ABSTRACT

Estimation and Control of Series Elastic Actuators for Decentralized Systems

by

James Holley

Robotic applications continue to move out of factories and laboratories into everyday applications. This movement drives a different paradigm for robotics; towards softer, compliant actuators intended for interacting with unknown environments and humans. The series elastic actuator, a robotic actuator with intentionally designed compliance, is a leading candidate for use in robots making the transition to human environments.

Compliance in actuation provides its own engineering challenges. Series elastic actuators provide an additional degree of freedom, resonances, and more complicated controllers to operate. A series elastic estimator is proposed that models additional degrees of freedom and provides disturbance rejection in order to provide more accurate signals to the control of the actuator, and to higher level systems that may be controlling several robotic actuators together.

Furthermore, any system that may interact with humans necessarily requires an element of safety. Passive systems are systems that do not produce any energy of their own. For this reason, providing a guarantee of passivity in a system is a safety measure for deeming a robotic system fit for human interaction. In this thesis, an extension of a novel series elastic torque controller, disturbance observer torque control, can be guaranteed passive with relatively little trade off.
Finally, a nonlinear torque controller is proposed that allows torque control to converge at a rapid exponential rate. Such a controller is not only important for torque control of the actuator, but leverages a larger proof on dynamics of a robot comprised of series elastic actuators. It separates actuator dynamics from robot dynamics, allowing a high level controller to assume it is comprised of rigid actuators, opening the door for more control schemes to be developed for robots without having to consider the extra complexity introduced by compliant actuators.

This thesis proposes several models, an estimation technique, and controller analysis that contribute to the goal of improving performance of robots with decentralized control systems.
Acknowledgments

After a long and storied career with graduate school, I have many people to thank for help over the years. Completing a masters while holding down a full time job was something I completely underestimated at the onset, and without the help, guidance, and motivation of everyone involved, I surely would not have completed it.

Firstly, I would like to thank Dr. Ames and Dr. O’Malley for getting me started and finished in this endeavor. The knowledge I have gained and experiences had working with each of you, and the members of the AMBER and MAHI labs, was invaluable.

I would like to especially thank Brice Howard for helping me push through classes early on and Jordan Lack for always offering a helping hand.

Thank you to all of my co-workers who supported my decision to continue striving for my degree. There were many times I needed to prioritize school work, and I greatly appreciate you allowing me to focus on coursework throughout the process.

Finally, I would not have been able to complete any of this without the support of my loving family, who I promise to alert first when I get word of the impending robot takeover.
### Contents

Abstract ii  
List of Illustrations viii  
List of Tables xv  

1 **Introduction** 1  
   1.1 Approaches to Actuator Control in Humanoid Robots . . . . . . . . . 1  
   1.2 Research Objective . . . . . . . . . . . . . . . . . . . . . . . . . . . 3  
   1.3 Outline and Contributions . . . . . . . . . . . . . . . . . . . . . . . 4  

2 **Literature Survey** 5  
   2.1 Series Elastic Actuator Control . . . . . . . . . . . . . . . . . . . . 5  
   2.2 Series Elastic Actuator Estimation . . . . . . . . . . . . . . . . . . . 7  
   2.3 Series Elastic Actuators in Complex Robots . . . . . . . . . . . . . . 7  
   2.4 Actuator Control and Robot Stability . . . . . . . . . . . . . . . . . 8  
   2.5 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9  

3 **Modeling of a Series Elastic Actuator as Part of a Multiple Degree of Freedom Robot** 10  
   3.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10  
   3.2 Position Input From Environment . . . . . . . . . . . . . . . . . . . . 10  
   3.3 External Force and Output Inertia . . . . . . . . . . . . . . . . . . . . 12  
   3.4 Time Differing Torque and Inertia . . . . . . . . . . . . . . . . . . . . 13  
   3.5 Modeling a Robot with SEAs . . . . . . . . . . . . . . . . . . . . . . 15
4 A Linear Estimator for Series Elastic Actuators

4.1 Introduction .................................................. 18

4.2 Estimator Design and Modeling ............................... 19
   4.2.1 Motor Side Disturbances .............................. 21
   4.2.2 Load Side Disturbance ............................... 22
   4.2.3 Composition of Disturbances into the Estimator Model .... 23

4.3 Updating the Estimator and Choosing Gains ................. 23
   4.3.1 Gain Matrix ............................................. 24

4.4 Analysis ...................................................... 25
   4.4.1 Reduction of the Gain Matrix ....................... 25
   4.4.2 Estimator Stability .................................... 26

4.5 Simulation .................................................. 27
   4.5.1 Optimization and Cost Function .................... 27
   4.5.2 Implementation Details .............................. 28
   4.5.3 Results ................................................ 30

4.6 Hardware Implementation ................................... 32

5 Disturbance Observer Torque Control Passivity

5.1 Introduction .................................................. 38

5.2 Disturbance Observer Torque Control ........................ 40
   5.2.1 Passivity of Disturbance Observer .................. 41
   5.2.2 Passivity of Practically Implemented Disturbance Observer ... 42

5.3 Analysis ...................................................... 44

6 Quadratic Program based Nonlinear Embedded Control
of Series Elastic Actuators

6.1 Introduction .................................................. 48

6.2 Rapidly Exponentially Stabilizing Control Lyapunov Functions .... 49
6.2.1 Min-Norm Controller ........................................ 51
6.3 RES-CLF Constructions ........................................ 52
  6.3.1 Robotic Systems ........................................... 52
  6.3.2 Output Dynamics .......................................... 53
  6.3.3 RES-CLF Construction ..................................... 54
  6.3.4 Relationship with Time-Scaling ......................... 56
6.4 Motor Dynamics and Control .................................. 58
  6.4.1 Error Dynamics ........................................... 58
  6.4.2 Control Law Construction ................................. 59
6.5 Robotic Systems with Series Elastic Actuators ............ 60
  6.5.1 SEA System Model ....................................... 60
  6.5.2 Application of RES-CLF Motor Controller .............. 61
  6.5.3 Singular Perturbation Perspective ...................... 63
  6.5.4 Connections with Previous Results ...................... 64
6.6 Simulation ...................................................... 65
  6.6.1 Implementation .......................................... 65
  6.6.2 Results .................................................. 66
6.7 Experimental Results ......................................... 67

7 Conclusion ......................................................... 75

Bibliography .......................................................... 76
Illustrations

3.1 A Series Elastic Actuator model where the environment drives the SEA output position as an input to the model. $\theta_m$ is the motor position, $\theta_L$ is the output position, $J_m$ is motor inertia, $b_m$ is motor damping, $k$ is the spring rate, and $\tau_m$ is the motor torque. . . . . . . 11

3.2 Torsional springs for Series Elastic Actuators in the Valkyrie robot are compliant elements allowing measurement of the actuator torques. This particular spring is used in the Valkyrie upper arm assembly in the elbow pitch and shoulder yaw joints. . . . . . . . . . 11

3.3 The model has been adjusted to include a constant output inertia $J_L$ and external torque $\tau_L$. This model is more applicable to more complicated robots comprised of several SEAs in a kinematic chain. . . . 13

3.4 The SEA model now includes a time varying output inertia $J_L(t)$. This is useful when modeling a state estimator for an SEA as part of a kinematic chain. Modeling the changes in inertia that may be caused by other actuators or loads elsewhere on the chain can be taken into account for better estimation of state variables. . . . . . 14

3.5 This simple 3-DOF manipulator demonstrates how different external loads and inertias can be experienced by a given actuator. Focusing on the base joint, the top configuration has near maximum inertia as experienced by the base joint with the arm fully extended. Retracting the arm close to the base link reduces the effective inertia, and would be a near minimum case, as shown in the bottom configuration. . . . 15
3.6 An illustration of the differences in robot models between an SEA robot and a rigid robot. The SEA robot is now much more complex as each actuator has two degrees of freedom separated by the elastic spring element. ......................................................... 16

4.1 The simulation output of $\theta_L$ showing the response of the system as a result of a motor torque chirp. ................................................................. 32

4.2 Error plots are shown for estimated signals $\theta_m$, $\dot{\theta}_m$, $\theta_L$ and $\dot{\theta}_L$. The estimator provides more accurate signals for each state and accounts for motor torques. ................................................................. 33

4.3 The simulation output of $\theta_L$ showing the response of the system as a result of a disturbance torque chirp. ......................................................... 33

4.4 Error plots are shown for estimated signals $\theta_m$, $\dot{\theta}_m$, $\theta_L$ and $\dot{\theta}_L$. The estimator provides more accurate signals for each state and accounts for output disturbances. ................................................................. 35

4.5 The simulation output of $\theta_L$ showing the response of the system as a result of both a motor torque chirp and a disturbance torque chirp. 35

4.6 Error plots are shown for estimated signals $\theta_m$, $\dot{\theta}_m$, $\theta_L$ and $\dot{\theta}_L$. The estimator provides more accurate signals for each state and accounts for motor torque and output disturbances. ................................................................. 37

4.7 The spring deflection velocity, $\dot{\theta}_m - \dot{\theta}_L$, as measured on hardware using both the estimator and a LPF is shown above. The estimator leads the LPF, implying less delay and cleaner signals reported to a controller or the high level system. ................................................................. 37
5.1 The block diagram for the disturbance observer torque controller. The PD controller and plant combined as the closed loop plant are denoted in blue as \( P_{CL} \). The closed loop disturbance observer block is denoted in red as \( DOB_{CL} \).

5.2 Passivity indices of the full DOB controller with an ideal PD controller. The passivity index is less than 1 for all frequencies, satisfying Equation 5.3, which shows the full DOB system is passive.

5.3 Passivity indices of the full DOB controller with a realistically implemented PD controller. The passivity index is greater than 1 for frequencies greater than 40 Hz, violating Equation 5.3, which shows the full DOB system is not passive.

5.4 In order to make the DOB passive, a feedforward term with static gain \( \gamma \) can be applied. This feedforward gain of the system is dependent on several parameters in the closed loop DOB system. If this system is required to be passive, \( \gamma \) should be minimized in order to reduce error propagation to controllers using the output signal \( \tilde{\tau} \).

5.5 Passivity indices of the full DOB controller with a realistically implemented PD controller and wrapped feed forward term. The passivity index is less than 1 for all frequencies, satisfying Equation 5.3, which shows the realistic DOB system is again passive.
5.6 The frequency responses of the PD controller, the DOB, and the passive DOB. Both DOB controllers are shown to have much better torque tracking especially at low frequencies. The cost of the passive DOB can be seen clearly in the magnitude response, as it can only reject high frequencies to the value of $\gamma$, or about -28 dB. This trade of passivity for rejection would have to be made when considering this approach. Understanding of the cost of using the feedforward gain $\gamma$ in order to make the DOB torque controller passive is left for future work.

6.1 A simulated series elastic actuator with initial conditions of $[\theta_\Delta, \dot{\theta}_\Delta] = [0, 0]$ and $\tau_d = 5$. The step response of the motor control is smooth and converges rapidly towards the desired value, in this case an output torque of 5 Nm. The output torque is shown in blue, with the desired torque denoted by a dashed line. The convergence rate is $\frac{1}{\epsilon} = 1100$.

6.2 The motor torque output of the simulated series elastic actuator is nonsmooth. The first section with non-zero torque is where the actuator must exert torque in order to satisfy the convergence boundary condition. Once the dynamics of the actuator are meeting or exceeding the convergence rate, the actuator no longer outputs motor torque, seen from the second section. The second, negative motor torque again is the actuator responding to conditions meeting the convergence boundary, this time to keep the actuator from overshooting the desired torque beyond the convergence rate. These sections match up with Figure 6.3 where the measured convergence rate contacts the convergence boundary.
6.3 The measured convergence rate and convergence boundary of the simulated series elastic actuator. This plot clearly shows where the actuator must exert effort in order to satisfy convergence conditions. The effects are most clearly matched up with Figure 6.2 where the effort is non-zero when the convergence rate is along the convergence boundary and zero when the convergence rate is exceeding the convergence boundary.

6.4 The experimental configuration for testing the controller under the locked output conditions. The motor controller was run at a rate of 5 kHz and is outlined in white. The SEA, a Valkyrie elbow actuator, was bolted to a grounding plate. It is shown outlined in yellow. A load cell, to verify torque output in addition to the spring measurement, is shown outlined in blue. The output is also bolted to the grounding plate.

6.5 Simulated data, denoted by the blue lines, corresponds with \( \epsilon \) values of \( \frac{1}{\epsilon} = 1100, 900, \) and 700 from left to right. Data collected from the experimental setup is denoted by red dots interpolated by straight lines between samples. The experimental data lines up closely to that of the simulated data.
6.6 Simulated motor torques are shown for the same values of $\epsilon$ from Figure 6.5. The flat, non-zero portions are areas where the motor torque was simulated to be saturated. Experimental motor torque was calculated and inferred by current measurements collected by the motor driver, and are shown by red dots with interpolated red lines between samples. Although these do not match up as cleanly as the results from Figure 6.5, they still demonstrate the desired behavior of the controller. That is, the output is nonsmooth and no output current is sent to the motor during periods where the output torque is still converging towards the desired value.

6.7 With fixed output, the controller tracked a sinusoidal $\tau_d$. This shows the controller is able to track time varying signals.

6.8 With fixed output, the controller was also tested against a chirp signal. This highlights the bandwidth of the controller, demonstrating the ability to track increasingly faster $\tau_d$.

6.9 The controller was also tested against step responses. This test highlights the stability of the controller as a step signal contains high frequency content.

6.10 The controller was able to track a sinusoid with a free output - the change from locked output to free output was not modeled into the controller. This result shows promising application of the controller in SEA robots. Further tests with a high level CLF providing the $\tau_d$ are needed to to verify this controller allows the rigid body dynamics to be assumed.

6.11 Human disturbance was applied during the tracking of the sinusoid. Some deviation can be seen from the desired signal. The controller remained stable throughout the testing and tracking error during this test is shown in Figure 6.12.
6.12 The error plot, $\tau - \tau_d$ highlighting the controllers performance.

Throughout the testing, the controller was able to regulate torque within 0.2 Nm of the desired torque.
## Tables

4.1 Actuator and Output Parameters ........................................... 20

4.2 The optimized gains for the proposed estimator and low pass filter from simulating the system with only motor torque. .......................... 31

4.3 The optimized gains for the proposed estimator and low pass filter from simulating the system by only adding disturbance $v_2(t)$. .......................... 34

4.4 The optimized gains for the proposed estimator and low pass filter from simulating the system with motor torque and disturbance. $v_2(t)$. .......................... 36

5.1 Controller parameters used for passivity analysis of the disturbance observer torque controller. ......................................................... 40
Chapter 1

Introduction

1.1 Approaches to Actuator Control in Humanoid Robots

Two prevailing paradigms drive the design and required performance of robotic actuators. Industrial automation robotics is geared towards maximizing end effector accuracy. Actuators are then designed with stiff elements and minimal backlash to minimize position errors. Robots comprised of these actuators are suited for performing repeatable tasks in a well defined environment. These robots can be dangerous or fail when operating outside of the location they were designed for, sacrificing compliance for bandwidth and accuracy. Soft robotics takes the opposite stance, focusing on compliance and stable interaction with the environment. Soft robots are typically low bandwidth and have low accuracy, especially when compared to their rigid counterparts. Research on robots and robotic actuators continues to bridge the gap, searching for the best of both paradigms - a compliant, high bandwidth and high accuracy actuator.

Series elastic actuators (SEAs) were introduced as an alternative to rigid robotic actuators. Introducing a spring element between the motor and output of the actuator provides a method for force measurement and inherent compliance while sacrificing bandwidth and stiffness. The design offers a means to higher accuracy force control and a reduction in friction, backlash, and torque ripple. SEAs have been used as an actuator topology in robots intended for unknown environments.
A need for robots that can interact with humans and operate in human engineered environments has become apparent. In 2013, the Defense Advanced Research Projects Agency (DARPA) hosted the DARPA Robotics Challenge Trials followed by the DARPA Robotics Challenge Finals in 2015. The event was designed to highlight robot use in disaster response. This event introduced the field of robotics to the challenge of “complex tasks in dangerous, degraded, human-engineered environments.” The challenge exposed the need for stable robotic locomotion, manipulation, and interaction with unknown environments.

NASA has a need for similar robotic technology. As plans are made for human exploration of the moon, asteroids, and potentially Mars, a robotic platform that can successfully navigate and operate in a human-engineered environment, or safely explore unknown terrain becomes necessary. NASA developed the Valkyrie robot for such a task. Research has continued on the platform to develop software and technology relevant to tasks and skills necessary to help astronauts or maintain habitats. The research conducted for this thesis covers torque control strategies intended for use on the Valkyrie platform to further its capabilities to operate and maintain stability in an unknown environment.

Choosing SEAs for a robot provides its own complications. Compliant elements at each actuator increase the complexity of a robot and can make modeling and state estimation computationally difficult, specifically in comparison to that of a rigid link robot. To this end, stability of each SEA and the ability of each SEA motor controller to provide accurate data about their state is an important area of research. Furthermore, it is important to try to decouple actuator and robot dynamics in order to simplify the controls problem at each level.

Work in this thesis is focused around NASA’s Valkyrie Robot. Valkyrie is com-
prised of 25 SEAs, including both rotational and linear actuators. Actuators are required to have accurate and high bandwidth torque control for the robot to maintain balance and perform accurate manipulation. This paradigm of compliant and safe operation also fits within NASA’s overall mission to develop robots to assist and work alongside astronauts.

1.2 Research Objective

The objective of this research is to further control and estimation strategies for series elastic actuators intended for use in complex robots with decentralized control systems. An estimator is developed for SEA motor controllers, allowing them to account for unmodeled portions of the plant or external disturbances. The estimator provides more accurate signals with less phase delay than traditional low pass filters. The overall goal of the estimator is to provide accurate signals for motor control and provide accurate information about state variables for a high level controller, operating on multiple SEAs.

Passivity of controllers is also an important requirement for SEAs as part of a kinematic chain. If an actuator controller is not passive, it is possible for it to cause instability in other actuators in the kinematic chain, which can in turn potentially cause global instability for the robot. An extension of the disturbance observer torque control architecture is proposed, showing a reasonable trade-off to ensure torque control passivity.

Decoupling actuator dynamics from robot dynamics is also developed in this thesis. Using time-scale separation, and understanding the rate at which an actuator can converge to its desired torque value, an SEA’s dynamics can be effectively decoupled from the full robot dynamics.
1.3 Outline and Contributions

This thesis is organized into the following chapters:

- **Chapter 2** is a literature review of SEA controllers and their advantages and disadvantages.

- **Chapter 3** provides the mathematical model of the SEA, specifically as it pertains to its place in a multiple degree of freedom humanoid robot.

- **Chapter 4** contains an approach of a linear estimator for SEAs intended for implementation at the motor driver level.

- **Chapter 5** provides an extension of the disturbance observer (DOB) control topology for SEAs to ensure passivity.

- **Chapter 6** contains an implementation of a nonlinear quadratic program based torque controller and a proof decoupling SEA actuator dynamics from rigid body dynamics of the full robotic system through time-scale separation.

- **Chapter 7** concludes the above contributions by providing potential applications and outlines related topics of future research.
Chapter 2

Literature Survey

This chapter provides an overview of actuator control techniques and their relation to overall robot stability.

2.1 Series Elastic Actuator Control

The subject of elasticity in robotic control has been studied thoroughly, even before the invention of the SEA. In 1985, a survey of force control methods was accumulated by Whitney [1]. Most of the analysis done in his review is still pertinent to controlling multiple degree of freedom robots today. Time delay in control systems is a principle factor in stability. Furthermore, interacting with hard contacts in the environment requires significant reduction of control gains for many of the control strategies. Whitney describes the implementations analyzed in his paper as ‘sluggish’ when interacting with stiff environments and suggests passive compliance as a means to improve the interaction. Furthermore, he states that the only solution allowing for fast response and low contact forces would be a system that deliberately used a compliant element. His work showed that this result was not an artifact of controller bandwidth, but related to the dynamic parameter inertia and actuator torque limits, of which each robot would have some maximum value.

Later, Pratt and Williamson proposed the series elastic actuator as a means to add this compliance to a robotic system [2]. The actuator was designed with a spring
element for just that purpose - to facilitate stable interaction with the environment and increase force control accuracy. SEA design has since been iterated and improved upon with advances in technology and analysis of SEA design choices [3].

Several approaches have been developed for force control of SEAs. Vallery proposed a cascaded torque-velocity controller for SEAs [4] that allowed integral action. Perhaps the most important aspect of this controller is its emphasis on passivity, a mathematically and experimentally proven stable interaction between the actuator and the environment. This control topology was expanded upon by Tagliamonte [5] to encompass impedance control as an outer loop to the controller. Impedance control was shown to remain passive as long as the rendered virtual stiffness of the impedance controller was less than that of the physical spring.

Similarly, disturbance observer (DOB) control was adapted for SEAs by Paine [6]. The DOB is designed to force output dynamics of the actuator torque controller and greatly reduce error. This controller was also designed for multiple degree of freedom robots as the DOB compensates for changes in inertia external to the actuator.

An impedance controller loop around the DOB torque controller was analyzed by Mehling [7] showing improved impedance renderings and passivity of the impedance controller. This technique also demonstrated the ability to render a virtual stiffness greater than that of the physical spring while remaining passive.

Other research has also sought to increase SEA impedance stiffness above that of the physical spring. Passive impedance behavior above the physical spring rate using time domain techniques was shown by Losey [8]. The use of acceleration feedback at the output of a SEA can also be used to render stiffness beyond the physical spring rate, as shown in [9].
2.2 Series Elastic Actuator Estimation

Estimation techniques for SEAs lack the body of research of SEA control. Typically the force estimation from direct measurement of spring deflection multiplied by spring stiffness is used. Estimators for flexible robots to determine force, such as [10] can be adapted for use in SEAs. This method has been expanded upon by Park [11] for force observation in SEAs, specifically for reaction force-sensing SEAs (RFSEA) which has the spring between the motor stator and ground instead of motor and output of the actuator. Similarly, DOBs have been used as a means to estimate external forces during control [12]. The estimator proposed in this thesis seeks to fill a gap in SEA estimation, providing accurate estimations of states for local control loops, as well as providing accurate output position and velocity estimation for a high level controller.

2.3 Series Elastic Actuators in Complex Robots

SEAs have been incorporated in robots since their conception in 1995. Some of the earlier robots used relatively few SEAs, such as Spring Flamingo [13] which used 6 SEAs. Series elastic robots quickly became more complex, for example M2 [14] built in 1998. M2 doubled the number of degrees of freedom to 12 and assumed a more humanoid shape.

Soon the adoption of SEAs in robots became more common, with a rich array of robots designed for different purposes using SEAs. Carnegie Mellon University has incorporated a modular SEA using a rubber spring into a snake robot [15] designed for navigating through diverse terrains, as well as a hexapod [16] constructed from actuators based on the same design. Humanoid robots utilizing SEAs have also become common. This includes upper body designs on stantions designed for manipulation,
like Baxter [17]. Humanoid robots with SEAs have also been designed with the intent of bipedal walking and navigation through dangerous or degrading environments. Robots like THOR and ESCHER [18] are humanoids using SEAs that were designed for the DARPA robotics challenge.

The National Aeronautics and Space Administration (NASA) is also interested in humanoid robots capable of assisting astronauts, navigating dangerous or unknown environments, and guaranteeing safe interaction with humans. SEAs were a critical part of the design of Robonaut 2, a humanoid deployed to the International Space Station. Robonaut 2 was designed to work side by side with astronauts and relied on the torque measurements from each actuator to provide safety functions and compliant interaction with astronauts [19]. NASA furthered its SEA design by building Valkyrie, a bipedal robot designed for the DARPA Robotics Challenge [20]. Valkyrie leverages a distributed computation architecture for SEA torque control to allow more robust high level control that can ignore the spring dynamics in each actuator [6]. NASA envisions a similar bipedal robot for use helping astronauts set up or maintain habitats on planetary surfaces. The humanoid form will allow it to reuse the same tools as astronauts, reducing the overall launch payload.

2.4 Actuator Control and Robot Stability

While the goal of a controller of a complex robot may require the assumption of rigid links, this is not a safe assumption for robots with SEAs. Spong developed an augmented controller for robots with flexible joints that is similar to a controller for rigid joints with an additional term [21]. This research showed improved performance, but more importantly showed that the flexibility of the robot should be considered in high level controllers when controlling motor torque. This is not ideal, and research
by Paine sought to effectively decouple actuator dynamics from robot dynamics [6]. Single perturbation methods and time-scale separation [22] within the framework of rapidly exponentially stabilizing control Lyapunov functions can be shown to decouple the actuator dynamics and robot dynamics [23].

2.5 Conclusion

There is a large breadth of research that has been conducted on SEAs and robots with compliant elements. This thesis first focuses on addressing SEA estimation in the context of systems with unknown output inertia. Research for SEA estimators has been focused around force observation, while the goal of the proposed estimator is to provide more accurate signals for signals that are used at the actuator level and at a high level controller. Passivity of the DOB torque controller has not been addressed previously, especially in the context of a realistically implemented controller with velocity filtering. Leveraging previous research, separation of actuator dynamics and robot dynamics is shown to be possible through the choice of the actuator control law.
Chapter 3

Modeling of a Series Elastic Actuator as Part of a Multiple Degree of Freedom Robot

3.1 Introduction

This chapter describes several models of SEAs, each having their own particular use. As SEAs can be used in single degree of freedom configurations, or chained together to build a more complex robot, different aspects of an SEA model can dominate the dynamics. Similarly, when designing, building, and implementing different estimation and control methods for SEAs, different models can provide insight to sensor placement, sensor resolution, spring rate, and inertia of components. This chapter will provide some insight into these choices as they relate to the different proposed models.

3.2 Position Input From Environment

SEAs have often been modeled neglecting output inertia [2], [24], [25], [5]. This model assumes the environment drives the output position of the spring, leaving a lumped spring-mass-damper system to be analyzed for stability, passivity, and control, shown in Figure 3.1.

In this model, the motor inertia, $J_m$ and motor friction, $b_m$ are the lumped sum from everything before the elastic element. Contributors to these parameters include the motor, gearbox, bearing, or other sources of inertia and friction. The elastic
Figure 3.1: A Series Elastic Actuator model where the environment drives the SEA output position as an input to the model. $\theta_m$ is the motor position, $\theta_L$ is the output position, $J_m$ is motor inertia, $b_m$ is motor damping, $k$ is the spring rate, and $\tau_m$ is the motor torque.

Figure 3.2: Torsional springs for Series Elastic Actuators in the Valkyrie robot are compliant elements allowing measurement of the actuator torques. This particular spring is used in the Valkyrie upper arm assembly in the elbow pitch and shoulder yaw joints.

Following the work done in [24] and [6], holding a fixed position at the output yields a useful transfer function. The equations describing the system and the forward transfer function with a fixed load position are
\[ \tau_m = J_m \ddot{\theta}_m + b_m \dot{\theta}_m + k \theta_m \]  
(3.1)

\[ P(s) = \frac{\tau_L(s)}{\tau_m(s)} = \frac{k}{J_m s^2 + b_m s + k} \]  
(3.2)

where \( \tau_L = k \theta_m \) is the force exerted on the output. With knowledge of the spring rate \( k \), the parameters of the second order system, \( J_m \) and \( b_m \) can be identified via system identification methods by performing tests on an actuator with a locked output.

This system is useful for constructing force or torque controllers. This plant is used as the nominal case for the development of the disturbance observer torque controller and is used in Chapter 5.

### 3.3 External Force and Output Inertia

This system can be extended to take output inertia of a SEA into account. This model has been studied previously as a multi-input, multi-output system by Mehling [26]. A diagram of the model is shown in Figure 3.3. This model ignores effects of gear ratio and assumes the motor torque, \( \tau_m \) as the lumped torque output of the motor and any gear box that would be part of the actuator. The equations of motion then become

\[ J_m \ddot{\theta}_m + b_m \dot{\theta}_m + k(\theta_m - \theta_L) = \tau_m \]  
(3.3)

\[ J_L \ddot{\theta}_L + b_L \dot{\theta}_L - k(\theta_m - \theta_L) = \tau_L \]  
(3.4)

with measured spring torque

\[ k(\theta_m - \theta_L) = \tau_k \]  
(3.5)
Mehling provided four transfer functions of this system in order to provide a sufficient analysis of torque controller behavior on the system [7], [26]. Of importance to this thesis, the transfer function between motor torque and spring torque is

\[
\frac{\tau_k(s)}{\tau_m(s)} = \frac{J_L k s^2 + b_L k s}{J_m J_L s^4 + (J_m b_L + b_m J_L) s^3 + (k J_L + b_m b_L + J_m k) s^2 + (k b_L + b_m k) s} \quad (3.6)
\]

This transfer function is used in Chapter 5 to analyze stability and passivity of the disturbance observer torque controller for a SEA as part of a kinematic chain.

### 3.4 Time Differing Torque and Inertia

The last model proposed in this thesis describes the output inertia as a time varying quantity. The model can be shown in Figure 3.4. The output inertia, \( J_L(t) \), in an SEA robot consisting of one or more kinematic chains will change for each actuator depending on robot configuration. A simple 3-DOF manipulator in a near minimum and near maximum inertia and external torque configuration is shown in Figure 3.5. Output inertia will be bounded and have a minimum and maximum value. It is
Figure 3.4: The SEA model now includes a time varying output inertia $J_L(t)$. This is useful when modeling a state estimator for an SEA as part of a kinematic chain. Modeling the changes in inertia that may be caused by other actuators or loads elsewhere on the chain can be taken into account for better estimation of state variables.

conceivable that $J_L$ could become very large if the robot came into contact with a hard surface - that is $J_L \to \infty$. If this case occurs, Equation 3.6 can be shown to reduce to Equation 3.2 for each actuator where this approximation is safe. Adding inertia as another time varying quantity adds additional complexity to the equations used to describe the dynamics.

To simplify the analysis of including a time varying inertia term, and to isolate nonlinearities, the equations can be written as:

$$J_m \ddot{\theta}_m + b_m \dot{\theta}_m + k(\theta_m - \theta_L) = \tau_m$$  \hspace{1cm} (3.7)

$$J_{L,\text{min}} \ddot{\theta}_L + b_L \dot{\theta}_L - k(\theta_m - \theta_L) + J_{L,\text{dist}} \dot{\theta}_L = \tau_L$$  \hspace{1cm} (3.8)

where $J_{L,\text{min}}$ is a constant, the minimum inertial load the actuator could experience, and $J_{L,\text{dist}}$ is any additional time varying inertia due to the robot configuration. By writing the equations this way, the change in inertia is isolated to a single nonlinear term. This model including changing output inertia and load torque is used in Chapter 4 to construct an estimator. The estimator is designed for a system where output
Figure 3.5: This simple 3-DOF manipulator demonstrates how different external loads and inertias can be experienced by a given actuator. Focusing on the base joint, the top configuration has near maximum inertia as experienced by the base joint with the arm fully extended. Retracting the arm close to the base link reduces the effective inertia, and would be a near minimum case, as shown in the bottom configuration.

inertia and torque are expected to change.

3.5 Modeling a Robot with SEAs

Constructing a robot out of SEAs, as described in Section 2.4, introduces an additional degree of freedom per actuator. It is often desired to develop controllers that do not take into account the spring dynamics. One promising approach to reducing an SEA robot's degrees of freedom back to that of a rigid robot is by using decentralized control. That is, placing motor drivers at each SEA controlling torque at a fast enough rate can effectively decouple actuator and robot dynamics.

There are many practical reasons for choosing a decentralized control approach for an SEA robot. Control algorithms for high level tasks of humanoid robots are often already complex. Adding additional terms to account for elasticity of actuators and motor dynamics can slow down the loop rate at which a high level controller can run. Even as computing power increases, it also increases for devices that can be
Figure 3.6: An illustration of the differences in robot models between an SEA robot and a rigid robot. The SEA robot is now much more complex as each actuator has two degrees of freedom separated by the elastic spring element.

placed directly at the actuator. Modern microcontrollers allow for complicated control methods to be run at the motor level. Actuators controlled by microcontrollers with greater computational capabilities still have to report data collected from sensors back to a global controller. The bandwidth at which this communication can take place is a limiting factor in many systems and can be further limiting by adding more actuators to a communication bus. To this end, a distributed architecture is a promising solution as microcontrollers can achieve fast loop rates for actuator control and time scale separation from high level controllers.

Figure 3.6 shows a four actuator robot comprised of series elastic actuators in comparison to a rigid link robot. The equations used for modeling the rigid robot are the Euler Lagrange equations and are shown in Equation 3.9. The elastic robot has similar equations with additional terms, shown in Equation 3.10 and Equation 3.11.

\[
M(q)\ddot{q} + H(q, \dot{q}) = u
\]  \hspace{1cm} (3.9)

\[
M(q)\ddot{q} + H(q, \dot{q}) = k(q_m - q) + b(\dot{q}_m - \dot{q})
\]  \hspace{1cm} (3.10)
\[ J\ddot{q} + b(\dot{q}_m - \dot{q}) + k(q_m - q) = u_m \] (3.11)

where \( M \) is the inertia matrix, \( H \) is a vector containing the Coriolis and gravity terms, \( u \) is actuator torque, \( q \) is the joint angle, \( q_m \) is the motor angle, \( u_m \) is the torque input to a given motor, \( k \in \mathbb{R}^{n \times n} \) is the diagonal matrix of the spring rates, \( b \in \mathbb{R}^{n \times n} \) is a diagonal matrix of damping constants, and \( J \in \mathbb{R}^{n \times n} \) is a diagonal matrix of motor inertias.

Chapter 6 describes a class of controllers that, through time scaling, allows a high level controller to approximate an SEA robot as a rigid robot.
Chapter 4

A Linear Estimator for Series Elastic Actuators

4.1 Introduction

Multiple degree of freedom robotic systems require accurate state estimation in order to achieve control objectives. For decentralized robots, this consists of kinematic chains, each joint of which is controlled independently by a motor controller. The motor controller reports state variable estimates to a global controller. These typically include output position, output velocity, and actuator torque. Achieving high accuracy and low latency of state variables is often difficult; actuators are equipped with position sensors, which means the motor controller is required to estimate velocity. This is further complicated by changing global model parameters. Output inertia and external torque upon an actuator change based on the configuration of other actuators in the kinematic chain. These values are not accessible by a motor controller, as it only has access to the sensors that are directly wired to it. Estimation of state variables is often calculated locally by performing finite difference calculations using some set of previously acquired position measurements or using low pass filtering [27], [28], [29]. Both methods have a clear trade-off of accuracy and delay, which in turn can have effects on a high-level controller’s performance. For robots incorporating SEAs, this problem is more complex due to the elastic element between the motor and output. This chapter proposes an SEA estimator allowing a motor controller to accurately estimate the local states of an SEA while accounting for changes in
external torque and inertial load. The estimator does not rely on global information describing these changes, although information passed to the estimator about inertial load, that may be known by a global controller, could increase accuracy. The estimator is also compared against simpler methods of calculating state variables and implemented at the embedded level on a Valkyrie motor driver.

Another important note about this chapter is the practical implementation and use of sensors that have higher fidelity information than others. Most motor drivers have a motor position sensor and, given that the motor is usually behind a gear train, this sensor has much higher resolution near the output than a sensor placed directly at the output. This helps tremendously with the estimation of output velocity. The actuator that this estimator is intentionally designed for is a Valkyrie actuator, which has higher resolution sensors on the motor position and spring deflection. The estimator is tuned to take advantage of this fact and exploit this actuator topology.

The parameters for the Valkyrie elbow actuator tested in this chapter are contained in Table 4.1. These were found by performing a system identification on the actuator with a locked output. This is performed by commanding a current chirp, then fitting parameters from the locked output model to the frequency response data.

### 4.2 Estimator Design and Modeling

Using the modeling Equations 3.7 and 3.8, the state space matrix representation can be constructed. In order to compensate for unknown time varying values exerted on the actuator, that is $\tau_L$ and $J_{L,\text{dist}}$, two additional states, $\delta_1$ and $\delta_2$, are added to the system of equations. These states are integrators designed to reach a constant value offsetting the disturbances. Because the motor controller runs at a much faster rate than output inertia or external force can change, the integrators can be tuned to adapt
quickly to effectively cancel them out, allowing accurate estimation of other state variables. The first integrator state affecting the motor side can also help account for unmodeled dynamics or errors in model identification. Similarly, it helps compensate for an imperfect current to torque mapping that an actuator may have that is modeled in the $B_H$ matrix.

This allows the formulation of the model matrices of the observer as

$$A_H = \begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 \\
\frac{k}{J_m} & -\frac{b_m}{J_m} & \frac{k}{J_m} & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
\frac{k}{J_{L,\text{min}}} & 0 & -\frac{k}{J_{L,\text{min}}} & -\frac{b_L}{J_{L,\text{min}}} & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} \quad (4.1)$$

$$B_H = \begin{bmatrix}
0 & \frac{1}{J_m} & 0 & 0 & 0 & 0
\end{bmatrix}^T \quad (4.2)$$

States and control inputs are defined to be
\[ x = \begin{bmatrix} \theta_m & \dot{\theta}_m & \theta_L & \dot{\theta}_L & \delta_1 & \delta_2 \end{bmatrix}^T \]  

(4.3)

\[ u = \begin{bmatrix} \tau_m \end{bmatrix} \]  

(4.4)

Taking a Valkyrie actuator and its sensor suite to construct the \( C \) matrix, it is best to choose the motor position sensor, the motor velocity calculation, and the spring deflection sensor. This is because, for this actuator topology, these are the highest resolution sensors and provide the most accurate readings of variables of interest. The Valkyrie actuator topology is well documented in [6]. The analysis presented after the introduction of the \( C \) matrix can be redone to fit a different actuator topology, but these sensors are used in this chapter for practicality and eventual implementation on hardware.

\[
C_H = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
1 & 0 & -1 & 0 & 0
\end{bmatrix}
\]  

(4.5)

### 4.2.1 Motor Side Disturbances

Supposing the modeling of the actuator has inaccuracies, the motor side equations can be rewritten as

\[
(J_m + J_{m, \text{dist}})\ddot{\theta} = -k(\theta_m - \theta_L) - (b_m - b_{m, \text{dist}})\dot{\theta}_m + (\tau_m + \tau_{m, \text{dist}})
\]  

(4.6)

where the additional terms \( J_{m, \text{dist}} \in \mathbb{R}^+, \ b_{m, \text{dist}} \in \mathbb{R}^+, \) and \( \tau_{m, \text{dist}} \in \mathbb{R} \) are terms describing the discrepancy between the modeled and actual inertia, friction, and
motor torque, respectively. Lumping these model uncertainties into a single term yields

\[ J_m \ddot{\theta} = -k(\theta_m - \theta_L) - b_m \dot{\theta}_m + \tau_m + \underbrace{J_{m,\text{dist}} \ddot{\theta} + b_{m,\text{dist}} \dot{\theta} + \tau_{m,\text{dist}}}_{v_1(t)} \] (4.7)

Performing a system identification with a locked output yields accurate values for model parameters, so \( J_{m,\text{dist}} \), \( b_{m,\text{dist}} \), and \( \tau_{m,\text{dist}} \) are likely small. It is important to note that any non-zero values for the above terms will be compensated for by one of the state estimator integrators.

### 4.2.2 Load Side Disturbance

The changing output inertia and external torque are taken into account in the same manner as motor side disturbances. The output equation can be rewritten

\[ (J_{L,\text{min}} + J_{L,\text{dist}}) \ddot{\theta} = k(\theta_m - \theta_L) - b_L \dot{\theta} + \tau_L \] (4.8)

where \( J_{L,\text{dist}} \in \mathbb{R}^+ \) is the additional inertia component that is unmodeled. Collecting the disturbances an rewriting the equation yields

\[ J_{L,\text{min}} \ddot{\theta} = k(\theta_m - \theta_L) - b_L \dot{\theta} + \underbrace{J_{L,\text{dist}} \ddot{\theta} + \tau_L}_{v_2(t)} \] (4.9)

For this analysis, \( b_L \) is assumed to be negligible, so disturbance of output friction is therefore not included. The integrator compensating for the unmodeled output dynamics will account for changing external inertia and torque. It should be noted that the gain values associated with this integrator will likely be larger, as we expect changes in external torques and inertias caused by robot motion, the effects of gravity, Coriolis effects, or other disturbances in the system.
4.2.3 Composition of Disturbances into the Estimator Model

The disturbance states can then be composed into the vector

\[ v = \begin{bmatrix} 0 & v_1(t) & 0 & v_2(t) & 0 & 0 \end{bmatrix}^T \] (4.10)

The state space then analyzed can be written concisely as

\[
\dot{x} = A_H x(t) + B_H u(t) + v(t) \] (4.11)

\[ y = C x(t) \] (4.12)

4.3 Updating the Estimator and Choosing Gains

To facilitate the implementation on an embedded system and reduce complexity, a simple discrete linear estimator update function is constructed. This is set up in discrete time to approximate the matrix exponential of the linear system. That is for a given system of equations

\[ \dot{x}(t) = A x(t) \] (4.13)

and initial condition \( x(0) = x_0 \) the solution for a given time step can be given with the matrix exponential

\[ x(t) = e^{tA} x_0 \] (4.14)

where \( e^{tA} \) is defined by the convergent power series

\[ e^{tA} = I + tA + \frac{t^2 A^2}{2!} + \ldots \] (4.15)
and can be approximated by

$$e^{tA} \approx I + tA$$  \hspace{1cm} (4.16)

For implementation on an embedded system, we are concerned about how the system has evolved over the time period between real time loops, denoted as \(T\) going forward.

Using this formulation, the state model \(x^*\) is updated based on the model, \(A_H\), \(B_H\), the control input, \(u\), the previous state \(\hat{x}\) and the time step, \(T\). It is important to note that upon startup, for this estimator implementation, the initial conditions of \(\hat{x}\) are set to 0. These update equations are then written in discrete time as

\[
x^*_n = \hat{x}_{n-1} + (A_H \hat{x}_{n-1} + B_H u_n)T \hspace{1cm} (4.17)
\]

\[
\hat{x}_n = x^*_n + K(x_{\text{meas}} - C_H x^*_n) \hspace{1cm} (4.18)
\]

where \(x_{\text{meas}}\) is a vector containing the motor position, motor velocity, and spring deflection measurements, \(K\) is the gain matrix described in Section 4.3.1, and \(n\) is the current time step.

This form of updating is similar to that of a Kalman filter with constant gain, only updating the state estimate propagation and the state estimate update. This was adapted from similar methods of discrete Kalman filtering, such as [30], [31], [32], [33], [34].

### 4.3.1 Gain Matrix

The last part of the estimator to define is the gain matrix, which consists of
This gain matrix is the standard form used in estimators and Kalman filters. Its size is determined by \( C \) and can be expanded or simplified by choosing a different set of sensors. This particular size is chosen to match the best set of sensors from a Valkyrie actuator, and because of this is able to be further simplified in Section 4.4.1 and are found through simulation in Section 4.5.

4.4 Analysis

The first concern of any estimator is observability, that is the ability to deduce all of the modes of a system by monitoring sensed outputs [35]. The estimator is found to be observable since the observability matrix is full rank.

\[
\text{rank}(\begin{bmatrix} C & CA_H & CA_H^2 & \ldots & CA_H^5 \end{bmatrix}) = 6
\]

(4.20)

4.4.1 Reduction of the Gain Matrix

The next step is to reason about choosing gains. Given that the system is observable, it is possible to choose gains such that the estimator’s poles are at any location. Unfortunately, for a given system with different sensor combinations and resolutions, gain selection can be a difficult task. To facilitate the ease of finding gains on a
real system, in this case a Valkyrie actuator, the gain matrix can be reduced by inspection of the $C$ matrix. Because the first row corresponds to the sensor measuring $\theta_m$, $K_{12}$ and $K_{13}$ can be set to 0. Similarly, $K_{21}$ and $K_{23}$ are set to 0 as the second row corresponds to $\dot{\theta}_m$. The third row corresponds to the sensor measuring spring deflection, or $\theta_m - \theta_L$, so $K_{32}$ can be set to 0. Finally, $K_{52}$ is set to 0, since the disturbance we want to account for manifests itself as a torque disturbance. The final row is left to allow all sensors to contribute to error accumulated by the second integrator. The resulting, simplified matrix is

$$K = \begin{bmatrix} K_{11} & 0 & 0 \\ 0 & K_{22} & 0 \\ K_{31} & 0 & K_{33} \\ K_{41} & K_{42} & K_{43} \\ K_{51} & 0 & K_{53} \\ K_{61} & K_{62} & K_{63} \end{bmatrix}$$  \hspace{1cm} (4.21)$$

This reduction allows for the gains to be searched for a given actuator with different mechanical parameters using an optimization search algorithm.

### 4.4.2 Estimator Stability

Discrete analysis of the system is necessary to ensure the estimator is stable. This is done by first substituting Equation 4.17 into 4.18 yielding

$$\hat{x}_n = \hat{x}_{n-1} + (A_H \hat{x}_{n-1} + B_H u)T + K(x_{\text{meas}} - C(\hat{x}_{n-1} + (A_H \hat{x}_{n-1} + B_H u)T))$$  \hspace{1cm} (4.22)$$

and then by distributing and collecting like terms, the equation can be written as
\begin{equation}
\hat{x}_n = (I + A_H T - KC - KCA_H T)\hat{x}_{n-1} + (B_H - KCB_H T)u + Kx_{\text{meas}}
\end{equation}

where $A_{\text{sys}}$ can be analyzed to understand the stability of the system. Since this is a discrete system, all eigenvalues of $A_{\text{sys}}$ must be contained by the unit circle for the system to be stable. While the symbolic representation of $A_{\text{sys}}$ is very complex, it is straightforward to calculate its eigenvalues once constants for all of the system parameters and gains have been identified.

4.5 Simulation

A simulation of an SEA with a minimum output inertia and time varying external inertia was constructed in MATLAB. Motivated by the desire to implement the estimator on hardware, emphasis was placed on making the sensors in the simulation realistic. Sensors were approximated by forcing discretization similar to their operation in reality. Gaussian noise was also applied to the sensor measurements to more realistically approximate their behavior.

4.5.1 Optimization and Cost Function

Because of the complexity of the system, discretization, and number of gain values to tune, a nonlinear optimization solver was chosen to optimize the gain matrix against error between the estimated signal and the actual state variable as produced by the simulation.

The cost function with error $e_k$ and input variables $x_k$, the exact variable determined by the simulation at a timestep, and $\hat{x}_k$, the estimated state variable at each time step, was defined as
\[ e_k = \sqrt{e_{k-1}^2 + (x_k - \hat{x}_k)^2} \] (4.24)

This function was applied to each of the state variables of interest - that is \( \theta_m \), \( \dot{\theta}_m \), \( \theta_L \), and \( \dot{\theta}_L \).

Each accumulated error was then weighted and summed by

\[ e_{tot} = w_1e_{\theta_m} + w_2e_{\dot{\theta}_m} + w_3e_{\theta_L} + w_4e_{\dot{\theta}_L} \] (4.25)

The weighting of the errors from each state, then providing \( e_{tot} \) as the optimizer cost function allows different states to be focused on over others. Depending on how the actuator is constructed, or the various resolutions of sensors, the weights can be tuned to get higher precision on certain states. This approach allows the 11 gains presented in Equation 4.21 to be reduced to 4 gains, each of which is intuitive and corresponds directly to a state of interest. For example, supposing it is desired to find gains that yield the best output velocity measurement, \( w_4 \) can be chosen to be much larger than \( w_1, w_2, \) and \( w_3 \). In this way, the optimizer will be biased to reduce the error of the \( \dot{\theta}_L \) measurement, and more aggressively search for gains that reduce this error.

### 4.5.2 Implementation Details

In order to fully test the designed estimator, a chirp signal was chosen for the signal \( v_2(t) \). This allows the full frequency response of the estimator’s integrators to be tested. The chirp was centered around zero since external torques and accelerations can take either positive or negative values.

Similarly, a chirp signal was chosen as an input torque for the modeled system. This allows the model section of the estimator to be evaluated. If the estimator is
working properly, the model portion will account for the known dynamics and produce an accurate signal.

To facilitate transfer to hardware, the estimators $B_H$ and $u$ were set to zero when gain tuning.

The estimator was compared against a first order low pass filter (LPF). Many filtering techniques could have been chosen for comparison, but this was chosen for specific characteristics of a first order LPF. First, the filter is easy to implement and is a common choice at the embedded level. A first order LPF also has a minimal amount of phase lag compared to higher order filters. The low pass filter implemented is written as

$$y_n = (1.0 - \alpha)x_n + \alpha y_{n-1}$$  \hspace{1cm} (4.26)

where $\alpha \in \mathbb{R}$ and $0 \leq \alpha \leq 1$ is a relative filtering index. This can be represented as the LPF in continuous time by converting using the backward rectangular rule of $z = \frac{1}{1-Ts}$ to be

$$\frac{(1-\alpha)}{\alpha T} = \frac{\omega}{s + \omega}$$  \hspace{1cm} (4.27)

where $T$ is the time between samples and $\omega = \frac{(1-\alpha)}{\alpha T}$ is the LPF frequency. For this simulation, the exact frequency that is achieved is not the focus, rather the optimization for reducing the cost function. For this reason, the $\alpha$ values are reported, but with the previous formula can be converted quickly into the relevant frequency. This conversion between discrete and continuous time is subject to errors in the backward rectangular approximation, but serves as a good approximation for the purpose of understanding the behavior of the LPF.
In order to draw a fair comparison between the proposed estimator and the first order LPF, the same weights, \( w_1, w_2, w_3, w_4 = 1 \), were given to the cost function and the optimizer solved for the cutoff frequency of the LPF resulting in the lowest error. Since \( \dot{\theta}_L \) is not directly measurable by the system, it was calculated by the finite difference of the filtered \( \theta_L \) estimate

\[
\dot{x}_n = \frac{\hat{x}_n - \hat{x}_{n-1}}{T} \tag{4.28}
\]

and the put through a low pas filter given in Equation 4.26. The LPF was also given the additional sensor input of output position. This allows the output velocity to be calculated by finite difference. The sensor was modeled after the sensors located at the output of each Valkyrie actuator.

4.5.3 Results

Motor Torque Only

First, the disturbance vector \( v(t) \) was set to zero and a linear chirp signal of amplitude 25 was chosen as a motor torque signal. Both filters were optimized against an equally weighted cost function. The resulting gains are shown in Table 4.2. To show the overall behavior of the system, the simulated output position is shown in Figure 4.1.

Disturbance Only

Next, the motor torque was set to zero and the disturbance signal \( v_2(t) \) was chosen to be a linear chirp signal of amplitude 25. Resulting gains after optimizing each filter are found in Table 4.3.
Table 4.2: The optimized gains for the proposed estimator and low pass filter from simulating the system with only motor torque.

<table>
<thead>
<tr>
<th>Optimized Gains for Motor Torque Only</th>
<th>( K_{\text{Estimator}} )</th>
<th>LPF</th>
</tr>
</thead>
<tbody>
<tr>
<td>( K_{11} )</td>
<td>0.0274</td>
<td>( \alpha_1 )</td>
</tr>
<tr>
<td>( K_{22} )</td>
<td>0.9966</td>
<td>( \alpha_2 )</td>
</tr>
<tr>
<td>( K_{31} )</td>
<td>0.0281</td>
<td>( \alpha_3 )</td>
</tr>
<tr>
<td>( K_{33} )</td>
<td>-0.0281</td>
<td>( \alpha_4 )</td>
</tr>
<tr>
<td>( K_{41} )</td>
<td>25.503</td>
<td>( \alpha_5 )</td>
</tr>
<tr>
<td>( K_{42} )</td>
<td>0.0613</td>
<td>( \alpha_6 )</td>
</tr>
<tr>
<td>( K_{43} )</td>
<td>-25.503</td>
<td>( \alpha_7 )</td>
</tr>
<tr>
<td>( K_{51} )</td>
<td>102.59</td>
<td>( \alpha_9 )</td>
</tr>
<tr>
<td>( K_{53} )</td>
<td>-102.59</td>
<td>( \alpha_{11} )</td>
</tr>
<tr>
<td>( K_{61} )</td>
<td>395.13</td>
<td>( \alpha_{13} )</td>
</tr>
<tr>
<td>( K_{62} )</td>
<td>-1.667</td>
<td>( \alpha_{15} )</td>
</tr>
<tr>
<td>( K_{63} )</td>
<td>-395.13</td>
<td>( \alpha_{17} )</td>
</tr>
</tbody>
</table>

**Final Cost:** 1.0726 \( \text{Final Cost:} \) 4.5442

**Motor Torque and Disturbance**

Finally, both motor torque and the disturbance were both chosen to be chirp signals. The resulting gains are shown in Table 4.4.

**Conclusion**

The estimator provides more accurate estimations of the state than a LPF. The last important information needed for the state estimator is the calculation of eigenvalues...
Figure 4.1: The simulation output of $\theta_L$ showing the response of the system as a result of a motor torque chirp.

of $A_{sys}$ by Equation 4.23 using the final gains used to account for both motor side and output side disturbances shown in Table 4.4. The resulting eigenvalues are $\lambda_1 = 0.004$, $\lambda_2 = 0.9738 + 0.1364i$, $\lambda_3 = 0.9738 - 0.1364i$, $\lambda_4 = 0.9671$, $\lambda_5 = 0.9991$, $\lambda_6 = 0.9999$. All eigenvalues lie within the unit circle, so the discrete system is stable.

4.6 Hardware Implementation

The estimator was implemented on a Valkyrie motor driver. It is difficult to evaluate the estimator against the LPF in the same manner as the simulation because the true states of the actuator are unknown. For this reason, a simple evaluation of phase of spring deflection velocity, $\dot{\theta}_m - \dot{\theta}_L$ was used to evaluate the estimator. Because of delay in the LPF and finite difference calculation, the estimator is shown to lead the LPF for this signal, shown in Figure 4.7. This shows the estimator is working as desired, but a further and more comprehensive evaluation of the estimator implemented on hardware is left for future work.
Figure 4.2: Error plots are shown for estimated signals $\theta_m$, $\dot{\theta}_m$, $\theta_L$ and $\dot{\theta}_L$. The estimator provides more accurate signals for each state and accounts for motor torques.

Figure 4.3: The simulation output of $\theta_L$ showing the response of the system as a result of a disturbance torque chirp.
Table 4.3: The optimized gains for the proposed estimator and low pass filter from simulating the system by only adding disturbance $v_2(t)$.

<table>
<thead>
<tr>
<th>Optimized Gains for Motor Torque Only</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Estimator</strong></td>
<td><strong>LPF</strong></td>
</tr>
<tr>
<td>$K_{11}$</td>
<td>$\alpha_1$</td>
</tr>
<tr>
<td>$K_{22}$</td>
<td>$\alpha_2$</td>
</tr>
<tr>
<td>$K_{31}$</td>
<td>$\alpha_3$</td>
</tr>
<tr>
<td>$K_{33}$</td>
<td>$\alpha_4$</td>
</tr>
<tr>
<td>$K_{41}$</td>
<td>$\alpha_5$</td>
</tr>
<tr>
<td>$K_{42}$</td>
<td>$\alpha_6$</td>
</tr>
<tr>
<td>$K_{43}$</td>
<td></td>
</tr>
<tr>
<td>$K_{51}$</td>
<td>101.09</td>
</tr>
<tr>
<td>$K_{53}$</td>
<td>-101.09</td>
</tr>
<tr>
<td>$K_{61}$</td>
<td>418.41</td>
</tr>
<tr>
<td>$K_{62}$</td>
<td>-1.5808</td>
</tr>
<tr>
<td>$K_{63}$</td>
<td>-418.41</td>
</tr>
<tr>
<td><strong>Final Cost: 2.9797</strong></td>
<td><strong>Final Cost: 7.615</strong></td>
</tr>
</tbody>
</table>
Figure 4.4: Error plots are shown for estimated signals $\theta_m$, $\dot{\theta}_m$, $\dot{\theta}_L$, and $\dot{\theta}_L$. The estimator provides more accurate signals for each state and accounts for output disturbances.

Figure 4.5: The simulation output of $\theta_L$ showing the response of the system as a result of both a motor torque chirp and a disturbance torque chirp.
Table 4.4: The optimized gains for the proposed estimator and low pass filter from simulating the system with motor torque and disturbance. $v_2(t)$.

<table>
<thead>
<tr>
<th>Optimized Gains for Motor Torque Only</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimator</td>
<td>LPF</td>
<td></td>
</tr>
<tr>
<td>$K_{11}$</td>
<td>$α_1$</td>
<td>0.4948</td>
</tr>
<tr>
<td>$K_{22}$</td>
<td>$α_2$</td>
<td>0.0035</td>
</tr>
<tr>
<td>$K_{31}$</td>
<td>$α_3$</td>
<td>0.5783</td>
</tr>
<tr>
<td>$K_{33}$</td>
<td>$α_4$</td>
<td>0.54711</td>
</tr>
<tr>
<td>$K_{41}$</td>
<td>$α_5$</td>
<td>0.59663</td>
</tr>
<tr>
<td>$K_{42}$</td>
<td>$α_6$</td>
<td>0.59332</td>
</tr>
<tr>
<td>$K_{43}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$K_{51}$</td>
<td>110.17</td>
<td></td>
</tr>
<tr>
<td>$K_{53}$</td>
<td>-110.17</td>
<td></td>
</tr>
<tr>
<td>$K_{61}$</td>
<td>482.99</td>
<td></td>
</tr>
<tr>
<td>$K_{62}$</td>
<td>-1.7472</td>
<td></td>
</tr>
<tr>
<td>$K_{63}$</td>
<td>-482.99</td>
<td></td>
</tr>
<tr>
<td>Final Cost: 2.908</td>
<td>Final Cost: 7.955</td>
<td></td>
</tr>
</tbody>
</table>
Figure 4.6: Error plots are shown for estimated signals $\theta_m$, $\dot{\theta}_m$, $\theta_L$ and $\dot{\theta}_L$. The estimator provides more accurate signals for each state and accounts for motor torque and output disturbances.

Figure 4.7: The spring deflection velocity, $\dot{\theta}_m - \dot{\theta}_L$, as measured on hardware using both the estimator and a LPF is shown above. The estimator leads the LPF, implying less delay and cleaner signals reported to a controller or the high level system.
Chapter 5

Disturbance Observer Torque Control Passivity

5.1 Introduction

Stability, in general, is not a strict enough criteria for complex robotic systems. As actuators are linked together, the stability of each system becomes coupled. Coupled systems are not necessarily stable, even if each individual part is. This introduces the concept of passivity, for which more general stability statements can be made. Passive systems are stable, and, passive systems coupled together in parallel or in feedback yield a passive system. In this way, ensuring an actuator’s control is passive is a principle concern in systems requiring additional safety factors.

The calculation of passivity is expressed in several different forms. Khalil defines a system \( y = h(t)u \) to be passive if \( u^T y \geq 0 \) [36]. From [37], for linear systems, the system \( G(s) \) is passive if all input-output trajectories \( y(t) = Gu(t) \) satisfy:

\[
\int_0^T y^T(t)u(t)dt > 0, \quad \forall T > 0
\]  

(5.1)

Equivalently, a system is passive if its frequency response is positive real. This means the entirety of the Nyquist plot is in the right half plane and can be written

\[
G(j\omega) + G(j\omega)^H > 0, \quad \forall \omega \in \mathbb{R}
\]  

(5.2)

Intuitively, this indicates that passive systems only consume or dissipate energy. The only way to get an increase in output is to increase the input. For robots
comprised of multiple actuators, introducing one or more non-passive actuators is a
stability concern as added energy to the system can quickly couple to other actuators
and further instability. If the robot is subject to an unknown environment, human
interaction, or some other disturbance, it is important for the robot to remain stable.
Although human input is an unknown quantity, humans are generally regarded as
passive in the literature.

From [37] calculating the passivity of a system is performed by noting the equiv-
alaney of Equation 5.2 and the small gain condition

$$\|(I - G(j\omega))(I + G(j\omega))^{-1}\| < 1, \quad \forall \omega \in \mathbb{R} \quad (5.3)$$

and then evaluating the peak gain as the measure of passivity

$$R := \|(I - G)(I + G)^{-1}\|_{\infty} \quad (5.4)$$

where $G$ is passive if $R < 1$.

Of particular importance later in the chapter, a system is input-feedforward pas-
sive if

$$u^T y \geq u^T \gamma(u) \quad (5.5)$$

for some function $\gamma$ [36]. This also provides a useful metric for determining how close
a system is to being passive or not passive called the input passivity index. The
passivity index is the largest $\gamma$ such that

$$\int_0^T y^T(t)u(t)dt \geq \gamma \int_0^T u^T(t)u(t)dt \quad (5.6)$$
for all trajectories \( y(t) = Gu(t) \) and \( T > 0 \). If \( \gamma > 0 \), the system is input strictly passive. If \( \gamma \) is less than zero, the system is said to have a shortage of passivity as it requires the static feedforward gain \( \gamma \) to become passive.

In this chapter, passivity indices are presented, showing the passivity of systems across frequencies. Given Equation 5.4, \( R \) can be calculated for a range of frequencies and plotted. If \( R < 1 \) for all frequencies, the system is passive. If \( R > 1 \) the system has a shortage of passivity that can be shown where \( R \) exceeds 1 in a given passivity plot.

### 5.2 Disturbance Observer Torque Control

Disturbance observer (DOB) torque control allows the accurate tracking of torques by comparing output to that of an ideal plant, then adjusting the input reference to account for any discrepancy. This method of torque control implemented on the Valkyrie robot is well documented in [6]. The controller is shown to accurately track torques, but further exploration of passivity analysis is needed. To achieve this, the DOB block diagram can be analyzed and shown to be passive. The full controller, with transfer functions to input and outputs, is shown in Figure 5.1.
5.2.1 Passivity of Disturbance Observer

The passivity of the controller is linked to the PD torque control loop and plant from Equation 3.6, denoted as $P$. The closed loop system for torque control can then be written

$$ G = \frac{FFP + PC}{1 + PC} \quad (5.7) $$

where $FF$ is the feed-forward component and $C$ is the PD controller transfer function. This analysis assumes input of $\tau_m$, or $FF = 1$. An ideal PD controller can be written as

$$ C(s) = K_P E(s) + sK_D E(s) \quad (5.8) $$

where $K_P$ and $K_D$ are control gains with values given by Table 5.2 and $E(s)$ is $\tau_r - \tau$.

From [6] the closed loop transfer function of the DOB is
\[
P_{DOB}(s) = \frac{G}{1 + Q(GP^{-1}_n - 1)} \tag{5.9}
\]

where \( Q \) is a second order Butterworth filter with transfer function

\[
Q(s) = \frac{1}{(s/\omega_q)^2 + \sqrt{2}(s/\omega_q) + 1} \tag{5.10}
\]

and \( \omega_q = 2\pi f_q \) as the cutoff frequency used in Table 5.2. The inverse of the nominal plant, \( P^{-1}_n \), is

\[
P^{-1}_n(s) = \frac{J_m s^2 + (b_m + kK_D)s + (k + k*K_P)}{kK_D s + k + kK_P} \tag{5.11}
\]

Analyzing the closed loop form of the DOB yields a passive system with the passivity plot shown in Figure 5.2. From the figure, it is seen that there is no excess passivity in the closed loop DOB system as the passivity index approaches 1 as the frequency increases.

### 5.2.2 Passivity of Practically Implemented Disturbance Observer

Supposing the controller is to be implemented, \( C(s) \) must take a different form since Equation 5.8 is not causal. A realistic implementation includes a pole for the derivative term at \( \omega \), yielding

\[
C(s) = K_P E(s) + \frac{sw_\tau K_D}{s + \omega_\tau} E(s) \tag{5.12}
\]

or equivalently in transfer function form

\[
\frac{C(s)}{E(s)} = \frac{(K_D + \frac{K_P}{\omega_\tau})s + K_P}{s + \omega_\tau + 1} \tag{5.13}
\]
Figure 5.2: Passivity indices of the full DOB controller with an ideal PD controller. The passivity index is less than 1 for all frequencies, satisfying Equation 5.3, which shows the full DOB system is passive.

where it can be seen that Equation 5.13 becomes Equation 5.8 as $\omega_r \to \infty$ and $\omega_r = 2\pi f_r$, or the cutoff frequency specified in hertz. The value used for analysis is found in Table 5.2.

Making this change to the PD controller has implications on passivity, failing the criteria specified in Equation 5.3. This can be seen in Figure 5.3 as the passivity index is greater than 1 for some frequencies.

This implementation of the DOB now has a shortage of passivity at frequencies above 40 Hz. To address this, a feed-forward term around the DOB, shown in Figure 5.4, can be added. Calculating the feed-forward term is straightforward for this system since it is single-input single-output. This can be done by calculating the minimum value of Equation 5.2, calculated as a static gain of $\gamma = 0.038$ for the system being analyzed. The passivity plot of the DOB with the feed forward term is shown in Figure 5.5. The feed forward term shifts the passivity plot down by value
Figure 5.3: Passivity indices of the full DOB controller with a realistically implemented PD controller. The passivity index is greater than 1 for frequencies greater than 40 Hz, violating Equation 5.3, which shows the full DOB system is not passive.

\[ \tilde{\tau} = \tau + \gamma \tau_d \]  

(5.14)

where \( \tau = \tau_d DOB_{CL} \), the torque output by the closed loop DOB torque controller.

5.3 Analysis

Adding the feed-forward term to the closed loop DOB system presents a cost of making the system passive as error passed to another controller using the torque controller. For the case of the system and parameters analyzed above, a small amount of the output measurement will be due to the desired signal passed in. Controllers working with this system will have to take in \( \tilde{\tau} \) as seen in Figure 5.4, which is not the actual torque output of the system. This inherent error, and since the system will
Figure 5.4: In order to make the DOB passive, a feedforward term with static gain $\gamma$ can be applied. This feedforward gain of the system is dependent on several parameters in the closed loop DOB system. If this system is required to be passive, $\gamma$ should be minimized in order to reduce error propagation to controllers using the output signal $\tilde{\tau}$.

only reject high frequencies to the value of $\gamma$ has to be considered when choosing the passive DOB. Further tuning and analysis, left for future work, to reduce the value of $\gamma$ can be done, as well as examining other methods of passifying the system. One such candidate is output-feedback, which would leave the torque signal intact for a higher level controller, but for this system, such an approach is not available through a static gain.

Despite this cost, using the DOB still provides an accurate means of tracking torque in a system with a free output over PD control. This is most visible by examining the frequency response of the system, shown in Figure 5.6. The DOB and passive DOB show accurate torque tracking up to about 50 Hz, showing more ideal behavior than that of the PD controller. From a practical perspective, systems are subject to noise and disturbances, often much larger than the theoretical value of $\gamma$ found for this system. It is clear that if a passive and low error torque tracking controller is desired, choosing to add this feedforward term is a small cost. This decision would have to be made within the context of a desired application, and further analysis of understanding the cost of introducing the feedforward term $\gamma$ on
Figure 5.5: Passivity indices of the full DOB controller with a realistically implemented PD controller and wrapped feed forward term. The passivity index is less than 1 for all frequencies, satisfying Equation 5.3, which shows the realistic DOB system is again passive.

other controllers, such as an impedance controller around the DOB torque controller, is left for future work.
Figure 5.6: The frequency responses of the PD controller, the DOB, and the passive DOB. Both DOB controllers are shown to have much better torque tracking especially at low frequencies. The cost of the passive DOB can be seen clearly in the magnitude response, as it can only reject high frequencies to the value of $\gamma$, or about -28 dB. This trade of passivity for rejection would have to be made when considering this approach. Understanding of the cost of using the feedforward gain $\gamma$ in order to make the DOB torque controller passive is left for future work.
Chapter 6

Quadratic Program based Nonlinear Embedded Control of Series Elastic Actuators

6.1 Introduction

Motivated by the foundational work of Ghorbel [21] and the time-scale separation of singularly perturbed systems [22], the framework of rapidly exponentially stabilizing control Lyapunov functions (RES-CLFs) [38], [39], is applied to the embedded level to achieve time-scale separation between the actuator and rigid body dynamics of the robot. RES-CLFs have proven effective in the control of robotic systems in the context of bipedal locomotion [39], [40] due to the fact that they guarantee exponential convergence at a desired rate of $\frac{1}{\epsilon} > 0$ i.e.,

$$\epsilon \dot{V}(x, u) \leq -\gamma V(x)$$

for a RES-CLF $V$ and the proper choice of control input $u$. With the observation that $\epsilon$ can be viewed as a time-scaling factor, $\tau = \frac{t}{\epsilon}$, the existence of a RES-CLF suggests the ability to create dynamics operating at different time-scales. Therefore, applying RES-CLFs to the actuator dynamics in a series elastic robotic system allows for the separation of the actuator dynamics from the rigid body dynamics.

Work in this chapter appeared in the 53rd IEEE Conference of Decision and Control, 2014. I would like to thank and acknowledge the contributions of Aaron D. Ames as a co-author.
To formally establish the decoupling afforded by RES-CLFs, properties of a specific class of RES-CLFs are formulated and applied to rigid body robotic systems (Section 6.3) after introducing the base definitions needed (Section 6.2). These RES-CLFs are then applied to the model of a series elastic actuator isolated from the rigid body dynamics of the system in Section 6.4; in particular, the RES-CLF naturally yields a quadratic program (QP) that minimizes actuator torque while achieving a desired convergence rate. Implementing this QP through its closed form solution yields a nonlinear nonsmooth embedded controller.

Returning to the full-order SEA model in Section 6.5 and, through the separation of time scales afforded by the RES-CLF based embedded controller, the dynamics of the full-order SEA model can be effectively decoupled into a rigid body system and isolated motor dynamics. In particular, by considering outputs in the joint coordinates and a RES-CLF controller for the rigid body dynamics that drives these outputs to zero exponentially, the outputs display the same convergence up to $O(\epsilon)$ for the full order SEA model with the RES-CLF embedded motor controller.

To demonstrate these results, the RES-CLF embedded controller is simulated and implemented on hardware in Section 6.6 and Section 6.7.

### 6.2 Rapidly Exponentially Stabilizing Control Lyapunov Functions

It is first important to consider a brief overview of rapidly exponentially stabilizing control Lyapunov functions (CLFs) in the context of nonlinear systems, which extend classical notions of CLFs [41], [42], [43] in order to achieve "rapid" exponential convergence.
Consider a nonlinear affine control system of the form

\[
\begin{align*}
\dot{x} &= f(x, z) + g(x, z)u \\
\dot{z} &= q(x, z)
\end{align*}
\]  

(6.2)

where \( x \in X \) are controlled (or output) states, \( z \in Z \) are the uncontrolled states, and \( U \) is the set of admissible control values for \( u \). In addition, it is assumed that \( f(0, z) = 0 \), or that the zero dynamics surface \( Z \) defined by \( x = 0 \) with dynamics given by \( \dot{z} = q(0, z) \) is invariant.

The focus of this chapter is on two forms of control Lyapunov functions: exponentially stabilizing and rapidly exponentially stabilizing. The interplay between these two types of CLFs will become clear in the context of time scaling as discussed in Section 6.5.

**Definition 6.1**

A continuously differentiable function \( V : X \to \mathbb{R} \) is an **exponentially stabilizing control Lyapunov function (ES-CLF)** if there exist positive constants \( c_1, c_2, c_3 > 0 \) such that

\[
\begin{align*}
\inf_{u \in U} [L_f V(x, z) + L_g V(x, z)u + c_3 V(x)] &\leq 0
\end{align*}
\]  

(6.4)

for all \( (x, z) \in X \times Z \).

Motivated by the desire to achieve rapid exponential convergence, the following augmented definition is introduced.

**Definition 6.2**

A continuously differentiable function \( V : X \to \mathbb{R} \) is a **rapidly exponentially stabilizing control Lyapunov function (RES-CLF)** if there exist positive constants
$c_1, c_2, c_3 > 0$ such that for all $0 < \epsilon < 1$

\[
c_1 \|x\|^2 \leq V(x) \leq \frac{1}{\epsilon^2} c_2 \|x\|^2
\]  

(6.5)

\[
\inf_{u \in U} \left[ L_f V(x, z) + L_g V(x, z)u + \frac{1}{\epsilon} c_3 V(x) \right] \leq 0
\]  

(6.6)

for all $(x, z) \in X \times Z$.

### 6.2.1 Min-Norm Controller

The existence of a RES-CLF yields a family of controllers that rapidly exponentially stabilize the system to the zero dynamics [44], [43]. In particular, consider the control values

\[
K_\epsilon(x, z) = \arg\min_{u \in U} \|u\|:
\begin{align*}
L_f V(x, z) + L_g V(x, z)u + \frac{1}{\epsilon} c_3 V(x) < 0
\end{align*}
\]  

(6.7)

wherein it follows that

\[
u_\epsilon(x, z) \in K_\epsilon(x, z) \Rightarrow \|x(t)\| \leq \frac{1}{\epsilon} \sqrt{\frac{c_2}{c_1}} e^{-\frac{1}{2c_3} t} \|x(0)\|
\]  

(6.8)

In addition, this yields specific feedback controllers, e.g., the min-norm controller

\[
m_\epsilon(x, z) = \arg\min_{u \in K_\epsilon(x, z)} \|u\|:
\begin{align*}
\psi_{0,\epsilon}(x) + \psi_{1,\epsilon}^T(x)u & \leq 0
\end{align*}
\]  

(6.9)

where

\[
\psi_{0,\epsilon}(x, z) = L_f V_\epsilon(x, z) + \frac{1}{\epsilon} \gamma V_\epsilon(x, z)
\]  

(6.10)

\[
\psi_{1,\epsilon}(x, z) = L_g V_\epsilon(x, z)^T
\]
While the controller \( m_\epsilon(x, z) \) that minimizes the control effort \( u \) can be stated in closed form as

\[
m_\epsilon(x, z) = \begin{cases} 
  \frac{-\psi_{0,\epsilon}(x,z)\psi_{1,\epsilon}(x,z)}{\psi_{0,\epsilon}(x,z) + \psi_{1,\epsilon}(x,z)} & \text{if } \psi_{0,\epsilon} > 0 \\
  0 & \text{if } \psi_{0,\epsilon} \leq 0
\end{cases}
\]

(6.11)

it is important to note that this closed form solution is the solution to the quadratic program (QP)

\[
m_\epsilon(x, z) = \arg\min_{u \in U} u^T u \\
\text{s.t. } \psi_{0,\epsilon}(x, z) + \psi_{1,\epsilon}^T(x, z)u \leq 0
\]

(6.12)

This formulation leads to a new class of controllers utilizing CLF based QPs; these have been applied to locomotion and manipulation in bipedal robots [45], and have been utilized to experimentally achieve robotic walking [46], [40].

### 6.3 RES-CLF Constructions

This section considers rigid body dynamics for a robotic system and, based on the desire to drive an output function \( y \rightarrow 0 \), constructs a specific class or RES-CLFs. A majority of this section is devoted to establishing the properties of this class of RES-CLFs. Specifically, the relationship between RES-CLFs and ES-CLFs that can be established through the use of time scaling is provided.

#### 6.3.1 Robotic Systems

Let \( Q \) be the configuration space of a robot with \( n \) degrees of freedom, i.e. \( n = \dim(Q) \) with coordinates \( q \in Q \). For the sake of definiteness, it may be necessary to choose \( Q \)
to be a subset of the actual configuration space of the robot so that global coordinates can be defined such that $Q$ is embeddable in $\mathbb{R}^n$, or more simply $Q \subset \mathbb{R}^n$.

Consider the equations of motion presented in Equation 3.9. For the sake of simplicity, the robot is considered to be fully actuated. Control inputs can then be considered to be $u \in \mathbb{R}^n$.

### 6.3.2 Output Dynamics

Consider an output function $y : Q \rightarrow \mathbb{R}^n$ and the control objective of driving $y(q) \rightarrow 0$ exponentially. Differentiating $y$ twice yields the output dynamics

$$
\ddot{y}(q) = D_y(q)\ddot{q} + \dot{D}_y(q, \dot{q})\dot{q} + A(q, \dot{q})u
$$

where $D_y(q) = \frac{\partial y}{\partial q}(q)$ is the Jacobian of the output $y$. Substituting in the dynamics of the robotic system in Equation 3.9 yields

$$
\ddot{y} = -D_yM^{-1}H + \dot{D}_y \dot{q} + D_yM^{-1}u
$$

where the dependence on $q$ and $\dot{q}$ was removed for notational simplicity. Letting $x = (y, \dot{y})$, then Equation 6.14 can be written in the general form of Equation 6.2.

$$
\dot{x} = \begin{bmatrix}
  x_2 \\
  L_7^2(x) \\
  f(x)
\end{bmatrix} + \begin{bmatrix}
  0 \\
  A(x) \\
  g(x)
\end{bmatrix} u
$$

where $L_7^2$ and A can be expressed in terms of the coordinates $x$ due to the standard diffeomorphism [47]. Also note that, in this case, the robot is assumed to be fully actuated, the zero dynamics in Equation 6.2 do not exist, leaving only the output dynamics.
Assume that the decoupling matrix, $A$, is invertible, i.e., that $y$ has (vector) relative degree 2, and choose the control law

$$u = A^{-1}(-L_f^2 + \mu) = MD_y^{-1}(\mu - \dot{D_y}q) + H$$

for some $\mu \in \mathbb{R}^n$ resulting in

$$\ddot{y} = \mu$$

This choice of control law allows us to write the output dynamics in Equation 6.15 as

$$\dot{x} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} \mu$$

### 6.3.3 RES-CLF Construction

With the error dynamics in hand, to construct a RES-CLF, first consider the continuous time algebraic Riccati equations (CARE)

$$F^T P + P F - P F F^T P + Q = 0$$

for any $Q = Q^T > 0$ and with solution $P = P^T > 0$. One can use $P$ to construct a RES-CLF that will stabilize the dynamics at user defined rate of $\frac{1}{\epsilon}$. In particular, define

$$V_\epsilon(x) = x^T I_\epsilon P I_\epsilon x, \text{with} \quad I_\epsilon = \text{diag}(\frac{1}{\epsilon}, I)$$

wherein it follows that
\[
\dot{V}_\epsilon(x) = L_F V_\epsilon(x) + L_G V_\epsilon(x) \mu
\]  
(6.21)

with

\[
L_F V_\epsilon(x) = x^T (F^T P_\epsilon + P_\epsilon F) x
\]
(6.22)
\[
L_G V_\epsilon(x) = 2 x^T P_\epsilon G
\]

It follows that, in this case, Equation 6.7 takes the form

\[
K_\epsilon(x) = \{ \mu \in \mathbb{R}^n : L_F V_\epsilon(x) + L_G V_\epsilon(x) \mu + \frac{1}{\epsilon} \gamma V_\epsilon(x) \leq 0 \} 
\]
(6.23)

For the dynamics given in Equation 6.18 and for \( P_\epsilon \) defined in Equation 6.20, \( V(x) = x^T P_\epsilon x \) is a RES-CLF with

\[
c_1 = \lambda_{\text{min}}(P), \quad c_2 = \lambda_{\text{max}}(P), \quad c_3 = \gamma = \frac{\lambda_{\text{min}}(Q)}{\lambda_{\text{max}}(P)}
\]
(6.24)

This can be seen by noting that from Equation 6.19 and the form of \( F \) and \( G \) that \( P_\epsilon \) solves the RES-CARE

\[
F^T P_\epsilon + P_\epsilon F - \frac{1}{\epsilon} P_\epsilon G G^T P_\epsilon + \frac{1}{\epsilon} I_\epsilon Q I_\epsilon = 0
\]
(6.25)

noting that \( \gamma P_\epsilon \leq I_\epsilon Q I_\epsilon \) and therefore

\[
\inf_{\mu} \left[ L_F V_\epsilon(x) + L_G V_\epsilon(x) \mu + \frac{1}{\epsilon} \gamma V_\epsilon(x) \right] \leq \inf_{\mu} \left[ x^T P_\epsilon G (\frac{1}{\epsilon} - G^T P_\epsilon x + 2\mu) \right] \leq 0
\]
(6.26)

These facts allow the following specific bounds on the convergence of \( y \).
Lemma 6.1

For the output dynamics given by Equation 6.15 and the RES-CLF $V_\epsilon = x^T P_\epsilon x$, with $P_\epsilon$ a solution to the RES-CARE in Equation 6.25, for any control law

$$u_\epsilon(x) = A(x)^{-1}(-L_f^2(x) + \mu_\epsilon(x))$$

with $\mu_\epsilon(x) \in K_\epsilon(x)$ Lipschitz continuous, it follows that

$$\left\| y(t) \right\| \leq \frac{1}{\epsilon} \sqrt{\frac{\lambda_{max}(P)}{\lambda_{min}(P)}} e^{\frac{\lambda_{min}(Q)}{2 \lambda_{max}(P)}} \left\| y(0) \right\|$$

Thus the output dynamics are exponentially stable at the origin.

6.3.4 Relationship with Time-Scaling

The original development of RES-CLFs was motivated by time-scaling based upon $\epsilon$ [38]. This original motivation can be revisited in the context of RES-CLFs to establish their relationship with ES-CLFs. For $\mu_\epsilon \in K_\epsilon(x)$, since $V_\epsilon$ is a RES-CLF it implies that

$$\epsilon \dot{V}_\epsilon(x, \mu_\epsilon(x)) \leq -\gamma V_\epsilon(x) \quad (6.27)$$

Therefore, choosing the time-scaling factor $\tau = \frac{1}{\epsilon} t$ we have

$$\frac{d}{d\tau} V_\epsilon(x(\tau), \mu_\epsilon(x(\tau))) \leq -\gamma V_\epsilon(x(\tau)) \quad (6.28)$$

Defining the state variables

$$x_\epsilon := I_\epsilon x = \begin{bmatrix} \frac{1}{\epsilon} y \\ \dot{y} \end{bmatrix} \quad (6.29)$$

it follows that
\[ V_\epsilon(x) = x_\epsilon^T P x_\epsilon =: V(x_\epsilon) \] (6.30)

with \( P \) as the solution to the CARE in Equation 6.19. As a result, it can be verified that

\[ \frac{d}{d\tau} V(x_\epsilon(\tau), \epsilon \mu_\epsilon(x(\tau))) \leq -\gamma V(x_\epsilon(\tau)) \] (6.31)

where, again \( \mu_\epsilon(x(\tau)) \in K_\epsilon(x(\tau)) \).

This allows the establishment of the time-scaling property of the class of RES-CLFs being considered

**Proposition 6.1**

\( V_\epsilon(x) = x^T P \epsilon x \) is a RES-CLF for the control system in Equation 6.18 if and only if \( V(x_\epsilon) = x_\epsilon^T P x_\epsilon \) is an ES-CLF for the system

\[ \epsilon \dot{x}_\epsilon = F \dot{x}_\epsilon + G \epsilon \mu \] (6.32)

Moreover,

\[ \mu \in \bar{K}_\epsilon(x) \quad := \quad \{ \mu \in \mathbb{R}^n : x^T P \epsilon G \frac{1}{\epsilon} G^T P x_\epsilon \mu \leq 0 \} \]

\[ \Downarrow \]

\[ \epsilon \mu \in \bar{K}(x_\epsilon) \quad := \quad \{ \mu \in \mathbb{R}^n : x_\epsilon^T P G (G^T P x_\epsilon + 2 \mu) \leq 0 \} \] (6.33)

Note that the motivation for considering the set \( \bar{K}_\epsilon(x) \) is that by Equation 6.26

\[ \bar{K}_\epsilon(x) \subset K_\epsilon(x) \] (6.34)

Therefore, control values in \( \bar{K}_\epsilon(x) \) will still rapidly exponentially stabilize the system Equation 6.18.
6.4 Motor Dynamics and Control

Consider a single series elastic actuator with the dynamics

\[ J\ddot{\theta}_\Delta + b\dot{\theta}_\Delta + k\theta_\Delta = \tau_m \]  

(6.35)

This is a similar formulation of the fixed output position model from Equation 3.1.

Given a desired torque as delivered from a “high-level” control value, \( \tau_d \) (assumed to be a constant value in this section), the goal is to drive the measured torque \( \tau_s = k\theta_\Delta \), obtained from the spring to the desired torque. In other words, the control objective is

\[
\text{Drive}\quad e := k\theta_\Delta - \tau_d \to 0 \quad \text{exponentially.}
\]

(6.36)

We will achieve this convergence, in a torque optimal fashion at a desired exponential rate, with a RES-CLF based QP.

6.4.1 Error Dynamics

With the control objective in hand, the error dynamics can be written as

\[
\frac{J}{k}\ddot{e} + \frac{b}{k}\dot{e} + e + \tau_d = \tau_m
\]

(6.37)

where \( \dot{e} = k\dot{\theta}_\Delta \) and \( \ddot{e} = k\ddot{\theta}_\Delta \). Picking the control law

\[
\tau_m = \frac{J}{k}\mu + \frac{b}{k}\dot{e} + e + \tau_d
\]

(6.38)

for a secondary controller, \( \mu \) implies that \( \dot{e} = \mu \) as in Equation 6.17.
The goal is to design the controller $\mu$ that will drive the error to zero. In this case, the error dynamics can be written by $z = (e, \dot{e})^T \in \mathbb{R}^2$ which is the form given in Equation 6.18.

6.4.2 Control Law Construction

Utilizing the RES-CLF $V_\epsilon(z) = z^TP_\epsilon z$, this RES-CLF can be converted back to both the coordinates of the original system, $\theta_\Delta$ and $\dot{\theta}_\Delta$, along with converting $\mu$ back to a control law in $\tau_m$.

Since the error coordinates, $z = (e, \dot{e})^T$, are functions of $\theta_\Delta$, $\tau_d$ and $\dot{\theta}_\Delta$ the CLF conditions become

$$\psi_{0,\epsilon}(\theta_\Delta, \dot{\theta}_\Delta, \tau_d) + \psi_{1,\epsilon}^T(\theta_\Delta, \dot{\theta}_\Delta, \tau_d)\mu \leq 0 \quad (6.39)$$

for $\psi_{0,\epsilon}$ and $\psi_{1,\epsilon}$ as defined in Equation 6.10 with $L_F V_\epsilon(\theta_\Delta, \dot{\theta}_\Delta)$ and $L_F V_\epsilon(\theta_\Delta, \dot{\theta}_\Delta)$ given in Equation 6.22. Finally, the inequality is converted back to an inequality in $\tau_m$ by noting that

$$\mu = \frac{k}{J}(\tau_m - b\dot{\theta}_\Delta - k\theta_\Delta) \quad (6.40)$$

Substituting Equation 6.40 back into Equation 6.39 yields the QP for motor control

$$m_\epsilon(\theta_\Delta, \dot{\theta}_\Delta, \tau_d) = \arg\min_{\tau_m \in \mathbb{R}} \tau_m^2$$

s.t. $\psi_{0,\epsilon}(\theta_\Delta, \dot{\theta}_\Delta, \tau_d) + \frac{k}{J} \psi_{1,\epsilon}^T(\theta_\Delta, \dot{\theta}_\Delta, \tau_d)(\tau_m - b\dot{\theta}_\Delta - k\theta_\Delta) \leq 0 \quad (6.41)$

The QP can then be defined in closed form by defining
\[
\begin{align*}
\bar{\psi}_{0,\epsilon}(\theta_\Delta, \dot{\theta}_\Delta, \tau_d) &= \psi_{0,\epsilon}(\theta_\Delta, \dot{\theta}_\Delta, \tau_d) - \frac{k}{J} \psi_{1,\epsilon}(\theta_\Delta, \dot{\theta}_\Delta, \tau_d)(b\dot{\theta}_\Delta + k\theta_\Delta) \\
\bar{\psi}_{1,\epsilon}(\theta_\Delta, \dot{\theta}_\Delta, \tau_d) &= \frac{k}{J} \psi_{1,\epsilon}(\theta_\Delta, \dot{\theta}_\Delta, \tau_d)
\end{align*}
\] (6.42)

wherein it follows that the min-norm controller takes the form

\[
m_\epsilon(\theta_\Delta, \dot{\theta}_\Delta, \tau_d) = \begin{cases} 
-\frac{\bar{\psi}_{0,\epsilon}(\theta_\Delta, \dot{\theta}_\Delta, \tau_d)}{\bar{\psi}_{0,\epsilon}(\theta_\Delta, \dot{\theta}_\Delta, \tau_d)} & \text{if } \bar{\psi}_{0,\epsilon}(\theta_\Delta, \dot{\theta}_\Delta, \tau_d) > 0 \\
0 & \text{if } \bar{\psi}_{0,\epsilon}(\theta_\Delta, \dot{\theta}_\Delta, \tau_d) \leq 0
\end{cases}
\] (6.43)

since \(\bar{\psi}_{1,\epsilon}\) is scalar valued. This form of the controller is simple enough to be implemented on a microcontroller at the embedded level.

### 6.5 Robotic Systems with Series Elastic Actuators

Utilizing the methods developed thus far, the dynamics of the full-order SEA model can be effectively decoupled into rigid body system and isolated motor dynamics. This is achieved through the time-scale separation implied by the results of Section 6.3 and the observation that the scaling afforded by \(\epsilon\) is naturally amenable to singular perturbation theory.

#### 6.5.1 SEA System Model

The models presented in Equations 3.10 and 3.11 can be rewritten as

\[
\begin{align*}
M(q)\ddot{q} + H(q, \dot{q}) &= k(q_m - q) + b(q_m - q) \\
J\ddot{q}_m + b(q_m - \dot{q}) + k(q_m - q) &= u_m
\end{align*}
\] (6.44)

where \(q\) is the joint angle, \(q_m\) is the motor angle, \(u_m\) is the torque input to each motor, \(k \in \mathbb{R}^{n \times n}\) is a diagonal matrix of spring constants for each SEA, \(b \in \mathbb{R}^{n \times n}\) is a diagonal matrix of damping constants for each SEA, and \(J \in \mathbb{R}^{n \times n}\) is a diagonal
matrix of motor inertias. Since \( k \) and \( b \) are diagonal matrices, their inverse is written \("1/k"\) and \("1/b"\) so as to denote the component wise inverse of the diagonal elements.

The controller developed in Section 6.4 was specific to a model which was not influenced by the global dynamics of the robotic system, in which case \( \theta_\delta = (q_m)_i - (q)_i \) for \( i = 1, \ldots, n \). Similarly, the rigid system model in Equation 3.9 assumed no compliance in the system. The remainder of this section will be devoted to understanding the coupling effects of these previously decoupled systems in the context of RES-CLFs.

### 6.5.2 Application of RES-CLF Motor Controller

Let \( u_d(q, \dot{q}) \) be a controller that, when applied to the system

\[
M(q)\ddot{q} + H(q, \dot{q}) = u_d(q, \dot{q})
\]

(6.45)
drives \( y \to 0 \). For example, \( u_d(q, \dot{q}) \) can be taken to be any controller in \( K_\varepsilon(x) \) as developed in Section 6.3.

Returning to the SEA model, Equation 6.44 can be written as

\[
J\ddot{q}_\Delta + b\dot{q}_\Delta + kq_\Delta = u_m - J\ddot{q}
\]

(6.46)

where \( q_\Delta = q_m - q \). Applying the motor controller Equation 6.38 in its vector form with \( \tau_d = u_d(q, \dot{q}) \) and

\[
u_m = \frac{J}{k}\mu + \frac{b}{k}z_2 + z_1 + u_d(q, \dot{q})
\]

(6.47)

with \( J, k, \) and \( b \) replaced by corresponding diagonal matrices with the inertia, damping and spring constant for each motor yields
\[
\dot{z}_2 = \mu - k\ddot{q} \quad \Leftrightarrow \quad k\ddot{q}_\Delta = \mu - k\ddot{q} \quad (6.48)
\]

where \(z_q = kq_\Delta - u_d(q, \dot{q})\) and \(z_2 = k\ddot{q}_\Delta\). Here \(\mu\) is an auxiliary control input that can be chosen to stabilize \(z_1\) and \(z_2\) to zero. Importantly, the goal is to achieve \(z_1 = 0\) where it will follow that \(kq_\Delta = u_d(q, \dot{q})\) yielding the desired behavior in the full-order SEA model Equation 6.44.

Writing \(z = (z_1, z_2)\), we have the following representation of the motor dynamics

\[
\begin{align*}
\dot{z}_1 &= z_2 - \dot{u}_d(q, \dot{q}) \\
\dot{z}_2 &= \mu - k\ddot{q}
\end{align*}
\]

\(\Rightarrow\) \(\dot{z} = Fz + G\mu - \begin{bmatrix} \dot{u}_d(q, \dot{q}, z) \\ k\ddot{q} \end{bmatrix} \) \( (6.49)\)

with \(F\) and \(G\) given as in Equation 6.18. Motivated by the choice of coordinates in Equation 6.29, the coordinate transformation \(z \mapsto I_\epsilon z =: z_\epsilon\) yields the following control system for the motor dynamics

\[
\epsilon \dot{z}_\epsilon = Fz_\epsilon + G\epsilon \mu - \epsilon \begin{bmatrix} \dot{u}_d(q, \dot{q}, z) \\ k\ddot{q} \end{bmatrix} \quad (6.50)
\]

Applying a feedback control law \(\mu_\epsilon(z) \in \bar{K}_\epsilon(z)\) yields the dynamics

\[
\epsilon \dot{z}_\epsilon = Fz_\epsilon + G\epsilon \mu_\epsilon(z) - \epsilon \begin{bmatrix} \dot{u}_d(q, \dot{q}, z) \\ k\ddot{q} \end{bmatrix} \quad (6.51)
\]

By Equation 6.33 this can be equivalently stated as

\[
\epsilon \dot{z}_\epsilon = Fz_\epsilon + G\mu(z) - \epsilon \begin{bmatrix} \dot{u}_d(q, \dot{q}, z) \\ k\ddot{q} \end{bmatrix} \quad (6.52)
\]

for \(\mu(z_\epsilon) \in \bar{K}(z_\epsilon)\). In this case, the direct mapping between \(\bar{K}_\epsilon(z)\) and \(\bar{K}(z_\epsilon)\) established in Proposition 6.1 implies that, given \(\mu_\epsilon(z) \in \bar{K}_\epsilon(z)\), \(\mu(z_\epsilon) = \epsilon \mu_\epsilon(I_\epsilon^{-1}z_\epsilon)\).
6.5.3 Singular Perturbation Perspective

The dynamics given in Equation 6.52 coupled with Equation 6.44 can naturally be viewed as a multitime-scale system. As a result, this system of equations is naturally amendable to singular perturbation theory. In this case, due to the RES-CLF controller applied at the motor control level, the actuator dynamics of the SEA are viewed as the fast dynamics, while the dynamics of the robot are the slow dynamics. It can be shown that this time-scale separation allows for the approximation of Equation 6.52 and Equation 6.44 by Equation 6.45.

Lemma 6.2

For the system in Equation 6.52, if \( \epsilon = 0 \) then \( z = 0 \) and thus \( kq_\Delta = u_d(q, \dot{q}) \).

Therefore, the quasi-steady state system is given by

\[
M(\bar{q})\ddot{\bar{q}} + H(\bar{q}, \dot{\bar{q}}) = u_d(\bar{q}, \dot{\bar{q}})
\] (6.53)

with boundary layer system

\[
\frac{d\eta}{d\tau} = F\eta + G\mu(\eta)
\] (6.54)

for \( \mu(\eta) \in \bar{K}(\eta) \).

Therefore, the quasi-steady state dynamics of the system are just the rigid body dynamics for which \( u_d \) was originally designed to stabilize.

Utilizing the result of singular perturbation analysis applied to the SEA model, it can now be established that for an \( \epsilon \) sufficiently small, the dynamics of the SEA model can be approximated as the rigid model; this follows from Tikhonov’s theorem [36], [22] coupled with Lemma 6.1 and Lemma 6.2.

Theorem 6.1

Consider the model of a SEA system given in Equation 6.44. For
\begin{align*}
    u_d(q, \dot{q}) &= A(q, \dot{q})^{-1}(-L_f^2(q, \dot{q}) + \mu_\epsilon(x)) \quad \text{(6.55)} \\
    u_m(z, u_d(q, \dot{q})) &= \frac{J}{k} \mu_\epsilon(z) + \begin{bmatrix}
        b \\
        k
    \end{bmatrix} z + u_d(q, \dot{q}) \quad \text{(6.56)}
\end{align*}

with \( \mu_\epsilon(x) \in K_\epsilon(x) \) differentiable and \( \mu_\epsilon(z) \in \bar{K}_\epsilon(z) \) Lipschitz continuous, it follows that there exists an \( \epsilon^* > 0 \) such that

\[
    \left\| \begin{array}{c}
        y(t) \\
        \dot{y}(t)
    \end{array} \right\| \leq \frac{1}{\epsilon} \sqrt{\frac{\lambda_{\text{max}}(P)}{\lambda_{\text{min}}(P)}} e^{-\frac{1}{2 \lambda_{\text{max}}(Q)} t} \left\| \begin{array}{c}
        y(0) \\
        \dot{y}(0)
    \end{array} \right\| + O(\epsilon) \quad \text{(6.57)}
\]

for all \( 0 \leq \epsilon \leq \epsilon^* \) and \( P \) satisfying Equation 6.19.

### 6.5.4 Connections with Previous Results

The importance of Theorem 6.1 is that it establishes an entire class of embedded level controllers, defined by the set \( \bar{K}_\epsilon(z) \), that yield approximations of the SEA model by the rigid model. To provide a connection with previous results, the controller given in [21] provides a single example of a controller in this class. In particular, picking

\[
    \mu(z) = -\frac{1}{2 \epsilon} G^T P_\epsilon z = -\frac{1}{2} \left( \frac{I}{\epsilon} z_1 + \frac{\sqrt{3} I}{\epsilon} z_2 \right) \quad \text{(6.58)}
\]

it follows that \( \mu_\epsilon(z) \in \bar{K}_\epsilon(z) \). Note that the form of this controller is a result of the simple form of \( F \) and \( G \) in Equation 6.18, from which it can be verified that the solution to the CARE in Equation 6.19, with \( Q = I \), is given by

\[
    P = \begin{bmatrix}
        \sqrt{3} I & I \\
        I & \sqrt{3} I
    \end{bmatrix} \Rightarrow P_\epsilon = \frac{1}{\epsilon} \begin{bmatrix}
        \sqrt{3} I & \frac{1}{\epsilon} I \\
        \frac{1}{\epsilon} I & \sqrt{3} I
    \end{bmatrix} \quad \text{(6.59)}
\]
Utilizing the control law in Equation 6.58 in Equation 6.56 yields the final form of the motor level controller

\[ u_m(q, \dot{q}, q_m, \dot{q}_m) = \frac{J}{k2\epsilon^2} u_d(q, \dot{q}) + (k - \frac{J}{2\epsilon^2})q_\Delta + (b - \frac{\sqrt{3}J}{2\epsilon})\dot{q}_\Delta \]  

(6.60)

This control law is, conceptually, very similar to the controller presented in [21].

6.6 Simulation

6.6.1 Implementation

From a practical implementation perspective, it is useful to have a simple closed form expression for the control law Equation 6.43; This allows for the controller to run in a known amount of time, and therefore be executed at real-time at the embedded level. Because of the simple form of \( F \) and \( G \) in Equation 6.18 and the solution to the CARE in Equation 6.19 with \( Q = I \), can be determined in closed form. This allows for the direct calculation of \( V_\epsilon \) as

\[ V_\epsilon(\theta_\Delta, \dot{\theta}_\Delta, \tau_d) = \sqrt{3} \frac{1}{\epsilon^2} (\tau_d - k\theta_\Delta)^2 + 2 \frac{1}{\epsilon} k(-\tau_d + k\theta_\Delta)\dot{\theta}_\Delta + \sqrt{3}k^2\dot{\theta}_\Delta^2 \]  

(6.61)

With these constructions, it is easy to explicitly calculate \( L_F V_\epsilon \) and \( L_G V_\epsilon \) in Equation 6.22. This allows for the direct calculation of \( \psi_{0,\epsilon} \) and \( \psi_{1,\epsilon} \) in Equation 6.10 with \( \gamma = \frac{1}{1+\sqrt{3}} \) and \( V_\epsilon \) given in Equation 6.61, which in turn allows for the final closed form expression of \( \tilde{\psi}_{0,\epsilon} \) and \( \tilde{\psi}_{1,\epsilon} \) as given in Equation 6.42. This yields the final form of the controller as given in Equation 6.43.

It is important to note that in Equation 6.61, the desired torque is viewed as a constant. This simplification was chosen for initial implementation to observe exponential convergence of the step response. In order to track more complex desired
signals, the derivative of desired torque would necessarily be added to the controller; the ability of the torque controller to achieve accurate tracking of signals and disturbances in the experiments can be attributed to the fast control loop rate of the embedded system relative to the rate at which commands can be sent to the system. That is, in the experiments, the embedded system is able to converge to the desired torque value faster than the discrete signal is changing. The addition of the torque derivative signal to the controller, is left as future work.

6.6.2 Results

The controller in Equation 6.43, representing the closed form solution of Equation 6.41, was implemented in simulation and provided a step input to observe the behavior of the controller. Data regarding output torque, motor torque effort ($\tau_m$) and convergence conditions used in the min-norm controller were collected to demonstrate the behavior of the controller.

By simulating a step response, the nonsmooth behavior of the controller is observed. In Figure 6.1, the torque output is smooth and converges to the control objective. The motor control effort is nonsmooth and has several intervals where no motor torque is exerted. This is shown in Figure 6.2. These intervals correspond with the areas where the dynamics of the system achieve or exceed the desired convergence rate, as shown in Figure 6.3. This is the desired behavior, from Equation 6.43.

A practical area for concern of the controller are the large torques requested to achieve convergence at the beginning of the step function. In reality, an actuator can only provide finite amounts of torques. This has several repercussions on the actual implementation of such a controller. The first is that changes in torque would be restricted to a certain size such that the maximum actuator output would not be
Figure 6.1: A simulated series elastic actuator with initial conditions of $[\theta_\Delta, \dot{\theta}_\Delta] = [0, 0]$ and $\tau_d = 5$. The step response of the motor control is smooth and converges rapidly towards the desired value, in this case an output torque of 5 Nm. The output torque is shown in blue, with the desired torque denoted by a dashed line. The convergence rate is $\frac{1}{\epsilon} = 1100$.

exceeded. This would constrain the high level controller to be less aggressive, or limit stability to a smaller set of external disturbances. Reduction of $\epsilon$ is another option for avoiding this situation, but, again, this limits the performance of the high level controller and compromises the time-scaling proposed in this chapter. Saturation, and therefore the violation of convergence, can be accepted as part of the behavior of the system, but can affect the system in the same way.

6.7 Experimental Results

The motor level control law was implemented on hardware, and is shown in video at [48]. The controller is shown to accurately track a variety of torque signals, even in the presence of unknown disturbances.

The actuators are composed of a BLDC motor, a harmonic drive gearset, and a torsional spring. The QP was implemented on the embedded motor controller and
Figure 6.2: The motor torque output of the simulated series elastic actuator is nonsmooth. The first section with non-zero torque is where the actuator must exert torque in order to satisfy the convergence boundary condition. Once the dynamics of the actuator are meeting or exceeding the convergence rate, the actuator no longer outputs motor torque, seen from the second section. The second, negative motor torque again is the actuator responding to conditions meeting the convergence boundary, this time to keep the actuator from overshooting the desired torque beyond the convergence rate. These sections match up with Figure 6.3 where the measured convergence rate contacts the convergence boundary.

ran at a control loop rate of 5 kHz. A software limitation of 45 Nm of torque was enforced. This implementation took effect at the current control level and is a safety feature intended to prevent damage to the motor, gearset, and spring. This limitation can be seen in Figure 6.6 as saturation. In order to draw a closer comparison between the simulation results and the experimental results, the simulation was updated to include this saturation, which can also be seen in Figure 6.6.

For the first set of tests, the SEA output was locked to a grounding plate. The experimental hardware for this configuration is shown in Figure 6.4.

This provides the closest match to the fixed position model, given in Equation 6.35. With the output fixed, a step response for different values of $\epsilon$ was measured. The results for the output torque are shown in Figure 6.5. The corresponding motor
Figure 6.3: The measured convergence rate and convergence boundary of the simulated series elastic actuator. This plot clearly shows where the actuator must exert effort in order to satisfy convergence conditions. The effects are most clearly matched up with Figure 6.2 where the effort is non-zero when the convergence rate is along the convergence boundary and zero when the convergence rate is exceeding the convergence boundary.

The torques required to achieve the response are shown in Figure 6.6.

To further demonstrate the potential of the controller as a candidate for implementation in a SEA robot, other desired torque signals were tested under locked output, as well as free output tests. The controller was also subjected to human provided disturbances to test torque tracking capabilities. The free output and human disturbance experiments can be viewed in video here [48]. The results of these tests are important to show the controller is capable of tracking torque, and desired torques, that are nontrivial.

With the locked output still attached to the actuator, the controller was tested against time varying $\tau_d$ signals. A sinusoid, chirp signal, and step response were chosen to apply an overall test of frequency response and stability. The controller was able to track all signals and remain stable, shown in Figure 6.7, Figure 6.8, and Figure 6.9.
Figure 6.4: The experimental configuration for testing the controller under the locked output conditions. The motor controller was run at a rate of 5 kHz and is outlined in white. The SEA, a Valkyrie elbow actuator, was bolted to a grounding plate. It is shown outlined in yellow. A load cell, to verify torque output in addition to the spring measurement, is shown outlined in blue. The output is also bolted to the grounding plate.

The output was then attached to a 4.5 kg weight at a distance of 0.3 m from the actuators center of rotation. The results tracking a sinusoid with this output is shown in Figure 6.10. Human disturbance, and error tracking during the disturbances, are shown in Figure 6.11 and Figure 6.12.
Figure 6.5: Simulated data, denoted by the blue lines, corresponds with $\epsilon$ values of $\frac{1}{\epsilon} = 1100, 900, \text{ and } 700$ from left to right. Data collected from the experimental setup is denoted by red dots interpolated by straight lines between samples. The experimental data lines up closely to that of the simulated data.

Figure 6.6: Simulated motor torques are shown for the same values of $\epsilon$ from Figure 6.5. The flat, non-zero portions are areas where the motor torque was simulated to be saturated. Experimental motor torque was calculated and inferred by current measurements collected by the motor driver, and are shown by red dots with interpolated red lines between samples. Although these do not match up as cleanly as the results from Figure 6.5, they still demonstrate the desired behavior of the controller. That is, the output is nonsmooth and no output current is sent to the motor during periods where the output torque is still converging towards the desired value.
Figure 6.7: With fixed output, the controller tracked a sinusoidal $\tau_d$. This shows the controller is able to track time varying signals.

Figure 6.8: With fixed output, the controller was also tested against a chirp signal. This highlights the bandwidth of the controller, demonstrating the ability to track increasingly faster $\tau_d$. 

Figure 6.9: The controller was also tested against step responses. This test highlights the stability of the controller as a step signal contains high frequency content.

Figure 6.10: The controller was able to track a sinusoid with a free output - the change from locked output to free output was not modeled into the controller. This result shows promising application of the controller in SEA robots. Further tests with a high level CLF providing the $\tau_d$ are needed to verify this controller allows the rigid body dynamics to be assumed.
Figure 6.11: Human disturbance was applied during the tracking of the sinusoid. Some deviation can be seen from the desired signal. The controller remained stable throughout the testing and tracking error during this test is shown in Figure 6.12.

Figure 6.12: The error plot, $\tau - \tau_d$, highlighting the controllers performance. Throughout the testing, the controller was able to regulate torque within 0.2 Nm of the desired torque.
Chapter 7
Conclusion

The goal of this thesis was to present several models, an estimator, and torque controllers that are applicable to robots with SEAs and distributed control architectures. The performance of any control system is limited by the quality and delay of signals it operates on. The estimator in this thesis was developed to provide better output signals for SEAs as part of a larger kinematic chain. The estimator shows improved performance over a low pass filter. Future work for the SEA estimator include a better way to tune the system and a metric for establishing how signals from each actuator affect a controller operating on them. The passivity of the DOB torque controller was shown to be passive in the ideal implementation, as well as input-feedforward passive in a realistic implementation. The cost of making the realistic system passive was shown to be relatively small. Understanding of each of controller parameters affects the passivity of the system, as well as other methods of making the DOB torque controller passive are left for future work. A RESCLF embedded torque controller was implemented and shown to allow for a high level controller to assume rigid dynamics for a SEA robot. The controller was shown to converge at a rapid exponential rate and was accurately simulated in the presence of actuator saturation. In practice, this saturation will be present in all actuators and cause the actuator to violate the convergence criteria. Understanding the limits of this saturation and their effect on the assumption of rigid actuators from the high level controller, as well as the convergence rate of the actuator torque control can be further investigated.
Bibliography


