Brigham Young University
BYU ScholarsArchive

All Theses and Dissertations

2018-04-01

Adaptive Control for Inflatable Soft Robotic Manipulators with Unknown Payloads

Jonathan Spencer Terry
Brigham Young University

Follow this and additional works at: https://scholarsarchive.byu.edu/etd

Part of the Mechanical Engineering Commons

BYU ScholarsArchive Citation

Terry, Jonathan Spencer, "Adaptive Control for Inflatable Soft Robotic Manipulators with Unknown Payloads" (2018). *All Theses and Dissertations*. 6769.
https://scholarsarchive.byu.edu/etd/6769

This Thesis is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in All Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.

Adaptive Control for Inflatable Soft Robotic Manipulators with Unknown Payloads

Jonathan Spencer Terry

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Marc D. Killpack, Chair
Randal W. Beard
Mark B. Colton

Department of Mechanical Engineering

Brigham Young University

Copyright © 2018 Jonathan Spencer Terry

All Rights Reserved

ABSTRACT

Adaptive Control for Inflatable Soft Robotic Manipulators with Unknown Payloads

Jonathan Spencer Terry
Department of Mechanical Engineering, BYU
Master of Science

Soft robotic platforms are becoming increasingly popular as they are generally safer, lighter, and easier to manufacture than their more rigid, heavy, traditional counterparts. These soft platforms, while inherently safer, come with significant drawbacks. Their compliant components are more difficult to model, and their underdamped nature makes them difficult to control. Additionally, they are so lightweight that a payload of just a few pounds has a significant impact on the manipulator dynamics. This thesis presents novel methods for addressing these issues.

In previous research, Model Predictive Control has been demonstrably useful for joint angle control for these soft robots, using a rigid inverted pendulum model for each link. A model describing the dynamics of the entire arm would be more desirable, but with high Degrees of Freedom it is computationally expensive to optimize over such a complex model. This thesis presents a method for simplifying and linearizing the full-arm model (the Coupling-Torque method), and compares control performance when using this method of linearization against control performance for other linearization methods. The comparison shows the Coupling-Torque method yields good control performance for manipulators with seven or more Degrees of Freedom. The decoupled nature of the Coupling-Torque method also makes adaptive control, of the form described in this thesis, easier to implement.

The Coupling-Torque method improves performance when the dynamics are known, but when a payload of unknown mass is attached to the end effector it has a significant impact on the dynamics. Adaptive Control is needed at this point to compensate for the model's poor approximation of the system. This thesis presents a method of layering Model Reference Adaptive Control in concert with Model Predictive Control that improves control performance in this scenario. The adaptive controller modifies dynamic parameters, which are then delivered to the optimizer, which then returns inputs for the system that take all of this information into account. This method has been shown to reduce step input tracking error by 50% when implemented on the soft robot.

Keywords: inflatable, soft robot, optimal control, adaptive control, pneumatic actuation, model predictive control

ACKNOWLEDGMENTS

My thanks go first and foremost to my wife, Lauren, who has given me an incredible amount of support as I've pursued my graduate studies. She has made many sacrifices and given me essential encouragement throughout the entire process. She's helped foster in our children an enthusiastic attitude about my schooling, who excitedly look forward to the day that they can celebrate the completion of this thesis. I could not have done this work without Lauren's help and support.

My gratitude also goes to BYU for providing the graduate school and necessary facilities for my studies. This also includes fellow members of the Robotics and Dynamics lab, who have assisted in working through ideas, troubleshooting software and hardware problems, and generally made the research an enjoyable process. Thanks also goes to the members of my graduate committee, including Mark Colton and Randy Beard.

Lastly, I'm grateful to Marc Killpack, my advisor. He took a risk in taking me on as a graduate student, considering my undergraduate degree was in a different engineering discipline and that I hadn't even been in school for several years. Marc has challenged me to more rigorous scholarship, while also fostering a relaxed and enjoyable atmosphere in his lab. I will always look back fondly on my time spent working as a graduate student under Marc's advisement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1. INTRODUCTION

There is a great amount of interest in developing robots that can effectively and safely interact with humans and our environments. This overall goal presents a number of smaller problems that first need to be addressed. For example, traditional robots, which are heavy, rigid, and actuated with motors, are inherently unsafe because of their high reflected inertia due to gearing. Another problem is that the environments that humans typically navigate are cluttered and difficult to model. A heavy traditional robot that makes unintentional contact with the environment will likely cause damage to the environment, itself, or both. Two general approaches to this problem are to either make the rigid robot more safe or to completely redesign the robot itself and the materials it is made out of to be more inherently safe.

In order to make a rigid robot more safe, several strategies have been proposed. Among the more straightforward ideas is simply constructing the robot out of lightweight material [1–3]. Developing responsive collision detection algorithms [4–6] as well as compliant actuation technology [7–9] will also greatly improve safety factors.

Another approach is an inflatable, pneumatically actuated soft robot, which is initially designed to be inherently safe. This approach, however, brings additional problems with it. Being compliant and underdamped, it is also more difficult to control with reasonable accuracy. For example, attaining dynamic performance metrics (such as rise time, overshoot and settling time) comparable to that of traditional arms is difficult, if not impossible. Also, deflating and reinflating makes repeatability between experiments difficult to achieve, due to components settling in different positions with each inflation. Part of this problem is due to the difficulty in modeling the robot manipulator itself. Standard dynamic models for a robotic manipulator assume rigid links and pin joint rotation, both of which are not the case for these soft, fully inflatable platforms. Additionally, because the robot is lightweight, when it picks up an object of only a pound or two, the object significantly changes the dynamics of the arm, and may even cause the compliant links to bend more than usual.

1.1 Need for Soft Robots

The unique advantages of soft, inflatable robots over rigid robots for specific applications are what motivate this research. In recent years, studies have been done to evaluate and quantify the dangers of unintentional impacts from rigid robots. In [10] the Head Injury Criterion (HIC) is identified as a limit for serious head injury upon robot impact, where inertia of the robot is a major driving contributor. This indicates with all operating parameters constant, a reduction in robot inertia, such as for the fabric-based soft robot joint in this thesis, directly reduces the HIC rating. Also in other research, [11] and [12], it has been shown that contact forces from inflatable links can be controlled with cable driven actuators. The ability to control forces can significantly reduce injuries or damage from unintentional impact.

One reason why robots with rigid joints and actuation have been used for so long is that low level system dynamics and sensing methods are well developed. With a soft robotic system, modeling the dynamics for model-based control is not straightforward and for each design new methods must be developed or learned through data to describe how the system behaves when actuated. An example of this is in [13–15] where it was found that planning is viable for elastomer actuators, when learned for each platform, using dynamic and constant curvature kinematic models. Designs for rotary, fabric-based, pneumatically actuated joints were proposed in [16] and the actuators used in this thesis are based on that design. One purpose of this thesis is to develop a model and more general method of control for this type of system. The lack of literature on the control of inflatable structures where there is a wide range of applications suggests a viable new area for research.

Potential applications for this research include health care, living assistance, space exploration, search and rescue, orthotics, and prosthetics [17–19]. As shown in [17], a variety of soft grippers, actuators, and structures have been designed and built with the intent of moving robotics into environments that are not currently feasible. Not all of the designs use gas as a working fluid or have inflatable components, but all are seeking an alternative approach to a rigid structural design.

1.2 Thesis Overview

The overall intent of the research presented in this thesis is to improve joint angle control performance for a soft, fully inflatable, pneumatically actuated robot. This is accomplished by the development of a linearization method for a model to be used in Model Predictive Control (MPC), and the implementation of Model Reference Adaptive Control (MRAC) that is incorporated into MPC.

The model linearization method, which we call the Coupling-Torque method (Chapter 3), enables us to decouple the dynamics of each joint in a robot manipulator and run independent controllers for each joint, as opposed to modeling and controlling the entire manipulator simultaneously. This makes model-based control more tractable to higher Degrees of Freedom.

Another advantage of the Coupling-Torque method is that it provides a simpler model and control scheme into which we can embed adaptive control. Adaptive control is needed for handling changes in the manipulator dynamics due to circumstances such as deflation/reinflation, fabric wear, and picking up an object of relatively significant mass. The adaptive control strategy presented in Chapter 4 is based on MRAC, which we incorporate into MPC. This gives us the advantage of utilizing the work already developed for controlling these robotic platforms. This strategy also allows the MPC optimization to determine the best control inputs to the manipulator, given the information it receives from the adaptive laws as well as the measured data from the physical platform.

1.3 Related Work

1.3.1 Soft Robot Design and Modeling

Several different strategies have been used to design and model soft robots. Work in [20] characterized different models for traditional rigid servo-pneumatic actuators which made different constant temperature assumptions. In [21] these assumptions were used to control force and stiffness for a linear pneumatic actuator. A different actuation approach utilizing McKibben Artificial Muscles, which emulate human muscles, was proposed in [22] and [23]. More recently, work has been completed on rotary elastic chamber actuators such as in [24] and [19], where two antagonistic bellows imparted torque on an armature rotating about a rigid rotary joint. In all of these cases the safety and robustness introduced by compliant joints was reduced by higher-inertia motors or rigid links between the joints. While these models were both useful and accurate for these rigid systems, they did not carry over well to systems with significant flexibility or non-rigid structure. The actuators used in this thesis are based on the design in [16].

Other approaches to soft robotics include small-scale, non-manipulator robots. These locomotive robots are often designed with the intent to imitate naturally occurring systems such as worms, octopi, or insects. One example is a four-legged multigate robot, made of elastomeric polymers and actuated with air [25]. Another robot, constructed of similar materials, uses the energy from small explosive chemical reactions to jump [26]. Another locomotive example has an "exoskeleton" and six legs like an insect, with actuation chambers at the joints, using air to drive locomotion [27]. These platforms, as currently constructed, are not capable of closed-loop control or manipulation. In contrast, we are interested especially in soft robots that are capable of payload manipulation and of utilizing sophisticated controls methods. Even so, it is possible that the methodology and strategies presented in this thesis may be applicable to these other types of soft robots as the researchers begin to incorporate closed-loop control.

1.3.2 Model Linearization

In Chapter 3, we compare three methods for linearizing a model for use in MPC. Little work has be done with the intent of making similar comparisons for the sake of control performance. Because of this, the related work for this section focuses on linearization methods used for robot manipulator control, especially in the context of MPC.

Hedjar et al. employed a Taylor Series approximation to simplify robot manipulator dynamics [28]. Various other simplification methods have been based on decoupling the dynamics [29], [30], as the Coupling-Torque method (presented in Chapter 3) does. Duchaine et al. have used a simplified predictive control, derived from velocity control schemes [31]. Lastly, feedback linearization is a commonly used method in many applications, with or without MPC [32–36]. However, using feedback linearization with MPC would require modifying the available torque as a constraint at each time step and it does not respond well to unmodeled disturbances or contacts. This is because the disturbances are not part of the model inversion, and so generally result in nonintuitive behavior. This defeats one of the main benefits of soft robots, which is that they are better able to safely navigate in an unmodeled environment because they can more safely make contact with it.

A strategy that has some similarity to the work presented in Chapter 3 is seen in work by Nakanishi et al. [37], where they compare various robotic manipulator controllers. Their paper compares operational space controllers, which are controllers that attempt to focus control on task variables while keeping redundant DoF as compliant as possible. Nakanishi et al. analyze different approaches to this control method on a theoretical and empirical basis, addressing formulations at the velocity, acceleration, and force levels. In the same paper, they also present a novel estimation algorithm for rigid body dynamics.

Most similar to our work on making MPC more tractable for control of nonlinear dynamical systems is work by Ling et al. [38], developed a segmented version of MPC, which they refer to as Multiplexed MPC (MMPC). MMPC is similar to the Coupling-Torque method used in this work, where both serve as a method for reducing complexity for computation speed. However the Coupling-Torque method shown in this work is shown to not only reduce computational complexity but is also a method for linearizing nonlinear systems (see section 3.2.3).

1.3.3 Adaptive Control

The adaptive control work in this thesis needs to be suitable for a robot manipulator, and it is desirable for it to work in conjunction with MPC. The literature we reviewed typically does not address both of these topics, so we first discuss related adaptive control work that focuses on MPC, and then for robot manipulators.

Different forms of adaptive control methods have been developed for use with MPC both on a general theoretical basis and also for specific platforms. Chowdary et al. [39] used a form of MRAC with MPC for aerospace applications with fast dynamics. Fukushima et al. [40] developed a general method for on-line system identification to work in conjunction with MPC. Adetola et al. [41] and Chen et al. [42] both developed Lipshitz-based adaptive MPC methods.

For platforms with robot manipulators, a typical adaptive control approach is on-line parameter estimation using a regressor matrix (see [23, 43–45]). MRAC has been used for individual soft actuators, with the intent of using a uniform control law despite individual actuator characteristics [46], as well as for robot manipulator joint angle control [47]. Tahia et al. [45] and Tonietti et al. [23], cited above, both worked on manipulators with antagonistic actuators, similar to the manipulators used in this thesis. Their manipulators, however, had rigid links with compliant actuation, whereas the platforms used in this thesis have compliant links as well as compliant actuators. Medrano-Cerda et al. [48] also did their work on manipulators with antagonistic actuators with rigid links, but the control in their work used an adaptive pole placement technique. Link compliance requires extra consideration in control development, especially when loads are added to the manipulator. This is a significant motivation for the research presented in this thesis.

A previous graduate student in the Robotics and Dynamics Lab developed an adaptive control method for King Louie [49]. This algorithm adapted joint stiffness when structural leaks were detected in order to prolong the robot's usefulness. It did not, however, address adapting for unknown loads.

The adaptive control research presented as a contribution in this thesis is unique because of the strategy used for combining MRAC with MPC. It is also, to our knowledge, the first attempt at developing adaptive control methods with the intent of improving control performance for a soft robot carrying an unknown load.

1.4 Contributions

This thesis documents the author's contributions to this body of research in generally improving joint angle control for underdamped robots. The following is a list of major achievements described within this thesis:

- Development of the Coupling-Torque method for linearizing (Section 3.2.3).
- Comparison of MPC for underdamped robots using three different methods of linearization (Sections 3.3 - 3.5).
- Formulation of MPC for robots with significant compliance at the joints using two methods of linearization (Sections 3.4.1 - 3.4.2).
- Implementation of Coupling-Torque method on two real robots (Section 3.6).
- Demonstrating instances where better knowledge of model parameters does not necessarily yield better control performance (Section 4.1).

- Development of a form of Model Reference Adaptive Control which couples with MPC (MRAC/MPC) for soft robots with antagonistic actuators and compliant links (Section 4.2.1).
- Implementation of MRAC/MPC on a soft inflatable platform, demonstrating improvement of control performance with unknown loads (Section 4.2.3).

The methodologies proposed in this thesis improve upon the work done in [50, 51], which utilized the same platforms as in this thesis. This research moves soft robot technology another step closer to implementation and usefulness in the real world, where these kinds of robots can safely interact with an unmodeled environment.

CHAPTER 2. BACKGROUND

This chapter provides background information with which one must be familiar in order to understand the work presented in this thesis. It contains a description of the robotic platforms used, a discussion on Model Predictive Control, and on Model Reference Adaptive Control.

2.1 Platform Description

This section describes two soft robot platforms, as well as one rigid robot platform, which were used for experiments in the research presented in this thesis.

2.1.1 Inflatable Platforms

The inflatable platforms used for this research are a single Degree of Freedom (DoF) joint called the Grub (Figure 2.1) and a ten DoF humanoid robot called King Louie (Figure 2.2). Both were developed and built by Pneubotics, an affiliate of Otherlab. Besides internal electronics such as IMU and pressure sensors, these platforms are entirely made of ballistic nylon fabric with internal bladders to prevent air leakage. The structure of these robots comes from an inflatable bladder which is pressurized to between 2-3 PSI gauge. At each joint, there are antagonistic actuators which can be filled to pressures between 0-20 PSI gauge. These actuators are similar to the designs mentioned in [16].

Air flow is controlled from the pressure source and vented to atmosphere through Enfield LS-v25 five port spool valves. For the purpose of the thesis, only one output port of the Enfield valves is used, making these valves act as three port spool proportional flow valves. As seen in Figure 2.3, each actuator or bladder has an individual valve for control, while sharing the same pressure source.

Communication between the sensors and controller is achieved using ROS (Robot Operating System) operating in nonreal time on an Ubuntu workstation as seen in Figure 2.4. Valves are

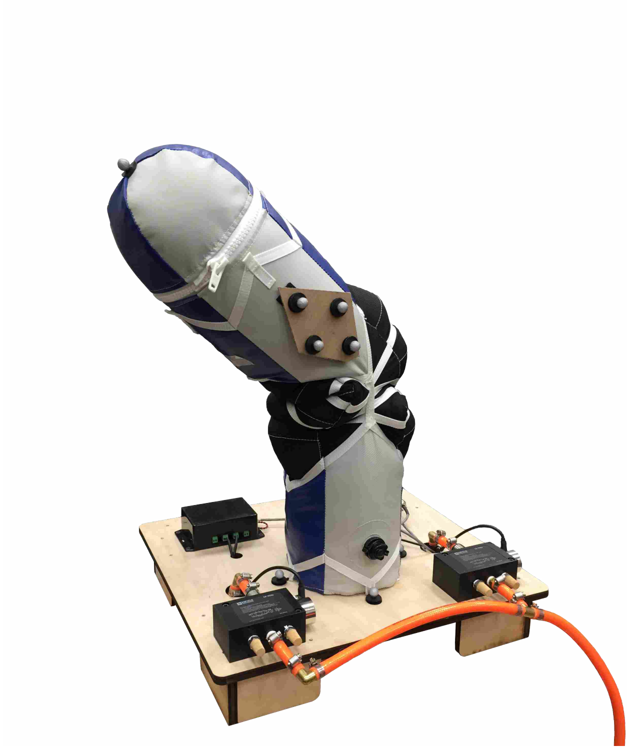


Figure 2.1: The single Degree of Freedom inflatable platform called the Grub.

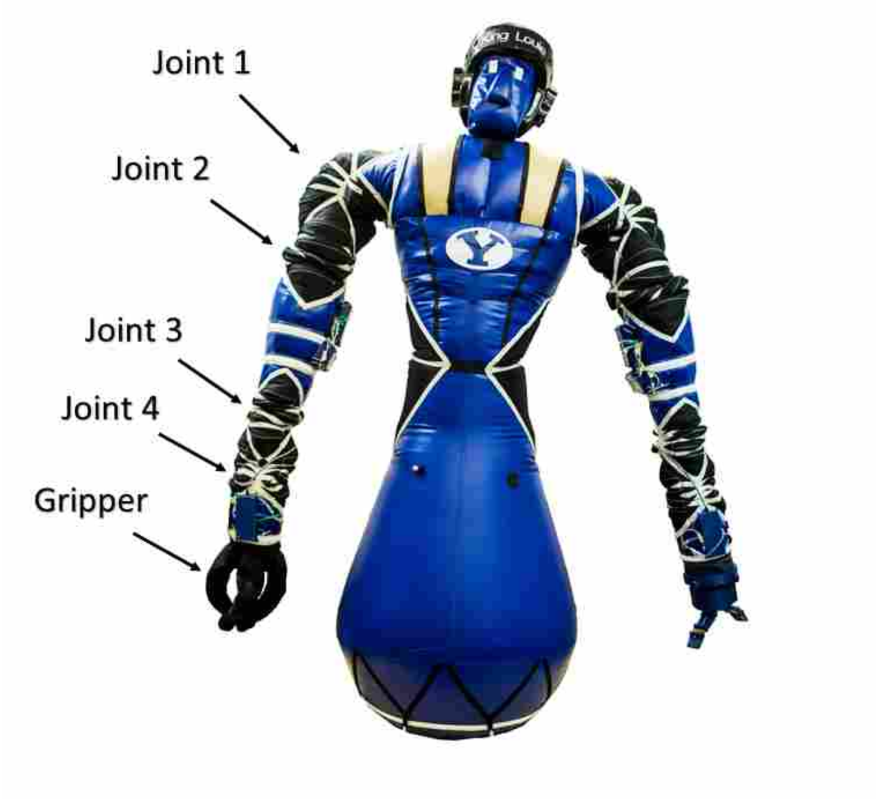


Figure 2.2: Inflatable humanoid robot named King Louie.

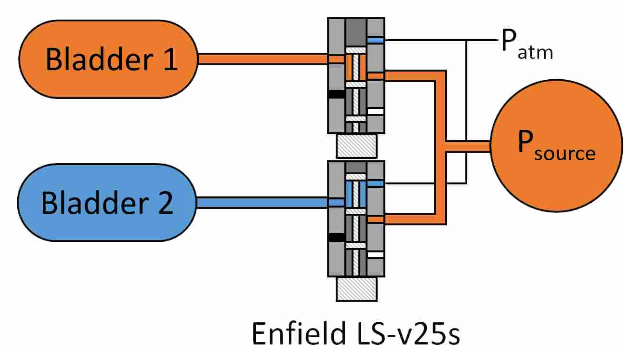


Figure 2.3: Representative figure of valve and bladder configuration.

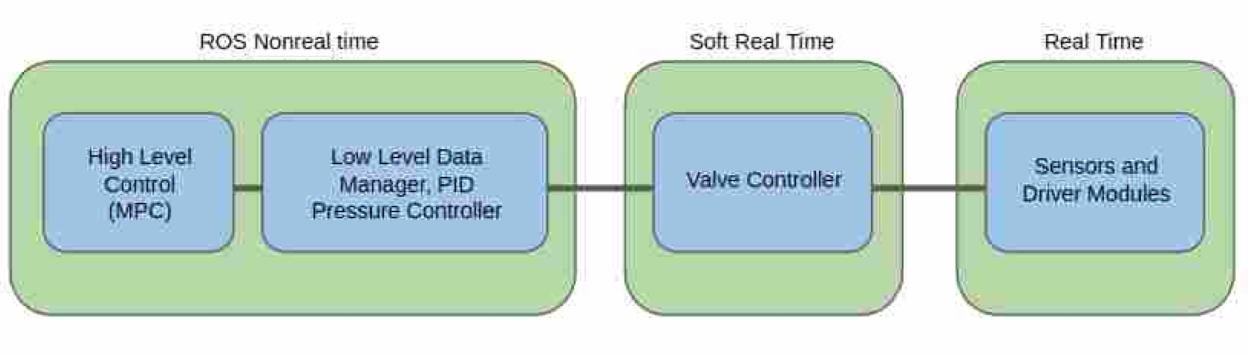


Figure 2.4: System chart showing nonreal time, soft real time, and real time operating system sections

controlled from a low level ROS node with a high thread priority effectively making the process soft real time. Sensors and driver modules are run in real time from embedded hardware. Pressure values are read and published at approximately 1 kHz. The pressure sensors are Honeywell HSC-DRNN100PGSA3 with a 0 to 100 PSI range. Joint angles for the Grub are determined using IMU (inertial measurement unit) located on the link. Joint angles are measured on King Louie by using HTC Vive Trackers. The HTC Vive system uses two emitter/sensor Lighthouses to determine the orientation of each Tracker. Trackers were placed on links of King Louie's arms and used to back out each joint angle. Pressure for each actuator is controlled by an underlying PID controller also operating at approximately 1 kHz. Desired pressures are published over ROS at the rate the controller is operating and the valves are actuated to achieve commanded pressure values.

2.1.2 Baxter

Baxter is a rigid humanoid robot with 7 DoF in each arm (Fig 2.5). Each arms' passive compliance comes from its series elastic actuators, which have a torsional spring between the actuator and the link load. We increase the compliance of the arms by implementing a low level, low gain, joint impedance controller which introduces active compliance at the joints. The impedance controller takes the form of Eq 2.1. Not only does Eq 2.1 introduce added compliance, but it helps to decouple the inertia of each link from the more distal links in the event of unexpected contact. Desired torque commands were communicated to Baxter using ROS.

$$\tau = \mathbf{K_p}(\mathbf{q_{des}} - \mathbf{q}) - \mathbf{K_d}\dot{\mathbf{q}} \tag{2.1}$$

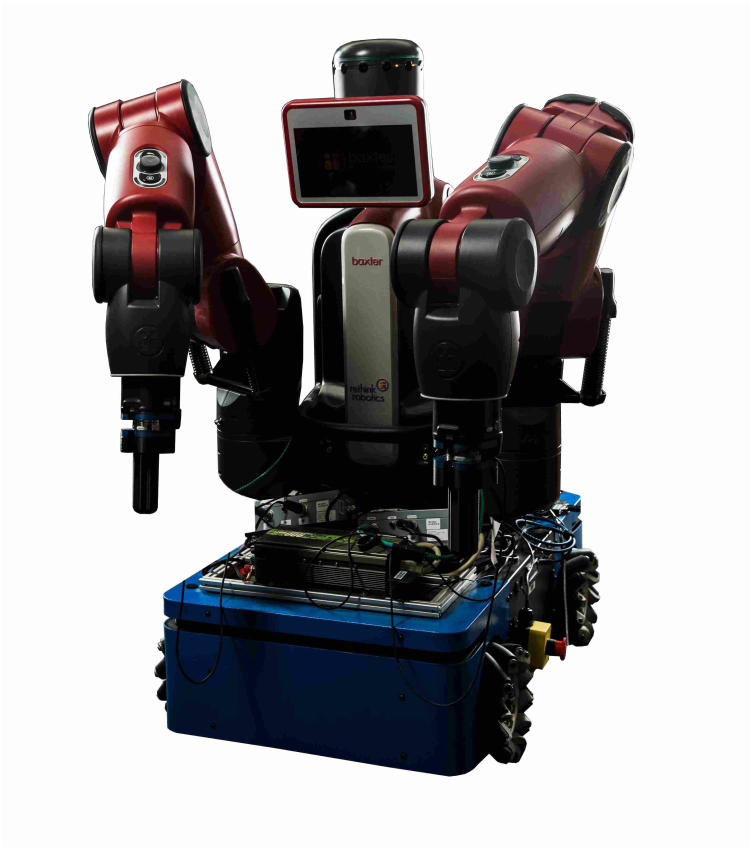


Figure 2.5: Baxter humanoid robot from Rethink Robotics.

2.2 Model Predictive Control

Model Predictive Control (MPC) has been used to successfully control systems in many different industrial applications, particularly those with slow, complex dynamics such as chemical plants [52]. More recently, as computer processing and efficient optimization methods have improved [53–56], MPC has been demonstrably useful to control systems with fast dynamics, such as robot manipulators [32, 57–63], and UAV research [64–66]. The effectiveness of MPC for the platforms used for this thesis was demonstrated in [36, 50, 51], where MPC performance in joint angle control was compared to LQR and input shaping control methods, amongst others. Because of this, MPC is used for joint angle control in this thesis.

MPC allows for the minimization of configurable costs across a discrete finite time horizon given specific constraints. This minimization is achieved through an optimization solver constrained by the system dynamics. MPC not only allows for model-based control but also the consideration of real-world constraints such as joint limits, pneumatic valve actuator restrictions, and hardware pressure limits.

A simple MPC setup, with cost function and constraints, that could be used for a platform like King Louie is shown in Eqs 2.2 - 2.5.

$$\underset{p_{des}^{+},p_{des}^{-}}{\arg\min} \sum_{k=0}^{T_c} \left(\left\| q_{goal} - q_k \right\|_{\alpha}^{2} \right) \tag{2.2}$$

$s.t.$

$$\begin{aligned} &\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k, \forall k = 0 \ldots T_c \\ &p_{min} \leq p_{des,k}^{+,-} \leq p_{max}, \forall k = 0 \ldots T_c \\ &q_{min} \leq q_{goal} \leq q_{max}, \forall k = 0 \ldots T_c \end{aligned} \tag{2.3}$$

where

$$\mathbf{x} = \begin{bmatrix} \dot{q} & q & p^{+} & p^{-} \end{bmatrix}^{T} \tag{2.4}$$

$$\mathbf{u} = \begin{bmatrix} p_{des}^{+} & p_{des}^{-} \end{bmatrix}^{T} \tag{2.5}$$

and q is the joint angle, T_c is the length of the optimization horizon, and p is pressure. The $+$ and $-$ superscripts on pressure terms indicate pressures that results in positive and negative changes in q, respectively. Eq 2.2 instructs the optimizer to minimize the difference between the current joint angle and the goal by manipulating the desired pressures. The optimizer's ability to do so is constrained by Eq 2.3, which consists of the dynamic model and bounds on pressure and joint angle. A diagram of this control scheme is shown in Fig 2.6.

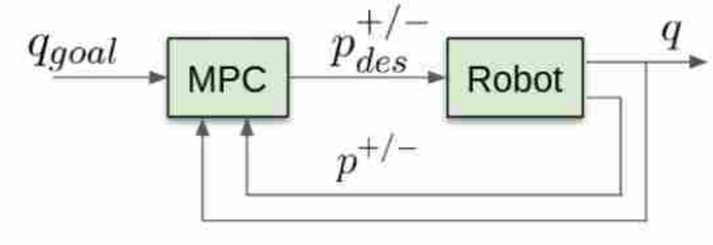


Figure 2.6: Generic MPC control diagram for King Louie.

At each time step, the MPC optimization solver takes in the current states and solves a convex cost function subject to the model dynamics and other specified constraints. For the purpose of this work, discretized, linearized dynamics were used within the solver to describe the plant. Model dynamics in this work were assumed linear and time-invariant as they were kept constant over the predictive horizon, despite being updated at every controller time step. Using plant dynamics, MPC predicts the states over a specific time horizon by varying the inputs, solving for an array of inputs that minimizes the specified cost function. Once a solution is found, the first time step of the solution is applied to the system and newly measured states are fed into the solver and the problem is solved again at the next time step.

An efficient solver for the MPC problems described in this work (see Sections 3.4.1 and 3.4.2) was generated using CVXGEN [67], a web-based tool for developing convex optimization solvers.

2.3 Model Reference Adaptive Control

Model Reference Adaptive Control (MRAC) is an adaptive strategy that modifies control inputs based on control performance feedback. The goal is to create a closed-loop controller that modifies parameters in response to system error in tracking performance. A reference trajectory is established, which is based on a model of the system, and system performance is compared to this reference trajectory.

The MRAC strategy presented by Lavretsky et al. [68] uses the modified model parameters to directly adjust the control input to the system. A simple version of this method, upon which we based the work in Chapter 4, is seen in Eqs 2.6 - 2.9.

$$\dot{q}_{ref} = \alpha(q_{goal} - q_{ref}) \tag{2.6}$$

$$\dot{q}_{err} = q_{ref} - q \tag{2.7}$$

$$\dot{\theta} = \Gamma\Phi q_{err} \tag{2.8}$$

$$u = u_c - \theta\Phi \tag{2.9}$$

where q_{ref} is a reference trajectory, Φ is the regressor, whose elements are the dynamic model parameters to be adapted, θ contains the adaptive parameters that modify the elements of Φ, u_c is the plant input as given by the nominal controller, u is the input modified by MRAC and sent to the plant, and α and Γ are gains. The purpose of q_{ref} is primarily to smooth step changes in q_{goal}, which otherwise can cause instability in θ. A diagram of this control scheme is shown in Fig 2.7, where the MRAC box represents Eqs 2.6 - 2.8 (after $\dot{\theta}$ has been integrated), the Nominal Controller box refers to some control scheme such as PID, and the Plant box represents the platform in use, such as a robot.

2.4 Conclusion

In this chapter, we have provided some context for the research presented in the rest of this thesis. The platforms used in this thesis (the Grub, King Louie, and Baxter) were introduced. We also gave a brief overview of Model Predictive Control and Model Reference Adaptive Control, which are used in the research presented in the following chapters.

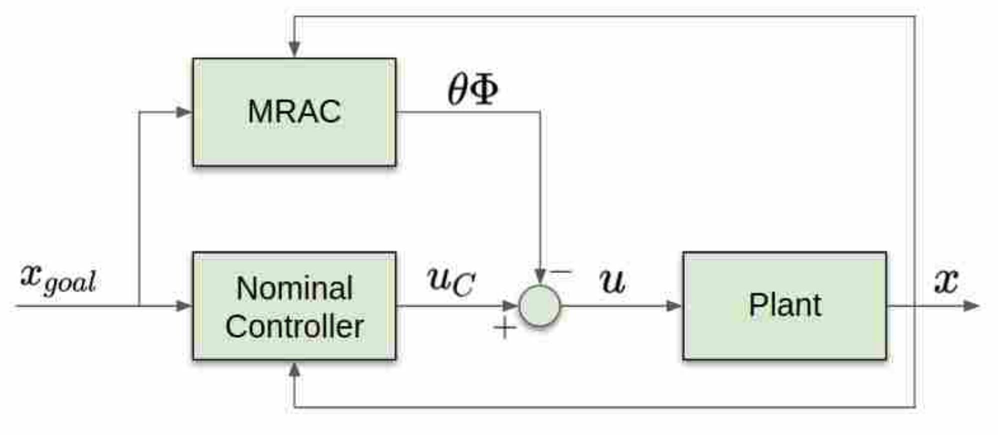


Figure 2.7: Generic MRAC control flow diagram.

CHAPTER 3. COUPLING-TORQUE LINEARIZATION

Model Predictive Control (MPC) requires a mathematical model of the system it is controlling, which model is used as an optimization constraint. When modeling a robot manipulator, the size of the model grows with each additional Degree of Freedom (DoF). For example, a Baxter manipulator has seven DoF and is modeled with two states per DoF. This results in a total of 14 states per arm, and a state-space **A** matrix of size 14 × 14. King Louie, with four DoF and four states per DoF, has a total of 16 states per arm, and a 16 × 16 **A** matrix (the gas dynamics introduce additional states for King Louie). This is a large space over which the optimizer must search for solutions, and the controller must run at fast enough rates to maintain good control performance if our objective is low-level position and pressure control.

One solution to this problem is to simplify the optimization search space so that control can be run at a fast enough rate. This chapter discusses the development of a method for linearizing the dynamic model of a robot manipulator, specifically when that model is used as a constraint for an MPC optimization. The result is a simplified model that decouples the links in the robot arm model. It retains enough model information to obtain good control performance, while remaining simple enough for control to run at sufficiently high rates.

A simplified model should also lend itself to a more straightforward adaptive control algorithm. The linearized model developed in this chapter is used in the adaptive controller discussed in Chapter 4.

3.1 Models

The models used in this work are all based on the standard equations of motion of a serial manipulator (see [69]):

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{F}(\dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) + \mathbf{J}(\mathbf{q})^{\mathbf{T}}\mathbf{f} = \tau \tag{3.1}$$

where $\mathbf{M}(\mathbf{q})$ is the mass matrix, $\mathbf{q}$, $\dot{\mathbf{q}}$, $\ddot{\mathbf{q}}$ are vectors of joint angles, velocities, and accelerations respectively, $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})$ is the Coriolis matrix, $\mathbf{F}(\dot{\mathbf{q}})$ is friction, $\mathbf{G}(\mathbf{q})$ is torque due to gravity, $\mathbf{J}(\mathbf{q})^{\mathbf{T}}\mathbf{f}$ is torque due to contact forces, and τ is the applied joint torques. Note that in this thesis all matrices are upper case bold, all vectors are lowercase bold and all constants are lowercase non-bold. We next treat the specifics of the dynamics for each robotic platform that we consider (either in simulation or real world) separately.

3.1.1 Baxter

In this work, for a Baxter 7-DoF humanoid robot (from Rethink Robotics), we assume that a low level controller accurately compensates for gravity, that friction is negligible (gearing friction is negligible because of the high-bandwidth joint torque controller based on feedback from the load-side series elastic actuators), and that the robot experiences no contact forces. Therefore we reduce Eq 3.1 to

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} = \tau \tag{3.2}$$

Eq 3.2 is nonlinear for any robot arm with rotary joints, and grows in complexity with the addition of each joint. For some of the nonlinear terms that we neglected in this case, they can easily be incorporated back into the Coupling-Torque method (see Section 3.2.3).

An impedance model is used to calculate torque from commanded joint angles ($\mathbf{q_{des}}$) with $\mathbf{K_p}$ and $\mathbf{K_d}$ as diagonal matrices containing proportional and derivative gains, and $\mathbf{q}$ and $\dot{\mathbf{q}}$ as the current joint angles and velocities. When controlling, the optimization for MPC will treat $\mathbf{q_{des}}$ as the input to manipulate, which acts as a virtual spring and damper pulling on the arm as follows:

$$\tau = \mathbf{K_p}(\mathbf{q_{des}} - \mathbf{q}) - \mathbf{K_d}\dot{\mathbf{q}} \tag{3.3}$$

For our experiments, we have set $\mathbf{K_p}$ and $\mathbf{K_d}$ to be very low, causing the arm to be intentionally underdamped. The motivation for this is twofold. First, underdamped dynamics are often more safe for scenarios where we expect the robot to interact with people or the environment. Second, many of the other robots we wish to control are pneumatically actuated soft robots that are inherently underdamped. Eq 3.3 is then substituted into Eq 3.2 to obtain Eq 3.4, which is the model we explore linearizing in this chapter.

$$\ddot{\mathbf{q}} = \mathbf{M}(\mathbf{q})^{-\mathbf{1}}[\mathbf{K_p}(\mathbf{q_{des}} - \mathbf{q}) - (\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{K_d})\dot{\mathbf{q}}] \tag{3.4}$$

The general state variable structure used in this work for Baxter is as follows:

$$\dot{\mathbf{x}} = \mathbf{A}(\mathbf{x})\mathbf{x} + \mathbf{B}(\mathbf{x})\mathbf{u} \tag{3.5}$$

where

$$\mathbf{x} = \begin{bmatrix} \dot{\mathbf{q}} & \mathbf{q} \end{bmatrix}^T \tag{3.6}$$

$$\mathbf{u} = \mathbf{q_{des}} \tag{3.7}$$

$$\mathbf{A}(\mathbf{x}) = \begin{bmatrix} -\mathbf{M}(\mathbf{q})^{-1}(\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{K_d}) & -\mathbf{M}(\mathbf{q})^{-1}\mathbf{K_p} \\ \mathbf{I} & \mathbf{0} \end{bmatrix} \tag{3.8}$$

$$\mathbf{B}(\mathbf{x}) = \begin{bmatrix} \mathbf{M}(\mathbf{q})^{-1}\mathbf{K_p} \\ \mathbf{0} \end{bmatrix} \tag{3.9}$$

3.1.2 King Louie

Each joint in King Louie has two antagonistic actuators that create torque as they are pressurized as shown in Figure 3.1. The states for this platform are not only the joint positions and velocities $(\dot{\mathbf{q}}, \mathbf{q})$ but also include the pressure in each of the actuators represented by $\mathbf{p}^+$ and $\mathbf{p}^-$, where $\mathbf{p}^+$ is a vector of all the pressures that cause positive actuation, referenced from each positive joint direction, and $\mathbf{p}^-$ is a vector of all pressures that cause negative actuation. All other coefficients or matrices that have the $+$ and $-$ notation correspond with actuation that produce either positive or negative torque as defined by the DH parameterization we use to approximate King Louie's kinematics. We start with Eq 3.2 and then add extra terms into the torque description as shown in the following equation:

$$\tau = \mathbf{K_p}\mathbf{q} - \mathbf{K_d}\dot{\mathbf{q}} + \tau^+(\mathbf{p}^+) - \tau^-(\mathbf{p}^-) + \mathbf{M}^{-1}(\mathbf{q})\mathbf{G}(\mathbf{q}) \tag{3.10}$$

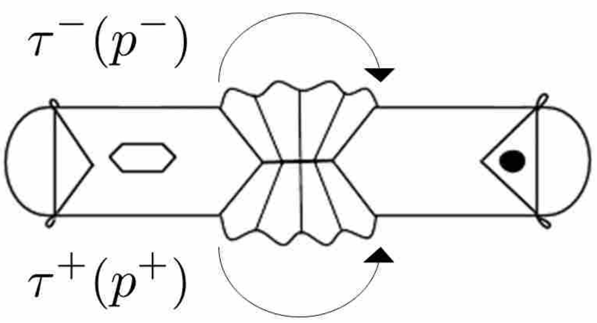


Figure 3.1: Antagonistic torque model for the antagonistic actuators.

where $\mathbf{K_p}$ and $\mathbf{K_d}$ are diagonal matrices that model the joint stiffness and damping (different than $\mathbf{K_p}$ and $\mathbf{K_d}$ in Eqs 3.3 and 3.4, which are used for impedance control for Baxter), $\tau^{+}(\mathbf{p}^{+})$ and $\tau^{-}(\mathbf{p}^{-})$ are the torques caused by all the positive or negative acting actuators respectively, and $\mathbf{M}^{-1}(\mathbf{q})\mathbf{G}(\mathbf{q})$ is the torque the actuators need to apply to oppose gravity. The only friction term that we include in the model of King Louie is viscous damping, which is included in the term $\mathbf{K_d}$, and we neglect any Coulomb friction because there is no gearing on this platform. The torque generated from pressure actuation is modeled to be linearly related to the pressure in the actuator as shown in Eq 3.11:

$$\tau^{+/-}(p^{+/-}) = \gamma p^{+/-} \tag{3.11}$$

For this work we assume the pressure in the actuators behave as first order systems as described by Eq 3.12 and neglect any nonlinearities.

$$\dot{p} = \alpha p + \beta p_{des} \tag{3.12}$$

where α, β and γ are constant values determined empirically.

The state variable equations used for this platform are as follows:

$$\dot{\mathbf{x}} = \mathbf{A}(\mathbf{x})\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{M}^{-1}(\mathbf{q})\mathbf{G}(\mathbf{q}) \tag{3.13}$$

where

$$\mathbf{x} = \begin{bmatrix} \dot{\mathbf{q}} & \mathbf{q} & \mathbf{p}^{+} & \mathbf{p}^{-} \end{bmatrix}^{T} \tag{3.14}$$

$$\mathbf{u} = \begin{bmatrix} \mathbf{p}_{des}^{+} & \mathbf{p}_{des}^{-} \end{bmatrix}^T \tag{3.15}$$

$$\mathbf{A} = \begin{bmatrix} -\mathbf{M}^{-1}(\mathbf{C}+\mathbf{K_d}) & -\mathbf{M}^{-1}\mathbf{K_p} & \mathbf{M}^{-1}\mathbf{K}_{\gamma}^{+} & -\mathbf{M}^{-1}\mathbf{K}_{\gamma}^{-} \\ \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{K}_{\alpha}^{+} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{K}_{\alpha}^{-} \end{bmatrix} \tag{3.16}$$

and

$$\mathbf{B} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{K}_{\beta}^{+} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{\beta}^{-} \end{bmatrix} \tag{3.17}$$

In Eq 3.16, the dependence of **A**, **M**, and **C** on the state **x** is not shown for readability. All **K** matrices are diagonal matrices that contain the different coefficients for the respective joint pressures. $\mathbf{K}_{\gamma}$ contains γ values that relate pressure to torque (see Eq 3.11). $\mathbf{K}_{\alpha}$ and $\mathbf{K}_{\beta}$ contain α and β values that model pressure dynamics (see Eq 3.12). Each of these **K** matrix values were determined empirically. These equations are used to control King Louie using both the Fixed-State and Coupling-Torque methods as described in section 3.6.

3.2 Discretization and Linearization Methods

For our work we use three different strategies to linearize the models described in the previous section. We begin with Taylor Series to obtain a benchmark linearization, and progressively simplify the model by assuming that terms which are functions of **q** and $\dot{\mathbf{q}}$ are constant over a given horizon (Fixed-State method), and then by decoupling the dynamics of each link from the other links to create a scalable linear model (Coupling-Torque method). Each of the linear state space models described below are discretized using the matrix exponential [70, Section 5-5] resulting in the following form for the dynamic model:

$$\mathbf{x}_{k+1} = \mathbf{A}_d\mathbf{x}_k + \mathbf{B}_d\mathbf{u}_k \tag{3.18}$$

3.2.1 Taylor Series Linearization

We first use Taylor Series linearization (TS), linearizing about the current states. The partial derivatives for the TS method were performed numerically due to practical limitations for symbolic differentiation in our simulation software. The difference in accuracy between the numerical and symbolic differentiation was checked on similar one and two DoF systems, and found to be negligible. TS is a standard method for linearizing equations and is primarily used in this work as a benchmark against which to compare the other two methods. TS linearizes the state space model as follows:

$$\delta\dot{\mathbf{x}} = \begin{bmatrix} \frac{\partial\ddot{\mathbf{q}}}{\partial\dot{\mathbf{q}}} & \frac{\partial\ddot{\mathbf{q}}}{\partial\mathbf{q}} \\ \mathbf{I} & \mathbf{0} \end{bmatrix} \delta\mathbf{x} + \begin{bmatrix} \frac{\partial\ddot{\mathbf{q}}}{\partial\mathbf{q_{des}}} \\ \mathbf{0} \end{bmatrix} \delta\mathbf{u} + f(x_0, u_0) \tag{3.19}$$

Generally the Taylor Series is used to linearize an equation about an equilibrium point, which results in $f(x_0,u_0) = 0$, where x_0 and u_0 are the equilibrium states and inputs, respectively. However, we are linearizing about the current states, so $f(x_0,u_0)$ must be retained. The first two terms in Eq 3.19 are discretized by using the matrix exponential as described before and the last term is discretized by evaluating Eq 3.5 at the beginning of the time horizon and multiplying by ΔT to get:

$$\Delta T \begin{bmatrix} \ddot{\mathbf{q}}_\mathbf{0} \\ \dot{\mathbf{q}}_\mathbf{0} \end{bmatrix} \tag{3.20}$$

where ΔT is the time step size. Eq 3.20 replaces $f(x_0,u_0)$ in Eq 3.19 for the discrete formulation. We only explore this method for controlling Baxter in simulation.

3.2.2 Fixed-State

The implementation of the Fixed-State (FS) method is dependent on the type of controller used. In this thesis the FS implementation is similar to that in [36, 57, 59]. For the FS method, **A** and **B** (Eqs 3.8-3.9 for Baxter, and Eqs 3.16-3.17 for King Louie) are evaluated at the current states, and are assumed to be constant for a given time horizon. The system is then discretized.

3.2.3 Coupling-Torque

The Coupling-Torque (CT) method is similar to the FS method but makes a further simplification. The FS and TS both use a dynamic model that includes cross-coupling in the joints. However, the CT method decouples the dynamics of the arm, allowing for the input of each link to be optimized independent of the other links for a given time horizon. Dynamic coupling occurs because of the torques generated from the motion of other links in the arm, and is expressed mathematically via the Coriolis and inertial terms. The CT method treats these coupling torques as a constant known disturbance over the horizon duration. This strategy makes the CT method computationally scalable to more degrees of freedom than the other two methods.

The general equation that MPC uses for its forward prediction when using the CT method are as follows for Joint i, $\forall i = 1 \ldots n$ where n is the number of joints:

$$\mathbf{x}_{k+1,i} = \mathbf{A}_{mpc,i}\mathbf{x}_{k,i} + \mathbf{B}_{mpc,i}\mathbf{u}_{k,i} + \mathbf{d}_{mpc,i} \tag{3.21}$$

For Baxter, each of the terms in Eq 3.21 are defined as follows (where $a_{i,j}$ and $b_{i,j}$ are the entries in the ith row and jth column of $\mathbf{A_d}$ and $\mathbf{B_d}$ from Eq 3.18):

$$\mathbf{A}_{mpc,i} = \begin{bmatrix} a_{i,i} & a_{i,j} \\ a_{j,i} & a_{j,j} \end{bmatrix} \tag{3.22}$$

$$\mathbf{B}_{mpc,i} = \begin{bmatrix} b_{i,i} \\ b_{j,i} \end{bmatrix} \tag{3.23}$$

$$\mathbf{x}_{k,i} = \begin{bmatrix} \dot{q}_{k,i} & q_{k,i} \end{bmatrix}^T \tag{3.24}$$

$$\mathbf{u}_{k,i} = q_{des,k,i} \tag{3.25}$$

$$\mathbf{d}_{mpc,i} = \begin{bmatrix} d_{\dot{q},i} & d_{q,i} \end{bmatrix}^T \tag{3.26}$$

where

$$d_{\dot{q},i} = \dot{q}_{1,i} - a_{i,i}\dot{q}_{0,i} - a_{i,j}q_{0,i} - b_{i,i}q_{des,0,i} \tag{3.27}$$

$$d_{q,i} = q_{1,i} - a_{j,i}\dot{q}_{0,i} - a_{j,j}q_{0,i} - b_{j,i}q_{des,0,i} \tag{3.28}$$

$$j = i + n$$

For King Louie, the $\mathbf{A}_{mpc,i}$ and $\mathbf{B}_{mpc,i}$ terms in Eq 3.21 are defined as

$$\mathbf{A}_{mpc,i} = \begin{bmatrix} a_{i,i} & a_{i,j_1} & a_{i,j_2} & a_{i,j_3} \\ a_{j_1,i} & a_{j_1,j_1} & a_{j_1,j_2} & a_{j_1,j_3} \\ a_{j_2,i} & a_{j_2,j_1} & a_{j_2,j_2} & a_{j_2,j_3} \\ a_{j_3,i} & a_{j_3,j_1} & a_{j_3,j_2} & a_{j_3,j_3} \end{bmatrix} \tag{3.29}$$

$$\mathbf{B}_{mpc,i} = \begin{bmatrix} b_{i,i} & b_{i,j_1} \\ b_{j_1,i} & b_{j_1,j_1} \\ b_{j_2,i} & b_{j_2,j_1} \\ b_{j_3,i} & b_{j_3,j_1} \end{bmatrix} \tag{3.30}$$

with

$$\begin{aligned} j_1 &= i + n \\ j_2 &= i + 2n \\ j_3 &= i + 3n \end{aligned} \tag{3.31}$$

The other terms are defined as follows:

$$\mathbf{x}_{k,i} = \begin{bmatrix} \dot{q}_{k,i} & q_{k,i} & p^+_{k,i} & p^-_{k,i} \end{bmatrix}^T \tag{3.32}$$

$$\mathbf{u}_{k,i} = \begin{bmatrix} p^+_{des,k,i} & p^-_{des,k,i} \end{bmatrix}^T \tag{3.33}$$

$$\mathbf{d}_{mpc,i} = \begin{bmatrix} d_{\dot{q},i} & d_{q,i} & d_{p^+,i} & d_{p^-,i} \end{bmatrix}^T \tag{3.34}$$

where

$$\begin{aligned} d_{\dot{q},i} = \dot{q}_{1,i} - a_{i,i}\dot{q}_{0,i} - a_{i,j_1}q_{0,i} - \\ a_{i,j_2}p^+_{0,i} - a_{i,j_3}p^-_{0,i} - b_{i,i}p^+_{des,0,i} - \\ b_{i,j_1}p^-_{des,0,i} + \mathbf{M}(\mathbf{q_0})^{-\mathbf{1}}\mathbf{G}(\mathbf{q_0})_i \end{aligned} \tag{3.35}$$

$$\begin{aligned} d_{q,i} = q_{1,i} - a_{j_1,i}\dot{q}_{0,i} - a_{j_1,j_1} q_{0,i} - \\ a_{j_1,j_2} p^+_{0,i} - a_{j_1,j_3} p^-_{0,i} - \\ b_{j_1,i} p^+_{des,0,i} - b_{j_1,j_1} p^-_{des,0,i} \end{aligned} \tag{3.36}$$

$$\begin{aligned} d_{p^+,i} = p^+_{1,i} - a_{j_2,i}\dot{q}_{0,i} - a_{j_2,j_1} q_{0,i} - \\ a_{j_2,j_2} p^+_{0,i} - a_{j_2,k_3} p^-_{0,i} - \\ b_{j_2,i} p^+_{des,0,i} - b_{j_2,j_1} p^-_{des,0,i} \end{aligned} \tag{3.37}$$

$$\begin{aligned} d_{p^-,i} = p^-_{1,i} - a_{j_3,i}\dot{q}_{0,i} - a_{j_3,j_1} q_{0,i} - \\ a_{j_3,j_2} p^+_{0,i} - a_{j_3,j_3} p^-_{0,i} - \\ b_{j_3,i} p^+_{des,0,i} - b_{j_3,j_1} p^-_{des,0,i} \end{aligned} \tag{3.38}$$

$a_{i,j}$ and $b_{i,j}$ are again the entries in the ith row and jth column of $\mathbf{A_d}$ and $\mathbf{B_d}$ from Eq 3.18. $\mathbf{d}_{mpc,i}$ is the torque generated from the motion of the other links. When implemented in the controller, $\mathbf{d}_{mpc,i}$ is calculated at the current states and then held constant over the MPC horizon.

To reiterate, the significance of this method is that it reduces the state variable matrix $\mathbf{A_d}$ ($mn \times mn$) to $\mathbf{A}_{mpc}$ ($m \times m$), where m is the number of states and n is the number of links. This greatly simplifies the MPC optimization, which improves computational tractability.

3.3 Model Accuracy

Each of these model simplifications were compared, in simulation, to the full nonlinear model for Baxter (Eq 3.4) to examine the impact of each of the simplifying assumptions (Fig 3.2). This also helps to determine a reasonable horizon length when these linearized models are implemented in MPC (assuming that the nonlinear model sufficiently represents reality). In these tests, Eq 3.4 (black line) is simulated with $\mathbf{q_{des}}$ (dark green line) as a step input and initial conditions of $\mathbf{q} = \dot{\mathbf{q}} = \mathbf{0}$. At t = 0.05 s, the states predicted by Eq 3.4 are passed to each of the three linearization model simulations (light green, red, and blue dotted lines), which independently predict the changes in states over a horizon length of about 0.15 s. At the end of the horizon the linearized

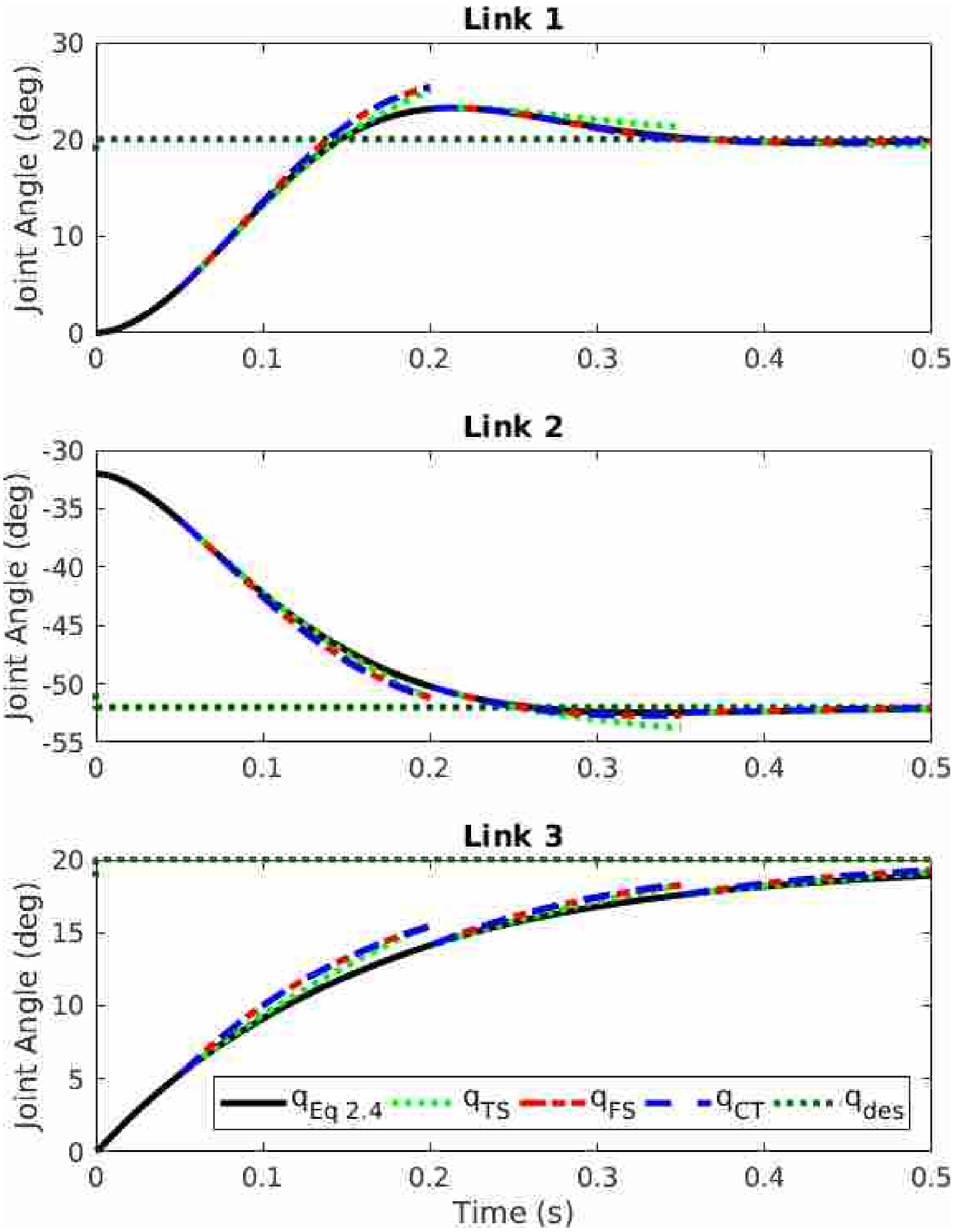


Figure 3.2: Comparing linearized models to the nonlinear dynamics for a simulation of a Baxter arm immediately after a 20 deg step in q_{des}. Two prediction horizons of 150 time steps (with 0.001 s per step) are shown.

model simulations are terminated and restarted with the current true states from the Eq 3.4 simulation. This gives an idea of the significance of the deviations introduced by each linearization method during the simulation time horizon. Greater deviations between the nonlinear and linear simulations imply that simplifying assumptions made for that linear model are more significant and will, presumably, negatively impact MPC performance.

Step changes of 20 deg were simultaneously commanded to three joints in simulation, as shown in Fig 3.2. The first horizon is from t = 0.05 - 0.2 s, the second horizon is from t = 0.2 - 0.35 s, and the final horizon is from t = 0.35 s - 0.5 s. All horizons are 150 time steps, with increments of 0.001 seconds. This gives a total horizon of 0.15 seconds, which is longer than the control horizons later used for the physical Baxter (see Section 3.6.1) and King Louie (see Section 3.6.2). Only the first three joints were used because of software/hardware limitations when running Taylor Series linearization on more than three joints.

In Fig 3.2, the TS linearization does not approximate the nonlinear model as well as FS and CT. This result is likely due to the fact that our TS approximation only uses the first term of the Taylor Series, so that the resulting approximation would be linear. By including more terms from the Taylor Series the TS method is expected to better approximate the nonlinear model, but then the approximation would not be linear and would not serve the intended purpose of this work to make MPC more tractable.

Determining the necessary accuracy for a model to perform well with MPC is a difficult problem. However, the key result here is that the accuracy of the CT model (which is the simplest model) is comparable to the TS model and that both give comparable closed-loop performance as shown in Section 3.4.

3.4 Control Performance

For this work, the MPC algorithm was formulated by first calculating the current states and model using the methods described previously. Then we send the model, current states, and cost function weights to an optimization routine which returns the control inputs and predicted corresponding states that minimize the cost function for a given horizon. The length of the horizon (in time) is determined by the number of time steps given to the optimizer, and the rate at which the controller is run. Only the inputs returned for the first time step are implemented on the platform,

Table 3.1: CT-MPC Cost Function Weights for Physical Baxter

	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6	Joint 7
α	3	10	10	5	10	5	5
β	0.02	0.1	0.08	0.05	0.08	0.05	0.01
γ	150	700	100	400	100	100	300

then the updated states (which are measured) are sent to the MPC optimization and the process is repeated. For continued reference in this thesis TS-MPC, FS-MPC, and CT-MPC refers to MPC based on linearized models using Taylor Series, Fixed-State, and Coupling-Torque methods, respectively.

3.4.1 Optimization Framework for Baxter

MPC for Baxter operated using the following cost function:

$$\underset{q_{des}}{\arg\min} \sum_{k=0}^{T_c} \left(\left\| \mathbf{q_{goal}} - \mathbf{q}_k \right\|_{\alpha}^{2} + \|\dot{\mathbf{q}}_k\|_{\beta}^{2} + \left\| \mathbf{q}_{\mathbf{des},k+1} - \mathbf{q}_{\mathbf{des},k} \right\|_{\gamma}^{2} \right) \tag{3.39}$$

where $\mathbf{q_{goal}}$ is the target angle, α, β and γ are weights, ΔT is the step size and T_c is the horizon length (set to 20 time steps in this work). The formulation shown in Eq 3.39 allows the optimizer to manipulate $\mathbf{q_{des}}$ as the system input, utilizing $\mathbf{K_p}$ and $\mathbf{K_d}$ as a virtual spring and damper on the system. The first term in Eq 3.39 drives the system to the desired joint angle, the second term prevents the link from moving too quickly, and the third term prevents the controller from changing $\mathbf{q_{des}}$ too aggressively. Values for CT-MPC cost function weights are shown in Table 3.1.

3.4.2 Optimization Framework for King Louie

The controller used for Baxter was modified for King Louie as given by the following equations:

$$\underset{\mathbf{p}^{+}_{\mathbf{des}},\mathbf{p}^{-}_{\mathbf{des}}}{\arg\min} \sum_{k=0}^{T_c} \Bigg(\left\|\mathbf{q}_{\mathbf{goal}} - \mathbf{q}_k\right\|^2_{\alpha} + \left\|\mathbf{p}^{+}_{\mathbf{des},k+1} - \mathbf{p}^{+}_{\mathbf{des},k}\right\|^2_{\beta} + \left\|\mathbf{p}^{-}_{\mathbf{des},k+1} - \mathbf{p}^{-}_{\mathbf{des},k}\right\|^2_{\beta} \cdots$$

$$+ \left\|\mathbf{p}^{+}_{\mathbf{des},k} - \mathbf{p}_{\mathbf{target}}\right\|^2_{\gamma} + \left\|\mathbf{p}^{-}_{\mathbf{des},k} - \mathbf{p}_{\mathbf{target}}\right\|^2_{\gamma} \Bigg) \qquad (3.40)$$

The first term in Eq 3.40 drives the joint angles to a desired position, the second and third terms limit the rate at which the pneumatic actuation can change, and the last two terms help to maintain a desired stiffness. The dynamic model of the arm and Eq 3.41 are used as optimization constraints, where Eq 3.41 restricts the search space to physically feasible actuator pressures. Cost function weights for CT-MPC for each link are α = 10, β = 0.7, and γ = 0.00005.

$$\mathbf{p}_{min} \leq \mathbf{p}^{+,-}_{\mathbf{des},k} \leq \mathbf{p}_{max}, \forall k = 0 \ldots T_c \qquad (3.41)$$

3.4.3 Simulation of Baxter Model and Controller

For simulations and modeling, system parameters such as inertia, mass, and center of mass were used as given by the manufacturer of the Baxter robot. Simulations were run in MATLAB, using the Robotics Toolbox developed by Peter Corke [69], with all simulated controllers running at 300 Hz with T_c = 30 time steps. MPC optimizations were performed using CVXGEN [53].

The results of the simulated MPC response are shown in Figure 3.3 and Table 3.2. The values in the table are an average over four step tests, while only one step test is shown in Fig 3.3. The results demonstrated by this simulation are remarkable. It is surprising that CT-MPC has comparable performance to TS-MPC, despite the simplicity of the method. These results show that CT-MPC has potential performance capability on physical systems, which is demonstrated in Section 3.6. FS-MPC does not perform as well here as the others.

Also shown in Figure 3.3 is control performance using a non-linear model-based controller. This method uses an optimization model that is identical to the simulation model, which means that the controller should be able to perfectly control the system. Because of this, the non-linear controller uses a single optimization to generate inputs over the entire horizon. The intent of including this comparison was to provide a best-case control scenario against which we can compare the other methods. Non-linear control performs the best for Links 1 and 2, but for Link 3 it has

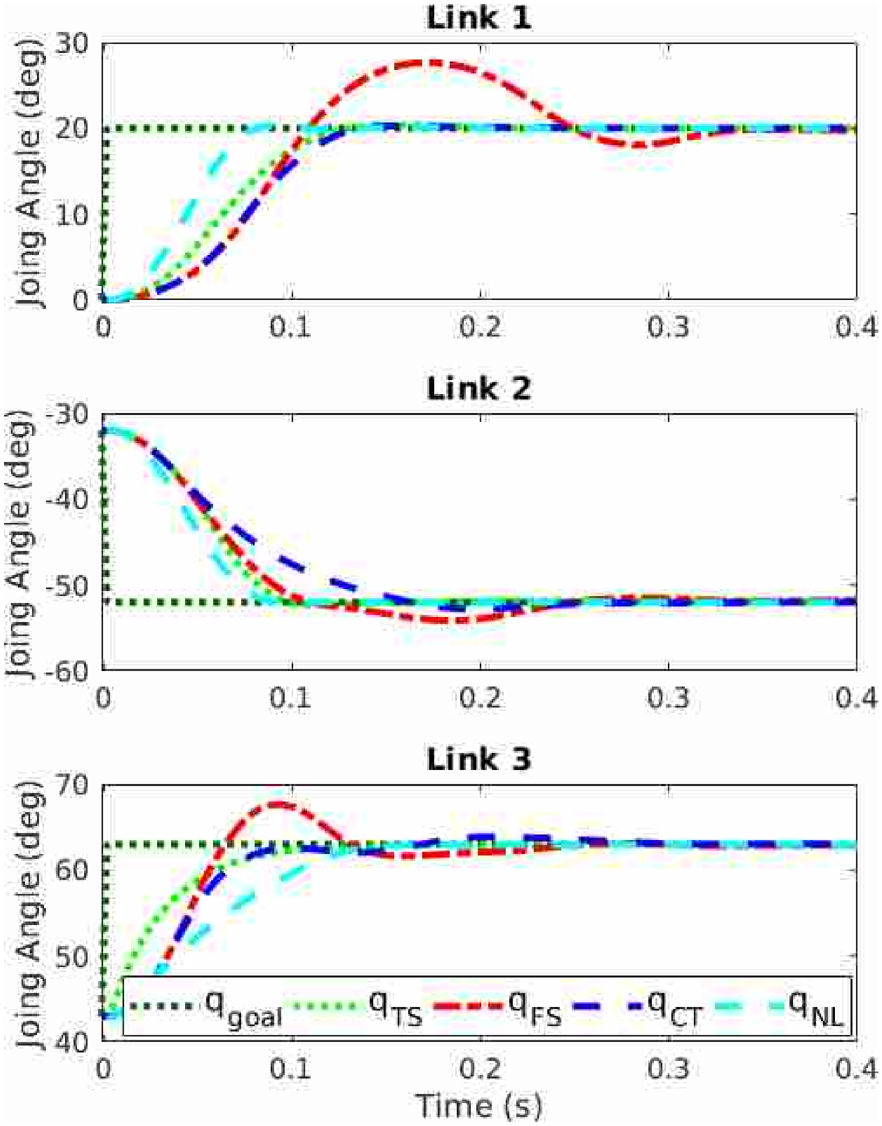


Figure 3.3: MPC on three links of a simulated Baxter robot, using three different linearization methods, as well as a non-linear optimized control method.

the slowest rise-time of all the methods. This simulation ran slowly and was very difficult to tune, when compared to the other control methods. It is likely that better non-linear control performance can be attained, but the practical difficulty in doing so speaks to the power of the other linearization methods, especially Coupling-Torque.

The process of tuning the multi-objective cost function weights presents a significant difference between the three linearization methods. The cost functions, as formulated in Eqs 3.39 and 3.40, have three weighting parameters (α, β, γ) per link. When running TS-MPC and FS-MPC on

Table 3.2: Step Test Results

	90% Rise Time (s)	±5% Settling Time (s)	Overshoot (%)
TS-MPC			
Link 1	0.11	0.12	0.6
Link 2	0.08	0.12	8.6
Link 3	0.07	0.09	0.0
FS-MPC			
Link 1	0.10	0.28	62.2
Link 2	0.10	0.23	5.0
Link 3	0.06	0.19	22.4
CT-MPC			
Link 1	0.11	0.16	11.7
Link 2	0.14	0.23	7.8
Link 3	0.07	0.23	7.6
NL-CTL			
Link 1	0.07	0.07	1.4
Link 2	0.07	0.08	0.7
Link 3	0.11	0.13	0.0

three links, this means that the optimizer is minimizing a cost function with nine different terms. In contrast, CT-MPC runs three optimizers independently (one for each link), each of which minimizes a cost function with only three terms. The formulation of TS-MPC and FS-MPC creates an interdependence between the tuning parameters in the cost function. Although there is still some interdependence when tuning CT-MPC (via the movements of the links only) it is to a significantly smaller degree than the other two methods. At least qualitatively, this makes the process of tuning the cost function weights easier for CT-MPC than for the other two linearization methods.

3.5 Controller Robustness

A large source of potential error for MPC is in how well the controller model matches reality. In addition to the comparison to the nonlinear model in Section 3.3, controller robustness was tested in simulation using the Baxter model by perturbing different model parameters. A first test was preformed by perturbing the dynamic parameters (link inertia and mass) of the Baxter arm model by up to ±100%. This perturbed model was then used to control Baxter in simulation using TS-MPC, FS-MPC, and CT-MPC. Parameters were left unchanged for the simulated robot to see how well MPC could perform for each method when the model used for control was significantly different than the system being controlled. The results of this test are shown in Fig 3.4. In this test, the TS-MPC quickly became unstable for all three links, whereas FS-MPC and CT-MPC showed almost no impact from the model perturbation. This is remarkable, considering that it is unlikely that actual dynamic parameters would be estimated with that degree of error.

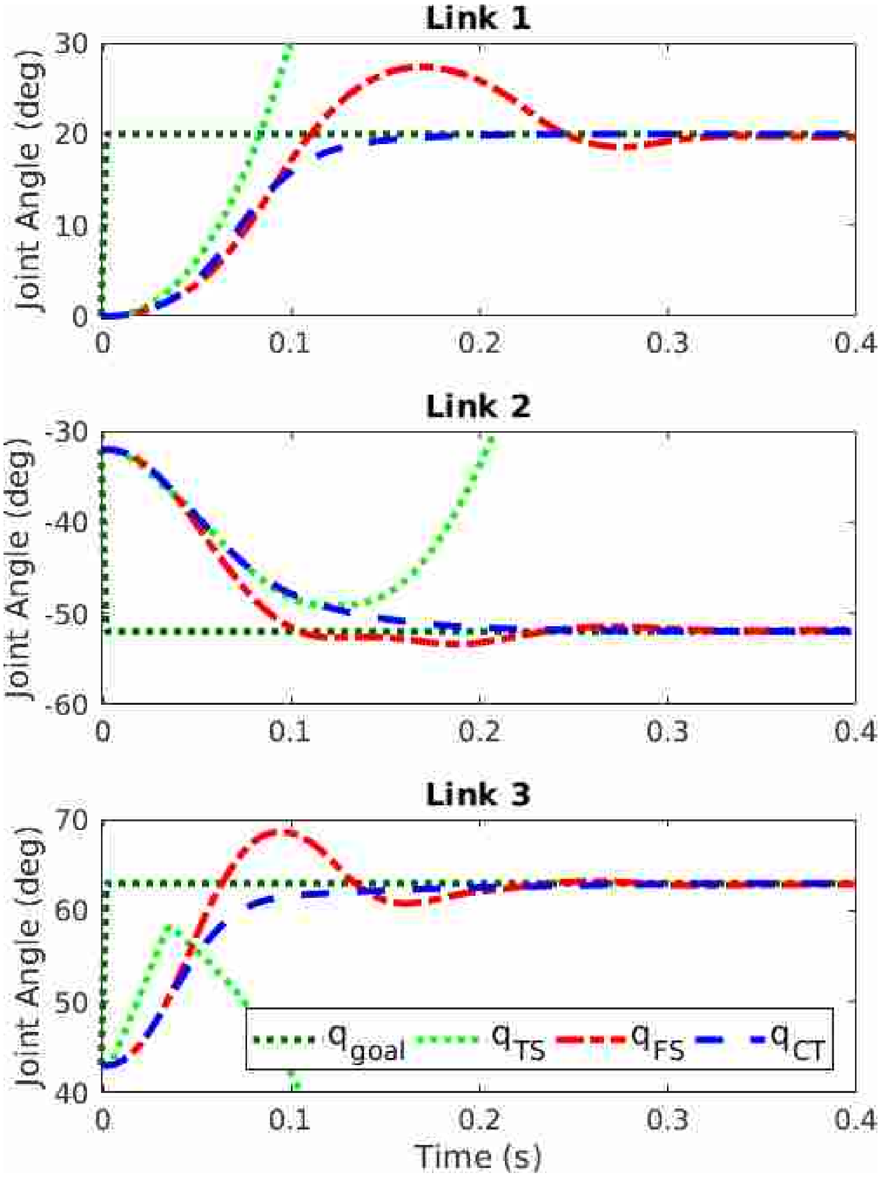


Figure 3.4: MPC on a simulated Baxter arm using the three linearization methods. Perturbed dynamic parameters by up to ±100% to simulate parameter mis-identification.

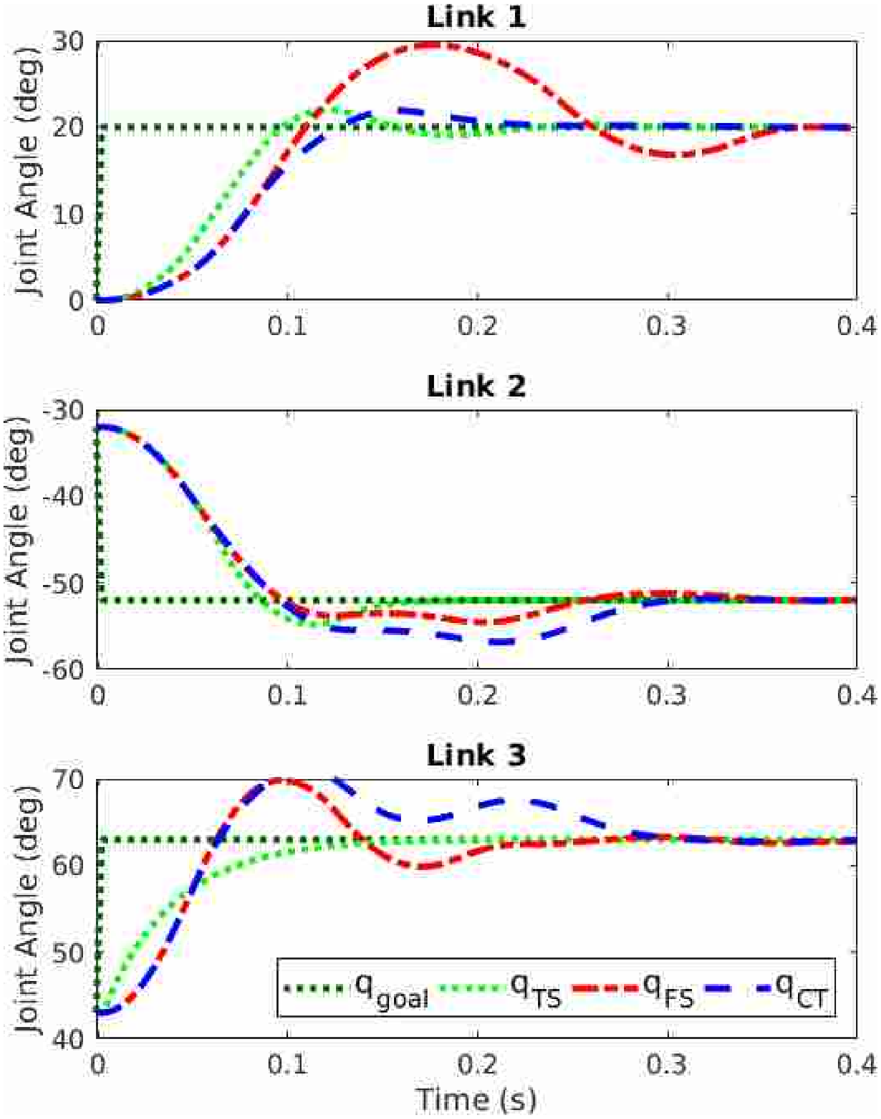
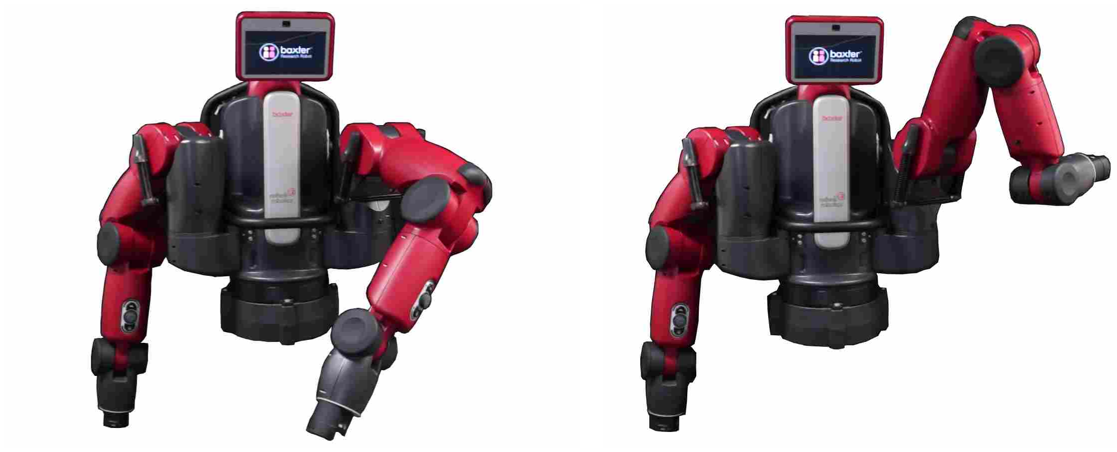


Figure 3.5: MPC with simulated Baxter robot using the three linearization methods and $\mathbf{K_d}$ increased by a factor of 3 to simulate parameter mis-identification.

For the second robustness test we increased the controller's values for $\mathbf{K_d}$ by a factor of 3 (which is like assuming larger viscous friction). The results are shown in Fig 3.5. Here, TS-MPC performed the best with the parameter mis-identification. Each of the methods were negatively impacted in terms of overshoot and settling time, but rise time was not significantly impacted. Again, this is an unrealistically large error in $\mathbf{K_d}$, showing the robustness of MPC to inaccurate parameters for all three linearization methods.



(a) Pose 1

(b) Pose 2

Figure 3.6: Poses used in tests referenced in Section 3.6.1

3.6 Hardware Implementation

3.6.1 MPC on Baxter

In [36], the authors compared MPC to other control methods for underdamped systems on a physical Baxter robot. The model was linearized with what was essentially the FS method but, due to the scalability limitations already mentioned for large DoF, the controllers operated on Cartesian position only. This allowed the full 7-DoF arm to be controlled while only optimizing for three DoF by controlling for end effector position but not orientation. End effector position error was measured while moving between two different poses (poses shown in Fig 3.6).

For comparison to the results presented in [36], CT-MPC was implemented on all 7 links of a physical Baxter robot, and was run at 200 Hz and T_c = 20 time steps (which gives a 0.095 second horizon). Figure 3.7 shows a comparison between the position error from [36] and CT-MPC from this work. CT-MPC performs similarly well to MPC from [36] for position error. This is remarkable, considering the extensive simplifications the CT method makes, and shows that the method is scalable to at least 7-DoF, allowing us to explicitly control full pose (orientation and position of the end effector) with all 7-DoF. Computational limitations that prevented the FS method from scaling very well in [36] are not a problem for the CT linearization method. The TS linearization has similar scalability problems to FS.

Due to difficulties with scalability, TS-MPC and FS-MPC were not implemented on the physical Baxter.

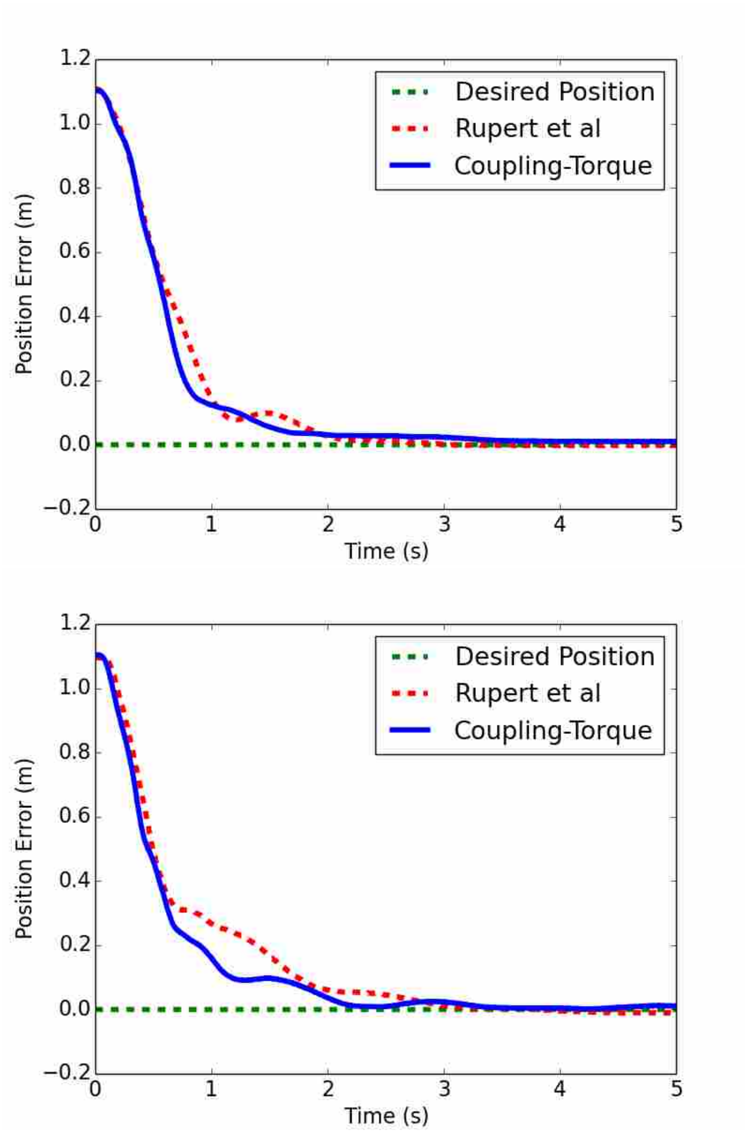


Figure 3.7: Distance from goal position versus time as Baxter arm steps from Pose 1 to Pose 2 in the top graph and from Pose 2 to Pose 1 in the bottom graph.

3.6.2 MPC on King Louie

The CT-MPC method was tested on three DoF of King Louie. Figure 3.8 shows the results of the CT-MPC on King Louie running at 300 Hz with T_c = 35 time steps (which gives a 0.117 second MPC horizon) and step commands of 40°. Control on King Louie was originally run at

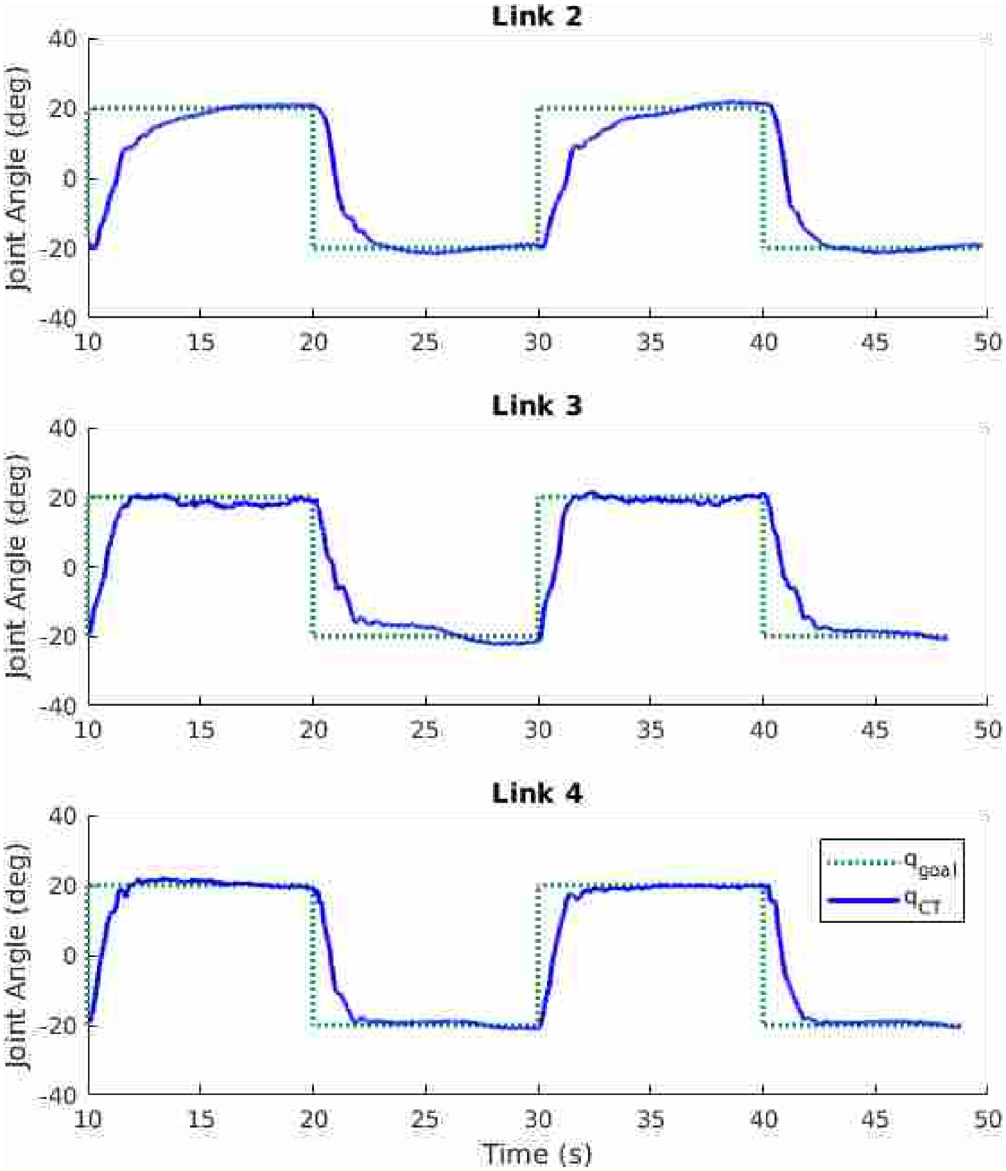


Figure 3.8: CT-MPC on King Louie for three joints.

200 Hz, but it was later discovered that 300 Hz gave the best performance, out of a range of rates tested. The same test was not done on Baxter. The CT-MPC has excellent performance with very little overshoot or residual oscillation.

When testing the FS-MPC we were not able to get any substantial data to show that it is an effective control method for high dimensional and complex systems with unmodeled nonlineararities like King Louie (ie. a three DoF robot with four states per DoF). FS-MPC was implemented by another graduate student on Nightcrawler, a robotic manipulator similar in design to

King Louie. After a significant amount of tuning the FS-MPC still had over 30° of steady state error for any step command and could only run at a maximum rate of 20 Hz, due to computational limits when solving the MPC problem. As the different weights for each joint were tuned they would adversely effect the performance of the other joints. With additional tuning the FS-MPC may have been able to perform as well as, or better than, the CT-MPC, but these results emphasize the strength of using a simpler model that is also simpler to tune.

3.7 Conclusion

In this chapter, we have presented the Coupling-Torque method, a novel method for linearizing and decoupling the dynamic model of a robot arm. We then compared its predictive performance and control performance when used in MPC to two other linearization methods. From these trials, we found the Coupling-Torque method to give good control performance, and to be scalable for systems with complex models and high DoF.

CHAPTER 4. ADAPTIVE CONTROL

The Coupling-Torque method provides a simple model upon which we develop an adaptive control method, which is discussed in this chapter. The simplicity of the model lends itself to a simpler process of incorporating adaptive control.

We expected that adapting model parameters to their ideal values would yield better control performance, as is typically the case. We first attempted several methods of online parameter estimation (Section 4.1), but discovered that accurate parameter values did not necessarily result in better control performance. We then shifted our focus to performance-based adaptive control, and developed a novel adaptive control method designed to work in conjunction with MPC of these soft robot platforms (Section 4.2).

4.1 Parameter Estimation

A common method for adaptive control is called Model Identification Adaptive Control (MIAC), where algorithms are designed to match experimental data with a model by manipulating model parameters. Generally, a more accurate model will result in improved model-based control performance. In initial work on adaptive control for soft robots, this strategy was employed to estimate the inertia of the Grub (Figure 2.1), which is modeled as an inverted pendulum. We focus on estimating inertia here because the end goal is to have King Louie pick up a mass of unknown weight, which will primarily impact the inertia of the arm.

4.1.1 Simulation

Three different MIAC-based methods were used to estimate the Grub's inertia during simulation. The methods used were Least Squares (LS), Recursive Least Squares (RLS), and Moving Horizon Estimation (MHE). Each simulation was run at 100 Hz, using the Robotics Toolbox for Matlab, developed by Peter Corke [69], where the simulated Grub had a three-pound weight added to the link's center of mass. FS-MPC (which is identical to CT-MPC for one DoF) was used to control the Grub's joint angle, and the estimated inertia value is used in the MPC model online as it is updated in simulation. Each method solved Equation 4.1 for I.

$$I\ddot{\mathbf{q}} = -K_d\dot{\mathbf{q}} - K_s\mathbf{q} + K_\gamma^+\mathbf{p}^+ - K_\gamma^-\mathbf{p}^- - G(\mathbf{q}) \tag{4.1}$$

where parameters are defined the same as in Equation 3.16. For each estimation method, the inertia value was initialized at I = 0.047 $kg \cdot m^2$, the rotational inertia of the simulated Grub when no weight is added. We then added a 3-pound weight to the simulated Grub's center of mass, which was not included in the control model.

The results for all three methods are shown in Figure 4.1. Note that the plots of control performance (q) for the three estimation methods were so similar that only one line could ever be seen, so only one line is shown in this plot. This indicates that adapting the inertia value (at least to the extent that it was changed here) has little, if any, impact on control performance. The simulated Grub, with extra weight added, has an inertia of 0.089 $kg \cdot m^2$, which is the value to which all three estimation methods settle.

Least Squares

LS estimation was implemented first because of its simplicity. We used all available data to estimate I, as shown in Eq 4.2:

$$I = (\ddot{\mathbf{q}}^T\ddot{\mathbf{q}})^{-1}\ddot{\mathbf{q}}^T RHS \tag{4.2}$$

where $\ddot{\mathbf{q}}$ is the second derivative of the joint angle and RHS is the Right Hand Side of Equation 4.1. Both $\ddot{\mathbf{q}}$ and RHS are vectors.

LS quickly settles to the correct inertia value, but causes the simulation to run fairly slowly so we pursued other estimation methods as well.

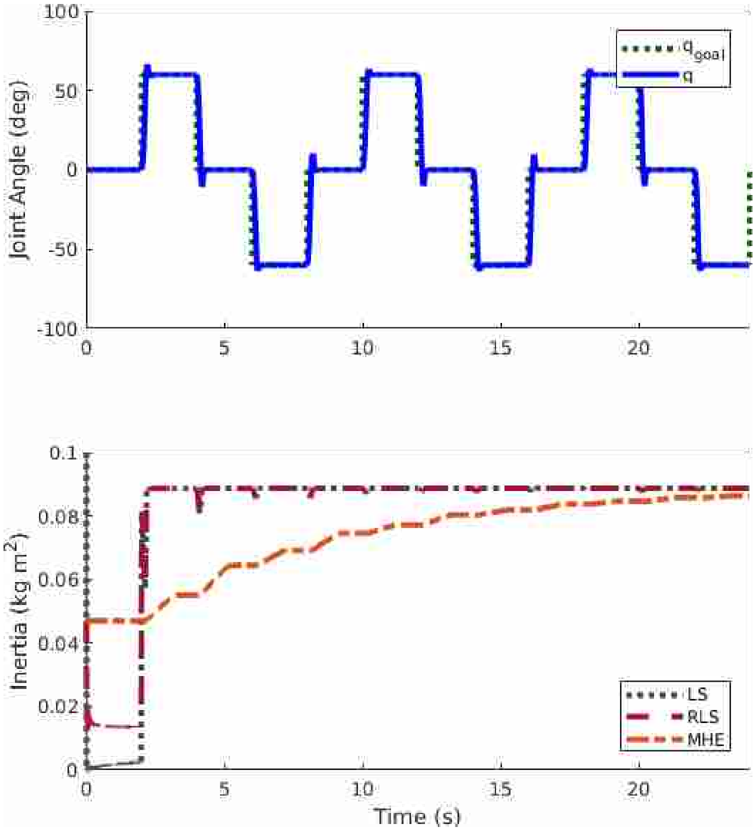


Figure 4.1: Control of a simulated Grub using MPC and varying the inertia value used in the MPC model.

Moving Horizon Estimation

The next estimation method implemented was MHE, which was implemented using CVXGEN [67] to minimize the cost function shown in Equation 4.3.

$$\arg\min_{I} \sum_{k=0}^{T_c} \left(\|I\ddot{q}_k - RHS_k\|_{\alpha}^{2} + \left\|I - I_{prev}\right\|_{\beta}^{2} \right) \tag{4.3}$$

$$s.t.$$

$$0 < I < I_{max}$$

where T_c is the length of the horizon (set to 100 time steps here) and α and β are weights, tuned to get the best balance between a quick rise time and reasonable settling at the correct inertia value. Fig 4.1 shows that MHE does settle to a particular inertia value, but does so slowly. We were unable to improve the rise time without negatively impacting MHE's ability to settle.

Recursive Least Squares

Lastly, RLS estimation was implemented. RLS is a powerful estimation method that typically estimates data as well as the LS method, but does so more efficiently than LS estimation. The RLS method estimates I at every time step using Equations 4.4 - 4.7.

$$\delta = RHS_k - \ddot{q}_k I_k \tag{4.4}$$

$$\varepsilon = \frac{\rho_k \ddot{q}_k}{\lambda + \ddot{q}_k^2 \rho_k} \tag{4.5}$$

$$\rho_{k+1} = \frac{\rho_k(1 - \varepsilon \ddot{q}_{k)}}{\lambda} \tag{4.6}$$

$$I_{k+1} = I_k + \delta \varepsilon \tag{4.7}$$

where k is the current time step, $\ddot{q}$ is the joint angle acceleration, λ is a forgetting factor (set to 1.0 here), and δ, ε, ρ are intermediate values.

RLS estimation did just as well as LS at estimating I. However, we saw very little difference in control performance regardless of the estimation method used. During this simulation process, we also discovered that inertia values an order of magnitude too small did improve control performance. This phenomenon is discussed further in the following section, and verified on the physical Grub.

4.1.2 Hardware

On the physical grub, MPC was used to control the joint angle, and the RLS algorithm (discussed in Section 4.1.1) was used to estimate the inertia value to be passed into the MPC model. The LS and MHE methods were not implemented on the physical Grub because they did not perform as well as RLS in simulation. The Grub's construction makes it difficult to predict an actual inertia value, so a three-pound weight was attached to the end, which should be the dominate contributor to the system's inertia (the Grub's link weighs about one pound). Treating the added weight as a point mass, a back-of-the-envelope calculation indicates that the system has an inertia of roughly 0.1 $kg \cdot m^2$, which is similar to the value the three estimation methods arrived at for the simulated Grub. The RLS algorithm settled at values to within ±5% of 0.1 $kg \cdot m^2$ within 0.4

seconds after the first step input, and the controller performance with the estimated inertia value can be seen in Figure 4.2 (blue dashed line).

However, in the development process, we discovered that manually setting an unrealistically low inertia value, such as 0.01 $kg \cdot m^2$, significantly improved control performance. Manually setting an unrealistically high inertia value, such as 1 $kg \cdot m^2$, significantly decreased control performance. Figure 4.2 shows control performance when $I = 0.01$ $kg \cdot m^2$ (red line), $I = 1$ $kg \cdot m^2$ (light green line), and when I is estimated using RLS (blue line). Table 4.1 shows the Root Mean Square (RMS) Error for each of these trials, which verifies the overall control improvement seen with very low inertia values. This indicates that, at least for this system, performance-based adaptive control would be more effective than a system identification strategy. This may be particularly true if our selected parameterization of the physical model is limited in what it can represent. Note that, in Figure 4.2, the Grub is at 0 deg when it is pointed straight up. The extreme overshoot seen at 0 deg, especially with $I = 0.01$ $kg \cdot m^2$, is caused by the torque due to gravity changing sign when q changes sign.

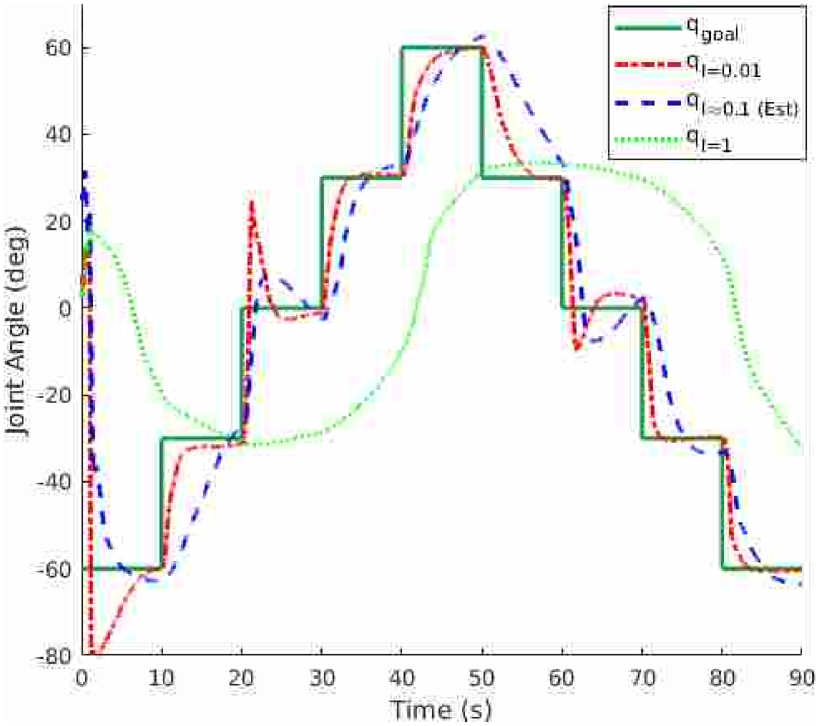


Figure 4.2: Joint angle control on the Grub, varying the model's inertia value, with a three-pound weight at the end of the link.

Table 4.1: RMS Error of MPC on the Physical Grub with respect to variant Inertia Value

Inertia Value ($kg \cdot m^2$)	RMS Error
1	41.8
0.1 (Est)	17.2
0.001	12.5

4.2 Model Reference Adaptive Control

We identified Model Reference Adaptive Control (MRAC) as a viable option for a performance-based adaptive control method and used Lavretsky's development for MRAC (see [68]) for our soft robot system. However, we also made adjustments so that we could continue to use MPC as the main control method.

4.2.1 Method Development

For this implementation of MRAC/MPC, Eq 3.21 is modified to

$$\mathbf{x}_{k+1,i} = \mathbf{A}_{mpc,i}\mathbf{x}_{k,i} + \mathbf{B}_{mpc,i}\mathbf{u}_{k,i} + \mathbf{d}_i - \Omega_i \tag{4.8}$$

The adaptive laws used to find Ω_i are

$$\dot{q}_{ref,i} = \alpha_i(q_{goal,i} - q_{ref,i}) \tag{4.9}$$

$$q_{err,i} = q_{ref,i} - q_i \tag{4.10}$$

$$\dot{\theta}_{i,j} = \begin{cases} \Gamma_{i,j}\Phi_{i,j}q_{err,i} - \sigma_{i,j}|q_{err,i}|\theta_{i,j} & \forall q_{err,i} \geq 1\ deg \\ 0 & \forall q_{err,i} < 1\ deg \end{cases} \tag{4.11}$$

$$\Omega_i = \sum_j \theta_{i,j}\Phi_{i,j} \tag{4.12}$$

$$\forall j = 1 \ldots J_i$$

where i designates a specific link, q_{goal} is the desired joint angle, q_{ref} is a desired reference trajectory, Φ is the regressor (which contains the dynamic parameters to be adapted and is defined in Eq 4.14 for the Grub and in Eq 4.15 for King Louie), J_i is the number of parameters to be adapted, and

α, Γ and σ are gains. The purpose of q_{ref} is to smooth out the transition between step changes in q_{goal}. Otherwise, abrupt step changes can cause stability issues for MRAC. Larger Γ values result in more quickly changing values for the adaptive parameters θ, whereas σ acts as a spring, preventing θ from growing too large. In Eq 4.11, the breakpoint at $q_{err} = 1$ deg serves as a deadband so that adaptive parameters do not respond to small changes in q. The parameters $\dot{q}_{ref}$ and $\dot{\theta}$ are numerically integrated to obtain q_{ref} and θ. The overall control scheme is summarized in Figure 4.3:

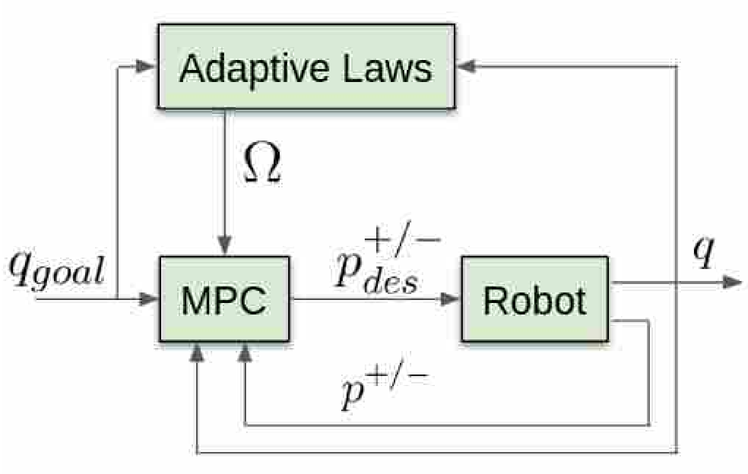


Figure 4.3: MRAC with MPC diagram.

In contrast, Lavretsky's MRAC formulation has the following method for incorporating the adapted parameters:

$$u_{mrac,i} = u_i - \Omega_i \tag{4.13}$$

where u_i is the input determined by some controller, and $u_{mrac,i}$ is the adapted input. Lavretsky's method modifies the input directly, whereas our method passes Ω in to MPC as a model parameter. This directly affects the solutions that MPC finds for control, while still respecting the constraints of the inputs and states.

4.2.2 Hardware Implementation on the Grub

Adaptive control was first implemented on the Grub (Figure 2.1). Adaptive gains (Γ and σ) were tuned with no extra mass attached, with Φ defined for the Grub as

$$\Phi_{Grub} = \begin{bmatrix} a_{1,1}\dot{q} & a_{1,2}q & a_{1,3}p^{+} & a_{1,4}p^{-} & I^{-1}G & I \end{bmatrix} \tag{4.14}$$

where $a_{m,m}$ are the entries in the mth row and nth column of $\mathbf{A}$ from Eq 3.16, x_i is the ith element of $\mathbf{x}$ from Eq 3.14, I is inertia (equivalent to I from Section 4.1) and G is gravity from Equation 3.13. These parameters were selected based on their impact on control performance during the tuning process.

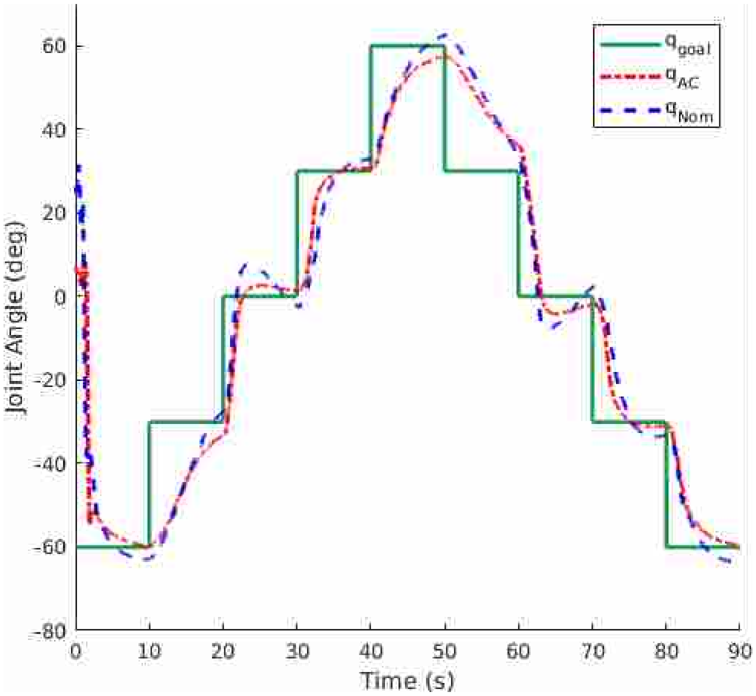


Figure 4.4: Adaptive vs Nominal control for the Grub with a three-pound weight added to the end of the link.

For comparison to the nominal controller, a three-pound mass was added to the end of the link after the initial adaptive parameter tuning. The results are shown in Figure 4.4 and Table 4.2, where the nominal case is the same as the I = 0.1 $kg \cdot m^2$ case in Figure 4.2 (the rotational inertia of the physical Grub with the three-pound weight added is about 0.1 $kg \cdot m^2$). In Fig 4.4 q_{goal} is the

Table 4.2: Grub Experiment Results

Controller	Mass (lb)	RMS Error (deg)	$\mathbf{q_{goal}}$ Error (deg)	95% Rise Time (s)
Nominal	3	17.22	2.94	6.29
MRAC/MPC	3	15.90	2.10	5.97

target joint angle, and q_{AC} and q_{Nom} are the resultant joint angles of MRAC/MPC and the nominal controller, respectively. Given that some of the step input responses never attain steady-state, the q_{goal} Error metric is shown instead of Steady-State Error in Table 4.2. This metric compares the q to q_{goal} at the final value, measured immediately before the next step input. Similarly, because some commands have significant q_{goal} Error, the 95% Rise Time metric shows rise time to the final value instead of rise time to q_{goal}. Root Mean Square (RMS) Error is also shown. Figure 4.4 indicates some improvement with MRAC/MPC, especially with regards to the overshoot when $q_{goal} = 0$ deg. Table 4.2 shows some improvement in all three metrics, but altogether we do not see as much improvement as we expected.

In contrast, MRAC/MPC on King Louie shows significant performance improvement, especially when weight is added to the end effector, as seen in Section 4.2.3. One explanation for this difference is that the success of MRAC/MPC on King Louie relies on adapting dynamic parameters that pertain to the motion of other links of the arm, whereas these dynamic couplings between joints do not exist with the Grub and therefore cannot easily be improved upon. This suggests that a different adaptive control approach may be more beneficial for improving performance of the Grub. However, this research aims to improve performance for a more complicated multiple degree of freedom system such as King Louie, so improving adaptive control further on the Grub was not pursued.

Despite the disappointment of MRAC/MPC's performance in this trial, it is remarkable that we can get this good of control using the one-pound link of the Grub to move a payload of three pounds. This high payload weight per robot weight ratio is one of the benefits of soft inflatable robots, and emphasizes the need for pursuing strategies, like adaptive control, that allow us to fully utilize these impressive platforms.

4.2.3 Hardware Implementation on the King Louie

Adaptive control was implemented on all four links of a King Louie ar. However, in preliminary testing it became clear that Link 1 and 2 (see Fig 2.2) had strong coupling effects, due to causes unique to King Louie's current setup. The problem is likely due to the way the HTC Vive trackers (described in Section 2.1.1) are being used to estimate joint angles. We have verified that the angle estimation works well for Links 2 - 4, but have not yet determined exactly what is causing the coupling between Links 1 and 2. We have used the HTC Vive trackers to estimate joint angles for other soft robotic platforms, and have not seen this problem in those trials. Because the focus of this thesis is not on soft robot configuration estimation, we instead focus on the contribution of this research and performed final experiments on the Links 2 - 4 only. For this reason, plots omit the time response of Link 1. Adaptive gains (Γ and σ) were tuned with no additional weight in the end effector, with Φ for King Louie defined in Equation 4.15, where the first column contains parameters for the Link 1, the second column contains parameters for Link 2, etc. Adaptive gains were tuned before the coupling effects between Links 1 and 2 were discovered, so the gains for Link 1 are shown in Equation 4.15 for the reader's information. The dynamic parameters adapted for King Louie are

$$\Phi_{KL} = \begin{bmatrix} a_{1,5}q_1 & a_{2,6}q_2 & a_{1,5}q_1 & a_{1,5}q_1 \\ a_{1,9}p_1^+ & g_2 & a_{1,9}p_1^+ & a_{1,9}p_1^+ \\ a_{1,13}p_1^- & 0 & a_{3,7}q_3 & a_{3,7}q_3 \\ a_{4,8}q_4 & 0 & a_{3,11}p_3^+ & a_{3,11}p_3^+ \\ g_2 & 0 & a_{3,15}p_3^- & a_{3,15}p_3^- \\ g_3 & 0 & a_{4,8}q_4 & a_{4,8}q_4 \\ 0 & 0 & a_{4,12}p_4^+ & a_{4,12}p_4^+ \\ 0 & 0 & a_{4,16}p_4^- & a_{4,16}p_4^- \\ 0 & 0 & g_2 & g_2 \\ 0 & 0 & g_3 & g_3 \end{bmatrix} \tag{4.15}$$

where $a_{m,n}$ and $b_{m,n}$ refer to the element in row m and column n of $\mathbf{A}$ from Eq 3.16 and $\mathbf{B}$ from Eq 3.17, respectively, and g_i is the result of the product of $\mathbf{M}^{-1}(\mathbf{q})\mathbf{G}(\mathbf{q})$ (see Eq 3.13) that corresponds to Link i. Note that the zeros in the first and second columns indicate that fewer dynamic parameters were adapted for Links 1 and 2. Also note that Columns 3 and 4 are identical. All of the parameters in Φ_{KL} were selected based on their impact on control performance during the tuning process.

We used MRAC/MPC to control arm movements for several minutes, allowing the adaptive parameters (θ) to settle. An example of parameter adaptation is shown in Figure 4.5. Note that each Φ_{KL} element has an accompanying θ value, so each link has the same number of θ values shown in Fig 4.5 as there are elements in the corresponding column of Φ_{KL} in Eq 4.15. Many of the θ values exhibit cyclical behavior which correlate with changes in p_{des}. This behavior demonstrates that some dynamic parameters included in Φ_{KL} do not encapsulate system behaviors that occur when p_{des} changes, and so the adaptive laws attempt to capture those dynamics by changing the corresponding θ value. Once the values for θ had settled, they were saved so that future trials could start with the adaptive gains found from this initial process, and are referred to here as $\theta_{m=0}$.

Figure 4.6 shows the control performance of an example trial using $\theta_{m=0}$, comparing nominal to adaptive control with no weight in the end effector. Also shown in Figure 4.6 is control performance with an integrator developed previously by a graduate student, which generally attained the best control performance to date [71, Page 37]. In Fig 4.6, and all following plots in this thesis, q_{goal} is the target joint angle, and q_{AC}, q_{Int} and q_{Nom} are the resultant joint angles of MRAC/MPC, control with the integrator, and the nominal controller, respectively.

Similar tests were run with one, two or three pounds grasped in the end effector (the arm itself weighs about five pounds, and three pounds is approximately the max payload of each arm). These trials were also started with the $\theta_{m=0}$ values from the no-weight scenario. For the trials with extra weight attached, the controller was given no indication of the extra weight added. Figure 4.7 shows data from the one-pound trial, Figure 4.8 shows data from the two-pound trial, and Figure 4.9 shows data from the three-pound trial.

For Link 2, Figures 4.6 - 4.9 show significant improvement with MRAC/MPC. In all four weight scenarios we see a decrease in rise time when compared against the integral and nominal controllers. This is especially encouraging as Link 2 traditionally has been the hardest to control well because it carries the weight of the rest of the arm. In the past, we observed that Link 1 and Link 2 behave quite similarly (due to the similarities in their size and construction), so it is likely that we will see similar performance improvements with MRAC/MPC on Link 1 when the angle estimation problem is addressed.

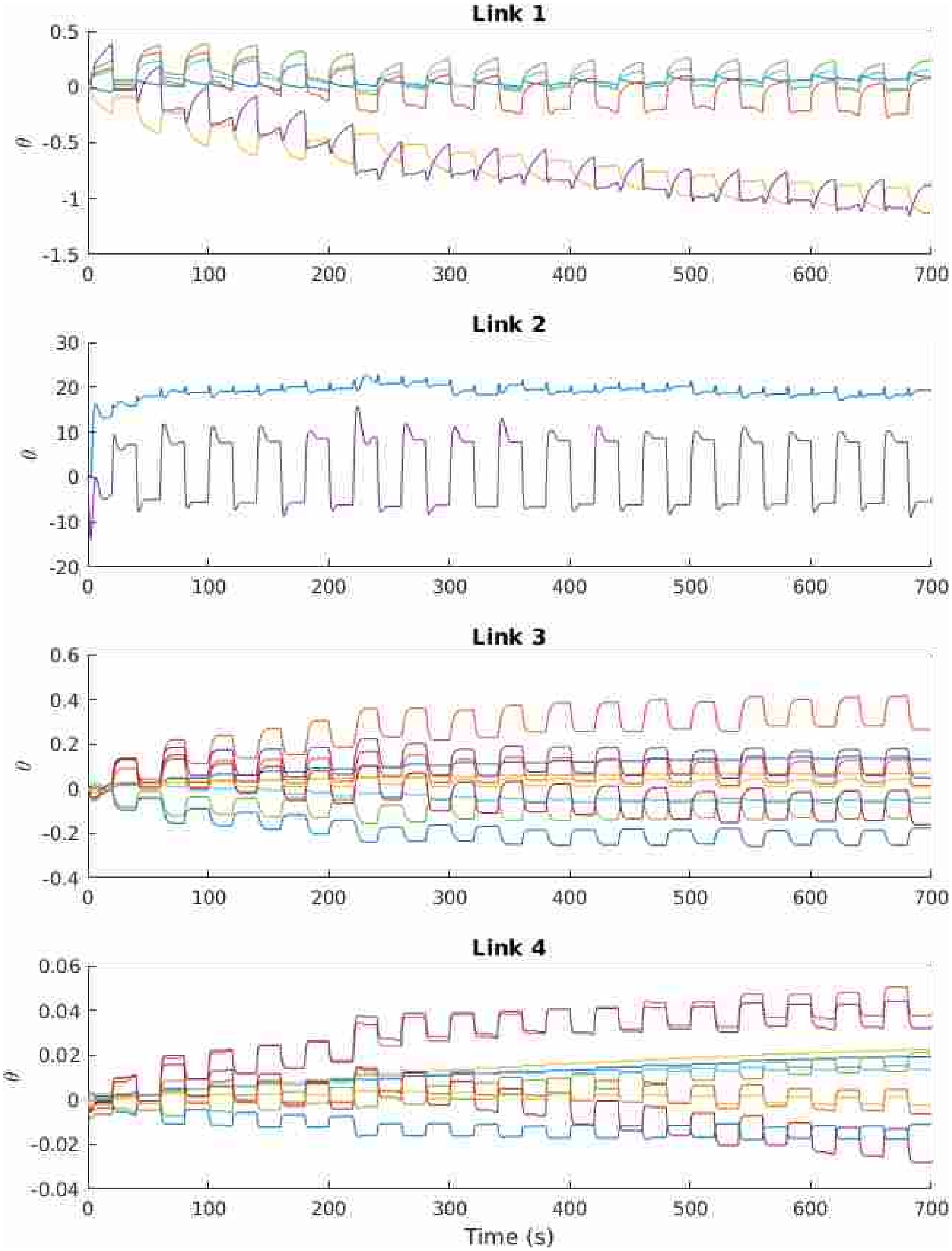


Figure 4.5: Growth of adaptive parameters over time with no weight in the end effector.

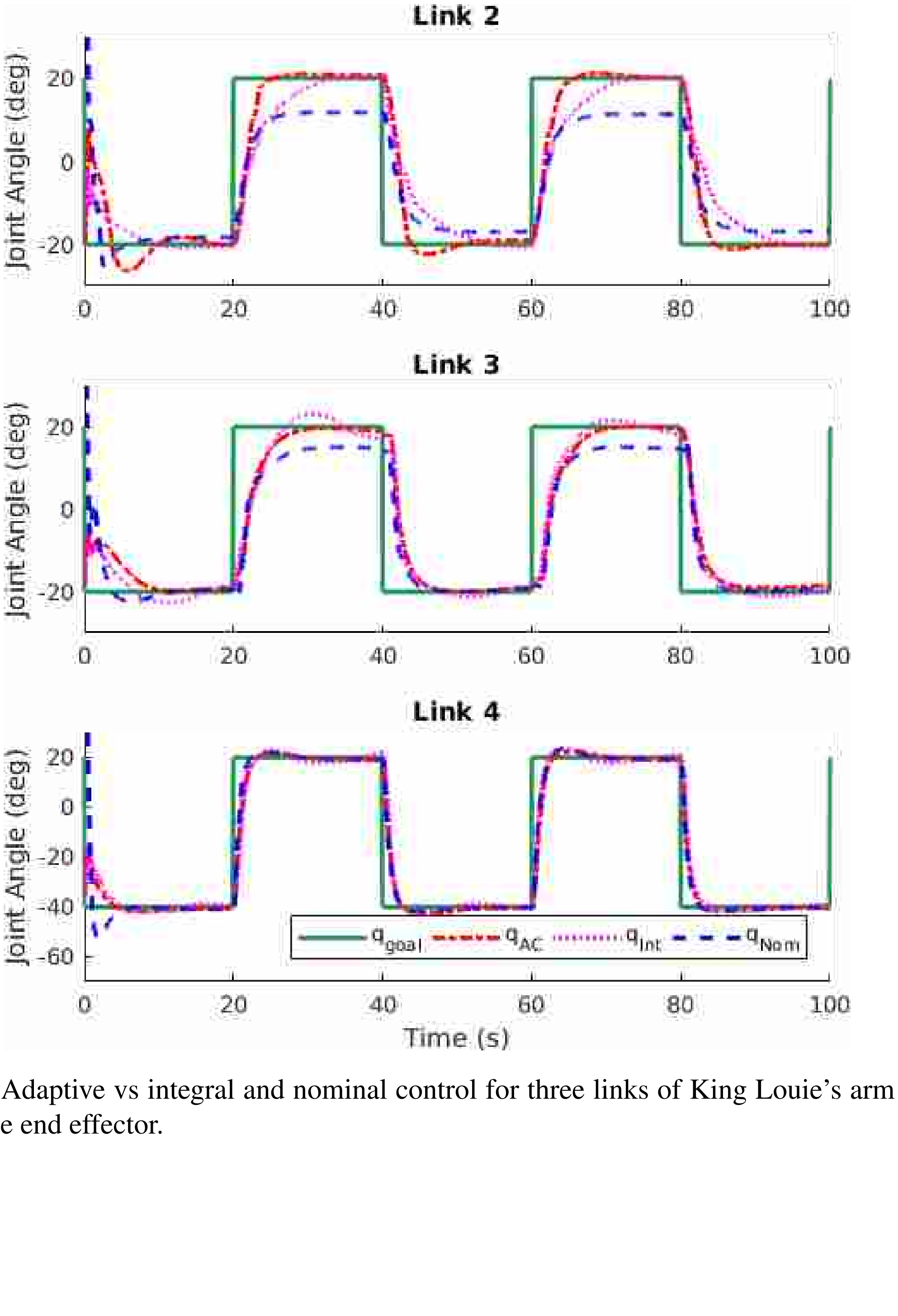


Figure 4.6: Adaptive vs integral and nominal control for three links of King Louie's arm with no weight in the end effector.

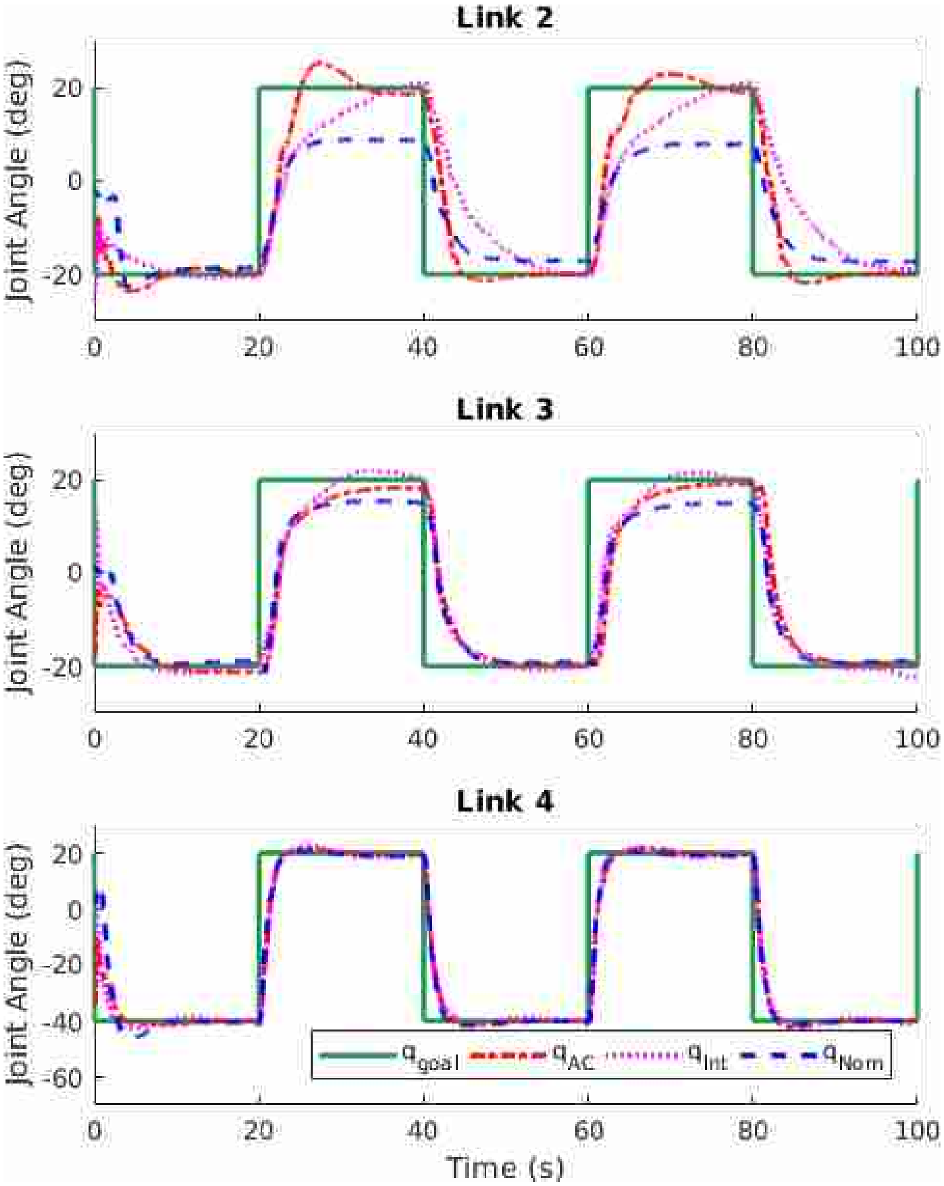


Figure 4.7: Adaptive vs integral and nominal control for three links of King Louie's arm with a one-pound weight (unknown to the controller) in the end effector.

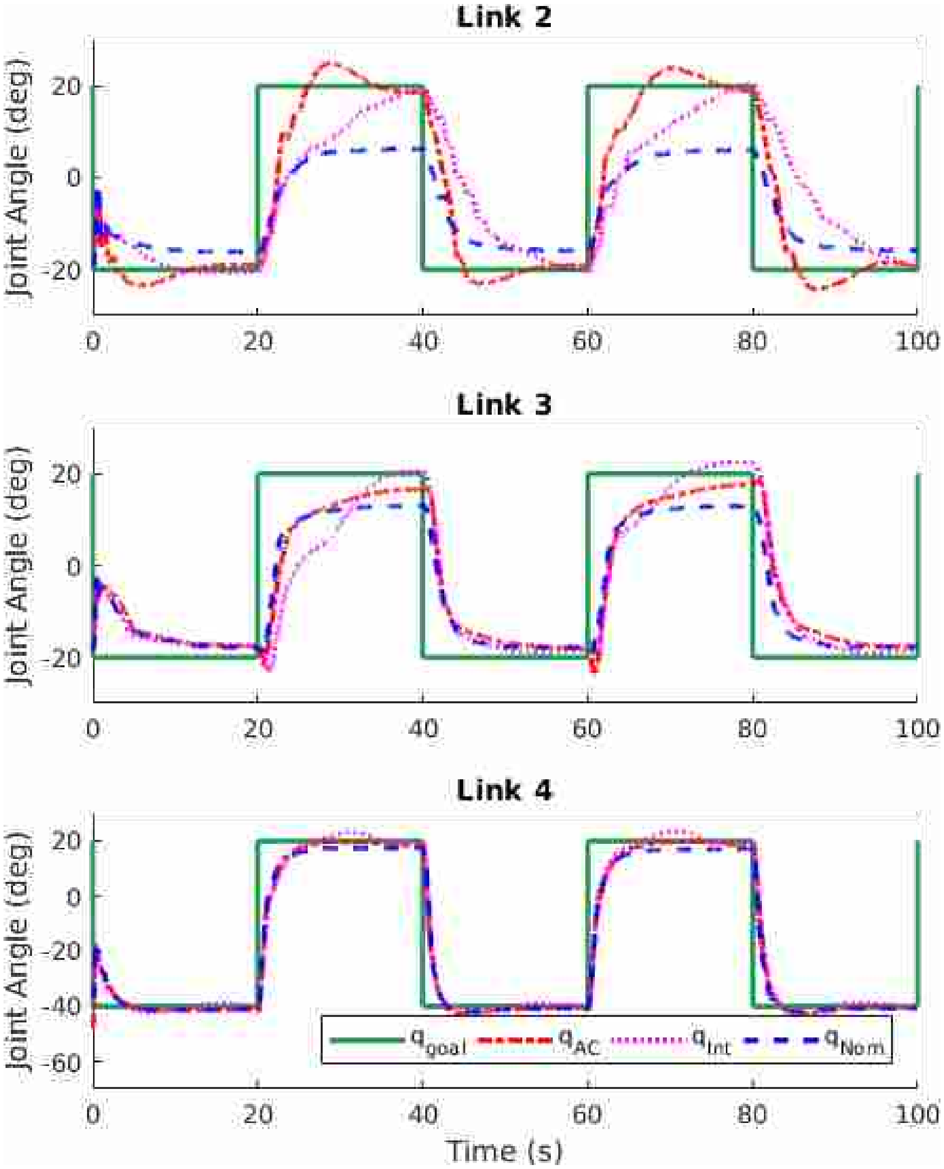


Figure 4.8: Adaptive vs integral and nominal control for three links of King Louie's arm with a two-pound weight (unknown to the controller) in the end effector.

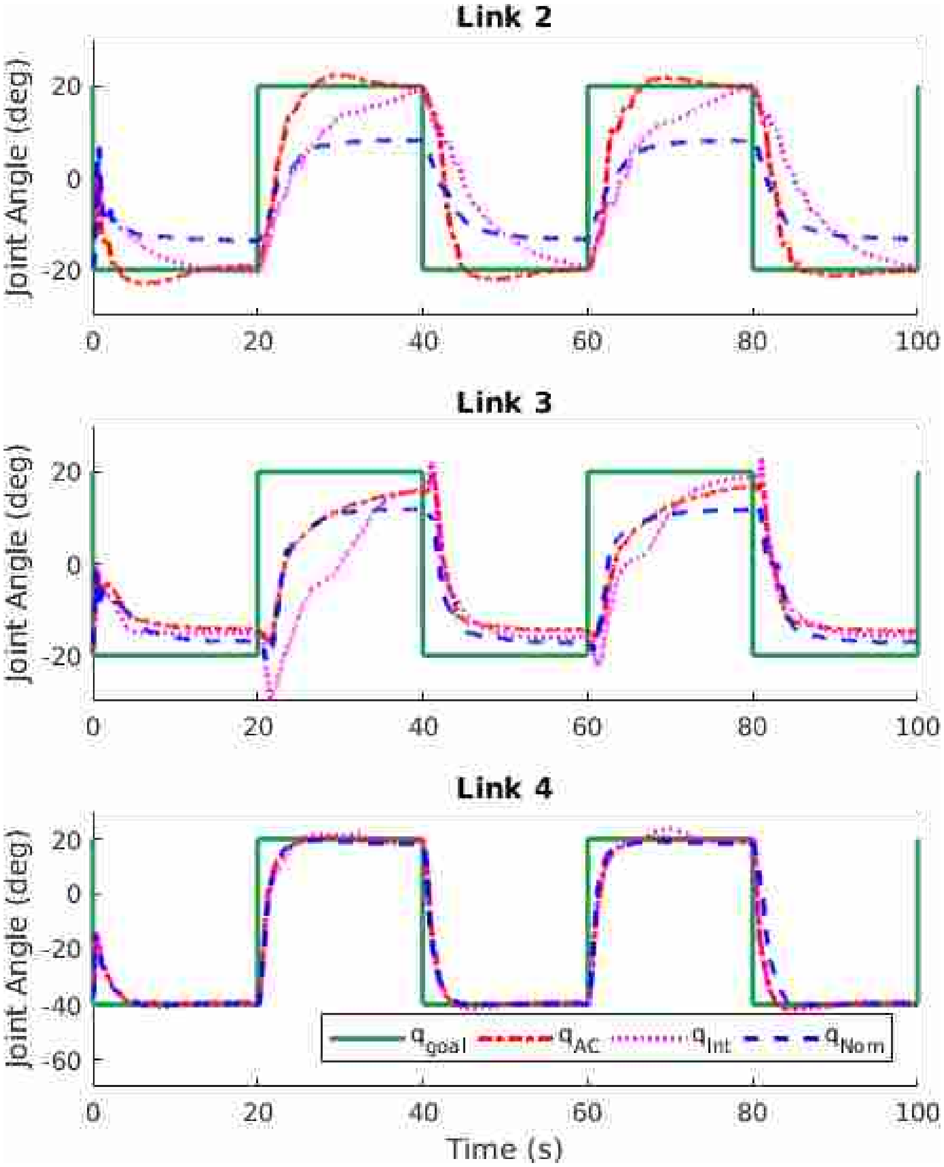


Figure 4.9: Adaptive vs integral and nominal control for three links of King Louie's arm with a three-pound weight (unknown to the controller) in the end effector.

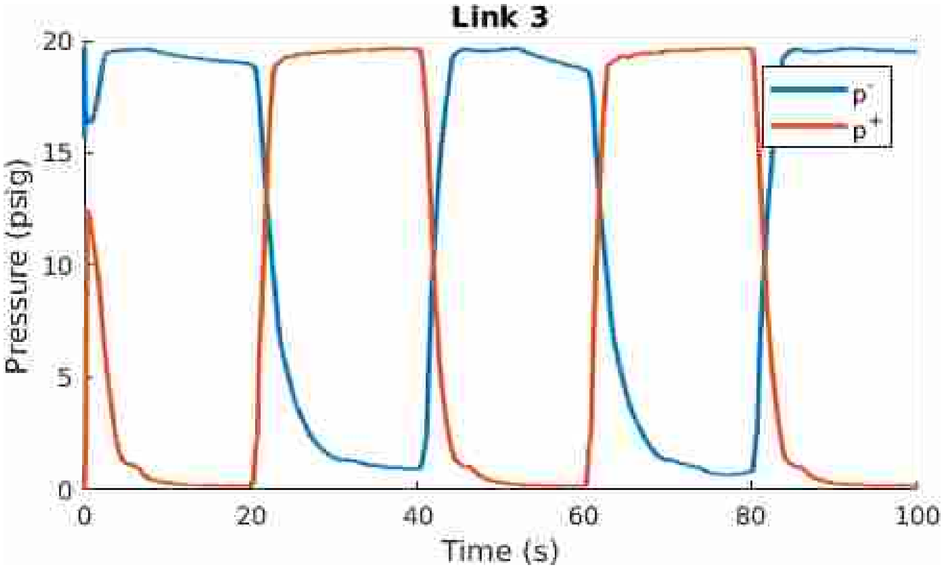


Figure 4.10: Actuation pressures for MRAC/MPC on King Louie with a three-pound weight.

For Link 3, MRAC/MPC shows improvement over the nominal control in steady-state offset, and smoother movement than the integral controller gives. However, the integral controller has less steady-state error than MRAC/MPC for most weight cases. For the three-pound case, the poor performance of all three controllers is at least partially due to actuator saturation where the commanded chamber pressures were near atmospheric pressure (see Fig 4.10).

For Link 4, control performance for the nominal case is already quite good, and neither the integral controller nor MRAC/MPC improve performance significantly.

For all links, but especially Links 2 and 3, MRAC/MPC does not improve rise time as much as we had hoped. It is possible that a different adaptive approach could make improvements here. It is also possible that this is a physical constraint of King Louie's construction, and that rise time cannot be decreased without introducing undesired oscillation.

Finally, a trial was run starting with $\theta_{m=0}$ and no weight in the end effector. At Time $\approx$ 90 seconds, a 3-pound mass is attached to the end effector, and was removed at Time $\approx$ 210 seconds. This gives an idea of the control performance with a heavy object in a pick-and-place task. The results of this trial are shown in Figure 4.11. The process of attaching and removing the mass was done manually by the operator, and so resulted in significant joint angle disturbances, which can be seen in Figure 4.11 around 90 and 210 seconds.

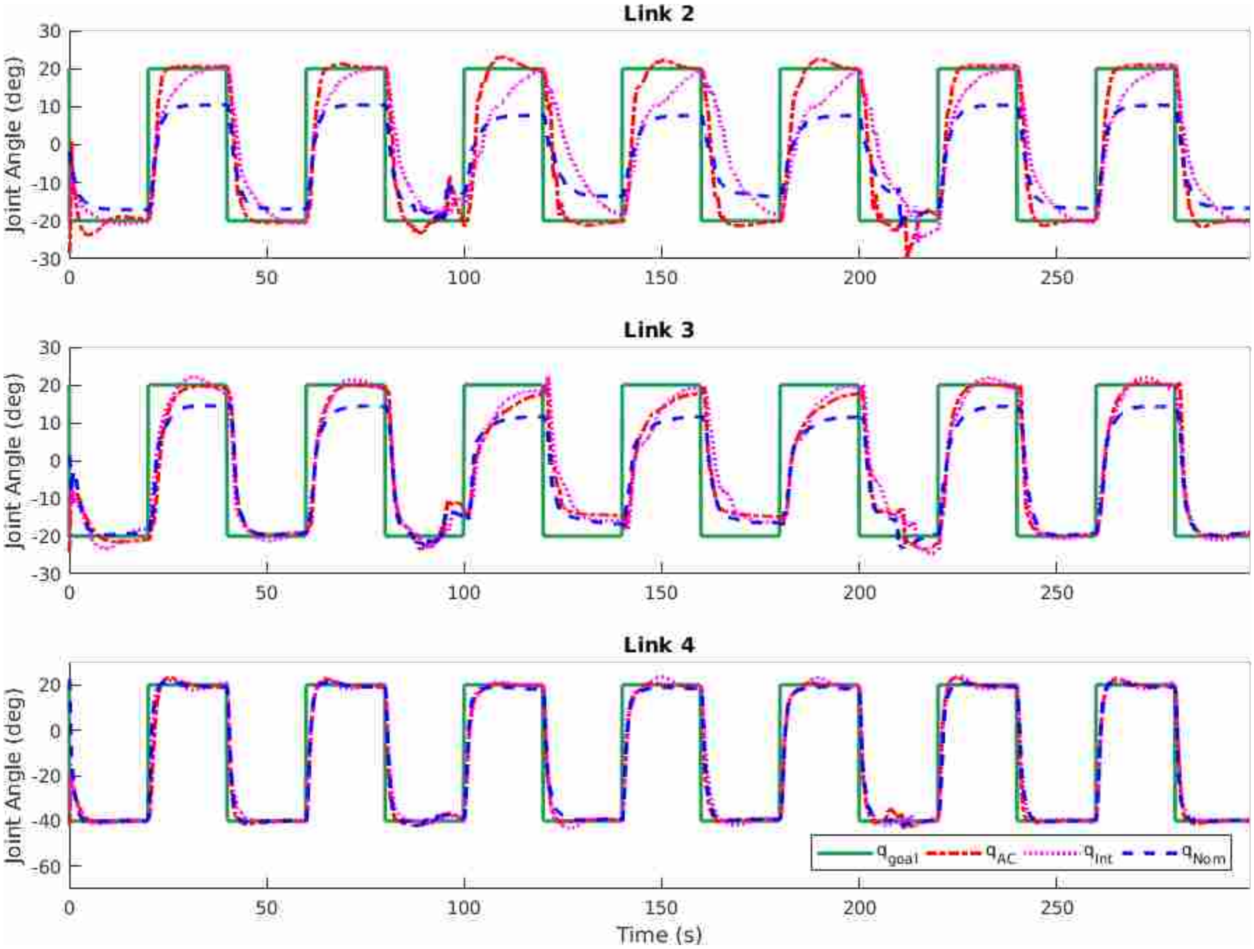


Figure 4.11: Adaptive vs integral and nominal control for three links of King Louie's arm. A 3-pound mass was attached to the end effector at about 90s and removed at about 210s. This resulted in significant disturbances at those times, which can be seen in the plot.

In addition to comparing controller performance with step inputs, we also compared performance when q_{goal} is a sine wave trajectory instead of step inputs (see Figure 4.12). Here, a three-pound mass is loaded on the end effector for the entire run. Again, the trial was started with the $\theta_{m=0}$ parameters. Adaptive control gives significant improvement for Link 2, while adaptive control performance is slightly worse for Link 3 and nearly identical for Link 4. Poor performance for both controllers on Link 3 at the positive peaks of the wave is due to actuator saturation. The same is true for Link 4, but to a smaller extent. Note that q_{ref} is set equal to q_{goal} for MRAC/MPC with a sine wave trajectory.

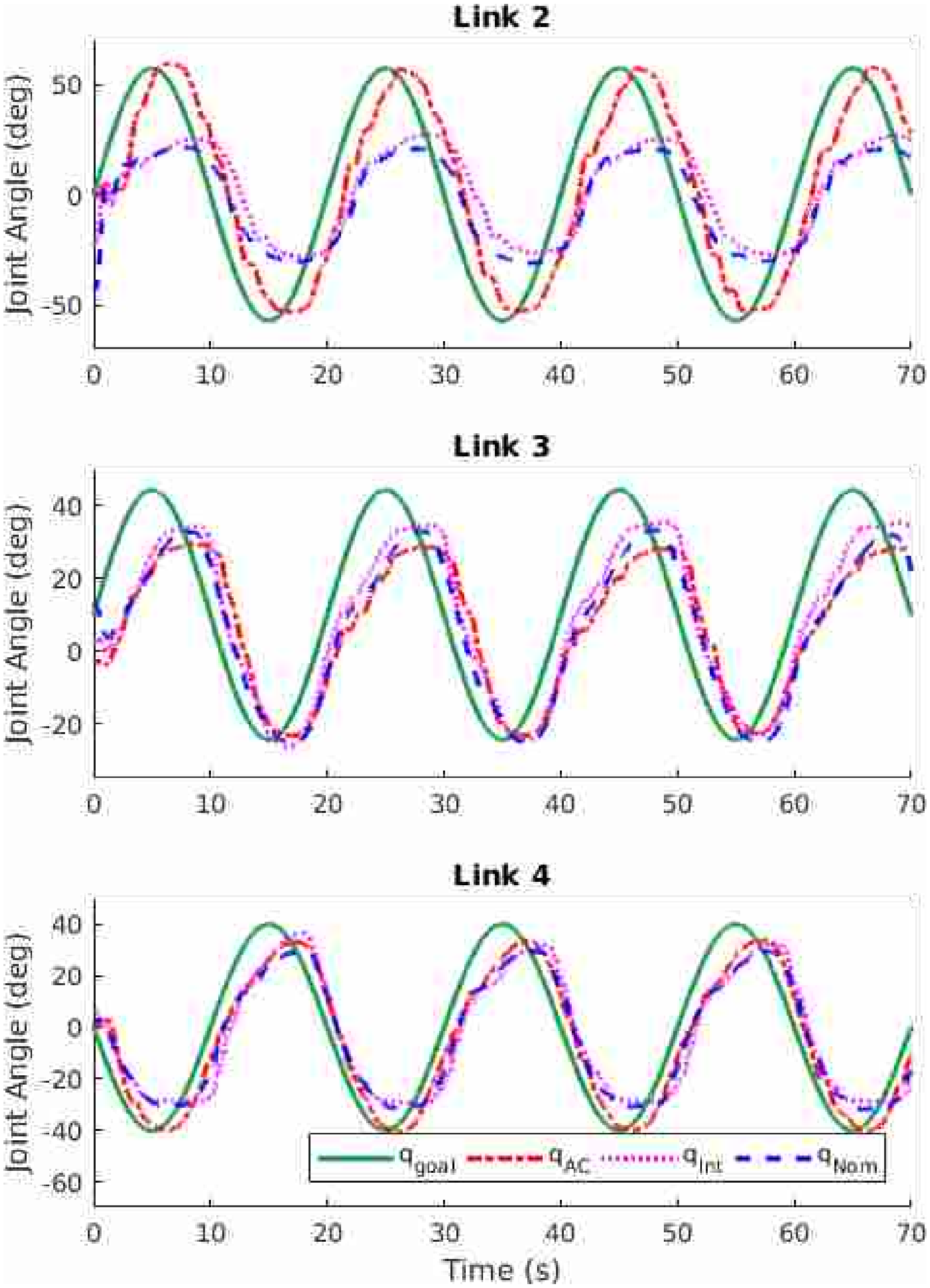


Figure 4.12: Adaptive vs integral and nominal control for three links of King Louie's arm. A 3-pound mass was attached to the end effector for the entire run.

The numerical data corresponding to Figures 4.6 - 4.12 is shown in Table 4.3. The results in Table 4.3 show general improvement using MRAC/MPC over the nominal and integral controllers.

The performance results demonstrate some strengths and weaknesses of this approach. Considerable improvements in joint angle control performance can be seen, especially in the case

Table 4.3: King Louie Experiment Results

Controller	**Trajectory**	**Mass (lb)**	**RMS Error (deg)**			**q_{goal} Error (deg)**			**95% Rise Time (s)**		
			Link 2	Link 3	Link 4	Link 2	Link 3	Link 4	Link 2	Link 3	Link 4
Nominal	Step	0	11.44	11.39	10.53	7.38	4.32	0.64	8.96	8.27	3.33
Integrator	Step	0	11.55	10.63	10.42	0.97	4.11	1.73	10.98	5.14	2.44
MRAC/MPC	Step	0	10.96	12.20	10.58	4.04	3.70	0.34	5.20	10.77	3.30
Nominal	Step	3	14.28	13.45	12.03	8.66	4.98	0.71	8.80	8.21	3.54
Integrator	Step	3	15.23	17.35	11.01	1.14	8.38	0.45	16.18	11.60	3.28
MRAC/MPC	Step	3	12.35	14.32	10.43	0.35	4.52	0.38	5.01	10.57	3.33
Nominal	Sine	3	25.28	14.57	11.47						
Integrator	Sine	3	30.89	13.81	14.92						
MRAC/MPC	Sine	3	20.88	16.78	10.82						

when significant weight (about half of the weight of the arm) is grasped with the end effector. We can also see from these results that there is room for improvement. It can be seen in Figure 4.5 that the adaptive parameters require several minutes to settle at ideal values. However, it is desirable to have adaptive control formulated such that parameters are adapted much more quickly. Additionally, it would be beneficial to decrease rise time generally for all experiments with step inputs. However, it is not clear at this point whether this is more of a current hardware limitation or a deficiency in the control schemes or adaptations used.

CHAPTER 5. CONCLUSION

This thesis has presented methodology that yields improved joint angle control for an inflatable robot manipulator with pneumatic actuation, including when an unknown load was grasped in the end effector. The reason the control of a robot of this type is of interest is because it is naturally lightweight and compliant, and the antagonistic nature of the actuators makes simultaneously controlling position and stiffness possible. Compliance and low inertia are desirable characteristics for the robots that operate around humans or in human environments. However, robots of this type are relatively new, and present difficulties in system modeling, sensing, and control that need to be addressed. This thesis addresses improving control for these robots, especially in the case of carrying objects of unknown weight.

5.1 Future Work

We expect that further investigation into the limits of Coupling-Torque MPC is important as the number of DoF for robot control increases. The most complex systems on which we have implemented Coupling-Torque MPC are the 14-state (7 Dof) Baxter manipulator, and the 16-state (4 DoF) King Louie manipulator. It is not clear, at this point, how complex a manipulator and model can be before the effectiveness of Coupling-Torque MPC begins to lessen.

We would also like to explore the theoretical bounds on approximation error for the three methods of linearization presented in Section 3.2. There is also the possibility to explore the energy consumption of the different methods and explore if one method tends towards more energy efficient control.

General model improvements could also greatly improve overall control performance. For example, control could be improved if the dynamic model also accounted for more of the gas dynamics. The gas dynamics are discontinuous in terms of choked and unchoked flow, and the behavior of the valve. This would require a controller that could account for the dramatic shift in dynamics as the system shifts from one mode to another. In this thesis, all of the gas dynamics are modeled as a first order pressure model. Simply increasing the order of the model and including cross coupling terms between the pressures in each actuator could improve performance rather than a direct model of the working fluid. Including these terms in the model and linearizing with the Coupling-Torque method may prove beneficial. This would also give the option of adapting those pressure dynamics with MRAC/MPC, or some other adaptive approach.

Improvements to adaptive control may be found in developing a method for faster parameter adaption. This might include reformulating the adaptive laws discussed in Section 4.2, or perhaps modifying a different adaptive strategy altogether, such as the methods discussed in Section 1.3.3. The lack of improvement when MRAC/MPC was applied to the Grub provides evidence that utilizing another adaptation method could be beneficial. For the adaptive scheme presented in this thesis, control performance improvement could be attained from a more rigorous method of identifying which dynamic parameters most need adaptation. Additionally, determining if improvements can be made to following a sine wave trajectory would also be beneficial.

5.2 Closing Remarks

In summary, in this thesis we compared three methods for linearizing the dynamic model of a robot arm, and showed how they affect MPC performance. The Taylor Series MPC performs the best overall in simulation, but is difficult to scale to higher DoF at high control rates with current optimization methods. We demonstrated the utility of Coupling-Torque MPC by showing that it has comparable performance to Taylor Series MPC in simulation, and that it performs well on Baxter and King Louie. Accomplishing similar performance with the Taylor Series MPC and Fixed-State MPC was not possible either due to computational limitations or difficulty in tuning.

We have also shown that by adapting dynamic parameters for soft robots online, we can significantly improve performance for model-based control. We demonstrated that knowing parameter values more precisely does not necessary lead to better control performance for soft robots with a given dynamic model parameterization. Instead, for these inflatable systems, adapting parameters based on control performance can be more effective than adapting parameters based on model accuracy. We also developed a form of MRAC/MPC that builds upon Coupling-Torque based MPC, and demonstrated its usefulness on three links of an inflatable soft robot with no load and while carrying an unknown load.

REFERENCES

[1] Tellez, R., Ferro, F., Garcia, S., Gomez, E., Jorge, E., Mora, D., Pinyol, D., Oliver, J., Torres, O., Velazquez, J., and Faconti, D., 2008. "Reem-B: An autonomous lightweight human-size humanoid robot." *2008 8th IEEE-RAS International Conference on Humanoid Robots, Humanoids 2008*, pp. 462–468. 1

[2] Kaneko, K., Kanehiro, F., Morisawa, M., Akachi, K., Miyamori, G., Hayashi, A., and Kanehira, N., 2011. "Humanoid robot HRP-4 - Humanoid robotics platform with lightweight and slim body." *IEEE International Conference on Intelligent Robots and Systems*, pp. 4400–4407. 1

[3] Haddadin, S., Suppa, M., Fuchs, S., Bodenmueller, T., Albu-Schäffer, A., Hirzinger, G., Bodenmüller, T., Albu-Schäffer, A., and Hirzinger, G., 2011. "Towards the Robotic Co-Worker." In *Robotics Research*, Vol. 70, pp. 261–282. 1

[4] De Luca, A., and Mattone, R., 2005. "Sensorless robot collision detection and hybrid force/motion control." *Proceedings - IEEE International Conference on Robotics and Automation,* **2005**(April), pp. 999–1004. 1

[5] De Luca, A., Albu-Schäffer, A., Haddadin, S., and Hirzinger, G., 2006. "Collision detection and safe reaction with the DLR-III lightweight manipulator arm." *IEEE International Conference on Intelligent Robots and Systems*, pp. 1623–1630. 1

[6] Haddadin, S., Albu-Schäffer, A., De Luca, A., and Hirzinger, G., 2008. "Collision detection and reaction: A contribution to safe physical human-robot interaction." *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pp. 3356–3363. 1

[7] Vallery, H., Veneman, J., van Asseldonk, E., Ekkelenkamp, R., Buss, M., and van Der Kooij, H., 2008. "Compliant actuation of rehabilitation robots." *IEEE Robotics and Automation Magazine,* **15**(3), pp. 60–69. 1

[8] Braun, D. J., Petit, F., Huber, F., Haddadin, S., Van Der Smagt, P., Albu-Schaffer, A., and Vijayakumar, S., 2013. "Robots driven by compliant actuators: Optimal control under actuation constraints." *IEEE Transactions on Robotics,* **29**(5), pp. 1085–1101. 1

[9] Zinn, M., Roth, B., Khatib, O., and Salisbury, J. K., 2004. "A new actuation approach for human friendly robot design." *International Journal of Robotics Research,* **23**(4-5), pp. 379–398. 1

[10] Bicchi, A., and Tonietti, G., 2004. "Fast and "soft-arm" tactics." *IEEE Robotics and Automation Magazine,* **11**(2), pp. 22–33. 2

[11] Sanan, S., Moidel, J. B., and Atkeson, C. G., 2009. “Robots with inflatable links.” *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009*, pp. 4331–4336. 2

[12] Sanan, S., Ornstein, M. H., and Atkeson, C. G., 2011. “Physical human interaction for an inflatable manipulator.” In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, pp. 7401–7404. 2

[13] Marchese, A. D., Komorowski, K., Onal, C. D., and Rus, D., 2014. “Design and control of a soft and continuously deformable 2D robotic manipulation system.” *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 2189–2196. 2

[14] Marchese, A. D., Katzschmann, R. K., and Rus, D., 2014. “Whole arm planning for a soft and highly compliant 2D robotic manipulator.” *IEEE International Conference on Intelligent Robots and Systems*(Iros), pp. 554–560. 2

[15] Marchese, A. D., Tedrake, R., and Rus, D., 2016. “Dynamics and trajectory optimization for a soft spatial fluidic elastomer manipulator.” *International Journal of Robotics Research,* **35**(8), pp. 1000–1019. 2

[16] Sanan, S., Lynn, P. S., and Griffith, S. T., 2014. “Pneumatic Torsional Actuators for Inflatable Robots.” *Journal of Mechanisms and Robotics,* **6**(3), p. 031003. 2, 4, 8

[17] Rus, D., and Tolley, M. T., 2015. “Design, fabrication and control of soft robots.” *Nature,* **521**(7553), pp. 467–475. 2

[18] Jones, J. A., 2001. “Inflatable Robotics for Planetary Applications.” *Jet Propulsion*(Figure 1), pp. 1–6. 2

[19] Gaiser, I., Wiegand, R., Ivlev, O., Andres, a., Breitwieser, H., Schulz, S., and Bretthauer, G., 2014. “Compliant Robotics and Automation with Flexible Fluidic Actuators and Inflatable Structures.” *Smart Actuation and Sensing SystemsRecent Advances and Future Challenges*(February), pp. 567–608. 2, 3

[20] Carneiro, J. F., and de Almeida, F. G., 2006. “Reduced-order thermodynamic models for servo-pneumatic actuator chambers.” *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering,* **220**(4), pp. 301–314. 3

[21] Shen, X., and Goldfarb, M., 2007. “Simultaneous force and stiffness control of a pneumatic actuator.” *Journal of Dynamic Systems, Measurement, and Control,* **129**(4), pp. 425–434. 3

[22] Bicchi, A., Rizzini, S. L., and Tonietti, G., 2001. “Compliant design for intrinsic safety: General issues and preliminary design.” In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, Vol. 4, IEEE, pp. 1864–1869. 3

[23] Tonietti, G., 2002. “Adaptive Simultaneous Position and Stiffness Control for a Soft Robot Arm.” In *Intelligent Robots and Systems*, no. October, pp. 1992–1997. 3, 5, 6

[24] Ivlev, O., 2009. “Soft fluidic actuators of rotary type for safe physical human-machine interaction.” *2009 IEEE International Conference on Rehabilitation Robotics, ICORR 2009,* **28359**, pp. 1–5. 3

[25] Shepherd, R. F., Ilievski, F., Choi, W., Morin, S. A., Stokes, A. A., Mazzeo, A. D., Chen, X., Wang, M., and Whitesides, G. M., 2011. "Multigait soft robot." *Proceedings of the National Academy of Sciences,* **108**(51), pp. 20400–20403. 4

[26] Shepherd, R. F., Stokes, A. A., Freake, J., Barber, J., Snyder, P. W., Mazzeo, A. D., Cademartiri, L., Morin, S. A., and Whitesides, G. M., 2013. "Using explosions to power a soft robot." *Angewandte Chemie - International Edition,* **52**(10), pp. 2892–2896. 4

[27] Nemiroski, A., Shevchenko, Y. Y., Stokes, A. A., Unal, B., Ainla, A., Albert, S., Compton, G., MacDonald, E., Schwab, Y., Zellhofer, C., and Whitesides, G. M., 2017. "Arthrobots." *Soft Robotics,* **4**(3), p. soro.2016.0043. 4

[28] Hedjar, R., Toumi, R., Boucher, P., and Dumur, D., 2005. "Finite horizon nonlinear predictive control by the taylor approximation: application to robot tracking trajectory." *International Journal of Applied Mathematics and Computer Science,* **15**(4), p. 527. 4

[29] Hamdi, S., Boucetta, R., and Bel Hadj Ali Naoui, S., 2016. "Dynamic modeling of a rigid-flexible manipulator using Hamilton's principle." *16th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering*, pp. 832–838. 4

[30] Mahmoud, M. S., and Bahnasawi, A. A., 1993. "Indirect discrete-time adaptive algorithm for manipulator control." *Applied Mathematical Modelling,* **17**(8), pp. 423–429. 4

[31] Duchaine, V., Bouchard, S., and Gosselin, C. M., 2007. "Computationally efficient predictive robot control." *IEEE/ASME Transactions on Mechatronics,* **12**(5), pp. 570–578. 4

[32] Poignet, P., and Gautier, M., 2000. "Nonlinear model predictive control of a robot manipulator." *6th International Workshop on Advanced Motion Control Proceedings*(2), pp. 401–406. 4, 12

[33] Makarov, M., Grossard, M., Rodriguez-Ayerbe, P., and Dumur, D., 2011. "Generalized Predictive Control of an anthropomorphic robot arm for trajectory tracking." *2011 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM),* **2**, pp. 948–953. 4

[34] Makarov, M., Grossard, M., Rodriguez-Ayerbe, P., and Dumur, D., 2012. "Active damping strategy for robust control of a flexible-joint lightweight robot." IEEE, pp. 1020–1025. 4

[35] De Luca, A., and Lucibello, P., 1998. "A general algorithm for dynamic feedback linearization of robots with elastic joints." *IEEE International Conference on Robotics and Automation,* **1**(May), pp. 504–510. 4

[36] Rupert, L., Hyatt, P., and Killpack, M. D., 2015. "Comparing Model Predictive Control and input shaping for improved response of low-impedance robots." *IEEE-RAS International Conference on Humanoid Robots,* **2015-Decem**, pp. 256–263. 4, 12, 21, 33

[37] Nakanishi, J., Cory, R., Mistry, M., Peters, J., and Schaal, S., 2008. "Operational space control: A theoretical and empirical comparison." *International Journal of Robotics Research,* **27**(6), pp. 737–757. 5

[38] Ling, K. V., Maciejowski, J., Richards, A., and Wu, B. F., 2012. "Multiplexed model predictive control ." *Automatica,* **48**(2), pp. 396–401. 5

[39] Chowdhary, G., Mühlegg, M., How, J., and Holzapfel, F., 2013. "Concurrent Learning Adaptive Model Predictive Control." In *Advances in Aerospace Guidance, Navigation and Control*, Q. Chu, B. Mulder, D. Choukroun, E.-J. Kampen, C. Visser, and G. Looye, eds. Springer Berlin Heidelberg, ch. 3, pp. 29–47. 5

[40] Fukushima, H., Kim, T.-H., and Sugie, T., 2007. "Adaptive model predictive control for a class of constrained linear systems based on the comparison model." *Automatica,* **43**(2), pp. 301–308. 5

[41] Adetola, V., and Guay, M., 2009. "Robust adaptive MPC for systems with exogeneous disturbances." *IFAC Proceedings Volumes (IFAC-PapersOnline),* **7**(PART 1), pp. 249–254. 5

[42] Chen, Y., Moore, K. L., Yu, J., and Xu, J.-X., 2008. "Iterative learning control and repetitive control in hard disk drive industry$–$A tutorial." *International Journal of Adaptive Control and Signal Processing,* **22**(4), pp. 325–343. 5

[43] Nguyen, K. D., and Dankowicz, H., 2015. "Adaptive control of underactuated robots with unmodeled dynamics." *Robotics and Autonomous Systems,* **64**, pp. 84–99. 5

[44] Slotine, J.-J. E., and Li, W., 1986. "On the Adaptive Control of Robot Manipulators." *ASME Winter Meeting,* **6**(1987), pp. 43–50. 5

[45] Taghia, J., Wilkening, A., and Ivlev, O., 2013. "Adaptive position control of fluidic soft-robots working with unknown loads." In *2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics:*, pp. 1121–1126. 5

[46] Skorina, E. H., Luo, M., Tao, W., Chen, F., Fu, J., and Onal, C. D., 2017. "Adapting to Flexibility: Model Reference Adaptive Control of Soft Bending Actuators." *IEEE Robotics and Automation Letters,* **2**(2), pp. 964–970. 5

[47] Khan, S. G., Herrmann, G., Pipe, T., Melhuish, C., and Spiers, A., 2010. "Safe adaptive compliance control of a humanoid robotic arm with anti-windup compensation and posture control." *International Journal of Social Robotics,* **2**(3), pp. 305–319. 5

[48] Medrano-Cerda, G. A. ., Bowler, C. J. ., and Caldwell, D. G. ., 1995. "Adaptive Position Control of Antagonistic Pneumatic Muscle Actuators." *Intelligent Robots and Systems,* **1**, pp. 378–383. 6

[49] Wilson, J. P., Best, C. M., and Killpack, M. D., 2017. "Variable stiffness adaptation to mitigate system failure in inflatable robots." *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 5844–5851. 6

[50] Best, C. M., Wilson, J. P., and Killpack, M. D., 2015. "Control of a pneumatically actuated, fully inflatable, fabric-based, humanoid robot." In *IEEE-RAS International Conference on Humanoid Robots*, Vol. 2015-Decem, pp. 1133–1140. 7, 12

[51] Best, C. M., Gillespie, M. T., Hyatt, P., Rupert, L., Sherrod, V., and Killpack, M. D., 2016. "A new soft robot control method: Using model predictive control for a pneumatically actuated humanoid." *IEEE Robotics & Automation Magazine,* **23**(3), pp. 75–84. 7, 12

[52] Qin, S. J., and Badgwell, T. A., 2003. "A survey of industrial model predictive control technology." *Control Engineering Practice,* **11**(7), pp. 733–764. 12

[53] Mattingley, J., and Boyd, S., 2012. "Cvxgen: A code generator for embedded convex optimization." *Optimization and Engineering,* **13**(1), pp. 1–27. 12, 28

[54] Domahidi, A., Zgraggen, A. U., Zeilinger, M. N., Morari, M., and Jones, C. N., 2012. "Efficient interior point methods for multistage problems arising in receding horizon control." In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, IEEE, pp. 668–674. 12

[55] Wang, Y., and Boyd, S., 2010. "Fast model predictive control using online optimization." *IEEE Transactions on Control Systems Technology,* **18**(2), pp. 267–278. 12

[56] Domahidi, A., and Zgraggen, A., 2012. "Efficient Interior Point Methods for Multistage Problems Arising in Receding Horizon Control." In *IEEE Conference on Decision and Control*. 12

[57] Killpack, M. D., and Kemp, C. C., 2013. "Fast reaching in clutter while regulating forces using model predictive control." In *Humanoid Robots (Humanoids), 2013 13th IEEE-RAS International Conference on*, IEEE, pp. 146–153. 12, 21

[58] Jain, A., Killpack, M. D., Edsinger, A., and Kemp, C. C., 2013. "Reaching in clutter with whole-arm tactile sensing." *The International Journal of Robotics Research*. 12

[59] Killpack, M. D., Kapusta, A., and Kemp, C. C., 2015. "Model predictive control for fast reaching in clutter." *Autonomous Robots*, pp. 1–24. 12, 21

[60] Erez, T., Tassa, Y., and Todorov, E., 2012. "Infinite-horizon model predictive control for periodic tasks with contacts." *Robotics: Science and Systems VII*, p. 73. 12

[61] Wilson, J., Charest, M., and Dubay, R., 2016. "Non-linear model predictive control schemes with application on a 2 link vertical robot manipulator." *Robotics and Computer-Integrated Manufacturing,* **41**, pp. 23–30. 12

[62] Koenemann, J., Del Prete, A., Tassa, Y., Todorov, E., Stasse, O., Bennewitz, M., and Mansard, N., 2015. "Whole-body model-predictive control applied to the hrp-2 humanoid." In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, IEEE, pp. 3346–3351. 12

[63] Kuindersma, S., Deits, R., Fallon, M., Valenzuela, A., Dai, H., Permenter, F., Koolen, T., Marion, P., and Tedrake, R., 2016. "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot." *Autonomous Robots,* **40**(3), pp. 429–455. 12

[64] Shim, D., Kim, H., and Sastry, S., 2003. "Decentralized nonlinear model predictive control of multiple flying robots." *IEEE International Conference on Decision and Control,* **4**(December), pp. 3621–3626. 12

[65] Leung, C., Huang, S., Kwok, N., and Dissanayake, G., 2006. "Planning under uncertainty using model predictive control for information gathering." *Robotics and Autonomous Systems,* **54**(11), pp. 898–910. 12

[66] Annamalai, A. S. K., Sutton, R., Yang, C., Culverhouse, P., and Sharma, S., 2015. “Robust Adaptive Control of an Uninhabited Surface Vehicle.” *Journal of Intelligent & Robotic Systems,* **78**(2), pp. 319–338. 12

[67] Mattingley, J., and Boyd, S., 2012. “Cvxgen: a code generator for embedded convex optimization.” *Optimization and Engineering,* **13**(1), pp. 1–27. 13, 39

[68] Lavretsky, E., and Wise, K. A., 2013. *Robust and Adaptive Control With Aerospace Applications*. Springer, London. 14, 42

[69] Corke, P., 2011. *Robotics, Vision and Control-Fundamental Algorithms in MATLAB*. Springer. 16, 28, 37

[70] Ogata, K., 1995. *Discrete-Time Control Systems*. Prentice Hall Englewood Cliffs, NJ. 20

[71] Gillespie, M. T., 2016. “Comparing Efficacy of Different Dynamic Models for Control of Underdamped , Antagonistic , Pneumatically Actuated Soft Robots.” PhD thesis, Brigham Young University. 47