\pi} v'(\tilde{\theta}_r^{a,e} + a \sin \sigma) \sin \sigma d\sigma < 0$. Manipulating this inequality and calculating the determinant of the Jacobian $J_r^a$ reveals that the average system is exponentially stable for small values of $a$. Therefore, the original system has a stable periodic solution.

For more details on this proof of stability, see [32].

### 2.3 Miscellaneous Variants

Though there are many variations and elaborations on the basic extremum seeking control scheme, it is common to classify an extremum seeking controller by the type of signal modulation: deterministic or stochastic. Other variants on the control scheme exist as well.

#### 2.3.1 Deterministic vs. Stochastic Perturbations

In a deterministic controller, the perturbation signal $P(t)$ follows some smooth periodic function. This function is usually a sinusoid. Such a perturbation has been chosen for the proposed adaptive exercise machine. In a stochastic controller, the perturbation signal $P(t)$ is some random noise-
generating function. This signal is generally required to be a zero-mean perturbation such as Gaussian noise [38].

One of the primary advantages of deterministic perturbations is the smoothness of parameter variations. There are many cases in which smooth parameter variation is helpful for ensuring that the dynamics of the system are not interfered with. Because ESC can only be used on steady-state (static mapping) calculations in a dynamic system, noisy or abrupt parameter variation may introduce dynamics that distort the output measurements.

However, if $\theta$ is a vector of parameters, care must be taken in choosing the perturbation vector $P(t)$. $P(t)$ would need to be a vector of periodic functions with unique frequencies to satisfy orthogonality requirements [38]. There must be a different perturbation frequency for each parameter $\theta_i$, so that the effects of the parameters on the output $y$ are decoupled. If multiple parameters are coupled, then the effect of the perturbation on one parameter estimate will be proportional to the effect on other parameters, and the parameter estimates will be constrained to a subset of values that may not contain the optimal parameter combination. It is therefore quite difficult to ensure orthogonality of the perturbation vector for high-dimensional systems.

One of the advantages of stochastic or otherwise rapid perturbations is the potential for much faster convergence [20]. When there is little or no chance of interfering with the dynamics of the plant, high frequency or noisy perturbations may be desirable for speed of convergence. This is difficult to implement, however, if the dynamics of the system can be excited under rapid parameter variation, which will cause the input-output mapping to become
Stochastic perturbations are also advantageous for systems where the predictability of deterministic perturbations may be problematic or unnatural. For example, for signal tracking via vehicle movement, it may be desirable to avoid the predictability of periodic perturbations so that the vehicle is not easily pursuable [38]. Also, many biological systems tend to follow random probing instead of periodic probing for optima tracking [36].

Because noisy perturbations would interfere with the dynamics of a human under loading of an adaptive exercise machine, deterministic perturbations were chosen for the current study.

2.3.2 Other Variants

There are several variations on the general ESC scheme. Two primary classes of perturbation were already discussed. In addition to those, relay ESC and sliding mode ESC are somewhat popular variants [43].

In relay ESC, the search direction is changed based on the calculated gradient $dy/d\theta$. When the gradient is calculated, its sign is fed to a switching law, which then outputs a positive or negative value (often a constant) that is fed to the integrator to update the parameter estimate. In this case, both the high- and low-pass filters may be removed from the scheme if desired. Removal of these components greatly simplifies the control scheme, both visually and conceptually. There is also no explicit load perturbation [15], because it is usually assumed that measurement noise or system disturbance produces some perturbations that can help the controller converge. One drawback of
this method is that tuning is often needed to prevent high-frequency switching once the extremum is nearly reached. Gradient estimation can also be difficult, depending on the application and the measuring equipment. Gradient estimation is often a major consideration when designing relay extremum seeking controllers.

Sliding mode ESC, though it may seem similar to relay ESC, is actually a distinct version of ESC. In sliding mode ESC, no gradient calculation is necessary [43]. Instead, a switching law is implemented, though not quite the same as in the relay ESC case. Sliding mode ESC uses sliding mode control theory to define a sliding function, as described in [9] and [43]. This sliding function can then be used to determine a switching law for the controller. Again, it is interesting to note that there are no intentional parameter perturbation signals outside of the stated update law. In addition, many recent advancements have incorporated other areas of controls and mathematics research into this branch of ESC. For example, there has been study into the use of fractional-order control for sliding mode ESC [57].

2.4 Example Problems

To visualize the effects of ESC and better understand its behavior, two simple examples are presented: optimization of a single degree-of-freedom system and optimization of a two degree-of-freedom system. Both cases involve simple quadratic input-output mappings to show that the actual known optimum is found by the controller.
2.4.1 Scalar Input (Single Degree of Freedom)

Here, a simple single-input problem statement is put forth and solved using ESC. Consider the plant equation

\[ y(\theta) = 1 - \theta^2 \]  

(2.16)

with known global maximum at \( \theta^* = 0 \). Such a simple scalar input-output mapping with a known maximum was selected to show that an extremum seeking controller is indeed able to converge to an expected maximum. Let the amplitude and frequency of input modulation be \( a = 0.5 \) and \( \omega = 3 \) respectively, and also let the low- and high-pass filter frequencies be \( \omega_l = \omega/5 \) and \( \omega_h = \omega \) and the estimator gain be \( k = 0.8 \).

The extremum seeking controller can now be initialized with the estimate \( \hat{\theta} = 4 \). \( \hat{\theta} \) can be seen to converge approximately to \( \theta^* \), as shown by the plot of \( \hat{\theta} \) in Figure 3.

![Figure 3: Example ESC inputs and outputs - 1 DOF](image-url)
Figure 3 displays three different measures with respect to time: the parameter estimate \( \hat{\theta} \), the actual perturbed input \( \theta = \hat{\theta} + a \sin(\omega t) \) that is passed to the plant, and the measured output \( y(\theta) \) that is passed back to the extremum seeking controller. It can be seen that \( \hat{\theta} \) does converge to approximately \( \theta^* \). Note that the actual input \( \theta \) to the system will continue to oscillate as long as the extremum seeking controller is perturbing the signal.

2.4.2 Vector Inputs

Here, a system with a vector of inputs is put forth for maximization via ESC. Consider the plant equation

\[
y(\theta) = 1 - \theta_1^2 - \theta_2^2
\]

with known global maximum at \( \theta^* = [0, 0] \). Let the modulation amplitude and frequency vectors be \( a = [0.4, 0.4] \) and \( \omega = [1, 2] \) respectively, and also let the low- and high-pass filter frequencies be \( \omega_l = \omega/5 \) and \( \omega_h = \omega \) and let the estimator gain be \( k = 0.5 \).

The extremum seeking controller can now be initialized with the estimate \( \hat{\theta} = [4, 4] \). \( \hat{\theta} \) can be seen to converge approximately to \( \theta^* \), as shown by the plot of \( \hat{\theta} \) in Figure 4.
It can be seen in Figure 4 that $\hat{\theta}$ does converge to approximately $\theta^*$.

### 2.5 Extremum Seeking Control Applications

Extremum seeking control has had a wide variety of applications. One of the most common uses is in anti-lock braking systems (ABS) in modern cars. Although it has been used for some time in cars [60], it is still an active area of research [37]. Extremum seeking control is used in ABS to find the instantaneous optimal braking torque based on current road and wheel conditions. ESC has also been applied to intensity tracking problems [61]. In intensity tracking, a vehicle or some device attempts to locate the source of...
a signal based upon the vehicle’s orientation and the strength of the signal. This problem has also been studied in the case of vehicles with a limited range of motion and other constraints [16]. Another vehicle-related application of ESC is in traffic light timing [33]. Many urban environments experience traffic changes from day to day, season to season, and year to year, which makes continual tuning of the light timing parameters necessary.

ESC is not only used for transportation-related control methods. In robotics, ESC has been applied to constrained motion tasks with parametric uncertainty [31]. Such a case requires continual updating of parameters (including control parameters) to adapt to an uncertain environment. Automatic flight formation optimization can also employ ESC [5]. There have even been recent studies to analyze and optimize energy consumption costs among a large group of users using ESC [56].

ESC is often used for application to renewable energy sources as well [17]. ESC is a very popular method for maximum power tracking in wind turbines via turbine angle modulation [28]. In fact, several different methods employing ESC have been attempted and implemented for turbine control. Some studies utilize an ESC model without explicit perturbations, instead relying on wind turbulence to produce a searching signal [41]. Other studies investigate using ESC in multi-turbine wind farms, where individual turbines can affect the efficiency of downwind turbines [62]. Some studies propose an ESC scheme where the model is partly known, and a specialized wind speed signal is needed to calculate the reference rotor speed [26]. In this case, the reference value was approximated using artificial neural networks. ESC has also been studied
for solar power collection, for use in finding the point of maximum power generation with photovoltaic panels [59]. Such an application can actually produce several local optima, and a special multi-modal form of ESC can be used to survey as many local optima as possible and find the global optima [6].

This is only a brief list of the applications of ESC, but it illustrates well the diversity of use cases of this control method.

\section*{2.6 Discussion}

In this chapter, extremum seeking control was introduced and discussed. The general control scheme was outlined, and a brief proof of stability was put forth after the intuitive convergence of the controller was shown. The two primary modes of parameter variation, deterministic and stochastic, were discussed, along with other variations on the extremum seeking control scheme. A simple example was shown to illustrate the convergence behavior of the extremum seeking algorithm. Finally, applications of extremum seeking control were surveyed to show the wide variety of use cases for this method of control. Now, before developing a final control scheme using ESC for an adaptive exercise machine, a human muscle model must be developed and analyzed to determine if any extrema could possibly exist.
CHAPTER III

SINGLE-MUSCLE MODEL

Before designing a muscle performance optimizing controller, a muscle-powered human model must be developed. This model should be used in simulation to approximate the behavior of a physical human in reaction to the proposed exercise machine. The first step in development of a human model is development of a single-muscle model. Fortunately, there has been previous work in mathematical modeling of human muscles. A Hill-type muscle model as described in [58] has been selected for use in this study. This model has been constructed with control applications in mind, and is therefore well suited for this study.

If extrema can be shown to exist for a single muscle, then it is certainly not implausible to suppose that extrema may exist in the case of a multi-muscle linkage. In order to show this, an analytical extremum could be derived, or a single muscle could be set up in simulation against an extremum seeking controller to numerically discover an optimum. In the case of a numerical simulation, the muscle model will require integration of a controller for trajectory tracking.
In Section 3.1, the basic structure of a Hill muscle model and its governing equations are presented. Possible measures of muscle effort with respect to selected load types are discussed in Section 3.2. In this section, the existence of extrema is proven or disproven for various performance measures (namely squared muscle activations, power, and squared power), generally assuming a spring-type load. A sliding mode controller is proposed to control the single-muscle model in Section 3.3, and simulation performance is discussed. The existence of extrema is verified by extremum seeking simulation results in Section 3.4. The chapter is concluded with a discussion.

3.1 Hill Muscle Model

This study assumes a Hill-type muscle model [58]. The Hill model was originally introduced by A. V. Hill in 1938 [23], but is still commonly used for muscle force analysis or estimation [21] and is still an active area of research [27] [54]. The basic Hill model was chosen both for its application to control theory and its simplistic (yet acceptably accurate) design. A Hill-type muscle setup is shown in Figure 5 (retrieved from [35]).

A Hill muscle consists of three primary components: the parallel/passive element $PEE$ or $PE$, the series elastic element $SEE$ (or tendon), and the

\[
\begin{array}{c}
\text{CE} \\
\text{SEE}
\end{array}
\]
contractile element $CE$. Generally, for dynamic equation derivation, a mass $m$ is attached to the end of the muscle (tendon side). In addition, the muscle must also be connected to an arbitrary restraining load $F$. The parallel and series elements are nonlinear springs. The contractile element produces the control input that drives the motion of the muscle. Although the control input is considered generally as the neural signal $n$ supplied to the muscle, for tractability in this very preliminary study, the contraction rate $u$ will be considered the control input. If the neural signals were to be considered, there would be a slightly more complex control scheme. The neural signal $n$ would pass through activation dynamics to produce an activation level $a$ (a value that is constrained to be between zero and one), and the contraction rate $u$ can then be calculated as a static mapping of $a$.

Because the contractile element can only produce force in the direction of contraction, an external force or antagonist muscle must always be acting in opposition to the muscle to achieve trajectory tracking. A muscle cannot resist compressive force on its own. In fact, in humans, muscles are arranged antagonistically or redundantly to allow for bi-directional (or three-dimensional) controlled movement. Antagonist muscle setups have even been shown to be more effective for stabilization of an inverted pendulum than certain types of direct torque control [48].

One of the advantages of the Hill model is the consideration of tendon elasticity and muscle excitation-contraction properties. Previously developed models only rarely included these properties when analyzing muscle force generation [58].
A Hill muscle as shown in Figure 5 can be described in state space as

\[
\begin{bmatrix}
\dot{x}_1 \\
\dot{x}_2 \\
\dot{L}_s
\end{bmatrix} = \begin{bmatrix}
x_2 \\
\frac{1}{m}(-\phi_s(L_s) + F) \\
x_2 + u
\end{bmatrix}
\] (3.1)

where \(x_1 = L_s + L_c\) is the length of the muscle, \(L_c\) is the CE length, \(L_s\) is the tendon length, \(\phi_s(L_s)\) describes the force produced by the tendon as a function of its length, \(m\) is the mass attached to the end of the muscle, \(F\) is the external tensile load, and \(u = -\dot{L}_c\) is the contraction rate of the CE.

A general, non-dimensionalized form of the nonlinear relationship \(\phi_s(L_s)\) between tendon force and tendon length is shown in Figure 6.

![Figure 6: Hill tendon force-length relationship](image)

Figure 6 reveals that a Hill-type tendon can operate in three regions: the slack region, the polynomial region, and the linear region. The exact placement of these regions varies from muscle to muscle, but the general shape is assumed for all muscles. In the slack region, zero or nearly zero force is produced. It is assumed that this region has zero slope; however, because the inverse of this function is needed for simulation, a very small slope can be introduced. In the linear region, the tendon force is a purely linear function of tendon length. In
the polynomial region, a 5th degree polynomial is constructed to allow for a smooth transition between the slack and linear regions.

The nonlinear equation being used to describe the relationship $\phi_p(L_c)$ between $PE$ force and $PE$ length $L_c$ is similar, and is illustrated in Figure 7.

![Figure 7: Hill PE force-length relationship](image)

The $PE$ operates in two regions: the slack region and the polynomial region. The slack region is treated similarly to the slack region in the tendon. In the polynomial region, a $3^{rd}$ degree polynomial extends smoothly from the end of the slack region and approaches positive infinity as $PE$ length increases.

These functions can be used to describe the total length $x_1$ of the system and the force relationship between elements.

$$x_1 = L_s + L_c$$  \hspace{1cm} (3.2)

$$\phi_s(L_s) = \phi_p(L_c) + F_{CE}$$  \hspace{1cm} (3.3)

In Equation 3.3, $F_{CE}$ is the force produced by the $CE$ element.
3.2 Parameter Selection, Extrema Existence

A simplified, single-muscle system allows for analytical discovery of extrema. If extrema can be shown to exist for a single muscle, then it would not be irrational to propose use of extremum seeking control to find optima for more complex multi-muscle systems. In addition, extremum seeking control can be applied to these simpler systems to verify the analytical solutions.

Although the eventual proposed exercise machine should be abstracted from the type of parameter $\theta$ and performance measure $y$, it would be time-consuming to prove existence of extrema for every possible case. Therefore, a subset of cases can be chosen, merely to verify that extrema can exist.

Although it is not necessary to analytically derive extrema for a system if ESC will be applied (because the goal of ESC is to numerically seek an extremum), this study aims to begin with a system that does have analytical extrema, as a proof-of-concept study. Future work will not necessitate the analytical derivation of muscle effort extrema.

3.2.1 Output Measures

Three possible performance measures $y$ are analyzed: average $CE$ power, mean-squared $CE$ power, and mean-squared muscle activations. Some power output methods yielded optima, while analysis of muscle activations did not.

Assuming perfect trajectory tracking, let the muscle have the following sinusoidal trajectory $x_1(t)$:

$$x_1(t) = A \sin(\omega t) + x_0$$  \hspace{1cm} (3.4)
Let it also be assumed that the parallel element is operating in the slack region where $\phi_{p,\text{slack}}(L_c) \approx 0$ and the tendon is operating in the linear region. If the tendon is operating in the linear region, then the tendon force length relationship will have the following structure:

$$\phi_s(L_s) = k_s L_s + \phi_{s0}$$  \hspace{1cm} (3.5)

This allows Equations 3.2 and 3.3 to be rewritten as follows:

$$A \sin(\omega t) + x_0 = L_s + L_c$$  \hspace{1cm} (3.6)

$$F_{CE} = k_s L_s + \phi_{s0}$$  \hspace{1cm} (3.7)

The second derivative of the trajectory can be set equal to the second Hill muscle state equation (see Equation 3.1) to obtain

$$-\omega^2 A \sin(\omega t) = \frac{1}{m} (- (k_s L_s + \phi_{s0}) + F)$$  \hspace{1cm} (3.8)

which can be reordered to show that $L_s(t)$ is

$$L_s(t) = \frac{d_0}{k_s} \sin(\omega t) - \frac{\phi_{s0}}{k_s} + \frac{1}{k_s} F$$  \hspace{1cm} (3.9)

where $d_0 = \omega^2 mA$. Applying Equation 3.6, $L_c(t)$ can be described as

$$L_c(t) = (A - \frac{d_0}{k_s}) \sin(\omega t) + \frac{\phi_{s0}}{k_s} + x_0 - \frac{1}{k_s} F$$  \hspace{1cm} (3.10)

and $\dot{L}_c(t)$, or $-u(t)$, would then be

$$\dot{L}_c(t) = -\frac{1}{k_s} \left( \omega (d_0 - Ak_s) \cos(\omega t) + \dot{F} \right)$$  \hspace{1cm} (3.11)

and according to Equation 3.7, $F_{CE}$ would be:

$$F_{CE} = d_0 \sin(\omega t) + F$$  \hspace{1cm} (3.12)
Mean Squared Muscle Activation Calculation

The CE force in a Hill muscle is dependent upon the activation level $a$, which takes on a value between zero (no muscle stimulation) and one (maximum stimulation). The maximum possible force that can be generated by the CE element is muscle-dependent, but can be generally denoted as $F_m$. The CE force is also dependent on the CE length $L_c$ and contraction rate $u = -\dot{L}_c$. This full relationship between states, activation level, and force is described in Equation 3.13.

$$F_{CE} = aF_m f(L_{cn})g(u_n)$$

(3.13)

In Equation 3.13, $f(L_{cn})$ is the length dependence function and $g(u_n)$ is the velocity dependence function. $L_{cn} = L_c/L_{cs}$ is the “normalized” CE length if $L_{cs}$ is the optimal force-producing length of the CE. The normalized contraction rate is $u_n = -\dot{L}_{cn}$. The full relationship can be rewritten to find the activation level $a$ as a function of the other parameters:

$$a = \frac{F_{CE}}{F_m f(L_{cn})g(u_n)}$$

(3.14)

The length and velocity dependence functions are defined as follows, adapted from [23] and [24]:

$$f(L_{cn}) = e^{-((L_{cn}-1)/W)^2}$$

(3.15)

$$g(u_n) = \begin{cases} 
\frac{A_g - A_g z_m + z_m (1 + A_g) u_n}{A_g - A_g z_m + (1 + A_g) u_n}, & \forall u_n < 0 \\
\frac{A_g - A_g u_n}{A_g + u_n}, & \forall u_n \geq 0 
\end{cases}$$

(3.16)

Here, $W$ is some width dependence factor, $A_g$ is a muscle constant, and $z_m$ is the maximum eccentric to isometric force ratio. Each of these values is a
positive constant.

The average squared muscle activation over the $r^{th}$ portion of a cycle of a cycle of motion can be described by Equation 3.17:

$$a_{avg}^2 = \frac{1}{rT} \int_{t_0}^{t_0+rT} \left( \frac{(d_0 \sin(\omega t) + F)}{F_m f(L_{cn})g(u_n)} \right)^2 dt$$ (3.17)

**Average Power Calculations**

It is important to specify how the power should be calculated. Here, it will be assumed that the power output of the contractile element (the primary force-producing element) is being measured. Then, multiple power calculations will be discussed: average power over one cycle, mean-squared power over one cycle, average power over a quarter cycle, and mean-squared power over a quarter cycle.

There is some justification for analyzing the average power over a quarter cycle. The specific region that will be analyzed in this study is the interval $\pi/2 \leq \omega t \leq \pi$, which is the portion of the concentric region of exercise in which the muscle is contracting and accelerating in the direction of contraction. A full cycle cannot be evaluated when measuring average muscle power because, as will be shown (for a spring-like load), the average power over one cycle will be zero. In addition, the average power over a half cycle will be zero. A quarter cycle, on the other hand, will yield nonzero results. For details, see Section 3.2.2.

Now $CE$ power can be defined as $P = F_C E \dot{L}_c$. Simplifying the notation of $\cos(n \omega t)$ and $\sin(n \omega t)$ to simply $c_n$ and $s_n$, respectively, the average $CE$ power and average squared $CE$ power over a fraction $r$ of a cycle would be:
\[ P_{\text{avg}} = \frac{1}{rT} \int_{t_0}^{t_0+rT} (d_0 s_1 + F) \left( -\frac{1}{k_s} \right) \left( -\omega A k_s c_1 + \omega d_0 c_1 + \dot{F} \right) dt \] (3.18)

\[ P_{\text{avg}}^2 = \frac{1}{rT} \int_{t_0}^{t_0+rT} (d_0 s_1 + F)^2 \left( \frac{1}{k_s^2} \right) \left( -\omega A k_s c_1 + \omega d_0 c_1 + \dot{F} \right)^2 dt \] (3.19)

For temporary simplification, letting

\[ \tau_1 = d_0 s_1 + F \] (3.20)

\[ \tau_2 = -\omega A k_s c_1 \] (3.21)

allows the average power as shown in equation 3.18 to be rewritten as

\[ P_{\text{avg}} = -\frac{1}{rT k_s} \int_{t_0}^{t_0+rT} (\tau_1 \dot{\tau}_1 + \tau_1 \tau_2) dt \] (3.22)

and partially evaluated as

\[ P_{\text{avg}} = -\frac{1}{rT k_s} \left( \frac{1}{2} \tau_1^2 + \int_{t_0}^{t_0+rT} \tau_1 \tau_2 dt \right) \] (3.23)

which can be expanded as

\[ P_{\text{avg}} = -\frac{1}{rT k_s} \left( \frac{1}{2} (d_0 s_1 + F)^2 \bigg|_{t_0}^{t_0+rT} - \omega A k_s \int_{t_0}^{t_0+rT} (d_0 s_1 + F)c_1 dt \right) \] (3.24)

and evaluated further to be

\[ P_{\text{avg}} = -\frac{1}{rT} \left( \frac{1}{2} \left( d_0 s_1 + F \right)^2 - d_0 A s_1^2 \right) \bigg|_{t_0}^{t_0+rT} - \omega A \int_{t_0}^{t_0+rT} c_1 F dt \] (3.25)

which will be used shortly for a case-by-case analysis of maximum power for various types of loads.

The average squared power as derived in Equation 3.19 involves many more terms upon expansion and will not yet be derived. Once load profiles are selected and some simplifications can be made, then the integral will be
3.2.2 Existence of Extrema

Below are examples of cases that do or do not produce extrema that can theoretically be achieved via an extremum seeking controller. For those that have theoretical extrema, their location will be derived and later (see Section 3.4) tested in simulation.

Note that all proofs below assume $F = F(t)$ to be a periodic function; that is, $F(t_0) = F(t_0 + T)$. More specifically, spring-like loads are analyzed, such that $F = c(\beta - x_1(t))$ where $c$ is the spring constant of the load and $\beta > x_0 + A$ is the equilibrium position of the spring (implying that the spring is always preloaded).

A very interesting yet simple phenomenon occurs when the load spring stiffness is $c = \omega^2 m$. Remembering that $d_0 = A\omega^2 m$, we can rewrite Equation 3.12 using the spring-like definition of the external load $F$:

$$F_{CE} = A\omega^2 m \sin(\omega t) + \omega^2 m (\beta - (A \sin(\omega t) + x_0))$$
$$= \omega^2 m (\beta - x_0)$$ \hfill (3.26)

Because $c = \omega^2 m$ produces something akin to passive vibration absorption, the force in the $CE$ (as well as the tendon, when the $PE$ force is zero) is constant, as shown in Equation 3.26. In addition, it can be seen that the tendon remains at a constant length:

$$L_s(t) = \frac{1}{k_s} \left( A\omega^2 m \sin(\omega t) - \phi_{s0} + \omega^2 m (\beta - (A \sin(\omega t) + x_0)) \right)$$
$$= \frac{1}{k_s} \left( \omega^2 m (\beta - x_0) - \phi_{s0} \right)$$ \hfill (3.27)
This occurs even though the CE is still in motion (yet, as stated previously, experiences constant force):

\[
L_c(t) = \frac{1}{k_s} \left( (Ak_s - A\omega^2m) \sin(\omega t) + \phi_{s0} + k_s x_0 - \omega^2m(\beta - (A \sin(\omega t) + x_0)) \right) \\
= \frac{1}{k_s} \left( Ak_s \sin(\omega t) + \phi_{s0} + k_s x_0 - \omega^2m(\beta - x_0) \right) \\
= x_1(t) + \frac{1}{k_s} (\phi_{s0} - \omega^2m(\beta - x_0))
\]

(3.28)

It will be seen that the case of \( c = \omega^2m \) will be significant throughout the derivations of extrema.

**Squared Muscle Activations - One Cycle**

Let us assume that the magnitude of oscillation \( A \) is very small. We can therefore linearize \( f(L_{cn}) \) and \( g(u_n) \):

\[
f(L_{cn}) \approx f'(L_{cn}(0))L_{cn} + f(L_{cn}(0)) = f_0' L_{cn} + f_0 \quad (3.29)
\]

\[
g(u_n) \approx g'(0)u_n + 1 \quad (3.30)
\]

Let \( f(L_{cn}) \) be further approximated as constant such that \( f(L_{cn}) \approx f(L_{cn}(0)) = f_0 \). This allows the average squared activation to be rewritten as follows:

\[
a_{avg}^2 = \frac{1}{F_m f_0 r T} \int_{t_0}^{t_0 + r T} \left( \frac{(d_0 \sin(\omega t) + F)}{\frac{g'(0)}{k_s} \left( \omega(d_0 - Ak_s) \cos(\omega t) + \dot{F} \right) + 1} \right)^2 dt \quad (3.31)
\]

If a simple spring load is applied such that \( F = c(\beta - x_1(t)) \) where \( c \) is the spring constant of the load and \( \beta > x_0 + A \) is the equilibrium position of the spring (implying that the spring load is always preloaded), then
\[ a_{avg}^2 = \frac{1}{F_m f_0 r T} \int_{t_0}^{t_0 + r T} \left( \frac{(d_0 - cA) \sin(\omega t) + c(\beta - x_0)}{g'(0) \omega (d_0 - Ak_s - cA) \cos(\omega t) + 1} \right)^2 \, dt \]  \hspace{1cm} (3.32)

Now, if Equation 3.32 is written in the form

\[ a_{avg}^2 = \frac{1}{F_m f_0 r T} \int_{t_0}^{t_0 + r T} \left( \frac{A_1 \sin(\omega t) + A_2}{A_3 \cos(\omega t) + 1} \right)^2 \, dt \]  \hspace{1cm} (3.33)

where \( A_n \) are constants, the integral can be evaluated as follows:

\[ a_{avg}^2 = \left( \frac{A_3(s_1(A_1^2(A_3^2 - 1) + A_2^2A_3^2) + 2A_1A_2(A_3^2 - 1))}{F_m f_0 r T A_3^2 \omega(A_3^2 - 1)(A_3c_1 + 1)} \right. \\
\left. + \frac{2A_1^2(A_3^2 - 1) - 2A_2^2A_3^2}{F_m f_0 r T A_3^2 \omega(A_3^2 - 1)^{3/2}} \tanh^{-1} \left( \frac{A_3 - 1}{\sqrt{A_3^2 - 1}} \tan \left( \frac{\omega t}{2} \right) \right) \right) \]  \hspace{1cm} (3.34)

Evaluating Equation 3.34 for \( r = 1 \) (full cycle), the average squared activation tentatively becomes:

\[ a_{avg}^2 = - \frac{A_1^2}{F_m f_0 \omega A_3^2} = - \frac{k_s^2(\omega^2 m - c)^2}{F_m f_0 \omega^2(g'(0))^2(\omega^2 m - k_s - c)^2} \]  \hspace{1cm} (3.35)

However, it can easily be shown that this solution is incorrect, because it states that the average squared activation is always negative (assuming a nonzero denominator), except for its maximum value, 0, at \( c = \omega^2 m \). The integral of a squared (real) function cannot be negative. The sign of this erroneous solution can be verified by setting the derivative (with respect to \( c \)) of this value equal to zero to test if extrema exist:

\[ \frac{d}{dc} a_{avg}^2 = 0 = \frac{-2k_s^3(\omega^2 m - c)}{F_m f_0 \omega^2(g'(0))^2(\omega^2 m - k_s - c)^3} \]  \hspace{1cm} (3.36)

This yields a solution at \( c = \omega^2 m \), as expected. To test whether this is a
maximum or minimum, let us find the sign of the second derivative at \( c = \omega^2 m \):

\[
\frac{d^2}{dc^2} a^2_{avg} = \frac{-2k_3^3(k_s + 2(\omega^2 m - c))}{F_m f_0 \omega^2 (g'(0))^2(\omega^2 m - k_s - c)^4}
\]

(3.37)

\[
\frac{d^2}{dc^2} a^2_{avg}(c = \omega^2 m) = \frac{-2}{F_m f_0 \omega^2 (g'(0))^2}
\]

(3.38)

We know that the value given in Equation 3.38 is negative. Therefore, a \textit{maximum} exists at \( c = \omega^2 m \), according to this solution.

It is not difficult to pinpoint the source of this apparent error. There are three cases that arise when evaluating the average squared muscle activation. The first is \( A_3 = \pm 1 \), which is unrealistic and can be discarded, as the denominators of some of the terms in 3.34 become zero. In the second case, \( \|A_3\| < 1 \), the denominator of all terms in Equation 3.34 will never be zero. In the third case, \( \|A_3\| > 1 \), the denominators will equal zero twice within one period. Let the case of \( \|A_3\| < 1 \) be analyzed first.

The first issue is the tangent function. At \( \omega t = \pi \), the term \( \tan(\omega t/2) \) becomes undefined. Simulation will reveal that the analytical solution diverges from a numerical solution only instantaneously at \( \omega t = \pi \). Therefore, assuming \( t_0 = 0 \) and \( \|A_3\| < 1 \), it can be said that, for a full cycle,

\[
a^2_{avg} = \frac{1}{T} \lim_{\epsilon \to 0} \left( \int_0^{T/2-\epsilon} a^2 dt + \int_{T/2+\epsilon}^T a^2 dt \right)
\]

(3.39)

If one is attentive to detail, the condition \( \|A_3\| < 1 \) will also cause some components of \( a^2_{avg} \) to be imaginary. However, using the identity \( \tanh^{-1}(z) = \frac{1}{i} \tan^{-1}(iz) \), these imaginary components cancel out (not shown here).

Equation 3.39, fully expanded, is extremely long and relatively complex, making evaluation extremely difficult and tedious. Similarly, the case of \( \|A_3\| > 1 \) is quite complex. There will be discontinuities (in addition to
the discontinuity from the tangent component) when $A_3 \cos(\omega t) = -1$, which translates to $t = t_{d1} = \frac{1}{\omega} \cos^{-1}(\frac{-1}{A_3})$ and $t = t_{d2} = T - t_{d1}$. Therefore, for $\|A_3\| > 1$,

$$a_{avg}^2 = \frac{1}{T} \lim_{\epsilon \to 0} \left( \int_0^{t_{d1}-\epsilon} a^2 dt + \int_{t_{d1}+\epsilon}^{T/2-\epsilon} a^2 dt + \int_{T/2+\epsilon}^{t_{d2}-\epsilon} a^2 dt + \int_{t_{d2}+\epsilon}^{T} a^2 dt \right)$$

(3.40)

which, fully expanded, is difficult and tedious to evaluate.

Therefore, because there are other muscle performance measures that could be used (such as average power) that will yield clear and simple optima, average squared muscle activation will be disregarded for the rest of this study.

**Average CE Power - One Cycle**

For a full period, $r = 1$. Because of the assumed periodicity of $F$, multiple terms drop out of the average power calculation in Equation 3.25, leaving only the integral term behind:

$$P_{avg} = \frac{\omega A}{T} \int_{t_0}^{t_0+T} \cos(\omega t) F dt$$

(3.41)

Again, a simple spring load is applied such that

$$F = c(\beta - x_1(t))$$

(3.42)

where $c$ is the spring constant of the load and $\beta > x_0 + A$ is the equilibrium position of the spring. The average power equation becomes

$$P_{avg} = \frac{\omega c A}{T} \int_{t_0}^{t_0+T} \cos(\omega t)(\beta - x_0 - A \sin(\omega t)) dt$$

(3.43)

which, upon integration, yields

$$P_{avg} = \frac{c A}{T} \left( (\beta - x_0) \sin(\omega t) + \frac{A}{4} \cos(2\omega t) \right) \bigg|_{t_0}^{t_0+T}$$

(3.44)
which evaluates to:

\[
P_{\text{avg}} = 0 \quad (3.45)
\]

Because this solution is a constant, it is concluded that it is not possible to vary load stiffness \( c \) to optimize the average power.

This result motivates partial cycle analysis. Because the CE power is defined as \( P = F_{CE} \dot{L}_c \), it can be seen that \( P \) will assume the form

\[
P = A_2 \sin(\omega t) \cos(\omega t) + A_1 \cos(\omega t) \quad (3.46)
\]

where all \( A_n \) are constants. From here, applying a double-angle formula reveals that the function has a double-frequency component:

\[
P = \frac{A_2}{2} \sin(2\omega t) + A_1 \cos(\omega t) \quad (3.47)
\]

Such a form produces components such as those seen in Figure 8, where, for example, \( A_1 = 1 \) and \( A_2 = 2 \) and \( \omega = 1 \).

![Frequency Components of P](image)

**Figure 8:** Double-frequency sinusoid example

It is known (and can be seen in Figure 8) that integrating any sinusoid over an entire period will yield zero. Also, integrating a cosine function over a half cycle, whether the region is \( 0 \leq \omega t \leq \pi \) or \( \pi \leq \omega t \leq 2\pi \), will yield zero, and integrating a double-frequency sinusoid over that same range will
also yield zero. Therefore, to extract a non-zero integration period, a quarter cycle \( n\pi/2 \leq \omega t \leq (n + 1)\pi/2 \) can be chosen.

**Average CE Power - Quarter Cycle**

For a quarter cycle, \( r = 1/4 \). The particular portion of the cycle analyzed here is the portion of concentric motion from the point of maximum extension \( (\omega t_0 = \pi/2) \) through the acceleration phase to the point of maximum speed \( (\omega t_f = \pi) \). If a spring-like load is again assumed where \( F = c(\beta - x_1(t)) \), then Equation 3.25 can be written as:

\[
P_{\text{avg}} = \frac{-2}{T} \left( \frac{1}{k_s}(d_0 s_1 + c(\beta - x_0 - A s_1))^2 - d_0 A s_1^2 \right) \bigg|^{t_0 + T/4}_t + \frac{4\omega c A}{T} \int_{t_0}^{t_0 + T/4} c_1(\beta - x_0 - A s_1) dt
\]

\[ (3.48) \]

where \( c_n \) and \( s_n \) again refer to \( \cos(n\omega t) \) and \( \sin(n\omega t) \), respectively. After several steps of simplification, this equation becomes

\[
P_{\text{avg}} = \frac{2(Ak_s - a_1)}{Tk_s} \left( a_1 s_1^2 + 2a_2 s_1 \right) \bigg|^{t_0 + T/4}_t
\]

\[ (3.49) \]

where \( a_1 = d_0 - cA \) and \( a_2 = c(\beta - x_0) \). Recall that the interval being analyzed begins at \( \omega t_0 = \pi/2 \). The average power then evaluates to

\[
P_{\text{avg}} = \frac{2(a_1 - Ak_s)}{Tk_s} (a_1 + 2a_2)
\]

\[ (3.50) \]

which can be expanded as follows:

\[
P_{\text{avg}} = \frac{-2A\omega}{\pi k_s} (c + k_s - \omega^2 m) \left( \left( \beta - x_0 - \frac{A}{2} \right) c + \frac{A\omega^2 m}{2} \right)
\]

\[ (3.51) \]

Notice the leading coefficient, if Equation 3.51 were to be expanded as a polynomial in terms of \( c \):
\[
\frac{\omega A}{\pi k_s} (A - 2(\beta - x_0))
\] (3.52)

In order for the average power to have a maximum with respect to \(c\), this term must be negative. This is not difficult to prove. Because \(\beta > x_0 + A\) must hold to keep the restraining spring applying tensile force, we can say that \(\beta - x_0 = A + \delta x\) where \(\delta x\) is some positive number. This allows the leading coefficient to be rewritten as
\[
-\frac{\omega A}{\pi k_s} (A + 2\delta x)
\] (3.53)

which is always negative for physically realizable systems.

For sake of compactness, let the following constants be defined:

\[
\begin{align*}
b_1 &= k_s - \omega^2 m \\
b_2 &= (\beta - x_0) - \frac{A}{2} = A + \delta x > 0 \\
b_3 &= \frac{d_0}{2} > 0 \\
b_4 &= \frac{2A \omega}{\pi k_s} > 0
\end{align*}
\] (3.54)

This allows the average power equation to be written as
\[
P_{\text{avg}} = -b_4 (c + b_1)(b_2c + b_3)
\] (3.55)

or, in polynomial form,
\[
P_{\text{avg}} = -b_4 (b_2c^2 + (b_1b_2 + b_3)c + b_1b_3)
\] (3.56)

which has a global maximum at:
\[
c^* = \frac{-b_1b_2 - b_3}{2b_2}
\] (3.57)

In order for this solution to be physically realizable, \(c^*\) must be positive. A negative stiffness would likely cause the system to become unstable. Because
the denominator is already positive, we must prove that the numerator is (or at least can be) positive. Therefore,

\[ b_1 < -\frac{b_3}{b_2} \]  

(3.58)

must be true. This statement can be rewritten as follows, if \( \delta_b \) is some positive constant:

\[ b_1 = -\frac{b_3}{b_2} - \delta_b \]  

(3.59)

Now, evaluating \( P_{avg} \) at \( c^* \) yields

\[ P_{avg} = -b_4 \left( -\frac{(b_1 b_2 + b_3)^2}{4b_2} + b_1 b_3 \right) \]  

(3.60)

And substituting in the value of \( b_1 \) from Equation 3.59 and simplifying, it can be shown that

\[ P_{avg} = b_4 \left( \frac{\delta_b^2}{4} b_2 + b_3 \left( b_2 + \delta_b \right) b_3 \right) \]  

(3.61)

which is always positive. Therefore, a positive global maximum will be produced at a positive load spring stiffness as long as

\[ k_s - \omega^2 m < \frac{-A\omega^2 m}{2\beta - 2x_0 - A} \]  

(3.62)

holds.

**Average Squared CE Power - One Cycle**

Again assume the load is \( F = c(\beta - x_1(t)) \), so \( \dot{F} = -cx_2(t) \):

\[ P_{avg}^2 = \frac{1}{T} \int_{t_0}^{t_0+T} \left( a_1 s_1 + a_2 \right)^2 \left( \frac{1}{k_s^2} \right) (\omega(a_1 - Ak_s)c_1)^2 dt \]  

(3.63)

using the previous definitions of \( a_1 \) and \( a_2 \). This can be rewritten as
\[ P_{\text{avg}}^2 = \frac{a_3^2}{rT} \int_{t_0}^{t_0+rT} \left( \frac{a_1^2}{8} + \frac{a_2^2}{2} + 2a_1a_2s_1c_1^2 + \frac{a_2^2}{2}c_2 - \frac{a_1^2}{8}c_4 \right) dt \]  

(3.64)

where \( a_3 = (a_1 - Ak_s)\omega/k_s \). This integral can be evaluated as:

\[ P_{\text{avg}}^2 = \frac{a_3^2}{rT} \left(\left( \frac{a_1^2}{8} + \frac{a_2^2}{2} \right) t - \frac{2a_1a_2}{3\omega}c_1^3 + \frac{a_2^2}{4\omega}s_2 - \frac{a_1^2}{32\omega}s_4 \right) |_{t_0}^{t_0+rT} \]  

(3.65)

From here, letting \( r = 1 \) for a full cycle, several terms drop out, and the equation becomes:

\[ P_{\text{avg}}^2 = \frac{a_3^2}{2} \left( \frac{a_1^2}{4} + a_2^2 \right) \]  

(3.66)

The constants can now be expanded and reordered to yield:

\[ P_{\text{avg}}^2 = \frac{1}{2} \left( \frac{A^2\omega}{2k_s} \right)^2 (c - (\omega^2m - k_s))^2 \left( \left( 1 + \frac{4}{A^2}(\beta - x_0)^2 \right) c^2 - 2\omega^2mc + \omega^4m^2 \right) \]  

(3.67)

From here, it is possible to analyze extrema with respect to the load parameter \( c \). It is assumed that \( \beta \) is held constant. Letting

\[ d_1 = \frac{4}{A^2}(\beta - x_0)^2 > 0 \]

\[ d_2 = \omega^2m - k_s \]

\[ d_3 = \omega^2m > 0 \]

\[ d_4 = \frac{4}{A^2}(\beta - x_0)^2 + 1 > 0 \]

(3.68)

we can see that there are four roots of the average squared power equation, if treated as a polynomial of \( c \). The first root is repeated, and the third and fourth have imaginary parts:

\[ c_{\text{roots}} = \left\{ \begin{array}{c} d_2, \\
            d_2, \\
            (d_3/d_4)(1 \pm i\sqrt{d_1}) \end{array} \right\} \]  

(3.69)
Note that, if \( P^2_{avg}(c) \) were fully expanded as a 4th degree polynomial with respect to \( c \), the leading coefficient would be

\[
\frac{1}{2} \left( \frac{A^2 \omega}{2k_s} \right)^2 \left( 1 + \frac{4}{A^2} (\beta - x_0)^2 \right)
\] (3.70)

which is always positive. Based on this information, the polynomial curve should have one global minimum at \( \omega^2 m - k_s \). This must be the global minimum as it is the only real zero crossing, and the leading coefficient of the quartic polynomial is positive. Depending on the value of the complex roots, there will be either no local maxima or one local maximum accompanied by a local minimum. To determine if a local maximum is possible, let us take the derivative of Equation 3.67 to find the location of each extremum.

\[
\frac{d}{dc} P^2_{avg} = 0 = \left( \frac{A^2 \omega}{2k_s} \right)^2 (c - d_2) \left( d_4 c^2 - (d_2 d_4 + 3d_3)c + d_3^2 + d_2 d_3 \right)
\] (3.71)

which has the following roots, beyond the obvious root of \( c = d_2 \):

\[
c_{\text{extrema}} = \frac{3d_3}{4d_4} + \frac{d_2}{4} \pm \frac{1}{4d_4} \sqrt{d_5}
\] (3.72)

where

\[
d_5 = (d_1 - 8)d_1 d_3^2 - (2k_s d_4 d_1) d_3 + k_s^2 d_4^2
\] (3.73)

is considered a polynomial with respect to \( d_3 \). The root \( c = d_2 \) corresponds to the first repeated root in Equation 3.69. This is the global minimum. If the next two extrema locations are imaginary, there is no local maximum. If the expression underneath the radical \( d_5 \) in Equation 3.72 is greater than or equal to zero, however, these roots will be real. If that expression is greater than zero, there will be a local maximum. To determine the sign of this expression, let us discover its roots with respect to \( d_3 \):
Because both roots are real and non-repeating, there will be conditions that cause \( d_5 \) to be negative or positive. Notice that the leading coefficient in \( d_5(d_3) \) is \((d_1 - 8)d_1\). Because \( d_1 = \frac{4}{\pi^2} (\beta - x_0)^2 \) is positive for all physically realizable systems, the sign of \((d_1 - 8)\) will be the deciding factor for the placement and existence of extrema. This produces two possible cases (excluding \( d_1 = 8 \)):

- **Case 1** (\( d_1 < 8 \)): \( d_5(d_3) \) is a downward facing parabola with one negative small-magnitude root and one positive larger-magnitude root. \( d_5 \) is positive for \( 0 < d_3 < k_s d_4/(\sqrt{d_1}(\sqrt{d_1} + \sqrt{8})) \).

- **Case 2** (\( d_1 > 8 \)): \( d_5(d_3) \) is an upward facing parabola with two positive roots. \( d_5 \) is positive for \( 0 < d_3 < k_s d_4/(\sqrt{d_1}(\sqrt{d_1} + \sqrt{8})) \) and \( d_3 > k_s d_4/(\sqrt{d_1}(\sqrt{d_1} - \sqrt{8})) \).

In either case, there may be conditions that produce a positive \( d_5 \). As long as these conditions are met, an extremum seeking scheme should be able to seek the local maximum - or infinity, if the initial estimate of \( c \) is significantly greater than the largest \( c \) value that produces a local extremum.

### 3.3 Controlled Single-Muscle System

Numerical validation of extrema requires that the muscle model be controlled for trajectory tracking. A sliding-mode controller was selected for this purpose.
3.3.1 Sliding Mode Control of a Single Muscle

Sliding mode control (SMC) was selected for control of the single-muscle system for its ability to obtain near-perfect tracking. This requirement is only needed for the single-muscle system to verify the existence of extrema for a single muscle. Once the multi-muscle linkage is developed, a different controller will be selected that is less robust. A less robust controller, such as an impedance controller, will sacrifice tracking accuracy for reduced control discontinuities. Such behavior should be more representative of a physical system. However, for preliminary extrema validation, the robust SMC will ensure tracking of the desired trajectory, effectively guaranteeing adherence to the analytically derived cases.

In SMC, a quadratic Lyapunov function is selected with respect to a sliding variable $s$:

$$V = \frac{1}{2} s^2 > 0$$  \hspace{1cm} (3.75)

$$\dot{V} = ss < 0$$  \hspace{1cm} (3.76)

The sliding variable is defined as

$$s = \left( \frac{d}{dt} + \lambda \right)^{n-1} e_1$$  \hspace{1cm} (3.77)

where $n$ is the number of states in the system, $e_1$ is the tracking error $x_1 - x_{1}^{des}$, and $\lambda$ is some error weight constant. In the case of the single-muscle system, $n = 3$, and the sliding variable becomes:

$$s = \ddot{e}_1 + 2\lambda \dot{e}_1 + \lambda^2 e_1$$

$$= e_3 + 2\lambda e_2 + \lambda^2 e_1$$  \hspace{1cm} (3.78)
where \( e_n = \frac{d^{n-1}}{dt^{n-1}} e_1 \). The Lyapunov derivative becomes negative definite by enforcing
\[
\dot{s} = -\eta \text{sgn}(s) \tag{3.79}
\]
where \( \eta \) is some positive constant gain and \( \text{sgn}(\cdot) \) is the sign function. In order to derive a control law, we must first take the derivative of \( s \) and manipulate the resulting equation. Using the definitions of \( s \) and \( \dot{s} \), we can state that:
\[
-\eta \text{sgn}(s) - 2\lambda e_3 - \lambda^2 e_2 + \ddot{x}_{\text{des}} = \ddot{x}_2 \tag{3.80}
\]
Because we already have the state equation for \( \dot{x}_2 \) in Equation 3.1, we can differentiate to find \( \ddot{x}_2 \) and substitute into Equation 3.80:
\[
-\eta \text{sgn}(s) - 2\lambda e_3 - \lambda^2 e_2 + \ddot{x}_{\text{des}} = \frac{1}{m} \left( -\phi_s'(L_s) \dot{L}_s + \dot{F} \right) \tag{3.81}
\]
By applying the third state equation from Equation 3.1 that states \( \dot{L}_s = x_2 + u \) and rearranging the equation, we can find our control law:
\[
u = \frac{-m}{\phi_s'(L_s)} \left( -\eta \text{sgn}(s) - 2\lambda e_3 - \lambda^2 e_2 + \ddot{x}_{\text{des}} \right) - x_2 + \frac{1}{\phi_s'(L_s)} \dot{F} \tag{3.82}
\]
This controller should effectively track a trajectory. However, it should be noted that this control law is intended primarily for simulation purposes when the load profile is known and differentiable. In practice, or in more complex simulations when the derivative of the load profile is difficult to determine, this exact control law may not be suitable. Therefore, this controller is only being used for the single-muscle system simulations, as a proof of concept for the existence of extrema.
3.3.2 Trajectory Tracking Simulation Example

Tracking was achieved by the sliding mode controller. For example, take the case of a simple spring-like load on a single-muscle system with the following parameter values:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Variable</th>
<th>Value (dimensionless)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oscillation Amplitude</td>
<td>$A$</td>
<td>0.004</td>
</tr>
<tr>
<td>Oscillation Frequency</td>
<td>$\omega$</td>
<td>$\pi$</td>
</tr>
<tr>
<td>Oscillation Mean</td>
<td>$x_0$</td>
<td>2.9952</td>
</tr>
<tr>
<td>Load Stiffness</td>
<td>$c$</td>
<td>2</td>
</tr>
<tr>
<td>Load Equilibrium</td>
<td>$\beta$</td>
<td>3.30</td>
</tr>
<tr>
<td>Tendon Slack Length</td>
<td>$s_n$</td>
<td>2</td>
</tr>
<tr>
<td>Tendon Stiffness</td>
<td>$k_s$</td>
<td>19.23</td>
</tr>
<tr>
<td>Mass</td>
<td>$m$</td>
<td>2.5</td>
</tr>
<tr>
<td>SMC Error Weight</td>
<td>$\lambda$</td>
<td>2</td>
</tr>
<tr>
<td>SMC Gain</td>
<td>$\eta$</td>
<td>90</td>
</tr>
</tbody>
</table>

Table I: Muscle example parameters

Under these conditions, the trajectory shown in Figure 9 is produced.
The tracking error converged to zero very quickly. Also, in Figure 10, it can be seen that the assumed conditions were met: $L_c$ remained in the slack region (below the slack limit, which is $L_c = 1$ in this case) and $L_s$ remained in the linear region (above $L_s = 2.04$ in this case). Also note that the activation level remained between 0 and 1, which indicates that the muscle was operating within a (theoretically) realizable range.
Therefore, our assumptions are valid, and the selected controller appears to be functional for the current system.

### 3.4 Muscle Extrema Validation

Now that a muscle model has been developed and has been shown to accurately track a desired trajectory within the proposed constraints, we can verify the theoretical extrema by implementing extremum seeking control to adapt the load parameters. Implementing ESC at this point and successfully discovering the analytical extrema (or a value in some neighborhood thereof) will both validate the analytical solution and showcase the effectiveness of ESC for this application. However, there are some preliminary concerns that must be met relating to the static input-output mapping assumption of ESC. These concerns include the repression of initial transients and the method of average performance calculation (discrete or sliding average) over a single period.

#### 3.4.1 Ensuring Assumptions are Met

The extremum seeking control scheme selected for this study requires that the input-output plant mapping is static. Therefore, transients in the signals must be minimized, and the method of parameter perturbation must be carefully selected.

**Rejecting Initial Transients**

In order to meet the controller’s assumption that the input-output mapping is static, we must reject disturbance and signal transients. Because a
human user will never be able to always start with the same ideal initial conditions, we must somehow reject the initial transients produced by the user while approaching the desired trajectory. To accomplish this, the controller was designed to be activated sometime after the beginning of the workout or simulation. This causes the load parameters to remain constant for some arbitrary amount of time before the controller begins to optimize them.

**Batch Processing vs. Sliding Average Calculation**

To find some average value - regardless of whether it be squared power, squared muscle activations, or some other measure - over each period to send as input to the controller, two options were considered: batch processing and sliding averages. Both approaches have merit and will be compared experimentally in later chapters.

In batch processing mode, the controller would operate in discrete time, with the sample time being one period of motion. Therefore, after one cycle of motion is completed, the average output value is measured and passed to the controller, and the controller then varies the input parameters accordingly. The advantage of this approach is parameter consistency. For each cycle, the load parameters remain constant. This, at face value, fulfills the static mapping condition. However, in practice and simulation, this will produce sharp transients at the beginning of each cycle. This will not only color the output measurements, but it will also cause the static mapping condition to be briefly violated. It is also possible that these sharp impacts could have detrimental effects on the user; however, such data is beyond the scope of
this study. To reduce the effect of transients, the discretized controller output (the system input parameter values) could be fed through a low-pass filter or otherwise interpolated to remove sharp discontinuities. Of course, the static mapping condition cannot be reasonably met with this modification.

In sliding average mode, the controller is varying parameters slowly but continuously and the average output is calculated over the most recent period. If the parameters are varied extremely slowly, then the effects of the approximately correct parameters are being evaluated over the most recent period. This will, theoretically, cause negligible error in the evaluation of any particular set of parameter values. Simulations confirm this hypothesis. This approach also reduces disturbance because of the smooth transitions during parameter variation. One drawback of this approach is the slow rate of target parameter estimation convergence. Because controller gains must be very low to ensure that the rate of parameter variation is very slow, the controller will spend a significant amount of time seeking the optimal value. In practice, this would mean that the user may not be able to reach their optimal working conditions before muscle fatigue sets in or the exercise session finishes. However, convergence in finite time to optimal load parameters is not necessarily required. As long as the controller is able to approach the ideal set of parameters and the initial parameter settings are not extremely far away from the ideal parameters, the output may be desirable. Also, the final conditions of the controller can be used for the initial load conditions for the next exercise session of the same user, or the final conditions can be extrapolated to some degree to find a new set of initial conditions that may be even closer to the
optimal set of parameters.

3.4.2 Simulation Results

All trials were conducted using a spring-like load, where the variable parameter $\theta$ was the spring stiffness $c$. In order to operate within the assumptions presented above, it is helpful to choose a small desired trajectory amplitude. Therefore, $A = 0.004$ was selected as the oscillation amplitude.

For the purposes of these simulations, all parameters and variables have been normalized. Therefore, all values (time, power, length, stiffness, etc.) are dimensionless. At face value, this choice may seem problematic - however, because the purpose of this portion of the study is to verify the existence of feasible extrema and not exactly derive locations of extrema, normalization of parameters is certainly allowable. Even though we are analytically deriving the location of each theoretical extremum in the following simulations, this is only to prove the validity of the ESC approach, and the motivation for analytically deriving the extremum is merely to verify its existence.

For brevity’s sake, the batch processing method and the sliding average will not both be applied to each optimization simulation in this section. However, the differences in output between the two methods will be discussed briefly in Chapter V, when both methods will be applied to optimization of load stiffness on a multi-muscle linkage.
Average CE Power - Quarter Cycle - Batch Processing

It was attempted to maximize the average $CE$ power over one quarter cycle using the batch processing method. Table II tabulates the system parameters used in this simulation.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Variable</th>
<th>Value (dimensionless)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oscillation Amplitude</td>
<td>$A$</td>
<td>0.004</td>
</tr>
<tr>
<td>Oscillation Frequency</td>
<td>$\omega$</td>
<td>$\pi$</td>
</tr>
<tr>
<td>Oscillation Mean</td>
<td>$x_0$</td>
<td>2.9952</td>
</tr>
<tr>
<td>Load Equilibrium</td>
<td>$\beta$</td>
<td>3.30</td>
</tr>
<tr>
<td>Tendon Slack Length</td>
<td>$s_n$</td>
<td>2</td>
</tr>
<tr>
<td>Tendon Stiffness</td>
<td>$k_s$</td>
<td>19.23</td>
</tr>
<tr>
<td>Mass</td>
<td>$m$</td>
<td>2.5</td>
</tr>
<tr>
<td>SMC Error Weight</td>
<td>$\lambda$</td>
<td>2</td>
</tr>
<tr>
<td>SMC Gain</td>
<td>$\eta$</td>
<td>90</td>
</tr>
<tr>
<td>ESC Estimation Gain</td>
<td>$k$</td>
<td>11000</td>
</tr>
<tr>
<td>ESC Perturbation Amplitude</td>
<td>$a$</td>
<td>0.2</td>
</tr>
<tr>
<td>ESC Perturbation Frequency</td>
<td>$\omega_{ES}$</td>
<td>$\pi/16$</td>
</tr>
<tr>
<td>ESC High-Pass Frequency</td>
<td>$\omega_h$</td>
<td>$\pi/16$</td>
</tr>
<tr>
<td>ESC Low-Pass Frequency</td>
<td>$\omega_l$</td>
<td>$\pi/80$</td>
</tr>
</tbody>
</table>

Table II: Muscle example parameters - power optimization

An optimum was reached by the extremum seeking controller after approximately 150 cycles of oscillation (muscle oscillation, not ESC perturbations).
First, it can be seen that the sliding mode controller was able to enforce tracking, even during load spring stiffness modulation. The entire simulation tracking performance is shown in Figure 11a, but because the full simulation data is rather meaningless visually, a smaller interval is expanded for closer examination in Figure 11b.

![Figure 11: ESC single muscle $P_{CE,avg}$ optimization - tracking](image)

(a) Full simulation  
(b) Selected region

The sliding mode controller clearly regulates the tracking error to zero.

Figure 12 demonstrates that the controls and element lengths remain approximately within acceptable limits. The $CE$ length $L_c$ never exceeds the slack length, the tendon length $L_s$ remains in the linear region. Both full and partial simulation results are shown so the general trend as well as instantaneous behaviors can be analyzed.
Figure 12: ESC single muscle $P_{CE,avg}$ optimization - muscle lengths and rates

In Figure 12b, it can be seen that there are sharp discontinuities that occur periodically in the control input $u$. This is a result of the combination of two factors: sudden load perturbations and the robustness of SMC. Because the extremum seeking controller operates in discrete time (sample time equal to the period of muscle motion), sudden changes in the load parameters cause
the simulated muscle to experience a sudden change in applied force. This change is accommodated for almost immediately by the robust sliding mode controller, and little to no visible tracking error is introduced, at the cost of extremely high control levels. In reality, a muscle would not be able to produce such robust control inputs. For the purposes of this simulation, however, these “hiccups” are acceptable, as they do not (significantly) alter the behavior of the system or existence of extrema.

The activation levels in Figure 12 exceed the maximum physically allowable level of 1. This occurs because the activation levels assume a normalized force-length relationship in the tendon and also assume that the maximum normalized force is not exceeded by the tendon. Whenever the tendon exceeds the maximum normalized force, the activations levels could potentially exceed 1. This may not necessarily have physical meaning, however, as this range may still be physically realizable in a non-normalized muscle.

The existence/absence of an extremum was predicted analytically at roughly $c = 2.6$. This optimal load stiffness was approximately located and can be seen in Figure 13, which produced power levels seen in Figure 14.
Figure 13: ESC single muscle $P_{CE,avg}$ optimization - load stiffness

Figure 14: ESC single muscle $P_{CE,avg}$ optimization - $P_{CE,avg}$

**Average Squared CE Power - One Cycle - Sliding Average**

It was attempted to maximize the average squared $CE$ power over one quarter cycle using the sliding average method. Table III tabulates the system parameters used in this simulation.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Variable</th>
<th>Value (dimensionless)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oscillation Amplitude</td>
<td>$A$</td>
<td>0.01</td>
</tr>
<tr>
<td>Oscillation Frequency</td>
<td>$\omega$</td>
<td>$8\pi/5$</td>
</tr>
<tr>
<td>Oscillation Mean</td>
<td>$x_0$</td>
<td>2.888</td>
</tr>
<tr>
<td>Load Equilibrium</td>
<td>$\beta$</td>
<td>28.04</td>
</tr>
<tr>
<td>Tendon Slack Length</td>
<td>$s_n$</td>
<td>2</td>
</tr>
<tr>
<td>Tendon Stiffness</td>
<td>$k_s$</td>
<td>19.23</td>
</tr>
<tr>
<td>Mass</td>
<td>$m$</td>
<td>2</td>
</tr>
<tr>
<td>SMC Error Weight</td>
<td>$\lambda$</td>
<td>10</td>
</tr>
<tr>
<td>SMC Gain</td>
<td>$\eta$</td>
<td>10</td>
</tr>
<tr>
<td>ESC Estimation Gain</td>
<td>$k$</td>
<td>400</td>
</tr>
<tr>
<td>ESC Perturbation Amplitude</td>
<td>$a$</td>
<td>0.2</td>
</tr>
<tr>
<td>ESC Perturbation Frequency</td>
<td>$\omega_{ES}$</td>
<td>$\pi/10$</td>
</tr>
<tr>
<td>ESC High-Pass Frequency</td>
<td>$\omega_h$</td>
<td>$\pi/10$</td>
</tr>
<tr>
<td>ESC Low-Pass Frequency</td>
<td>$\omega_l$</td>
<td>$\pi/50$</td>
</tr>
</tbody>
</table>

Table III: Muscle example parameters - squared power optimization

An optimum was reached by the extremum seeking controller after approximately 230 cycles of oscillation. First, it can be seen that the sliding mode controller was able to enforce tracking, even during load spring stiffness modulation. The entire simulation tracking performance is shown in Figure 15a, and a small interval of the simulation is expanded for closer examination in Figure 15b.
The controls and element lengths also remain within acceptable limits. However, it is curious that the muscle activation levels are far from ideal (Figure 16a). The activation levels should not exceed 1. However, as stated previously, this should not be taken necessarily as an indicator of infeasibility. Because the model is normalized, activation levels above 1 simply mean that the muscle exceeded the maximum “normalized” length, which may or may not be possible once the model is de-normalized. That being said, it may still be that the high activation levels are an indicator of infeasibility.

The $CE$ length $L_c$ never exceeds the slack length, and the tendon length $L_s$ remains in the linear region. Figure 16a illustrates this, and a small interval of the simulation is expanded for closer examination in Figure 16b.
The existence of an extremum was predicted analytically at roughly $c = 13.7$. Though not achieved exactly, the optimal load stiffness was approximately located and can be seen in Figure 17, which produced power levels seen in Figure 18.

Figure 16: ESC single muscle $P_{CE,avg}^2$ optimization - muscle lengths and rates
In this chapter, the basic Hill muscle model was analyzed and set up for simulation. Properties of the muscle were discussed, and analytical extrema were successfully derived for multiple output optimization cases. These results validate the hypothesis that there are at least some parameter combination selections (both input $\theta$ and output performance $y$) that will cause an individual

3.5 Discussion

In this chapter, the basic Hill muscle model was analyzed and set up for simulation. Properties of the muscle were discussed, and analytical extrema were successfully derived for multiple output optimization cases. These results validate the hypothesis that there are at least some parameter combination selections (both input $\theta$ and output performance $y$) that will cause an individual
muscle to perform optimally.

These results are very valuable for future use. The methods and formulae described can be easily followed for analysis of parameter combinations not discussed in this work. More importantly, the very possibility of extrema existence motivates extremum seeking for more complex multi-muscle systems where analytical solutions are difficult or impossible to find.
CHAPTER IV

MULTI-MUSCLE LINKAGE MODEL

It would be very helpful for simulation to have a muscle model that models an entire section of a human. This is much more helpful than a single muscle model because human muscles do not operate in a vacuum - they coordinate with each other and act upon human “links” instead of directly acting upon external loads. In addition, this will help move toward the goal of a multi-muscle power optimization strategy when implementing the extremum seeking controller through the adaptive exercise machine.

What must be developed is a generalized model that allows for simulation of an arbitrary number of links or joints with an arbitrary number of muscles. Therefore, all code must be scalable and abstracted from model-specific parameters. Any system-specific parameters (link length, muscle parameters, etc.) must not be hard-coded in the simulation. This will allow for the simulation to model any part (or the entirety) of a human, which will in turn make generalization of the exercise machine type (load parameters) much simpler.

In this chapter the development of a generalized, multi-muscle, robotics-motivated linkage model is discussed. The joint torque and muscle force
control techniques are outlined in Section 4.1, followed by a description of
robotics equation algorithmic modifications for numerical simulation in MAT-
LAB Simulink in Section 4.2. Section 4.3 presents results from simulation
of a sample model developed with the proposed framework. The chapter is
concluded with a discussion in Section 4.4.

4.1 Linkage Control Design

A robotic linkage (open kinematic chain) with external force applied fol-

\[ M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + J_F^T(q)F_e = \tau \]  

In Equation 4.1, \( M(q) \) refers to the mass matrix of the robotic linkage; \( C(q, \dot{q}) \)
is the Coriolis matrix; \( g(q) \) is the gravity vector; \( J_F^T(q) \) is the transpose of the
velocity Jacobian \( J_F(q) \) of the point at which external force is applied, which
for this study will be assumed to be the end effector; \( F_e \) is the external force
applied; \( \tau \) is a vector of joint torques; and \( q \) is a vector of joint positions.

Note that \( M(q), C(q, \dot{q}), g(q), \) and \( J_F(q) \) are dependent on the state vari-
able. However, for sake of compactness and readability, a small change of
notation will be made. \( M(q), C(q, \dot{q}), g(q), \) and \( J_F(q) \) will be simplified to \( M,\)
\( C, g, \) and \( J_F, \) respectively.

It is desired to control the joint angles \( q \) using the joint torques \( \tau \). To ac-

accomplish this, two different controllers will be analyzed - an inverse dynamics
controller and an impedance controller. Impedance control was selected be-
cause there is some evidence that humans employ something akin to impedance control during motion [51]. Inverse dynamics was also selected as an alternative control method, to analyze the effects of different control schemes on the system.

Direct control using joint torques is not possible with a muscle actuated linkage. This is due to the fact that the muscle inputs are not joint torques, but rather contraction rates of the contractile element in each muscle. The muscle inputs must therefore be calculated once the desired joint torques $\psi$ are calculated, adding one more level of complexity to the controller.

To accomplish this extra step of control, a backstepping controller has been designed to regulate the muscle outputs $u$ such that the actual joint torques $\tau$ approach the desired joint torques $\psi$.

The simulated linkage will adhere to the following pipeline per timestep in simulation. The instantaneous joint error $e$ is passed to the synthetic controller, which calculates the desired joint torques $\psi$ for convergence to zero error. The desired joint torques $\psi$ are then passed to the backstepping controller, which calculates the muscle inputs $u$. The muscle inputs $u$ are then used to calculate the equations of state.

It is important to note, as stated in the previous chapter, that the parallel element of the muscle is assumed to be operating in the slack region. That is, negligible force is produced by the parallel element. This assumption simplifies the linkage model and speeds up simulation time. It was shown in the previous chapter that this assumption is not unrealistic. Although future versions of the linkage design framework will be modified to include modeling of the parallel
element, such elaboration is not necessary for the preliminary proof of concept methods set forth in this work.

4.1.1 Synthetic Control: Inverse Dynamics

Inverse dynamics is one of the two proposed methods for calculating the joint torques required for the linkage to track a specified trajectory. Inverse dynamics is a non-robust, non-adaptive control method which calculates output joint torques by finding joint torques that will effectively cancel the dynamics of the system to follow a specified trajectory. In general, basic non-robust and non-adaptive inverse dynamics control is impractical, as it is impossible to know the parameters of a system exactly, and it is therefore impossible to accurately calculate joint torques that will exactly eliminate the dynamics of the system. However, in the case of this study, this controller is being used in a simulated human, and it is assumed that the simulated human has perfect knowledge of its own system properties. It can be assumed that any desired trajectory can be achieved exactly by the human. The human model is also abstracted from the exercise machine model, which needs no knowledge of the dynamics of the human. Therefore, a non-robust and non-adaptive method of control for human joint torque calculation is acceptable.

The general calculation of joint torque control $\psi_{id}$ using inverse dynamics, adapted from [50], is described by Equation 4.2.

$$\psi_{id} = Ma + C\dot{q} + g$$

(4.2)

Here, $a$ is the synthetic acceleration, calculated as $a = \ddot{q}^d - K_d\dot{\tilde{q}} - K_p\tilde{q}$, where $K_p$ and $K_d$ are diagonal matrices of positive gains and $\tilde{q} = q - q^d$ is the joint
position error if \( q^d \) is the desired joint angle vector. If no external forces are present, substituting \( \psi_{id} \) for the joint torques \( \tau \) in the general robotics equation (Equation 4.1) yields

\[
M \ddot{q} + C \dot{q} + g = Ma + C\dot{q} + g \tag{4.3}
\]

which, using the definition of \( a \), simplifies to Equation 4.4:

\[
M \ddot{q} = M(\ddot{q}_{\text{des}} - K_d \dot{\tilde{q}} - K_p \tilde{q}) \tag{4.4}
\]

Assuming the mass matrix \( M(q) \) is invertible, the joint torque error can be described by the differential equation

\[
\ddot{\tilde{q}} + K_d \dot{\tilde{q}} + K_p \tilde{q} = 0 \tag{4.5}
\]

which is stable (shown formally in section 4.1.4). The error dynamics can be rewritten in state-space form as

\[
\dot{e} = \begin{bmatrix} 0 & I \\ -K_p & -K_d \end{bmatrix} e \tag{4.6}
\]

where \( e = [\tilde{q}, \dot{\tilde{q}}]^T \) and \( I \) is the identity matrix.

Now \( \psi_{id} \) can be supplied to the backstepping controller as the synthetic (desired) control.

### 4.1.2 Synthetic Control: Impedance Control

Impedance control is generally used for implementation in manipulators that will be interacting with the environment and require some compliance such that a desired trajectory is less strictly enforced. This would allow a controller to “back off” on input application (force, torque, etc.) when obstacles are met, such as a rigid wall. With less robust trajectory tracking, the manipulator is less likely to experience damage or inflict damage on the
environment.

There are many different styles of impedance control, but the approach that will be analyzed here is inverse-dynamics motivated. In this approach, it is again assumed that the controller has perfect knowledge of the plant (and is therefore non-robust and non-adaptive). The controller will again effectively cancel out the true dynamics of the plant, but in the case of impedance control, a new mechanical “impedance” (having a similar mathematical meaning as the electrical sense of the word) will be enforced that reacts to the tracking error. These dynamics are generally modeled as follows:

\[ I_c \ddot{\tilde{q}} + B_c \dot{\tilde{q}} + K_c \tilde{q} = -J_T^T F_e \]  

(4.7)

where \( I_c, B_c, \) and \( K_c \) are the desired impedance parameters - decoupled/diagonal inertia, damping, and stiffness matrices (consisting of positive values on the diagonal). Note that, in the absence of external forces, this system is stable (shown formally in Section 4.1.4). The error dynamics can also be written in state-space form:

\[
\begin{bmatrix}
0 & 1 \\
-I_c^{-1} K_c & -I_c^{-1} B_c
\end{bmatrix} e + \begin{bmatrix} 0 \\ -I_c^{-1} \end{bmatrix} J_T^T(q) F_e
\]

(4.8)

Substituting \( \ddot{q} = \ddot{\tilde{q}} + \ddot{\bar{q}} \) into the general robotics equation and reordering terms yields the synthetic control \( \psi_{imp} \):

\[
\psi_{imp} = -MI_c^{-1}(B_c \dot{\tilde{q}} + K_c \tilde{q} + J_T^T F_e) + M\ddot{\bar{q}} + C\dot{\bar{q}} + g + J_T^T F_e
\]

(4.9)

Now \( \psi_{imp} \) can be supplied to the backstepping controller as the synthetic (desired) control.
4.1.3 Backstepping Controller

Backstepping control has been selected to enforce that the desired control demanded by the synthetic controller is met by the muscle outputs. First, an outline of backstepping control is given, and then the specific control scheme used for this study is put forth.

Backstepping Control

Backstepping control is used to synthetically suppress undesirable dynamics of a system by “forcing” a certain component or state - referred to as the synthetic control \( \xi \) - to converge to a certain desired trajectory \( \psi = \psi(x) \). The error between these two values is denoted as \( \omega = \xi - \psi \).

This technique “backsteps” through the integrator (recursively, for multi-state systems) by assuming that the system can be represented in the following form, taken from [30]:

\[
\begin{bmatrix}
\dot{\eta} \\
\dot{\xi}
\end{bmatrix} =
\begin{bmatrix}
f(\eta) + g(\eta)\xi \\
u
\end{bmatrix}
\] (4.10)

If the synthetic control is effectively regulated to the desired control, that is, \( \xi = \psi \), and if \( \psi(0) = 0 \), then for a positive-definite Lyapunov function \( V(\eta) \), the Lyapunov derivative \( \dot{V}(\eta) \) must be bounded by \( -W(\eta) \) to ensure stability, where \( W(\eta) \) is positive definite.

\[
\dot{V}(\eta) = \frac{\partial V}{\partial \eta} [f(\eta) + g(\eta)\psi(\eta)] \leq -W(\eta)
\] (4.11)

Note that the controller that stabilizes the \( \eta \) subsystem is irrelevant, as long as it is stabilizing.

From here, \( \xi \) must be regulated to \( \psi \), which is accomplished by first rewrit-
ing the $\eta$ subsystem using the definition of the synthetic control error $\omega$:

$$\dot{\eta} = f(\eta) + g(\eta)\omega + g(\eta)\psi$$ (4.12)

Now a new, augmented Lyapunov function $V_a(\eta, \omega) = V(\eta) + \frac{1}{2}\omega^2$ can be defined, whose derivative is:

$$\dot{V}_a(\eta, \omega) = \frac{\partial V}{\partial \eta} [f(\eta) + g(\eta)\omega + g(\eta)\psi] + \omega(u - \dot{\psi})$$

$$= \frac{\partial V}{\partial \eta} [f(\eta) + g(\eta)\psi] + \omega \left[ \frac{\partial V}{\partial \eta} g(\eta) + u - \dot{\psi} \right]$$ (4.13)

Choosing

$$\frac{\partial V}{\partial \eta} g(\eta) + u - \dot{\psi} = -\gamma \omega$$ (4.14)

allows the new augmented Lyapunov function derivative to be negative definite and asymptotically stable:

$$\dot{V}_a(\eta, \omega) \leq -W(\eta) - \gamma \omega^2$$ (4.15)

**Linkage Backstepping Controller**

We can now apply backstepping control to the linkage model being developed. The general robotic equation can be modified by declaring $\xi$ to be our synthetic input $\tau$ which should converge to the desired input $\psi$. We can also say that $\xi = \xi - \psi + \psi$. This creates the following modified robotic equation:

$$M\dddot{q} + C\dot{q} + g + J_F^TF_e = \xi - \psi + \psi$$ (4.16)

If we now define our synthetic control error as $\omega = \xi - \psi$, we obtain the following equation:

$$M\dddot{q} + C\dot{q} + g + J_F^TF_e = \omega + \psi$$ (4.17)
Using the synthetic control law derived via inverse dynamics, we can substitute in the desired joint control $\psi_{id}$ and manipulate the equation to yield

$$\omega = M(\ddot{q}_{des} + K_d \dot{q} + K_p q) + J_F^T F_e$$

which allows the error $e$ to be written in state-space form as follows:

$$\dot{e} = \begin{bmatrix} 0 & I \\ -K_p & -K_d \end{bmatrix} e + \begin{bmatrix} 0 \\ M^{-1} \end{bmatrix} \omega + \begin{bmatrix} 0 \\ -M^{-1} \end{bmatrix} J_F^T F_e$$

If impedance control is selected for our synthetic input, then substituting $\psi_{imp}$ for the synthetic control in Equation 4.17 yields

$$\omega = M(q) \ddot{q} + M(q) I_c^{-1} (B_c \dot{q} + K_c \dot{q} + J_F^T F)$$

We can now write the error $e$ in state-space form as

$$\dot{e} = \begin{bmatrix} 0 & 1 \\ -I_c^{-1} K_c & -I_c^{-1} B_c \end{bmatrix} e + \begin{bmatrix} 0 \\ M^{-1} \end{bmatrix} \omega + \begin{bmatrix} 0 \\ -I_c^{-1} \end{bmatrix} J_F^T(q) F_e$$

### 4.1.4 Composite Controller Stability

Before implementation or simulation, the stability of the cascaded synthetic control to backstepping control algorithm must be analyzed. However, one of the requirements of backstepping control is that the system be stable when the synthetic control error $\omega$ is zero (that is, the synthetic control $\xi$ is equal to the desired control $\psi$). Therefore, the stability of the inverse dynamics or impedance controller alone must first be proven. It is interesting to note that, regardless of whether the impedance controller or the inverse dynamics controller is selected for generation of the desired synthetic control, the proof of stability remains the same.

If the error states as derived in Equation 4.6 or 4.8 can be compactly
written as $\dot{e} = Ae$ in the absence of external forces, then a Lyapunov function $V$ is proposed using $P = P^T > 0$:

$$V = e^T P e$$  \hspace{1cm} (4.22)

This Lyapunov function has the derivative $\dot{V} = e^T (A^T P + PA)e = -e^T Q e$. If $P$ can be chosen such that $Q = Q^T > 0$, then the system is stable.

Now that the stability of the synthetic controller has been proven (in the presence of negligible external forces), the stability of the composite controller can be analyzed. If the error states as seen in Equation 4.19 or 4.21 can be written more compactly as $\dot{e} = Ae + B\omega$, then an augmented Lyapunov function $V^b$ is proposed using $D = D^T > 0$:

$$V^b = e^T P e + \omega^T D \omega$$ \hspace{1cm} (4.23)

Taking the derivative of $V^b$ with respect to time yields the following equation:

$$\dot{V}^b = -e^T Q e + 2\omega^T (B^T P e + D(\dot{\xi} - \dot{\psi}))$$ \hspace{1cm} (4.24)

One method to ensure stability is to enforce that $\dot{V}^b$ be negative definite by enforcing that

$$B^T P e + D(\dot{\xi} - \dot{\psi}) = -\Gamma \omega$$  \hspace{1cm} (4.25)

where $\Gamma = \Gamma^T > 0$. If this is enforced, the Lyapunov derivative becomes

$$\dot{V}^b = -e^T Q e - 2\omega^T \Gamma \omega$$ \hspace{1cm} (4.26)

which is negative definite. This will cause the system to be globally asymptotically stable.

Equation 4.25 produces the following control law for $\dot{\xi}$:

$$\dot{\xi} = D^{-1}(-\Gamma \omega - B^T P e) + \dot{\psi}$$ \hspace{1cm} (4.27)
This is not, however, immediately realizable. It must be remembered that \( \xi = \tau \), which is a vector of joint torques, and the actual control is muscle contraction speeds. We must therefore derive a relationship between the two to compute the necessary muscle inputs. This is not difficult. By applying the chain rule of differentiation, we obtain the following relationship between the synthetic control \( \xi_i \) at each joint and the muscle contraction rates \( u_i \):

\[
\dot{\xi}_i = \dot{\tau}_i = \left( \frac{\partial \tau_i}{\partial q_i} \right) \dot{q}_i + \left[ \frac{\partial \tau_i}{\partial L_{si}} \right]^T [\dot{x}_i + u_i] \quad (4.28)
\]

Reordering this equation yields a solution for \( u_i \):

\[
u_i = \left[ \frac{\partial \tau_i}{\partial L_{si}} \right]^T \ast \left( \dot{\xi}_i - \left( \frac{\partial \tau_i}{\partial q_i} \right) \dot{q}_i - \left[ \frac{\partial \tau_i}{\partial L_{si}} \right]^T [\dot{x}_i + u_i] \right) \quad (4.29)
\]

In eq. 4.29, the \( (\ast) \) in the term \( \left[ \frac{\partial \tau_i}{\partial L_{si}} \right]^T \ast \) denotes the pseudoinverse, which calculates the least squares solution to the equation. This is necessary because the system is generally underdefined when there are multiple redundant muscles. This will fail if \( \frac{\partial \tau_i}{\partial L_{si}} = 0 \). However, this case is only possible if every muscle at a particular joint is in a singular configuration, which is improbable.

### 4.2 Algorithmic Adaptations for Simulation

A generalized robotic model for simulation is needed. That is, the Matlab simulation should be capable of handling manipulators with arbitrary numbers of links and arbitrary parameter values. This makes it necessary, for ease of use, to avoid hard-coding model-specific values in the simulation.

Unfortunately, many of the derivations (as shown in \cite{50}) of the common parameters we use in robotics theory - the mass matrix, Coriolis matrix, velo-
ity Jacobian, etc. - involve symbolic differentiation. Because Simulink cannot properly handle online symbolic computation, we must develop a numerical approach for solving these values when the robot parameters are generalized. Below is an outline of the major algorithmic changes implemented. Note that there is no change in the theoretical derivation of these parameters; these changes are only algorithmic modifications.

4.2.1 Transformation Matrices

All required algorithmic modifications for this model assume the ability to calculate the derivative of transformation matrices $H_i^{i-1}$ with respect to the joint variable(s). Therefore, it becomes necessary to show that these can easily be calculated.

According to [50], we know that any transformation matrix $H_n^0$ is a product of the transformations $H_i^{i-1}$ from frame to frame:

$$H_n^0 = H_1^0 H_2^1 ... H_n^{n-1}$$  \hspace{1cm} (4.30)

This can be written more compactly as:

$$H_n^0 = \prod_{i=1}^{n} H_i^{i-1}$$  \hspace{1cm} (4.31)

To find the gradient with respect to the $k^{th}$ joint variable $q_k$, we can apply the product rule as follows:

$$\frac{\partial H_n^0}{\partial q_k} = \sum_{i=1}^{n} \left( \left( \prod_{j=1}^{i-1} H_j^{j-1} \right) \frac{\partial H_i^{i-1}}{\partial q_k} \left( \prod_{j=i+1}^{n} H_j^{j-1} \right) \right)$$  \hspace{1cm} (4.32)

Fortunately, because $\frac{\partial H_i^{i-1}}{\partial q_k} = 0$ for all $i \neq k$, eq. 4.32 simplifies to

$$\frac{\partial H_n^0}{\partial q_k} = \left( \prod_{j=1}^{k-1} H_j^{j-1} \right) \frac{\partial H_k^{k-1}}{\partial q_k} \left( \prod_{j=k+1}^{n} H_j^{j-1} \right)$$  \hspace{1cm} (4.33)
or, with even simpler notation,

\[
\frac{\partial H_0^0}{\partial q_k} = H_0^{k-1} \frac{\partial H_{k-1}^k}{\partial q_k} H_n^k
\]  

(4.34)

This formula also applies when breaking down \( H_{k-1}^k \) into the four sub-transformations following the Denavit-Hartenberg convention (see [50]) as:

\[
H_{k}^{k-1} = H_{\text{rot}Z} H_{\text{tran}Z} H_{\text{tran}X} H_{\text{rot}X}
\]  

(4.35)

Again in this case, only one sub-transformation contains the variable \( q_k \). Therefore, if we can calculate the (element-by-element) derivative of each fundamental transformation with respect to the joint variable, then all other transformations/parameters can be calculated numerically. For the sake of space not all four fundamental transformation derivatives will be derived. Only \( \frac{d}{dq} H_{\text{rot}Z} \) is shown as an example:

\[
\frac{d}{dq_k} H_{\text{rot}Z}(q_k) = \frac{d}{dq_k} \begin{bmatrix}
\cos(q_k) & -\sin(q_k) & 0 & 0 \\
\sin(q_k) & \cos(q_k) & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix} = \begin{bmatrix}
-\sin(q_k) & -\cos(q_k) & 0 & 0 \\
\cos(q_k) & -\sin(q_k) & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix}
\]  

(4.36)

Note that the derivative of a rotation matrix \( \frac{d}{dq_k} \mathbf{R}_{ij} \) is simply the upper-left 3 \times 3 block of the full transformation matrix derivative.

### 4.2.2 Gravity Vector

The gravity vector is defined in [50] as the gradient of the potential energy \( P \) of a manipulator with respect to the joint variables \( q \):

\[
g(q) = \frac{dP}{dq}
\]  

(4.37)

where
\[ P = \sum_{i=1}^{n} m_i g_v^T r_{ci}^0 \]  

(4.38)

and \( m_i \) and \( r_{ci} \) are, respectively, the mass and position of the center of mass of the \( i^{th} \) link, and \( g_v \) is the world gravity vector. Noting that \( r_{ci}^0 = H_i^0 r_{ci}^i \), where \( r_{ci}^i \) is the \( i^{th} \) center of mass, we can derive each term \( g_k(q) \) in \( g(q) \) to be:

\[ g_k(q) = \frac{\partial P}{\partial q_k} = \sum_{i=1}^{n} m_i g_v^T \frac{\partial H_i^0}{\partial q_k} r_{ci}^i \]  

(4.39)

Noting that \( \frac{\partial H_i^0}{\partial q_k} = 0 \) for all \( i < k \), the lower limit of the summation in Equation 4.39 can be modified for computational efficiency:

\[ g_k(q) = \frac{\partial P}{\partial q_k} = \sum_{i=k}^{n} m_i g_v^T \frac{\partial H_i^0}{\partial q_k} r_{ci}^i \]  

(4.40)

### 4.2.3 Coriolis Matrix

The Coriolis matrix \( C(q, \dot{q}) \) in the robotic equation is normally derived element-by-element as follows:

\[ c_{kj} = \frac{1}{2} \sum_{i=1}^{n} \left\{ \frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k} \right\} \dot{q}_i \]  

(4.41)

where \( d_{ij} \) is an element in the mass matrix \( M(q) \) and \( c_{kj} \) is an element of the Coriolis matrix. Therefore, this necessitates finding derivatives of \( M(q) \). No other algorithmic modifications need to be made to the derivation of the Coriolis matrix.

### 4.2.4 Mass Matrix Derivative

The algorithm for calculating the mass matrix \( M(q) \) is shown in [50] to be
\[ M(q) = \sum_{i=1}^{n} m_i J_{vi}^T J_{wi} + J_{wi}^T R_0^i I_i R_0^{0r} J_{wi} \]  \hspace{1cm} (4.42)

where \( J_{vi} \) and \( J_{wi} \) describe the linear and angular velocity Jacobians of the center of mass of the \( i^{th} \) link, \( R_0^i \) is the rotational transformation matrix for the \( i^{th} \) link, and \( I_i \) is the inertia tensor of the \( i^{th} \) link. From here it can be stated that

\[
\frac{\partial M(q)}{\partial q_k} = \sum_{i=1}^{n} m_i \frac{\partial}{\partial q_k} (J_{vi}^T J_{vi}) + \frac{\partial}{\partial q_k} \left( J_{wi}^T R_0^i I_i R_0^{0r} J_{wi} \right) \hspace{1cm} (4.43)
\]

and we can again note that \( \frac{\partial H_0}{\partial q_k} = 0 \) for all \( i < k \) to allow us to modify the lower limit of the summation from \( i = 1 \) to \( i = k \) for computational efficiency:

\[
\frac{\partial M(q)}{\partial q_k} = \sum_{i=k}^{n} m_i \frac{\partial}{\partial q_k} (J_{vi}^T J_{vi}) + \frac{\partial}{\partial q_k} \left( J_{wi}^T R_0^i I_i R_0^{0r} J_{wi} \right) \hspace{1cm} (4.44)
\]

From here, applying the product rule reveals a need for the derivatives of the velocity Jacobian.

### 4.2.5 Velocity Jacobian Derivative

Although human joints can be approximated as revolute joints, the velocity Jacobian derivative will be derived for both revolute and prismatic joints for the sake of generality. For revolute joints, the \( i^{th} \) column of the velocity Jacobian can be calculated according to [50] to be:

\[
\begin{bmatrix}
J_{vr_i} \\
J_{wr_i}
\end{bmatrix} = \begin{bmatrix}
z_{i-1}^0 \\
n 	imes (P_j^0 - O_i^0) \\
z_{i-1}^0
\end{bmatrix} \hspace{1cm} (4.45)
\]

For prismatic joints, the Jacobian column is much simpler:

\[
\begin{bmatrix}
J_{vp_i} \\
J_{wp_i}
\end{bmatrix} = \begin{bmatrix}
z_{i-1}^0 \\
0
\end{bmatrix} \hspace{1cm} (4.46)
\]

In Equations 4.45 and 4.46, \( z_i^0 \) refers to the \( i^{th} \) link’s z vector in world coor-
coordinates, \( P^0_j \) is the global position of the point of interest, and \( O^0_i \) is the global position of the \( i^{th} \) frame’s origin. Noting that the product rule applies to cross products, the derivatives of the columns of the Jacobian become

\[
\frac{\partial}{\partial q_k} \begin{bmatrix} J_{vr_i} \\ J_{wr_i} \end{bmatrix} = \left[ \left( \frac{\partial R^0_{i-1}}{\partial q_k} z_0 \right) \times (P^0_j - O^0_{i-1}) + z^0_{i-1} \times \left( \frac{\partial H^0_{i}}{\partial q_k} P^j_j - \frac{\partial H^0_{i-1}}{\partial q_k} O_0 \right) \right] \tag{4.47}
\]

\[
\frac{\partial}{\partial q_k} \begin{bmatrix} J_{vp_i} \\ J_{wp_i} \end{bmatrix} = \frac{\partial}{\partial q_k} \left[ \begin{bmatrix} \frac{\partial R^0_{i-1}}{\partial q_k} z_0 \\ 0 \end{bmatrix} \right] \tag{4.48}
\]

where we note that \( z_0 = [0, 0, 1]^T \) and \( O_0 = [0, 0, 0, 1]^T \), although all 4-element vectors will need the one “padding” to be removed to compute the cross product.

### 4.3 Simulation Example

A roughly arm-like, 2-link linkage was proposed to test the model. Four muscles were placed at the first joint and three were placed at the second joint. Figure 19 illustrates this setup when both joint angles are zero.
Figure 19: Two-link setup

Table IV displays the placements $A$ and $B$ of the muscles about each joint.

<table>
<thead>
<tr>
<th>Link $i$</th>
<th>Muscle $j$</th>
<th>$A_x$</th>
<th>$A_y$</th>
<th>$A_z$</th>
<th>$B_x$</th>
<th>$B_y$</th>
<th>$B_z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>-0.01</td>
<td>0.01</td>
<td>0.005</td>
<td>-0.15</td>
<td>0.005</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0.03</td>
<td>0</td>
<td>-0.13</td>
<td>0.01</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>0</td>
<td>-0.09</td>
<td>0</td>
<td>-0.17</td>
<td>-0.005</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>-0.01</td>
<td>-0.05</td>
<td>0</td>
<td>-0.13</td>
<td>-0.015</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>-0.07</td>
<td>0.03</td>
<td>0</td>
<td>-0.18</td>
<td>0.03</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>-0.04</td>
<td>-0.02</td>
<td>0</td>
<td>-0.22</td>
<td>-0.02</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>-0.03</td>
<td>-0.04</td>
<td>0</td>
<td>-0.23</td>
<td>-0.04</td>
<td>0</td>
</tr>
</tbody>
</table>

Table IV: Two-link manipulator example muscle placement

The manipulator was commanded to track an arbitrary vector of sine waves
\[ q_{des} = A \sin(\omega t + \phi) + A_0. \]
The trajectory parameters and the muscle constants are outlined in Table V. The linkage was simulated with zero initial conditions.
Both impedance control and inverse dynamics were used for synthetic control generation.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oscillation Amplitude</td>
<td>$A$</td>
<td>$[0.1,0.1]^T$ rad</td>
</tr>
<tr>
<td>Oscillation Frequency</td>
<td>$\omega$</td>
<td>$[1,1]^T$ rad/s</td>
</tr>
<tr>
<td>Oscillation Mean</td>
<td>$A_0$</td>
<td>$[-0.1,0.1]^T$ rad</td>
</tr>
<tr>
<td>Oscillation Phase</td>
<td>$\phi$</td>
<td>$[-\pi/2,\pi/2]^T$ rad</td>
</tr>
<tr>
<td>Tendon Slack Length</td>
<td>$s_n$</td>
<td>0.04 m</td>
</tr>
<tr>
<td>Tendon Stiffness</td>
<td>$k_s$</td>
<td>6 N/m</td>
</tr>
</tbody>
</table>

Table V: Two-link manipulator example parameters

4.3.1 Impedance Control

In the case of impedance control, the impedance parameters were chosen as follows: $I_{imp} = 3I_n$, $B_{imp} = 4I_n$, and $K_{imp} = 20I_n$ where $I_n$ is the identity matrix.

The simulated human’s joint variables $q_1$ and $q_2$ clearly converged to the desired trajectory $q_1^d$ and $q_2^d$, as seen in Figure 20.
Figure 20: Two-link manipulator under impedance control - tracking

Figure 21 shows the convergence of joint torques $\tau$ (solid lines) to the desired joint torques $\psi$ (dashed lines) and also displays all of the muscle contraction rates $u$.

Figure 21: Two-link manipulator under impedance control - controls

The muscle contraction rates, though not necessarily sinusoidal, are still reasonably smooth after the initial transient. The resulting tendon lengths and rates are shown in Figure 22.
4.3.2 Inverse Dynamics

Selecting inverse dynamics as the method of synthetic control yields similar results with shorter transients. The control gains $K_p$ and $K_d$ were both selected to be $5I_n$ where $I_n$ is the identity matrix. The simulated human’s joint variables $q_1$ and $q_2$ clearly converged close to the desired trajectory $q_1^d$ and $q_2^d$, as seen in Figure 23.

Figure 22: Two-link manipulator under impedance control - lengths and rates
Figure 23: Two-link manipulator under inverse dynamics control - tracking

Figure 24 shows the convergence of joint torques $\tau$ (solid lines) to the desired joint torques $\psi$ (dashed lines) and also displays all of the muscle contraction rates $u$.

Figure 24: Two-link manipulator under inverse dynamics control - controls

The muscle contraction rates $u$ clearly experience a much shorter transient here than in the case of an impedance controlled system. When the inverse dynamics controller is selected, there is not a synthetic dynamic effect that causes the transient to be more pronounced. The resulting tendon lengths
and rates are shown in Figure 25.

![Figure 25: Two-link manipulator under inverse dynamics control - lengths and rates](image)

The tendon lengths clearly remain positive, which is desirable. Negative simulated tendon lengths would not be physically realizable, and may cause the simulation to behave in such a way that is not physically possible.

### 4.4 Discussion

A muscle-actuated human linkage model has been developed for use in simulation. The primary goal was to create a generalized linkage model that could accommodate an arbitrary number of links without involving online differentiation. All explicit differentiations were successfully replaced with numerical computation methods. A secondary goal was the design of a muscle actuation controller for joint trajectory tracking. A cascaded synthetic and
backstepping control scheme was successfully devised for this purpose. The controller used a synthetic controller to calculate the desired joint torques for trajectory tracking, and a backstepping controller was used to enforce convergence of joint torques produced by the muscles to the desired joint torques.

Sample simulations were run to test the model. A two-link manipulator with arbitrary parameters was constructed using the proposed framework. The model performed satisfactorily and acted within feasible limits. At this point, the difference between selecting impedance control or inverse dynamics for the synthetic controller was not significant - there was very little noticeable difference in the outputs. However, Chapter V will reveal that the choice of synthetic control has significant and important effects on the output once a virtual load is introduced.

It is concluded that this model should be valid and beneficial for further use. Because the associated code is streamlined for arbitrary selection of linkage parameters as well as number of links and muscles, this model can be employed for a variety of simulation applications.
CHAPTER V

SIMULATION OF FULL CONTROLLED LINKAGE MODEL

A full, multi-link muscle-actuated linkage can now be prepared for simulation against the extremum seeking controller to optimize exercise load parameters. This batch of simulations will be very useful for demonstrating the existence of extrema for a multi-muscle system. It must also be verified that the system operates within the assumptions set forth by extremum seeking control.

In this chapter, a full multi-muscle model is simulated against an extremum seeking controller to seek optimal exercise machine load parameters. First, a generalized mass-spring-damper virtual load generator is derived and described in Section 5.1. This virtual load generator can be replaced by other designs if desired, but it is the model used for load simulation in this chapter. Second, Section 5.2 reports on the behavior of the system under a constant load, to verify and analyze the effect of the load generator on the human linkage. The extremum seeking controller is finally implemented in Section 5.3, where the
effects of load stiffness modulation are analyzed and the existence of extrema is investigated. The chapter is concluded with a discussion.

5.1 General Load Setup

If the linkage is some open kinematic chain connected to a virtual load at the end effector $E^0$ in $\mathbb{R}^3$, then the point at which force is applied to the linkage is described by

$$E^0 = H_n^0 q$$

(5.1)

where $q$ is the vector of joint positions. Let it also be assumed that the virtual mass-spring-damper load is measured with respect to a distance $x$ from a point $P^0$ in $\mathbb{R}^3$ to $E^0$, such that:

$$\Delta = E^0 - P^0$$

(5.2)

$$x = \|\Delta\| = \sqrt{\sum_{i=0}^{3} \Delta_i^2}$$

(5.3)

If $\|\Delta\| = g(f(\Delta)) = (f(\Delta))^{1/2}$, then it can be said that $\frac{d}{dt}\|\Delta\| = \frac{1}{2} (f(\Delta))^{-1/2} \dot{f}(\Delta)$.

In addition, it can be noted that $\dot{\Delta} = \dot{E}^0$ because $P^0$ is assumed constant.

This allows us to find the first and second derivatives of $x$:

$$\dot{x} = \frac{1}{x} \Delta^T \dot{E}^0$$

(5.4)

$$\ddot{x} = \left( -\frac{x}{x^2} \Delta^T + \frac{1}{x} \left( \dot{E}^0 \right)^T \right) \dot{E}^0 + \frac{1}{x} \Delta^T \ddot{E}^0$$

(5.5)

And the derivatives of $E^0$ can be calculated as

$$\dot{E}^0 = J_{v,e}(q) \dot{q}$$

(5.6)

$$\ddot{E}^0 = J_{v,e}(q) \ddot{q} + J_{v,e}(q) \dddot{q}$$

(5.7)
where $J_{v,e}(q)$ is the velocity Jacobian of the end effector. Using this model, the force of the load on the linkage will be described as

$$\|F\| = m\ddot{x} + b\dot{x} + cx$$

(5.8)

$$F = -\|F\| \frac{\Delta}{x}$$

(5.9)

as long as $x \neq 0$ (which is guaranteed if the equilibrium position $P^0$ is chosen to be outside of the workspace of the end effector).

When considering sign conventions, it is important to note that this force is defined as the force applied to the linkage by the virtual load. The force vector that represents the force applied to the virtual load by the linkage acts in the opposite direction of the force derived above.

### 5.2 Constant Load Parameters

It is first desired to validate the load model. To test it, two models will be analyzed: a simplistic single-link manipulator actuated by two antagonistic muscles, and the two-link model proposed in the previous chapter. Both setups will be tested using a spring-like virtual load, with virtual stiffness of $c = 5$ N/m.

#### 5.2.1 Single-Link Manipulator

A single-link manipulator, being relatively simple, can allow for easy visualization and verification of the three-dimensional load model. Because the motivation of this section is only to verify the load model and ensure that the linkage model reacts appropriately, only the nonzero trajectory case of the
multi-link model will be discussed, while both nonzero and setpoint trajectory simulations will be discussed for the single-link model.

A link of length 1 m was constructed that operated in the \((x, y)\) plane without gravitational effects. The two muscles were placed symmetrically about the link (when \(q = 0\)) as shown in Figure 26:

![Linkage Setup](Image)

Figure 26: One-link setup

The muscle attachment points are outlined in Table VI.

<table>
<thead>
<tr>
<th>Link (i)</th>
<th>Muscle (j)</th>
<th>(A_x)</th>
<th>(A_y)</th>
<th>(B_x)</th>
<th>(B_y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.2</td>
<td>-0.7</td>
<td>0.1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>0</td>
<td>-0.2</td>
<td>-0.7</td>
<td>-0.1</td>
</tr>
</tbody>
</table>

Table VI: One-link manipulator example muscle placement

The point of load origin \(P^0\) was located at \((1 \text{ m}, -1 \text{ m})\) in the world frame. The initial state was set to be the configuration shown in Figure 26.
Inverse Dynamics: Setpoint Regulation

If the synthetic control is chosen to be inverse dynamics where the control gains $K_p$ and $K_d$ are both selected to be $5I_n$ (where $I_n$ is the identity matrix) and the desired trajectory is just a setpoint ($q^d = 0$), then the trajectory of Figure 27 is observed.

Figure 27: One link, inverse dynamics (setpoint), constant $c$ - tracking

Knowing that $P^0$ (from which the virtual load is referenced) is located directly below the end effector when $q = 0$, it is expected that the external load should have no $x$ component at that state, a $y$ component acting in the negative direction, and no $z$ component. This is verified by Figure 28.
Figure 28: One link, inverse dynamics (setpoint), constant $c - F$

Figure 29 displays the resulting control inputs, with data from each muscle.

Figure 29: One link, inverse dynamics (setpoint), constant $c -$ controls
Impedance Control: Setpoint Regulation

If the synthetic control is chosen to be impedance control where the impedance parameters are chosen as $I_{imp} = 3I_n$, $B_{imp} = 4I_n$, and $K_{imp} = 20I_n$ (where $I_n$ is the identity matrix) and the desired trajectory is just a setpoint ($q^d = 0$), then the trajectory of Figure 30 is observed.

Figure 30: One link, impedance control (setpoint), constant $c$ - tracking

Figure 30 illustrates an obvious steady-state error. This is, of course, simply a result of the impedance controller. The controller attempts to imitate a mass-spring-damper, which will exhibit steady-state error in the presence of nonzero steady-state loading force.

Again, it is expected that the external load should have no $x$ component at $q = 0$ and a $y$ component acting in the negative direction. This is verified by Figure 31.
Figure 31: One link, impedance control (setpoint), constant $c - F$.

Figure 32 displays the resulting control inputs.

Figure 32: One link, impedance control (setpoint), constant $c - \text{controls}$
Inverse Dynamics: Nonzero Trajectory

If the synthetic control is chosen to be inverse dynamics and the desired trajectory is a sine wave $x^d = 0.2\sin(t)$, then the trajectory of Figure 33 is observed.

![Tracking Performance](image)

Figure 33: One link, inverse dynamics (trajectory), constant $c$ - tracking

Here, there is slight deviation from tracking after the initial transient. This is to be expected, as the inverse dynamics controller is not necessarily robust to external loads. The deviation from the desired trajectory does have the advantage of being slightly more physically realistic, however - humans rarely follow an externally specified trajectory exactly when prompted.

It is again expected to see no $x$ component of force and a $y$ component acting in the negative direction when $q = 0$. This is verified by Figure 34.
Figure 34: One link, inverse dynamics (trajectory), constant $c - F$

Figure 35 displays the resulting control inputs.

Figure 35: One link, inverse dynamics (trajectory), constant $c - \text{controls}$
Impedance Control: Nonzero Trajectory

If the synthetic control is chosen to be impedance control and the desired trajectory is a sine wave $x^d = 0.2\sin(t)$, then the trajectory of Figure 36 is observed.

![Tracking Performance](image)

Figure 36: One link, impedance control (trajectory), constant $c$ - tracking

Here, there is clear DC error after the initial transient. In this study, however, this is not necessarily a problem. If the performance (power, etc.) can still be optimized, enforcement of trajectory tracking is not (necessarily) an important issue.

It is again expected to see no $x$ component of force and a $y$ component acting in the negative direction when $q = 0$. This is verified by Figure 37.
Figure 37: One link, impedance control (trajectory), constant $c - F$

Figure 38 displays the resulting control inputs.

Figure 38: One link, impedance control (trajectory), constant $c - \text{controls}$
5.2.2 Two-Link Manipulator

The linkage analyzed in Chapter IV is selected for use again, and has been reproduced in Figure 39.

![Figure 39: Two-link setup](image)

The manipulator is again commanded a vector of sine waves, with the same values (and muscle parameters) as specified earlier in Table V. The manipulator’s muscle placement points can be found in Table IV. The linkage was simulated with zero initial conditions. Both impedance control and inverse dynamics were used for synthetic control generation. The virtual spring load was given a stiffness of 5 N/m and $P^0$ was set to (0.7, 0, 0) m.

Inverse Dynamics: Nonzero Trajectory

In this simulation, the synthetic control was chosen to be inverse dynamics, with the same control gains as in the single-link manipulator. The trajectory of Figure 40 was observed.
As shown in Figure 40, there is only slight deviation from tracking. As stated previously, this is not necessarily of consequence. Because \( P^0 \) is located along the x-axis, it is expected to see primarily an \( x \) component of force with little to no \( y \) component when the linkage achieves acceptable tracking. This is verified by Figure 41.
Figure 41: Two links, inverse dynamics, constant $c - F$

Figure 42 displays the resulting control inputs.

Figure 42: Two links, inverse dynamics, constant $c - \text{controls}$
Impedance Control: Nonzero Trajectory

In this simulation, the synthetic control was chosen to be impedance control, with the same control gains as in the single-link manipulator. The trajectory of Figure 43 was observed.

![Tracking Performance](image)

Figure 43: Two links, impedance control, constant $c$ - tracking

There is more tracking error when using impedance control than inverse dynamics, but the error is still relatively low.

The load magnitude (shown in Figure 44) is very similar to the load profile experienced when inverse dynamics control is utilized.
Figure 44: Two links, impedance control, constant $c - F$

Figure 45 displays the resulting control inputs.

Figure 45: Two links, impedance control, constant $c - $ controls
5.3 Extremum Seeking Simulations

Now that the load model has been validated and linkages have been shown to behave properly in the presence of external load, extremum seeking control can be applied to the load generator. The manipulators in the previous section will be simulated against loads with varying virtual stiffness. All simulations will utilize linkage trajectory tracking instead of setpoint regulation. To analyze the effects of batch processing and sliding averages, both approaches will be attempted and compared.

For multi-muscle systems, it becomes difficult to extract the average outputs (be it power, squared power, etc.) for a specific quarter cycle. At any selected instant, every muscle is (or at least could be) in a different operating range of motion. Because of this, it becomes difficult to determine exactly when each muscle is in the desired sector of motion.

There are two possible ways around this. First, the selected quarter cycle could simply be specified as a quarter cycle of motion of the linkage. However, this could pose stability issues. If, for example, the output performance is defined as power (not squared power), the power output could be measured as negative or nearly zero in that particular sector. If the output power is negative (assuming that the metric being used is not a nonnegative output such as squared power), it is likely that either the ESC scheme needs to seek a minimum instead of a maximum, or the power output in that region has no meaning and therefore is not useful to optimize. If the output is approximately zero, it is likely that a poor choice was made in the selection of the region for
analysis. For example, if the power output is described by $P = A \sin(t)$ where $A$ can be optimized, choosing to optimize the average power from $\pi/2$ to $3\pi/2$ would be fruitless (because the output would always be zero). A second method to implement quarter-cycle analysis on a multi-muscle system could be to approximate the sectors of motion of each muscle and have a separate filter for each muscle to extract the power output during that time. However, this approach can be difficult in implementation, and therefore was not used in this work. Instead, simply full cycle squared power optimization was investigated.

5.3.1 Single-Link Manipulator

Inverse Dynamics Synthetic Control

When an inverse dynamics controller is selected for linkage control, with the same control gains as stated previously in this chapter, the tracking of Figure 46 is achieved:

![Tracking Performance](image)

(a) Batch Processing  (b) Sliding Average

Figure 46: One link, inverse dynamics, ESC - tracking

It can be seen in Figure 46 that there is a slight difference in tracking
between the batch processing and sliding average method of load perturbation. Regardless of whether the tracking error is acceptable, however, an interesting phenomenon occurs in the extremum seeking controller, shown in Figure 47.

![ES Results](image)

(a) Batch Processing  (b) Sliding Average

Figure 47: One link, inverse dynamics, ESC - ESC results

Momentarily ignoring the initial transient in the power output, it can be seen that there is significant “creep” that occurs in the parameter estimate \( \hat{\theta} \). This “creep” is seemingly inexplicable. It does not seem to be approaching an actual extrema, because it will be seen that the impedance controlled linkage reaches an extrema and settles. This “creep” is also evident in the multi-link manipulator. In both cases, extending the simulation will show that the parameter will continue to creep in the same direction until the muscles can no longer operate normally. Therefore, there does not seem to be a physically realizable extremum.

It is possible that the type of linkage control (and its ability to track a reference trajectory) will affect the power output measure, or even cause no extrema to exist. A breakdown of each method of control and the extrema it
tends to create for the selected linkages will be presented in Section 5.3.3.

It could be that there is some dynamic effect that is introduced when using inverse dynamics control that causes the system to fail the static mapping condition of ESC. If such is the case, we would have to disregard data acquired using inverse dynamics, much like we must disregard the initial transient in power output (Figure 47) that seems to imply that there is a maximum that is not achieved by the extremum seeking controller. In the case of this transient, it is important to remember that the desired performance measure is a steady-state value. Because the initial transient is not a static mapping of input parameter (spring stiffness $c$) to the steady-state performance measure ($P_{avg}^2$), any result during that period is not representative of the objective function and must be discarded.

It is also interesting that there is more oscillation in the batch processing estimate of $c$. This is possibly due to the transients introduced after each cycle, producing dynamic effects that alter the estimation.

It can be seen in Figure 48a that the control inputs to the muscles show sharp discontinuities (which are most likely not physically realizable) whenever the synthetic load parameters are varied in the batch processing method. The sliding average method, on the other hand, exhibits no such phenomenon.
Figure 48: One link, inverse dynamics, ESC - muscle controls

Although it appears that the discontinuous load variation did not have a significant impact on the outcome, the sudden load changes may theoretically color the results in other cases if the transient is long enough relative to the period of motion.

**Impedance Control Synthetic Control**

When an impedance controller is selected for linkage control, with the same control gains as stated previously in this chapter, the following tracking is achieved, shown in Figure 49.
It can be clearly seen from Figure 49 that there is a significant deviation from tracking. Steady-state tracking error is an inherent quality of an impedance controlled linkage in the presence of nonzero external load. This brings out an interesting quality of the extremum seeking controller for this setup. Because the desired trajectory is not strictly enforced, it can be said that the ESC is not only finding an optimal load parameter, but is rather producing a trajectory-load combination that produces optimal performance. Technically the modified trajectory is a result of the load and can therefore not be varied independently by the extremum seeking controller, but changes in the load do produce changes in the trajectory. This realization scales to the physical world as well. In reality, a human will not follow a specified trajectory exactly. In the presence of changing external loads, it is probable that their trajectory will deviate even more from the desired trajectory (or at least change in some way). Therefore, it is unlikely that the same extrema will be detected as should be produced during perfect tracking. These extrema can
still be considered valid, however. Because the changes in periodic trajectory are small between each period, the static mapping condition of ESC still holds.

It is interesting to see in Figure 50 that there is little to no discernable “creep” in the ESC parameter estimation when impedance control is used by the linkage. The extremum seeking controller seems to have reached a stationary value - although it remains to be seen if this value is truly a maximum.

Again, we must disregard the initial power transient when analyzing the success of the extremum seeking controller. Keeping that in mind, both the batch processing and sliding average methods converge to roughly - though not exactly - the same physically realizable value. It is clear here how fulfillment of the static mapping criterion plays an important role in the success of ESC. Even slight differences in mapping can cause noticeable output differences.

The resulting control inputs can be seen in Figure 51.
The sudden load changes in the batch processing motivated ESC simulation produce instantaneous discontinuities in the control levels. However, with the exception of these discontinuities, all controls appear to remain within reasonable levels.

5.3.2 Two-Link Manipulator

Inverse Dynamics

When an inverse dynamics controller is selected for linkage control, with the same control gains as stated previously in this chapter, the following tracking is achieved, shown in Figure 52.
It can be seen in Figure 52 that the tracking error is acceptable in both the batch processing and sliding average cases. However, it is again noted that the extremaum seeking controller is unable to locate a maximum in Figure 53.

Extension of the simulation would yield similar results as with the single link manipulator - the parameter estimation continues to climb (until the link-
age controller is no longer robust enough to the load and becomes unstable, not shown here).

Again it can be seen in Figure 54a that the control inputs to the muscles experience sharp discontinuities (which are most likely not physically realizable) whenever the synthetic load parameters are varied in the batch processing method.

Figure 54: Two links, inverse dynamics, ESC - muscle controls

**Impedance Control**

When an impedance controller is selected for linkage control, with the same control gains as stated previously in this chapter, the tracking of Figure 55) is achieved.
Because the oscillation amplitude is somewhat small and the linkage is in a configuration that minimizes the steady-state error tendencies of impedance control, there is not significant steady-state error. This “acceptable” tracking error produces a situation not unlike when inverse dynamics is used as the method of synthetic joint control, seen in Figure 56.

Again, we must disregard the initial power transient when analyzing the
success of the extremum seeking controller. Keeping that in mind, both the batch processing and sliding average methods fail to reach an optimum. Extension of the simulation yields similar results as the inverse dynamics case, where the parameter estimation will continue to climb along a roughly linear trendline until the system becomes unstable.

The resulting control inputs can be seen in Figure 57.

![Figure 57: Two links, impedance control, ESC - muscle controls](image)

The sudden load changes in the batch processing motivated ESC simulation clearly produced discontinuities in the control levels. However, aside from these discontinuities, all controls appear to remain within reasonable levels.

5.3.3 Interpretation of Results

It was observed that the only case that seemed to produce an optimum was the impedance-controlled single link manipulator. Further inspection reveals that the value found was in fact not an optimum of any sort, but rather a result of poor tuning. If the ESC estimation gain were to be increased, the “creep”
that was observed in the other simulations would have also been observed.

To validate the results of the previous simulations, the systems were simulated in the absence of the extremum seeking controller, for various load stiffness values. The resulting average power values were then plotted against the corresponding load stiffness, and the results are reproduced in Figure 58. This is done to examine the location of any possible extrema and evaluate the performance of the ESC simulations.

Figure 58: Evaluation of multi-muscle extrema existence

As can be seen in Figure 58, the two-link manipulator chosen produces a
roughly parabolic performance (squared power) to parameter (load stiffness) relationship for both synthetic control cases, which yields no maximum. The “creep” experienced in the ESC simulations is therefore justifiable because the extremum seeking controller was attempting to find a maximum when there was in fact none. The single link inverse dynamics controlled linkage also did not exhibit a maximum.

The single-link manipulator under impedance control, on the other hand, is a more interesting case. First, it can be clearly seen that the region of operation during the ESC simulation (20 N/m ≤ c ≤ 35 N/m) should have produced a continuously increasing parameter estimate, as the output-to-parameter relationship does not have a maximum in that region. Therefore, it is concluded that care must be taken in tuning the extremum seeking controller. The previous simulations probably required higher estimation gains to avoid the appearance of arriving at a stable value. High estimation gains should have caused the parameter estimation to change more rapidly. Second, closer examination of the performance-to-parameter relationship (Figure 59) reveals the existence of a maximum.

![Figure 59](image.png)

Figure 59: Close-up: one link, impedance control extrema
The maximum occurs at slightly below $c = 5 \text{ N/m}$. It was attempted to employ ESC again, this time with $c = 6$ as the initial estimate (and using the sliding average method of average power calculation). The actual maximum did in fact seem to be reached, as seen in Figure 60.

![Parameter Estimation](image)

Figure 60: One link (impedance control) ESC results

Therefore, it is concluded that it is indeed possible for extrema to exist, and more importantly, it is possible to locate these extrema.

## 5.4 Discussion

Several simulations were conducted to model human linkage response to an adaptive exercise machine. After initial setup of the proposed loading conditions, two human models were analyzed. One model was a simple one-link manipulator with two antagonistic muscles placed symmetrically about the joint. The second model was a slightly more complex, arm-like two-link manipulator actuated by several redundant muscles. Both models were first tested against a constant load, and then extremum seeking control was applied to the system to find an optimal virtual load spring stiffness that maximized the average squared power of the muscles in the linkage.
It was found that the choice of synthetic linkage control - or, rather, the trajectory produced by the linkage - had an effect on the ability of the controller to seek an extremum. For certain controllers and trajectories, the average muscle power did not have a physically realizable extremum. In the case of impedance control with significant deviation from the desired trajectory, however, an extremum was located. This extremum was also physically realizable in that the load stiffness was nonnegative and the muscles still behaved reasonably.
CHAPTER VI

CONCLUSIONS AND FUTURE WORK

In conclusion, the feasibility of an extremum seeking controlled exercise machine has been demonstrated. A framework for generating muscle-actuated linkages of arbitrary size has been demonstrated to exhibit optimal power outputs in at least one case when a spring-like load is varied by an extremum seeking controller. This process was completed in four steps. First, the basic extremum seeking control scheme was put forth and analyzed. Second, the muscle model used for simulation was demonstrated to have optimal loading conditions during sinusoidal motion. Third, this muscle model was implemented in a multi-link model and evaluated in simulation. Finally, this full model was simulated against the extremum seeking controller, and optima were shown to exist for some cases. The optima found in the final simulations were found when the muscles’ contractile element average squared power was being maximized by varying load stiffness. More specifically, an optimum was found when the linkage utilized impedance control as its synthetic control method and the tracking error was significant. The existence of this extremum supports the feasibility of using extremum seeking control in prac-
tice in an adaptive exercise machine. These results also motivate analysis of other parameter combinations that could potentially be used to optimize some other muscle performance measure not discussed in this work.

To allow for simulation of the extremum seeking controller, a multi-link, muscle-actuated model development framework was developed. By using existing robotics kinematics theory, human limbs can be approximated as links. These links are actuated by multiple redundant muscles. For the developed framework, inverse dynamics and impedance control were both cascaded separately with a backstepping controller to achieve linkage trajectory tracking. The first level of control, the inverse dynamics or impedance controller, was used to calculate the desired joint torques for trajectory tracking. Backstepping control was then selected to regulate the muscle inputs to produce the desired joint torques. Simulations were run using both the inverse dynamics and the impedance control mode of synthetic control to compare results. The basic robotic linkage equations of motion were then adapted for numerical use in simulation. A single-link and a two-link manipulator were tested in simulation using this framework, and they were shown to operate within acceptable limits.

The linkages developed were then used to simulate a human against an adaptive exercise machine that varies load stiffness for squared muscle power optimization. The resulting data led to two important conclusions. The first conclusion of the simulations is that extrema can exist for multi-muscle systems, and this implies that a physical human will exhibit optimal performance under certain loading conditions. However, the simulations also revealed that
there may be control methods or trajectories that do not produce extrema. This supports the idea that proper parameter and trajectory selection is vital, because some combinations may not yield extrema.

**Future Work**

There are several opportunities and necessities for future research and development. First, more realistic linkages should be selected for simulation. That is, the linkage and muscle parameters (such as link mass, tendon stiffness, etc.) should be selected to more accurately represent specific components of the human body. The models simulated in this work use somewhat arbitrarily selected parameters. These models, though acceptable for proof of concept, are not necessarily representative of real human limbs. Real limbs not only have different masses and dimensions and such, but also act in three dimensions. Fortunately, the framework developed in this work allows for very simple modification of system parameters - including scaling to three dimensions - so parameter modification is a rather simple task.

The linkage model must also be made to include the dynamics of the parallel element and neural signal of the Hill muscle model. This will allow for more accurate model performance. Although the parallel element has been shown to often operate within the slack region, it should not always be assumed that such is true. The inclusion of the parallel element dynamics may alter the position of extrema, or even create or eliminate extrema. If the neural signals and activation dynamics are also included, more accurate muscle operating ranges should be obtained. This should limit the extremum seek-
ing controller’s tendencies to seek extrema in ranges of motion that are not physically realizable.

The choice of synthetic linkage control for the linkage model should also be analyzed more deeply. Because different control methods produce different tracking error and different control levels, the position and existence of extrema will vary between synthetic control methods. This was demonstrated in simulation, when the impedance controlled linkage reached an optimum, while the inverse dynamics controller (which had smaller tracking error) did not seem to. Therefore, more simulation studies must be conducted to analyze this phenomenon.

More loading cases must also be analyzed. Different virtual load parameters should be investigated for optimization, such as a full mass-spring-damper load or some nonlinear resistive load. Different performance measures can also be analyzed. It may be the case that certain combinations of outputs may be more likely to yield extrema. Additionally, different trajectory profiles must be investigated. Not all human motion tasks are perfectly sinusoidal, and therefore the existence of extrema for various load profiles must be investigated.

The current study does not account for the effects of fatigue. Muscle fatigue certainly may alter the input-output mapping. This would cause the static mapping condition to fail. It is possible that the controller will adapt to the slowly changing system, although it is also theoretically possible for the controller to become unstable and unsafe. It may be possible to model the effects of muscle fatigue [54] to obtain more accurate simulations.

Finally, physical implementation of the proposed extremum seeking con-
troller is the end goal. Once the simulated human linkages are properly mod-
eled and exhibit extrema under realistic trajectories, transitioning to a physical
exercise machine will allow for future customization and process optimization.
The algorithm may then be used for rehabilitation, as well as for strengthening
exercises in microgravity.
BIBLIOGRAPHY


APPENDIX
APPENDIX A

CODE REPOSITORY

For access to the repository containing code developed and used for contribution to this work, please contact Brahm Powell (author) or Dr. Hanz Richter (thesis chairperson and research advisor) at the following email addresses:

- Brahm Powell: brahm.t.powell@gmail.com
- Dr. Hanz Richter: h.richter@csuohio.edu